SURFACE RECONSTRUCTION AND LATERAL ADSORBATE INTERACTIONS ON COBALT-BASED CATALYSTS: AN INVESTIGATION USING LATTICE GAS MODELS AND DENSITY FUNCTIONAL THEORY

By

GREGORY BRANDON COLLINGE

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSPHY

WASHINGTON STATE UNIVERSITY
The Gene and Linda Voiland School of Chemical Engineering and Bioengineering

DECEMBER 2019

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of GREGORY BRANDON COLLINGE find it satisfactory and recommend that it be accepted.

____
Jean-Sabin McEwen, Ph.D., Chair

____
Norbert Kruse, Ph.D.

____
Cornelius Ivory, Ph.D.

____
Kirk Peterson, Ph.D.

ACKNOWLEDGMENT

This dissertation is not just the culmination of the five years I have spent as a graduate student in the research group of Jean-Sabin McEwen in the Voiland School of Chemical Engineering and Bioengineering at Washington State University. No, this dissertation is truly the physical embodiment of a sinuous, halting, and, at times, marred with existential crisis 16-year journey from an idealistic and naïve childhood to a—well—slightly *less* idealistic and naïve adulthood…which I like to think also has some highlights of newly-acquired pragmatism and wisdom. It has been easy to downplay how important this accomplishment would mean to me, but now that I am about to attain something that I have been working toward for so long, I must admit I am humbled, incredibly proud, and both excited and frightened about what the future holds. So! I have a great number of people, organizations, and things to be thankful for, and I want to briefly enumerate them here.

First, I must thank the Achievement Rewards for College Scientists (ARCS), Seattle chapter, the National Science Foundation (NSF) East Asia and Pacific Summer Institutes (EAPSI, grant number 1613890), the NSF Graduate Research Fellowship Program (GRFP, grant number 1347973), and Belgian American Education Foundation (BAEF, Boat '18) Fellowship program for funding and opportunities to do the research that ultimately makes up this dissertation. I could not have done this without them.

It perhaps goes without saying, but I am immeasurably thankful for my mother, my sisters, and my grandparents; who effectively raised me and instilled within me a desire to always grow and become a better person while never forgetting that I am a fallible human like everyone else (yes, even those who *don't* have a Ph.D.! Crazy, I know…). I wouldn't be here,

both literally and metaphorically, if not for you guys, so again, I cannot begin to explain, especially with so little room available, how grateful I am to be part of this family.

Next, I need to thank the many teachers and professors who saw within me something worth cultivating and encouraging. I won't enumerate all of them here—if you suspect you might be one of them, you probably are (looking at you, Steven Saunders and Kirk Peterson!)—but I credit my desire to become a professor over just a researcher to these amazing people. I can only hope to be as good a teacher as they are

I want to extend my appreciation to the entirety of my Ph.D. committee: Jean-Sabin McEwen, Norbert Kruse, Cornelius Ivory, and Kirk Peterson. Their presence on my committee is not just a service to the university but also a service to me. I sincerely appreciate the time they have taken to assess my progress as a graduate student and Ph.D. candidate. Unequivocally, I can say I have zero regrets asking these individuals to be on my Ph.D committee, and I hope they have zero (or at least a number that comfortably rounds down to zero) regrets acquiescing!

I want to thank my Ph.D. advisor, Jean-Sabin McEwen, for the many and varied opportunities to improve my CV and professional experiences during graduate school: the myriad letters of recommendation, award nominations, and introductions that ultimately landed me the scholarships and fellowships I earned during my tenure as a graduate student here at WSU. I must also thank Jean-Sabin for the latitude he provided to choose projects that truly interested and excited me; it is no exaggeration to say that the latter half of this dissertation would not have happened without this freedom. For this and more, I am incredibly appreciative.

I want to also provide a special thanks to my experimental collaborator and Ph.D. committee member, Norbert Kruse, who has been an amazing and selfless advocate, mentor, and indeed, friend. Norbert was the first person in academia to make me feel like I belonged there, that my work was exceptional, and that my ideas were worth exploring. Whether during formal meetings or impromptu ones over coffee at Thomas Hammer, his influence, both scientific and professional, has been and continues to be enjoyable and truly noteworthy.

Renqin Zhang, Fanglin Che, Jacob Bray, Kyle Groden, and Neeru Chaudhary amongst all the others in the McEwen group are owed a particularly strong thank you here. It is no exaggeration to say that my time in the laboratory/office would have been both less enjoyable and less intellectually productive without these people. Having now spent time in a number of other research groups, I can say with certainty that the research environment we cultivated was special and enviable. More importantly, at least one or two ideas in this dissertation only came about because of my interactions these amazing people! I am glad to have called them mentors, colleagues, and friends.

For their collaborations, the professional opportunities, and my own personal growth, I also wish to thank Catherine Stampfl at the University of Sydney in Australia and Mark Saeys at Ghent University in Ghent, Belgium. In both cases, my time working with these excellent researchers was too short, but I gained a great deal from my exposure to their research and ideas, and of course, the culture of the amazing countries they work in. Not everyone gets such opportunities, and I want to express how grateful I am; I will not take the time they spent on me for granted for an instant.

As cliché as it is, I want to thank the many friends I have made throughout the various “eras” of my life. Matthew Bellmer and Beth Robinette are high school friends that subsequently helped me climb out of the pit of complete social deficit I managed to dig after the six years that followed my cancer diagnoses and subsequent treatments and hospitalizations that deprived me of the social growth one is supposed to experience between the ages of 18 and 24. Starting at

Spokane Falls Community College, Jake Gray was my first real college friend, responsible for me majoring in Chemical Engineering, and truly a worthy foil; which is still true to this day. The best parts of my undergraduate years at WSU were due to my roommates, Robert Franklin and Evelyn Moos, who were steadfast and wonderful friends that supported me during my most overly responsible and overly *ir*responsible moments there; who also kept me grounded, reminded me why the arts and humanities are important, and started me down the road to loving spicy food. Nearly all the fun and fulfilling moments in graduate school I owe to Alyssa and Heydon Hensley. I want to specifically thank Heydon for all the proof reading, late-night musings about the role of men in Feminism (hint: it is large), talking endlessly about languages and linguistics with me, and of course being my guide to the world of Dungeons and Dragons. Alyssa deserves special mention here, too, as she started as the best TA that I had ever had, then the best mentor I had ever had, but slowly and surely became one of the best friends that I have ever had. Whether staying up late at Rico's, rafting recklessly down the north fork of the Clearwater, playing Dungeons and Dragons for 6+ hours at a time (special shout out to Matt Ramsey, Sean Stubbe, and Kevin Kinkle, here!), or just sitting over coffee and talking science and our careers, Alyssa is no doubt single-handedly responsible for the vast majority of the personal and professional growth responsible for this dissertation. For this, I am eternally grateful; and I look forward to what the future holds for our friendship.

For keeping me sane and in incredibly good company while in Ghent, Belgium, I want to also thank Lars Grabow and Günes Avci. It was completely by chance that we became friends, but boy am I glad we did! Even amongst the excellent Belgian beers and frites, I can definitely say I miss these guys the most. That Lars was willing to learn about my research purely so he could be a comprehensive recommender on my future applications was above and beyond anything I would have expected. Sincerely, I honestly hope we don't go a year not seeing one another (at a conference or otherwise).

My fiancé, Marin Hatcher, needs mentioning here, as well, because she is truly the most amazing, intelligent, kind, and beautiful person I know; but perhaps most importantly because she agreed to keep dating me after I suddenly announced I was going to spend a year in Belgium. Even when things were tough and frustrating, she continued and continues to be a source of strength and compassion in my life. I could not have written this dissertation without an aneurism (a hopefully metaphorical one) without her in my life, and I will spend the rest of it reaffirming my gratitude for all of this. I could be grossly more effusive about how much I cherish and appreciate Marin for all 350 pages of this dissertation, but since this is not a place for poetry, I will simply state that I love her very, very much, and that I cannot wait to see what our future holds! Thank you so much, Marin!

There are many more who deserve mentioning, but I suppose they will have to wait for my memoirs. A final thank you to everyone at Washington State University, especially fiscal specialist, Jennifer Keller, administrative assistant, Nicole Cannon, academic coordinator, Samantha Bailey, and IT system administrator, Kate Konen, for making my life infinitely easier while there: you guys are truly unsung heroes.

SURFACE RECONSTRUCTION AND LATERAL ADSORBATE INTERACTIONS ON COBALT-BASED CATALYSTS: AN INVESTIGATION USING LATTICE GAS MODELS AND DENSITY FUNCTIONAL THEORY

Abstract

by Gregory Brandon Collinge, Ph.D.
Washington State University
December 2019

Chair: Jean-Sabin McEwen

This dissertation deals with the problem of computationally determining the structure and catalytically active phase of cobalt-based Fischer-Tropsch (FT) catalysts driven toward oxygenates. Cobalt-copper based catalysts are examined first using density functional theory calculations. The interactions of carbon monoxide (CO) with cobalt-copper nanoparticle surfaces are analyzed for CO's possible role in restructuring these nanoparticles and subsequent creation of the catalytically active phase. It is shown that cobalt-copper catalysts preferentially form Co@Cu core-shell nanoparticles, and that CO can induce up to half of step edge and one quarter of terrace copper atoms to be substituted with cobalt atoms from the core. Cobalt enrichment is limited due to the formation of cobalt subcarbonyl complexes. These complexes are shown to be capable of rupturing and diffusing across the surface, providing the ingredients needed for nanoisland formation and facet reconstruction. This work points to new models that are likely important to cobalt-copper catalyzed FT.

New lattice gas (LG) cluster expansion (CE) tools are also developed in this dissertation via the development of the *ab initio* Mean-field Augmented Lattice Gas Modeling (AMALGM) code. A new theoretical reformulation of the CE formalism, which is more appropriate for the LG paradigm, is detailed. AMALGM is designed to use a newly developed convergent version of the leave-multiple-out cross-validation (LMO-CV) score as the objective function for optimization of LG CEs. A new method for quantifying the errors of the effective cluster interactions of CEs is also developed. This error quantification is shown to capture the uncertainty in CEs due to geometric relaxations in DFT data. AMALGM and these new methods are then utilized to investigate the differences in CO adsorption energetics on the face centered cubic (fcc) and hexagonal close packed (hcp) phases of cobalt relevant to FT. It is shown that the first nearest neighbor (1NN) dominates and is very repulsive in both but significantly more repulsive for fcc cobalt. At high chemical potentials, 1NN CO pairings may be allowed on hcp cobalt where they are still energetically prohibited on fcc cobalt, suggesting a potential source for the reduced activity observed on hcp cobalt catalysts.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ACRONYMS

1NN: 1st Nearest Neighbor

1TT: 1st Nearest Neighbor “Triangular Trio”

AMALGM: *Ab initio* Mean-field Augmented Lattice Gas Modeling (code)

ANOVA: Analysis of Variance

APT: Atom Probe Tomography

ATAT: Alloy Theoretic Automated Toolkit

BO: Born-Oppenheimer

CE: Cluster Expansion

CESS: Cluster Expansion Similarity Score

CI: Confidence Interval

CS: Construction Set

CV: Cross Validation

DFT: Density Functional Theory

DOS: Density of States

ECI: Effective Cluster Interaction

EXAFS: Extended X-ray Absorption Fine Structure

FBZ: First Brillouin Zone

fcc: face centered cubic

FIM: Field Ion Microscopy

FS: Final State

FT: Fischer-Tropsch

GGA: Generalized Gradient Approximation

hcp: hexagonal close packed

HF: Hartree-Fock

IFP: *Institut Francais du Petrol*

IR: Infrared (spectroscopy)

IS: Initial State

KPPRA: K-Points Per Reciprocal Atom

KS: Kohn-Sham

LEED: Low Energy Electron Diffraction

LG: Lattice Gas

LMO-CV: Leave-Multiple-Out Cross-Validation (score)

LOO-CV: Leave-One-Out Cross-Validation (score)

MEP: Minimum Energy Pathway

MF: Mean Field

ML: Monolayer

NN: Neural Network (chapter 6) or Nearest Neighbor (chapter 7 and 8)

PAW: Projector Augmented Wave

PBE: Perdew-Burke-Enzerhof

pDOS: Projected Density of States

PES: Potential Energy Surface

RME: Root Mean Error

RMSR: Root Mean Squared Residual

SCF: Self Consistent Field

STM: Scanning Tunneling Microscopy

TEM: Transmission Electron Microscopy

TS: Transition State

UNCLE: UNiversal CLuster Expansion (code)

VASP: Vienna *Ab initio* Simulation Package

VS: Validation Set

XPS: X-ray Photoelectron Spectroscopy

Dedication

To my mother, Diana, and

my grandparents, Clifford and Karyl.

For raising me, trusting me, and

unconditionally loving me.

CHAPTER ONE:

INTRODUCTION

1.1 Background and Motivation

The research presented in this dissertation is motivated by the need for a rational strategy in the design of cobalt-based Fischer-Tropsch (FT) catalysts tuned toward the synthesis of so-called "oxygenates"—long-chain hydrocarbons with terminal oxygen functionality, e.g. alcohols and aldehydes. Oxygenates are incredibly desirable chemical products that can be used in the production of pharmaceuticals, detergents, plastics, etc., with industry typically synthesizing them in the mega-process of homogeneous hydroformylation. There are a number of reasons to work toward synthesizing these oxygenates in a heterogeneous process like FT and we detail these summarily here. Firstly, hydroformylation is incredibly energy intensive due to its homogeneous nature, where the reactants, products, and catalyst all coexist in the same phase and must therefore be separated from each other after reaction: FT does not suffer nearly as severely in this regard due to its reactants, products, and catalysts largely existing in different phases (gas, gas and/or liquid, and solid, respectively). Secondly, typical hydroformylation catalysts are made from rare and expensive noble metals (e.g. rhodium): FT cobalt-based catalysts are considerably less rare and expensive. Lastly, hydroformylation requires that sophisticated olefin feed stocks be maintained, which are typically produced via petroleum cracking, making hydroformylation inextricably reliant on the extraction and refinement of crude oil: FT, on the other hand, uses synthesis gas, or "syngas", which can be produced from any number of renewable or fossil resources. Despite these drawbacks, hydroformylation remains an industry standard due to its superior activity and selectivity, which are simply much too great to be offset by the major drawbacks summarized above. This therefore motivates the design of oxygenate-driven FT catalysts that are active and selective enough to compete with hydroformylation; and in order to do this in a rational way (i.e. beyond the typical guess-and-check methodology), we must determine what factors lead to increased selectivity and activity in oxygenate-driven FT catalysts.

This dissertation concerns itself primarily with perhaps the first and arguably most important factor in the performance of catalysts: its structure/environment. By this we mean the configuration of the catalyst's components along with that of the reactants and intermediates in the catalytically active phase. This information must be known before any reactions can be elucidated from a fundamental perspective because exploration of the potential reaction network of FT will not yield meaningful results if the structure of the catalyst is merely "guessed at". Indeed, the absolute and relative rates of elementary reaction steps within the reaction network—which manifest as activity and selectivity, respectively—are determined in major part by the composition and configuration of the catalytically active phase. Briefly, since the discovery of the FT reaction in 1926 by the eponymous Fischer and Tropsch[1], the reaction mechanism of the FT reaction has been in continual debate. Whether the typical mechanisms are at play in oxygenate-driven FT is not explored within this dissertation, but we argue that this question cannot be addressed without knowing what exactly these catalysts look like at an atomistic level in the first place.

1.2 The Cobalt-Copper Oxygenate-Driven Fischer-Tropsch Catalyst

While oxygenate synthesis was observed as a minority product in the original reactions performed by Fischer and Tropsch[1], it was not until the *Institut Francais du Petrol* (IFP) developed a CoCu-based FT catalyst that they were specifically targeted for synthesis[2, 3]. The rationale for mixing cobalt with copper was the assumption that one could combine the chain-lengthening properties of cobalt with the alcohol synthesis properties of copper. Despite appearing naïve in its conception, the IFP CoCu catalyst was reportedly successful in producing long-chain terminal alcohols; at least up to C_6. In terms of its success, the IFP asserted that intimate contact between Co and Cu was achieved in their preparation and essential to the performance of their catalyst. However, thermodynamic phase diagrams show that Co and Cu are immiscible (with a maximum of ~6% mixing being achievable)[4], making this assertion quite curious. However, regardless of why the IFP catalyst works, their process was unfortunately not good enough to replace hydroformylation as the primary method for oxygenate synthesis. This failure was due at least in part to the inability of the IFP CoCu-based catalyst to synthesize past C_6. It was therefore rather encouraging when, more recently, the Kruse group was able to demonstrate oxygenate synthesis past C_6 and to arbitrary chain length (depending on choice of pressure and temperature) using a new formulation of CoCu-based catalysts[5-9].

Despite the encouraging results of the Kruse group, we still do not know *why* Cu increases selectivity to oxygenates in any of these CoCu-based catalysts. To determine this, we need to know what the surfaces of CoCu-based catalysts look like under reaction conditions. That is, we need to know what the composition and configuration of the catalytically active phase is. This is especially important for CoCu catalysts since experimental investigations have shown significant restructuring upon exposure to CO and/or syngas[10-14]. Based on simple surface energy arguments (based on Cu having a lower surface energy), and atom probe tomography of CoCuMn catalysts[5], we know that CoCu catalysts will have a tendency to form Co@Cu core-shell nanoparticles. In particular, as-prepared Cu@Co core-shell nanoparticles (i.e. the inverse of its preferred morphology) have been shown to experience significant, and largely irreversible Cu enrichment upon exposure to $O_{2[10, 11]}$.Exposure to similar CoCu catalysts results in Co enrichment at the surface of these nanoparticles[12, 13], and perhaps even more interesting, it has been shown that exposure to syngas ultimately depletes CoCu catalysts entirely of Co, leaving behind a Cu framework[14]. This is not observed with exposure to only CO or H_2. Combined with the fact that NiCu catalysts suffer similar Ni depletion ("valorization") due to gaseous Ni tetracarbonyl, $Ni(CO)_4$, formation[15, 16], it can be posited that this Co depletion could be due to the formation of cobalt's analogous gaseous carbonyl complex: cobalt tetracarbonyl hydride, $Co(CO)_4H$. $Co(CO)_4$ is not a stable gas phase complex making valorization unlikely with exposure to CO alone. Such results demonstrate that CoCu is highly susceptible to reconstruction upon exposure to an FT environment. This means that if one wishes to explore reaction pathways on CoCu, accurately modeling the catalytically active phase is of paramount importance since it is liable to change upon exposure to reactants. In chapters three through five of this dissertation, the structure of CoCu is explicitly explored as a function of CO coverage and pressure. We find that CO induces a significant reversal of the segregation tendency on both flat and stepped surfaces, but that at experimentally relevant temperature and pressures, the Co surface concentration is more limited than would have been expected. This limitation is due to the formation of cobalt subcarbonyls, or $Co(CO)_x$ species. We also demonstrate that these $Co(CO)_x$ species are capable of forming at and rupturing from step sites, diffusing across the surface, and dimerizing on CoCu terraces: the precise ingredients needed to explain large-scale morphological reconstruction of CoCu catalysts.

1.3 Lattice Gas Cluster Expansions

Chapters six through eight of this dissertation are concerned with the reformulation and implementation of multi-component lattice gas (LG) cluster expansions (CEs)[17-21]. The primary motivation here is to create a systematic capability to quantitatively compare systems of interest to FT. Typical models of the cobalt catalyst surface often assume the environment in which the FT elementary reaction steps take place based on varying degrees of information available from experiment. At one end is the assumption that the reactions take place at metallic sites in the absence of spectators (i.e. at the "low coverage" limit), an assumption completely at odds with experiments that routinely show evidence for a "crowded" surface[22-24]. At the other end, some computational work incorporates coverage effects, but usually in a limited manner—for instance, including self-coverage dependence of CO and H binding energies, but for no other species nor dependent on other species[25]. Regardless of incorporation of binding energy coverage dependence, to our knowledge transition states are always obtained at the low coverage limit. This is in large part due to the computational cost of obtaining transition state energies and the difficulty in identifying what kinds of environments would be relevant enough to warrant such an expense. This is unfortunate, because the presence or participation of other reaction intermediates can be expected to alter the stability of one or all of the initial states, final states, and/or transition states. As a result, there is a significant need to identify relevant configurations of reactants and intermediates in a methodologically consistent manner.

LG CEs have the potential to provide a means of describing lateral interaction between arbitrarily complex configurations of adspecies. From these LG CEs, Monte Carlo simulations can be performed to identify stable phases potentially relevant enough to warrant new, altered transitions state calculations. The issue here is that all software available for this endeavor force the user into using the Ising variable paradigm[26-29]. In chapter six of this dissertation, a reformulation of the existing multi-component CE formalism[17-21] is established and a thorough review of the CE formulation available in the literature is provided. With significant geometric relaxations away from ideal lattice positions present in surface-adsorbate systems and large data sets using varying supercell sizes and shapes, errors can be expected that may affect the convergence and reliability of CEs[20, 21, 30]. A method for quantifying these errors is presented in chapter seven. Creation of LG CEs which incorporate this new formalism and error estimation technique is finally presented in chapter eight and used to compare the CO/Co(0001) and CO/Co(111) systems. These systems are important to the FT reaction because below ~20 nm, cobalt nanoparticles experience a phase change from its native hexagonal close packed (hcp) to face centered cubic (fcc)[31-36], and this change is associated with reduced activity[37-39]. Our work shows that this may be due in part to surface strain induced by the smaller lattice constant of the fcc crystal. This strain is shown to be associated with significantly larger short-range lateral interactions, which may be responsible for disrupting important surface phases of CO in FT.

REFERENCES

[1] F. Fischer, Tropsch, H. , Brennstoff-Chemie 7 (1926) 319-344.

[2] A. Sugier, E. Freund, US 4,122,110, (1978)

[3] D. D. P. Courty, E. Freund, A. Sugier, J. Mol. Catal. 17 (1982) 241-254.

[4] T. Nishizawa, K. Ishida, Bull. Alloy Phase Diagr. 5 (1984) 161-165.

[5] Y. Xiang, V. Chitry, P. Liddicoat, P. Felfer, J. Cairney, S. Ringer, N. Kruse, J. Am. Chem. Soc. 135 (2013) 7114-7117.

[6] Y. Xiang, R. Barbosa, N. Kruse, ACS Catal. 4 (2014) 2792-2800.

[7] Y. Xiang, R. Barbosa, X. Li, N. Kruse, ACS Catal. 5 (2015) 2929-2934.

[8] Y. Xiang, N. Kruse, Nature Communications 7 (2016) 13058.

[9] J. M. Voss, Y. Xiang, G. Collinge, D. E. Perea, L. Kovarik, J.-S. McEwen, N. Kruse, Top. Catal. 61 (2018) 1016-1023.

[10] S. K. Beaumont, S. Alayoglu, V. V. Pushkarev, Z. Liu, N. Kruse, G. A. Somorjai, Farad. Discuss. 162 (2013) 31-44.

[11] S. Alayoglu, S. K. Beaumont, G. Melaet, A. E. Lindeman, N. Musselwhite, C. J. Brooks, M. A. Marcus, J. Guo, Z. Liu, N. Kruse, G. A. Somorjai, J. Phys. Chem. C 117 (2013) 21803-21809.

[12] M. L. Smith, N. Kumar, J. J. Spivey, J. Phys. Chem. C 116 (2012) 7931-7939.

[13] N. D. Subramanian, C. S. S. R. Kumar, K. Watanabe, P. Fischer, R. Tanaka, J. J. Spivey, Catal. Sci. Tech. 2 (2012) 621-631.

[14] S. Carenco, A. Tuxen, M. Chintapalli, E. Pach, C. Escudero, T. D. Ewers, P. Jiang, F. Borondics, G. Thornton, A. P. Alivisatos, H. Bluhm, J. Guo, M. Salmeron, J. Phys. Chem. C 117 (2013) 6259-6266.

[15] D. B. Liang, G. Abend, J. H. Block, N. Kruse, Surf. Sci. 126 (1983) 392-396.

[16] V. K. Medvedev, R. Börner, N. Kruse, Surf. Sci. 401 (1998) L371-L374.

[17] J. M. Sanchez, F. Ducastelle, D. Gratias, Physica A: Statistical Mechanics and its Applications 128 (1984) 334-350.

[18] J. M. Sanchez, Phys. Rev. B 48 (1993) 14013-14015.

[19] J. M. Sanchez, Phys. Rev. B 81 (2010) 224202.

[20] J. M. Sanchez, Journal of Phase Equilibria and Diffusion 38 (2017) 238-251.

[21] J. M. Sanchez, Phys. Rev. B 95 (2017) 216202.

[22] J. Schweicher, A. Bundhoo, N. Kruse, J. Am. Chem. Soc. 134 (2012) 16135-16138.

[23] J. P. den Breejen, P. B. Radstake, G. L. Bezemer, J. H. Bitter, V. Frøseth, A. Holmen, K. P. d. Jong, J. Am. Chem. Soc. 131 (2009) 7197-7203.

[24] W. T. Ralston, G. Melaet, T. Saephan, G. A. Somorjai, Angew. Chem. Int. Ed. 56 (2017) 7415-7419.

[25] P. van Helden, J.-A. v. d. Berg, M. A. Petersen, W. Janse van Rensburg, I. M. Ciobîcă, J. van de Loosdrecht, Farad. Discuss. 197 (2017) 117-151.

[26] A. van de Walle, M. Asta, G. Ceder, Calphad 26 (2002) 539-553.

[27] A. van de Walle, G. Ceder, Journal of Phase Equilibria 23 (2002) 348.

[28] A. v. d. Walle, M. Asta, Modell. Simul. Mater. Sci. Eng. 10 (2002) 521.

[29] D. Lerch, O. Wieckhorst, G. L. W. Hart, R. W. Forcade, S. Müller, Modell. Simul. Mater. Sci. Eng. 17 (2009) 055003.

[30] A. H. Nguyen, C. W. Rosenbrock, C. S. Reese, G. L. W. Hart, Phys. Rev. B 96 (2017) 014107.

[31] O. Kitakami, H. Sato, Y. Shimada, F. Sato, M. Tanaka, Phys. Rev. B 56 (1997) 13849-13854.

[32] O. S. Edwards, H. S. Lipson, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 180 (1942) 268-277.

[33] C. G. Granqvist, R. A. Buhrman, J. Appl. Phys. 47 (1976) 2200-2219.

[34] S. Gangopadhyay, G. C. Hadjipanayis, C. M. Sorensen, K. J. Klabunde, IEEE Trans. Magn. 28 (1992) 3174-3176.

[35] E. Anno, Phys. Rev. B 50 (1994) 17502-17506.

[36] J. L. Maurice, J. Briático, J. Carrey, F. Petroff, L. F. Schelp, A. Vaurès, Philos. Mag. A 79 (1999) 2921-2934.

[37] D. I. Enache, B. Rebours, M. Roy-Auberger, R. Revel, J. Catal. 205 (2002) 346-353.

[38] M. Sadeqzadeh, H. Karaca, O. V. Safonova, P. Fongarland, S. Chambrey, P. Roussel, A. Griboval-Constant, M. Lacroix, D. Curulla-Ferré, F. Luck, A. Y. Khodakov, Catal. Today 164 (2011) 62-67.

[39] M. K. Gnanamani, G. Jacobs, W. D. Shafer, B. H. Davis, Catal. Today 215 (2013) 13-17.

CHAPTER TWO:
METHODOLOGY

In this section, the methods used in this dissertation are enumerated. First, the general underlying theory that underpins the majority of the research, quantum mechanics, is curated so that only the details most pertinent to the work are presented. Next, a detailed overview of the statistical mechanical treatment applied to the calculations is provided. Each contribution is treated separately and explained in detail. Finally, the workhorse theory of this dissertation, density functional theory, is explained in detail along with its implementation in the specific software used to perform the research herein.

2.1 General Theory

The central tool used in the theoretical calculations presented in this dissertation is quantum mechanics, the foundation of which is the Schrödinger Equation. The most generalized version of the Schrödinger Equation is time-dependent, but since we are not interested in describing the transient behavior of chemical systems at the level of quantum mechanics, we immediately turn to the time-independent Schrödinger Equation, which is shown in Equation 2.1 below:

$$\widehat{\mathrm{H}}\psi_{\mathrm{n}} = \mathrm{E}_{\mathrm{n}}\psi_{\mathrm{n}} \tag{2.1}$$

where $\widehat{H}$ is the Hamiltonian operator, ψ_n is the wavefunction in its n^th excitation state, and E_n is the energy of that excitation state. In its most generalized version, the Hamiltonian operator can be written as the sum of the kinetic energy operator, $\hat{T}$, and potential energy operator $\hat{V}$. The kinetic energy operator is established from quantum mechanics, but the potential energy operator varies depending on the specific interparticle interactions present within the system being modeled. Explicitly, this is written as

$$\left(-\frac{\hbar^2}{2}\sum_{i}^{N}\left[\frac{1}{m_i}\nabla_i^2\right]+\hat{V}(\{\mathbf{r}\})\right)\psi_n(\{\mathbf{r}\})=E_n\psi_n(\{\mathbf{r}\}) \tag{2.2}$$

where $\hbar$ is the reduced Planck constant, m_i is the mass of the i^{th} particle in the system, ∇_i^2 is the Laplacian of the i^{th} particle, and $\{\mathbf{r}\}$ is the set of all vector positions for each of the N particles in the system (3N total spatial components and N total spin components). Note that we have also made explicit the functional dependence of the wavefunction on particle positions. The Schrödinger Equation is an eigenvalue problem. Given a description of the potential energy for any arrangement of the N particles and set of boundary conditions, a corresponding set of $\psi_n(\{\mathbf{r}\})$ and E_n can be found—albeit rarely as an exact analytical solution, but numerical solutions can often be found to good approximation.

The basic assumption of all the quantum chemical calculations performed in this dissertation is that the wavefunction describing the state of any chemical system can be decoupled and partitioned into some combination of nuclear, electronic, vibrational, rotational, and/or translational contributions. The majority of this decoupling is made possible by invoking the well-known and widely-accepted Born-Oppenheimer (BO) approximation which states that each nucleus in a system can be viewed as fixed within the reference frame of the system's electrons. As nuclei are significantly more massive than electrons (by at least three orders of magnitude), their motion can be shown to be essentially decoupled mathematically. Physically, this means that if the nuclei of a chemical system are at some position $\mathbf{R_0}$ and the electrons are at their simultaneous minimum energy states around them, for any small displacement of the nuclei, d**R**, the electrons will never "lag behind,"— that is, they will always be capable of moving to their new minimum energy states around the nuclei at the new position $\mathbf{R_0} + \mathrm{d}\mathbf{R}$. Because of the BO approximation, the nuclei can be viewed as independent of the electrons and the nuclear wavefunction thus decoupled from any remaining contributions. The electronic wavefunction and energy can then be found for any arrangement of nuclei, including arrangements that trace out the path of the nuclei as they translate (i.e. move through space), vibrate, or rotate. In this case, the electronic wavefunction is usually described as *parameterized* by nuclei positions, and its energy solution then plays the role of the potential in the (adiabatic) BO time-independent Schrödinger equation:

$$\left(-\frac{\hbar^2}{2}\sum_{j}^{N_{nuc}}\left[\frac{1}{m_j}\nabla_j^2\right] + E_{elec}(\{\mathbf{R}\})\right)\psi_{tot}(\{\mathbf{R}\}) = E_{tot}\psi_{tot}(\{\mathbf{R}\}) \qquad (2.3)$$

where m_j is the mass of the j^{th} nucleus, N_{nuc} is the number of nuclei in the system, $E_{elec}(\{\mathbf{R}\})$ is the electronic energy found, in principle, from the solution of Equation 2.2 wherein the nuclei positions, $\{\mathbf{R}\}$, are held constant (which eliminates the kinetic energy terms for the nuclei), $\{\mathbf{r}\}$ is the set of electron positions, $\psi_{tot}(\{\mathbf{R}\})$ is the total BO time-independent wavefunction (only a function of nuclei positions now), and E_{tot} is its total energy. $E_{elec}(\{\mathbf{R}\})$ defines a potential energy surface (PES) of $3N_{nuc}$ degrees of freedom. We note for thoroughness' sake that $\psi_{tot}(\{\mathbf{R}\})$ implicitly contains the quantum nuclear state wavefunction.

Strictly speaking, the BO approximation does not justify the decoupling of translations, vibrations, and rotations of nuclear motion; however, assuming these contributions to be mutually decoupled is a common and reasonable approximation at the level of theory employed in this dissertation. In this case, the subsets of $3N_{nuc}$ degrees of freedom corresponding to translations, vibrations, and rotations can all be assigned a separate PES (solutions of $E_{elec}(\{\mathbf{R}\})$). If the nuclei are in a local minimum on these PES, the electronic energy of this state can be separated as well, and the remaining degrees of freedom described either explicitly or implicitly. These considerations result in a total, time-independent wavefunction, ψ_{total}, and total energy, $\mathrm{E}_{\mathrm{total}}$, of

$$\psi_{\mathrm{total}} = \psi_{\mathrm{nuc}}\psi_{\mathrm{elec}}\psi_{\mathrm{trans}}\psi_{\mathrm{vib}}\psi_{\mathrm{rot}} \tag{2.4.1}$$

$$\mathrm{E}_{\mathrm{total}} = \mathrm{E}_{\mathrm{nuc}} + \mathrm{E}_{\mathrm{elec}} + \mathrm{E}_{\mathrm{trans}} + \mathrm{E}_{\mathrm{vib}} + \mathrm{E}_{\mathrm{rot}} \tag{2.4.2}$$

where $\psi_{\mathrm{nuc}}/\mathrm{E}_{\mathrm{nuc}}$, $\psi_{\mathrm{elec}}/\mathrm{E}_{\mathrm{elec}}$, $\psi_{\mathrm{trans}}/\mathrm{E}_{\mathrm{trans}}$, $\psi_{\mathrm{vib}}/\mathrm{E}_{\mathrm{vib}}$, and $\psi_{\mathrm{rot}}/\mathrm{E}_{\mathrm{rot}}$ are the quantum nuclear state, electronic, translational, vibrational, and rotational time-independent wavefunctions/energies, respectively. The wavefunctions in Equation 2.4.1 are acted on only by operators of the subset of $3\mathrm{N}_{\mathrm{nuc}}$ coordinates associated with its degree of freedom (i.e. translations, vibrations, rotations). All wavefunctions can be found explicitly via solution of their own version of Equation 2.3 and thereby we make determining their energetic contributions much easier to accomplish.

2.2 Statistical Mechanical Treatment of Decoupled Contributions

In the absence of added energy (e.g. electromagnetic radiation or heat), the wavefunctions in Equation 2.4.1 that describe a system will remain in their mutual ground states. However, at finite temperatures, their excitation states can become occupied and their energetic contributions relevant to the total energy of the system. The number and distribution of occupied ground and excited states is actually the source of a system's entropy and is thus critically important to the determination of thermodynamic quantities.

Statistical mechanics allows for one to determine meso- and macroscopic thermodynamic and kinetic quantities from the atomistic information derived from Equations 2.2 – 2.3. This is the main goal of many of the calculations and simulations presented in this dissertation as these quantities determine the stability of reactants, reaction intermediates, and products; as well as the reaction rates connecting them. Thankfully, due to our assumption that the contributions described in Equations 2.4 are decoupled, we can treat each contribution separately with the statistical mechanical tools most appropriate for that contribution and then subsequently combine them to produce the final thermodynamic quantities of interest. Note that in the work done in this dissertation, only *equilibrium* statistical mechanics is used.

The central quantity of equilibrium statistical mechanics that must be found for all contributions is the partition function. There are many statistical mechanical "ensembles" to choose from with, in principle, any particular choice being a matter of convenience. However, in order to discuss the most important details and assumptions made, here we will stick to the canonical ensemble, which treats the number of particles, temperature, and spatial extent as constant. The canonical partition function, Q, is given by

$$\mathrm{Q}(\mathrm{N},\mathcal{X},\mathrm{T}) = \sum_{\mathrm{j}} \mathrm{e}^{-\beta \mathrm{E}_{\mathrm{j}}(\mathrm{N},\mathcal{X})} \tag{2.5}$$

where β is the "thermodynamic beta" equivalent *to* $\frac{1}{\mathrm{k_B T}}$, $\mathrm{k_B}$is the Boltzmann constant, T is the absolute temperature, $\mathrm{E_j}(\mathrm{N},\mathcal{X})$ is the energy of quantum state j, and the sum runs over all possible and allowed quantum states for the system. Each term in the summation is proportional to the probability that the system will be in quantum state j; the partition function therefore contains all the probabilistic information of the entire system. Since the work presented in this dissertation is for surfaces, it is also important to note that the spatial extend, $\mathcal{X}$, is often represented by nanoparticle surface area, $\mathcal{A}$, or equivalently the number of surface unit cells, $\mathrm{N_{u.c.}}$, where we define $\mathcal{A} = \mathrm{a_{u.c.}N_{u.c.}}$, with $\mathrm{a_{u.c.}}$ the surface area of a single surface unit cell. This relationship is often stated in terms of the number of adsorption sites, $\mathrm{N_s}$ (see Clark [1]), and is equivalent provided there is one adsorption site per surface unit cell; we choose a more general form by defining the relationship $\mathrm{N}_{\mathrm{S,i}} = \mathrm{n}_{\mathrm{S,i}}\mathrm{N}_{\mathrm{u.c.}}$, where $\mathrm{N}_{\mathrm{S,i}}$ is the number of sites of type i and $\mathrm{n}_{\mathrm{S,i}}$ is the number of sites of type i per surface unit cell. For gas phase species, $\mathcal{X}$ is the more commonly used volume, $\mathcal{V}$, which is the typical canonical variable used in the literature. However, here we will leave the canonical partition function in the more general form to avoid confusion when moving between systems of different spatial dimensions (such as gas phase vs. surface adsorbed phase).It is also important to note that the energy of each quantum state of the system is *extensive*, dependent on the number of particles, N, and the spatial extent, $\mathcal{X}$, of the system. While this has little consequence when evaluating most molecular partition functions, this is highly relevant when having to consider two or more separate systems or when switching to the grand canonical ensemble.

Without some simplification and assumptions, evaluating the partition function in Equation 2.5 is nearly impossible (or incredibly onerous). In many cases, the first assumption is that the Hamiltonian of the system is not only separable into energy contributions from its various degrees of freedom but also energy of each particle or quasi-particle in the system. This means that the energy of quantum state j can be expressed as a sum of particle energies over all particles, each with its own separate set of quantum states to be summed over. This produces the following rearrangement of Equation 2.5:

$$\begin{aligned} \mathrm{Q} &= \sum_{\mathrm{j}_1}\sum_{\mathrm{j}_2}\sum_{\mathrm{j}_3}\cdots\sum_{\mathrm{j}_\mathrm{N}} \mathrm{e}^{-\beta\left[\mathrm{E}_{\mathrm{j}_1}(\mathrm{N},\mathcal{X})+\mathrm{E}_{\mathrm{j}_2}(\mathrm{N},\mathcal{X})+\mathrm{E}_{\mathrm{j}_3}(\mathrm{N},\mathcal{X})+\cdots+\mathrm{E}_{\mathrm{j}_\mathrm{N}}(\mathrm{N},\mathcal{X})\right]} \\ &= \sum_{\mathrm{j}_1}\mathrm{e}^{-\beta[\mathrm{E}_{\mathrm{j}_1}(\mathrm{N},\mathcal{X})]}\sum_{\mathrm{j}_2}\mathrm{e}^{-\beta[\mathrm{E}_2(\mathrm{N},\mathcal{X})]}\sum_{\mathrm{j}_3}\mathrm{e}^{-\beta[\mathrm{E}_{\mathrm{j}_3}(\mathrm{N},\mathcal{X})]}\dots\sum_{\mathrm{j}_\mathrm{N}}\mathrm{e}^{-\beta\left[\mathrm{E}_{\mathrm{j}_\mathrm{N}}(\mathrm{N},\mathcal{X})\right]} \\ &= \mathrm{q}_1\mathrm{q}_2\mathrm{q}_3\dots\mathrm{q}_\mathrm{N} \end{aligned} \tag{2.6}$$

where a unique index, j_1 through j_N is assigned to each particle (molecule) in the system and lowercase q represent the molecular partition functions which are not necessarily identical for every particle of the same type. This is because, even amongst systems of like particles, the summations above must still be restricted to allowable states (fermions, for instance, cannot simultaneously occupy the same quantum state). Further, if the particles are indistinguishable, the permutation of any set of $j_1 - j_N$ states does not produce a new state and thus only one unique combinations of states should be allowed. Thankfully, due to the presence of either configurational or translations degrees of freedom in systems of distinguishable and indistinguishable particles, respectively, we can safely assume all the particles in our systems obey Boltzmann statistics. This assumption says that when the number of quantum molecular states is much, much greater than the number of particles, we can safely ignore restrictions on the summation over allowed states in Equation 2.6 due to their overwhelming scarcity compared to other states. For indistinguishable particles, it is then rather straightforward to account for the overcounting of permuted states since there are N! of them for each type of particle. The unrestricted summations in Equation 2.6 now produce identical molecular partition functions for all particles of the same type (e.g. CO molecules and H_2 molecules) and these can be combined as [2]

$$Q = \frac{[q_a]^{N_a}[q_b]^{N_b} \dots [q_m]^{N_a}}{N_a!\, N_b! \dots N_m!} \text{ (all particles indistinguishable)} \tag{2.7}$$

$$Q = [q_a]^{N_a}[q_b]^{N_b} \dots [q_m]^{N_a} \text{ (all particles distinguishable)} \tag{2.8}$$

where subscripts denote the type of particle. While not explicitly shown, each partition function still depends on temperature, number of particles, and spatial extent. It should also be noted that while Equations 2.7 and 2.8 denote systems where particles are either all indistinguishable or all distinguishable, they can be combined according to the distinguishability of each particle type as needed. Thus, Equations 2.7 and 2.8 can be generalized as

$$Q = Q_a Q_b \dots Q_m \tag{2.9}$$

where each total molecular partition function, Q_i, is

$$Q_i = \frac{[q_i]^{N_i}}{N_i!} \text{ or } Q_i = [q_i]^{N_i} \tag{2.10}$$

depending on whether the i^{th} particle type is indistinguishable or distinguishable, respectively.

With all of the above assumptions clarified, we can now turn to the equations which define the connection between the partition function and thermodynamic state functions that are used in this dissertation, which are true regardless of assumptions made above. Without fanfare, these are:

$$F = -k_B T \ln Q \tag{2.11}$$

$$U = k_B T^2 \frac{\partial(\ln Q)}{\partial T} \tag{2.12}$$

$$S = \frac{U - F}{T} = k_B T \frac{\partial(\ln Q)}{\partial T} + k_B \ln Q \tag{2.13}$$

$$\mu_a = \frac{\partial F}{\partial N_a} = -k_B T \frac{\partial(\ln Q)}{\partial N_a} \tag{2.14}$$

where F is the Helmholtz free energy, U is the internal energy, S is the entropy, and μ_a is the chemical potential of particle type “a”. Transformations to Gibbs free energy, G, and enthalpy, H, can be made via thermodynamic relationships or, as is done in this dissertation, via evaluation of the isothermal-isobaric ensemble partition function, Δ:

$$\Delta = \sum_{\mathcal{V}=0}^{\infty} \mathrm{Q}(\mathrm{N}, \mathcal{V}_{\mathrm{i}}, \mathrm{T}) \mathrm{e}^{-\beta \mathrm{p} \mathcal{V}} \approx \int_{0}^{\infty} \mathrm{Q}(\mathrm{N}, \mathcal{V}, \mathrm{T}) \mathrm{e}^{-\beta \mathrm{p} \mathcal{V}} \mathrm{C} \mathrm{d}\mathcal{V} \tag{2.15}$$

$$\mathrm{G} = -\mathrm{k_B} \mathrm{T} \ln \Delta \tag{2.16}$$

$$\mathrm{H} = \mathrm{k_B} \mathrm{T}^2 \frac{\partial (\ln \Delta)}{\partial \mathrm{T}} \tag{2.17}$$

where p is the total pressure, $\mathcal{V}$ is the volume and C is a proportionality constant with units of inverse volume needed to make the partition function dimensionless. In Equation 2.15, an integral replaces the summation since volume, $\mathcal{V}$, is not actually countable; the proportionality constant C must be inserted to counter the introduction of dimensionality when moving to the integral formulation. We follow the work of Sack [3] and set this to the constant $\mathrm{C} = \beta\mathrm{p}$, which is chosen for mathematical convenience and is otherwise inconsequential in the thermodynamic limit.

In Equations 2.11 – 2.17, the partition function is always found within a logarithm. As a result, when the total partition function Q in Equations 2.7 and 2.8 is substituted into any of these equations, the molecular partition function of each particle type can be easily partitioned into sums of thermodynamic contributions. To demonstrate this, we substitute Equation 2.7 into Equation 2.11 to get

$$\begin{aligned} \mathrm{F} &= -\mathrm{k_B} \mathrm{T} \ln[\mathrm{Q_a} \mathrm{Q_b} \dots \mathrm{Q_m}] \\ &= (-\mathrm{k_B} \mathrm{T} \ln[\mathrm{Q_a}]) + (-\mathrm{k_B} \mathrm{T} \ln[\mathrm{Q_b}]) + \cdots + (-\mathrm{k_B} \mathrm{T} \ln[\mathrm{Q_m}]) \\ &= F_a + F_b + \cdots + F_m \end{aligned} \tag{2.18}$$

where we can see that the total Helmholtz free energy, F, can be written as a sum over the Helmholtz free energies of each particle type—again, typically a molecular species. This result allows us to consider the molecular partition function for each particle type separately, determine its thermodynamic contribution(s), and then perform the summation at the end according to stoichiometry or other reaction-specific considerations, which is incredibly useful.

Because of the first assumption we have made concerning the separability of degrees of freedom, each molecular partition function, q, in Equation 2.8 can be found by evaluating the partition functions of each of its constituent degrees of freedom as

$$\mathrm{q} = \mathrm{q}_{\mathrm{nuc}}\mathrm{q}_{\mathrm{elec}}\mathrm{q}_{\mathrm{vib}}\mathrm{q}_{\mathrm{rot}}\mathrm{q}_{\mathrm{trans}} \tag{2.19}$$

which can be substituted into Equation 2.10 to give the total partition function of the i^{th} particle type as

$$\mathrm{Q}_{\mathrm{i}} = \frac{\left[\mathrm{q}_{\mathrm{i,nuc}}\mathrm{q}_{\mathrm{i,elec}}\mathrm{q}_{\mathrm{i,vib}}\mathrm{q}_{\mathrm{i,rot}}\mathrm{q}_{\mathrm{i,trans}}\right]^{\mathrm{N_i}}}{\mathrm{N_i}!} \text{ (indistinguishable particle)} \tag{2.20.1}$$

or

$$\mathrm{Q}_{\mathrm{i}} = \left[\mathrm{q}_{\mathrm{i,nuc}}\mathrm{q}_{\mathrm{i,elec}}\mathrm{q}_{\mathrm{i,vib}}\mathrm{q}_{\mathrm{i,rot}}\mathrm{q}_{\mathrm{i,trans}}\right]^{\mathrm{N_i}} \text{ (distinguishable particle)} \tag{2.20.2}$$

To be explicit about how the preceding equations are used, we note that, in this dissertation, the only “particle types” that are distinguishable are the adsorbates in lattice gases. In this case, the particles are considered fixed at lattice points and thus have no translational degrees of freedom. However, the adsorbates gain *configurational* degrees of freedom, which play a similar role. We thus have two separate total partition functions for the i^{th} particle type as

$$\begin{aligned}\mathrm{Q}_{\mathrm{i}} &= \left[\mathrm{q}_{\mathrm{i,nuc}}\right]^{\mathrm{N_i}}\left[\mathrm{q}_{\mathrm{i,elec}}\right]^{\mathrm{N_i}}\left[\mathrm{q}_{\mathrm{i,vib}}\right]^{\mathrm{N_i}}\left[\mathrm{q}_{\mathrm{i,rot}}\right]^{\mathrm{N_i}}\frac{\left[\mathrm{q}_{\mathrm{i,trans}}\right]^{\mathrm{N_i}}}{\mathrm{N_i}!}\\ &= \mathrm{Q}_{\mathrm{i,nuc}}\mathrm{Q}_{\mathrm{i,elec}}\mathrm{Q}_{\mathrm{i,vib}}\mathrm{Q}_{\mathrm{i,rot}}\mathrm{Q}_{\mathrm{i,trans}}\end{aligned} \tag{2.21}$$

$$\mathrm{Q}_{\mathrm{i,lg}} = \left[\mathrm{q}_{\mathrm{i,nuc}}\right]^{\mathrm{N_i}}\left[\mathrm{q}_{\mathrm{i,elec}}\right]^{\mathrm{N_i}}\left[\mathrm{q}_{\mathrm{i,vib}}\right]^{\mathrm{N_i}}\left[\mathrm{q}_{\mathrm{i,rot}}\right]^{\mathrm{N_i}}\mathrm{q}_{\mathrm{i,config}} \tag{2.22}$$

$$= Q_{i,nuc} Q_{i,elec} Q_{i,vib} Q_{i,rot} Q_{i,config}$$

where $Q_{i,lg}$ is the lattice gas (distinguishable particles) total partition function for its i^{th} component. Note that we have also made explicit (1) the important association of the permutation-correcting $N_i!$ with the translational molecular partition function in Equation 2.21, and (2) the fact that the configurational partition function is typically assessed for all N_i particles at once in Equation 2.22.

We now turn our attention to how each contribution is treated in this dissertation.

2.2.1 The Nuclear Contribution

While the quantum nuclear state wavefunction can be very complicated (and firmly in the purview of particle physics) its energetic contribution is typically unimportant to chemical systems. This is because we are only interested in energy *differences* in chemical systems. That is, while E_{elec}, E_{trans}, E_{vib}, and E_{rot} may (and likely will) change over the course of a chemical transformation, E_{nuc} will not. This means E_{nuc} will always cancel out in any subsequent calculations, and we can safely ignore the nuclear terms in all the preceding partition function equations.

2.2.2 The Electronic Contribution

It is not an exaggeration to say that the electronic wavefunction and energy define the vast majority of research efforts in computational chemistry. This is certainly the case in this dissertation where most computational resources have been used to obtain the electronic energy in Equation 2.4. There is no simple method for creating a closed form partition function for electronic energies either, meaning that the summation over electronic states in Equation 2.6 must, in principle, be performed manually. Mercifully, electronic excitation states tend to be high in energy (with respect to the lowest electronic state) and are thus negligible at the temperatures considered in this dissertation. Therefore, we can make a very good approximation of the electronic partition function as a “sum” over a single state, i.e. its ground state:

$$q_{i,elec} = \sum_{j=0}^{0} e^{-\beta E_{i,0}} = e^{-\beta E_{i,0}} \tag{2.23}$$

$$Q_{i,elec} = \left[e^{-\beta E_{i,0}}\right]^{N_i} \tag{2.24}$$

Equation 2.23 can be plugged into the thermodynamic Equations 2.11 – 2.17 to produce very simple expressions for the electronic contribution to them:

$$\begin{gathered} F_{i,elec} = -k_B T \ln\left[e^{-\beta E_{i,0}}\right]^{N_i} \\ \Rightarrow F_{i,elec} = N_i E_{i,0} \end{gathered} \tag{2.25}$$

$$\begin{gathered} U_{i,elec} = k_B T^2 \frac{\partial\left(\ln\left[e^{-\beta E_{i,0}}\right]^{N_i}\right)}{\partial T} \\ \Rightarrow U_{i,elec} = N_i E_{i,0} \end{gathered} \tag{2.26}$$

$$S_{i,elec} = \frac{U - F}{T} = 0 \tag{2.27}$$

$$\begin{gathered} \mu_{i,elec} = \frac{\partial F_{i,elec}}{\partial N_i} = -k_B T \frac{\partial(\ln Q)}{\partial N_i} \\ \Rightarrow \mu_{i,elec} = E_{i,0} \end{gathered} \tag{2.28}$$

$$\Delta_{i,elec} = \int_0^\infty \left[e^{-\beta E_{i,0}}\right]^{N_i} e^{-\beta p \mathcal{V}} \beta p d\mathcal{V} = \left[e^{-\beta E_{i,0}}\right]^{N_i} \tag{2.29}$$

$$G_{\mathrm{i,elec}} = -\mathrm{k_B T} \ln\left[\mathrm{e}^{-\beta \mathrm{E_{i,0}}}\right]^{\mathrm{N_i}}$$
$$\Rightarrow G_{\mathrm{i,elec}} = \mathrm{N_i E_{i,0}} \tag{2.30}$$

$$H_{\mathrm{i,elec}} = \mathrm{k_B T^2} \frac{\partial\left(\ln\left[\mathrm{e}^{-\beta \mathrm{E_{i,0}}}\right]^{\mathrm{N_i}}\right)}{\partial \mathrm{T}}$$
$$\Rightarrow H_{\mathrm{i,elec}} = \mathrm{N_i E_{i,0}} \tag{2.31}$$

where we see that in the absence of any volume dependence in $\mathrm{Q_{i,elec}}$, the Gibbs free energy and enthalpy are in fact equivalent to the Helmholtz free energy and internal energy, respectively. This is a general result that will be replicated for any contribution without a volume dependence.

The ground state electronic energy in all the calculations performed in this dissertation were computed using density functional theory (DFT) as implemented in the Vienna *ab initio* Simulation Package (VASP).[4-6] Due to the central role it plays in this dissertation, DFT will be discussed in its own section. An electronic ground state energy can be calculated, in principle, for any set of nuclei positions with all energies for each possible set of nuclei positions (a 3N-dimensional space) making up the systems potential energy surface (PES). However, when specific "structures" made up of the same N nuclei are considered, we are referring specifically to one of the local energy minima in that 3N-dimensional PES. We find (and thus define) these structures by providing an initial guess for the nuclei positions and then running a variety of well-known force minimization procedures included in the VASP software. All structures shown in the figures included in this dissertation are the final result of this process unless otherwise stated. We will denote the nuclei positions of each structure as $\mathbf{R_0}$ in the following sections.

2.2.3 The Translational Contribution

The translational contribution to the total energy is the simplest contribution to determine so long as ideal gas (either 3D or 2D) assumptions hold, and one of the most difficult contributions to determine otherwise. In this dissertation, all gas phase molecules are assumed to remain in ideal gas conditions and adsorbate translations are treated as *either* free translators (ideal two-dimensional gas) *or* as vibrations due to confinement to a potential energy well. We deal with the ideal gas translational contributions in this subsection.

In principle, one can determine the energy states of a system of non-interacting freely-translating particles from direct solution of Equation 2.2 or 2.3 and application of Equation 2.5. While this is often instructive, in this dissertation, the high-T limit is always imposed on the translational partition function (where it is unequivocally the most justified due to vanishingly small translational energy spacings). It is thus much more relevant to start from the classical partition function which is equivalent to Equation 2.5 in the high-T limit. The classical partition function is an integral over all "phase space" which is defined by all possible momenta and positions of particles in a system as

$$\mathrm{Q} = \frac{1}{\mathrm{h^S}} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \int_{\text{all space}} \dots \int_{\text{all space}} \mathrm{e}^{-\beta\mathcal{H}(\boldsymbol{p},\boldsymbol{q})} \mathrm{d}p_{1,1} \dots \mathrm{d}p_{\mathrm{N,S}}\, \mathrm{d}q_{1,1} \dots \mathrm{d}q_{\mathrm{N,S}} \tag{2.32}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are the vectors of, respectively, single-dimension momenta and positions (in Cartesian coordinates) of the N particles in the system of S spatial dimensions, and $\mathcal{H}(\mathbf{p},\mathbf{q})$ is the classical Hamiltonian. Each of the 2SN integrals run over either all of momentum or real space. While the extent of momentum space is easily defined outright ($-\infty$ to ∞), real space is defined by the problem under consideration. The factor of $\frac{1}{\mathrm{h^S}}$ converts the classical partition function from classical phase space to quantum phase space [2] and is needed here to make results using Equation 2.32 correspond to those using Equation 2.5 in the high-T limit. From here, we will assume three dimensions and use x, y, and z to specify them. Our assumption that the Hamiltonian can be described by a summation over individual particle contributions has the same effect on Equation 2.32 that it had on Equation 2.5 to yield Equation 2.6. Thus, we can write the molecular classical partition function as

$$\mathrm{q} = \frac{1}{\mathrm{h}^3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \iiint_{\text{all space}} \mathrm{e}^{-\beta\mathcal{H}(p_x,p_y,p_z,\mathrm{x},\mathrm{y},\mathrm{z})} \mathrm{d}p_x \mathrm{d}p_y \mathrm{d}p_z \mathrm{dxdydz} \tag{2.33}$$

where integration over "all space" is a conceptual shorthand for integration over real-space coordinates up to whatever length(s) or volume is relevant to the problem at hand.

To apply Equation 2.33 to the translational contribution, we must define the Hamiltonian and the limits of integration over all space. For non-interacting particles freely moving through space, no potential energy is gained or lost by each particle and only kinetic energy remains:

$$\mathcal{H}(p_x, p_y, p_z, \mathrm{x}, \mathrm{y}, \mathrm{z}) = \frac{p_x^2}{2\mathrm{m_i}} + \frac{p_y^2}{2\mathrm{m_i}} + \frac{p_z^2}{2\mathrm{m_i}} \tag{2.34}$$

where $\mathrm{m_i}$ is the (total) mass of the $\mathrm{i^{th}}$ particle or molecule. Because Equation 2.34 does not depend on particle positions, the integration over "all space" will now simply integrate up to the volume of the system, $\mathcal{V}$. Combining this and Equation 2.34, Equation 2.33 becomes

$$\begin{aligned} \mathrm{q_{trans}} &= \frac{1}{\mathrm{h}^3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathrm{e}^{-\beta\left[\frac{p_x^2}{2\mathrm{m_i}} + \frac{p_y^2}{2\mathrm{m_i}} + \frac{p_z^2}{2\mathrm{m_i}}\right]} \mathrm{d}p_x \mathrm{d}p_y \mathrm{d}p_z \iiint_{\mathcal{V}} \mathrm{dxdydz} \\ &= \left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \mathcal{V} \end{aligned} \tag{2.35}$$

For distinguishable particles this is the final result. However, free translation implies that the particles are ultimately indistinguishable, and we must account for the factorial associated with the translational partition function shown in Equation 2.21. Specifically, this is

$$\mathrm{Q_{i,trans}} = \frac{[\mathrm{q_{trans}}]^{\mathrm{N_i}}}{\mathrm{N_i}!} = \frac{1}{\mathrm{N_i}!}\left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \mathcal{V}\right]^{\mathrm{N_i}} \tag{2.36}$$

For values of N of even modestly large size, the Sterling approximation is incredibly accurate and can be used here without reservation. This yields

$$\begin{aligned} \mathrm{Q_{i,trans}} &= \left(\frac{e}{\mathrm{N_i}}\right)^{\mathrm{N_i}}\left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \mathcal{V}\right]^{\mathrm{N_i}} \\ &= \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \frac{\mathcal{V}}{\mathrm{N_i}} e\right]^{\mathrm{N_i}} \end{aligned} \tag{2.37}$$

It is tempting (and common) to replace $\frac{\mathcal{V}}{\mathrm{N}}$ in Equation 2.37 with the ideal gas equivalent $\frac{\mathrm{k_B T}}{\mathrm{P}}$. However, this inevitably leads one to call the resultant free energy the Gibbs free energy instead of the Helmholtz free energy due to the seeming dependence on pressure. This is an incorrect equivalence, and in this dissertation, we opt for a more rigorous treatment of the two quantities that avoids this potential pitfall.

Inserting Equation 2.37 into Equations 2.11 through 2.17 produce the translational contribution to the thermodynamic state functions of the system of interest:

$$\mathrm{F_{i,trans}} = -\mathrm{k_B T} \ln\left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \frac{\mathcal{V}}{\mathrm{N_i}} e\right]^{\mathrm{N_i}} \tag{2.38}$$

$$\Rightarrow \mathrm{F}_{\mathrm{i,trans}} = -\mathrm{k_B T N_i} \ln\left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \frac{\mathcal{V}}{\mathrm{N_i}} e\right]$$

$$\mathrm{U}_{\mathrm{i,trans}} = \mathrm{k_B T}^2 \frac{\partial\left(\ln\left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \frac{\mathcal{V}}{\mathrm{N_i}} e\right]^{\mathrm{N_i}}\right)}{\partial \mathrm{T}} \tag{2.39}$$

$$\Rightarrow \mathrm{U}_{\mathrm{i,trans}} = \frac{3}{2} \mathrm{N_i k_B T}$$

$$\mathrm{S}_{\mathrm{i,trans}} = \frac{\mathrm{U} - \mathrm{F}}{\mathrm{T}} \tag{2.40}$$

$$\mu_{\mathrm{i,trans}} = \frac{\partial \mathrm{F}_{\mathrm{i,elec}}}{\partial \mathrm{N_i}}$$

$$\Rightarrow \mu_{\mathrm{i,trans}} = -\mathrm{k_B T} \ln\left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \frac{\mathcal{V}}{\mathrm{N_i}} e\right] \tag{2.41}$$

The evaluation of the isothermal-isobaric partitions function deserves special attention here since it depends on volume and must be dealt with within the integral over volume.

$$\Delta_{\mathrm{i,trans}} = \int_0^{\infty} \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}} \frac{\mathcal{V}}{\mathrm{N_i}} e\right]^{\mathrm{N_i}} \mathrm{e}^{-\beta \mathrm{p} \mathcal{V}} \beta \mathrm{p} \mathrm{d}\mathcal{V}$$

$$= \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}}\right]^{\mathrm{N_i}} \frac{\beta \mathrm{p}}{\mathrm{N_i}!} \left(\int_0^{\infty} \mathcal{V}^{\mathrm{N_i}} \mathrm{e}^{-\beta \mathrm{p} \mathcal{V}} \mathrm{d}\mathcal{V}\right)$$

$$= \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}}\right]^{\mathrm{N_i}} \frac{\beta \mathrm{p}}{\mathrm{N_i}!} \left(\mathrm{N_i}! \left[\frac{1}{\beta \mathrm{p}}\right]^{\mathrm{N_i}+1}\right) \tag{2.42}$$

$$= \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2}\right)^{\frac{3}{2}}\right]^{\mathrm{N_i}} \left[\frac{1}{\beta \mathrm{p}}\right]^{\mathrm{N_i}}$$

$$= \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2} \right)^{\frac{3}{2}} \frac{\mathrm{k_B}T}{\mathrm{p}} \right]^{\mathrm{N_i}}$$

$$G_{\mathrm{i,trans}} = -\mathrm{k_B T} \ln \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2} \right)^{\frac{3}{2}} \frac{\mathrm{k_B}T}{\mathrm{p}} \right]^{\mathrm{N_i}}$$

$$\Rightarrow G_{\mathrm{i,trans}} = \mathrm{k_B T N_i} \ln \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2} \right)^{\frac{3}{2}} \frac{\mathrm{k_B}T}{\mathrm{p}} \right] \tag{2.43}$$

$$H_{\mathrm{i,trans}} = \mathrm{k_B T^2} \frac{\partial}{\partial \mathrm{T}} \left(\ln \left[\left(\frac{2\pi \mathrm{m_i k_B T}}{\mathrm{h}^2} \right)^{\frac{3}{2}} \frac{\mathrm{k_B}T}{\mathrm{p}} \right]^{\mathrm{N_i}} \right)$$

$$\Rightarrow H_{\mathrm{i,trans}} = \frac{5}{2} \mathrm{N_i k_B T} \tag{2.44}$$

where now it can be seen that there is indeed a difference between the Helmholtz free energy and the Gibbs free energy, albeit not drastically.

2.2.4 The Vibrational Contribution

Unlike the translational contribution, the vibrational contribution has significant quantum effects that should not be ignored. We thus use Equation 2.3 and find the total energy of the vibrational contribution in Equation 2.4.2 by assuming that the electronic energy, $\mathrm{E_{elec}}(\{\mathbf{R}\})$, is harmonic. Explicitly, we assume, at least locally, that the PES of a structure made up of M nuclei can be described by a multidimensional Taylor series centered at some local energetic minimum $\mathbf{R_0}$ and truncated after the 2nd order term:

$$\mathrm{E_{elec}}(\{\mathbf{R}\}) \approx \mathrm{E_{elec}}(\mathbf{R_0}) + \boldsymbol{\nabla}^T(\mathbf{R} - \mathbf{R_0}) + (\mathbf{R} - \mathbf{R_0})^{\mathrm{T}}\mathrm{H}(\mathbf{R} - \mathbf{R_0}) \tag{2.45}$$

where $\mathbf{R}$ is the nuclei position vector of 3M (usually Cartesian) components, $\mathrm{E_{elec}}(\mathbf{R_0})$ is the energy of the structure at $\mathbf{R_0}$, which we set to zero since this value is already associated with the electronic contribution; $\boldsymbol{\nabla}$ is the gradient vector, which is zero since the structure is at an energy minima; H is the Hessian matrix, and $\mathbf{R} - \mathbf{R_0} = \Delta\mathbf{R}$ is the displacement vector. With the first two terms in Equation 2.32 set to zero, the harmonic PES becomes

$$\mathrm{E_{elec}}(\{\mathbf{R}\}) \approx (\Delta\mathbf{R})^{\mathrm{T}}\mathrm{H}(\Delta\mathbf{R}) \tag{2.46}$$

After insertion into the BO time-independent Schrödinger equation (Equation 2.3) and diagonalization of the resultant “mass-weighted” Hessian matrix, Equation 2.46 becomes a system of 3M 1-dimensional harmonic oscillators and the wavefunction and energy associated with each 3M-component eigenvector (the so-called normal mode) can be found analytically. The energy for each normal mode is then the same as for a 1-dimensional quantum harmonic oscillator:

$$\mathrm{E_{vib,k,n}} = \mathrm{h}\nu_{\mathrm{k}}\left(\mathrm{n} + \frac{1}{2}\right), \qquad \mathrm{n} \in \mathbb{N} \tag{2.47}$$

where $\mathrm{E_{vib,k,n}}$ is the energy of the $\mathrm{k^{th}}$ normal mode in its $\mathrm{n^{th}}$ excitation state, ν_{k} is the frequency of the $\mathrm{k^{th}}$ normal mode, and n is a natural number corresponding to the $\mathrm{n^{th}}$ excitation state of the normal mode. The total energy associated with all vibrations is then a simple summation over all normal modes:

$$\mathrm{E_{vib,n_i}} = \sum_{\mathrm{k}=1}^{3\mathrm{M}} \mathrm{h}\nu_{\mathrm{k}}\left(\mathrm{n_k} + \frac{1}{2}\right) \tag{2.48}$$

where the normal mode index is added to the natural number to reflect that the excitation states of each normal mode are independent.

Equation 2.48 can now be inserted into the partition function with the summation running over all possible excitation states:

$$
\begin{aligned}
q_{i,vib} &= \sum_{n=0}^{\infty} e^{-\beta\left[\sum_{k=1}^{3N} h\nu_k\left(n_k+\frac{1}{2}\right)\right]} \\
&= \prod_{k=1}^{3M} e^{-\beta\left[\frac{1}{2}h\nu_k\right]} \sum_{n=0}^{\infty}\left(e^{-\beta[h\nu_k]}\right)^{n_k}
\end{aligned}
\qquad (2.49)
$$

where a few rearrangements reveal a geometric series with constant multiple $e^{-\beta[h\nu_k]}$. This series has a closed form expression that can be inserted into Equation 2.49 to give

$$
q_{i,vib} = \prod_{k=1}^{3N} e^{-\beta\left[\frac{1}{2}h\nu_k\right]}\left(\frac{1}{1-e^{-\beta[h\nu_k]}}\right) \qquad (2.50)
$$

which is a product of 3M 1-dimensional harmonic oscillator partition functions. In principle, only 3M-5, 3M-6, or 3M-3 degrees of freedom actually correspond to proper vibrational degrees of freedom for linear, non-linear, and periodic systems. However, in practice, all 3M degrees of freedom are assessed, and one must manually sift out the erroneous "extra" modes corresponding to rotations and translations. Equation 2.50 can be inserted into Equation 2.21 to give the total vibrational partition function for particle type i as

$$
Q_{i,vib} = \left[\prod_{k=1}^{3M}\left(\frac{e^{-\beta\left[\frac{1}{2}h\nu_k\right]}}{1-e^{-\beta[h\nu_k]}}\right)\right]^{N_i} \qquad (2.51)
$$

and thermodynamic quantities:

$$
F_{i,vib} = -k_B T \ln\left[\prod_{k=1}^{3M}\left(\frac{e^{-\beta\left[\frac{1}{2}h\nu_k\right]}}{1-e^{-\beta[h\nu_k]}}\right)\right]^{N_i} \qquad (2.52)
$$

$$\Rightarrow F_{i,vib} = N_i \sum_{k=1}^{3M} \left[\frac{1}{2} h\nu_k - k_B T \ln\left(\frac{1}{1 - e^{-\beta[h\nu_k]}}\right)\right]$$

$$U_{i,vib} = k_B T^2 \frac{\partial}{\partial T} \ln\left[\prod_{k=1}^{3M}\left(\frac{e^{-\beta\left[\frac{1}{2}h\nu_k\right]}}{1 - e^{-\beta[h\nu_k]}}\right)\right]^{N_i}$$

$$\Rightarrow U_{i,vib} = N_i \sum_{k=1}^{3M} \left[\frac{1}{2} h\nu_k + \frac{h\nu_k}{e^{\beta[h\nu_k]} - 1}\right] \tag{2.53}$$

$$S_{i,vib} = \frac{U - F}{T} \tag{2.54}$$

$$\mu_{i,vib} = \sum_{k=1}^{3M} \left[\frac{1}{2} h\nu_k - k_B T \ln\left(\frac{1}{1 - e^{-\beta[h\nu_k]}}\right)\right] \tag{2.56}$$

$$G_{i,vib} = F_{i,vib} \tag{2.57}$$

$$H_{i,vib} = U_{i,vib} \tag{2.58}$$

where we've taken advantage of the result found previously that the Gibbs free energy and enthalpy are equivalent to the Helmholtz free energy and internal energy, respectively, when there is no volume dependence on the partition function.

2.2.5 The Rotational Contribution

Like the translational contribution, the (eigen)energy states of a system undergoing rotation can be determined from solution of the time independent Schrödinger Equation using Equation 2.2. However, as was the case there, it is unnecessary to do so here since we consistently operate in the “high-T” limit of every system studied. It is thus advantageous (and, it turns out, much more straightforward) to use the classical partition function when considering the general case of a 3-dimensional polyatomic molecule.

While it is possible to determine the rotational partition function from Equation 2.32 in the classical coordinates of phase space, a conceptually simpler approach is to simply realize that for a rotating body of particles in some fixed configuration, the integral in Equation 2.32 can better be represented by an integration over all *angular* momenta (L_x, L_y, L_z) and over all rotational invariant configurations of the molecule. Thus, we can replace the differentials with the molecule's three angular momenta—whose principle moments of inertia (I_x, I_y, I_z) we assume have been identified—and three Euler angles: pitch (θ), yaw (ϕ), and/or roll (χ)). Every configuration is energetically degenerate (i.e. the Hamiltonian is independent of these angles) and therefore the Hamiltonian for the i$^{\mathrm{th}}$ rotating body is given as

$$\mathcal{H}_{\mathrm{i}}\left(L_x, L_y, L_z, \theta, \phi, \chi\right) = \left(\frac{\mathrm{L}_{\mathrm{x}}^2}{2I_x} + \frac{\mathrm{L}_y^2}{2I_y} + \frac{\mathrm{L}_z^2}{2I_z}\right)_i \qquad (2.59)$$

As before, we assume that the Hamiltonian for a collection of bodies can be expressed as a sum over each body's energetic contribution allowing for the partition function to be separated into a product of single-body molecular partition functions. With this consideration, combining Equation 2.59 with integration over all rotationally accessed configurations converts Equation 2.32 to

$$\mathrm{Q_{rot}} = \mathrm{q_{rot}^N} = \left[\frac{1}{\mathrm{h}^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{2\pi}\int_{0}^{2\pi}\int_{0}^{\pi} \mathrm{e}^{-\beta\left[\frac{\mathrm{L_x^2}}{2I_x}+\frac{\mathrm{L}_y^2}{2I_y}+\frac{\mathrm{L}_z^2}{2I_z}\right]} \mathrm{d}L_x \mathrm{d}L_y \mathrm{d}L_z \sin\theta \mathrm{d}\theta \mathrm{d}\phi \mathrm{d}\chi\right]^{\mathrm{N}} \qquad (2.60)$$

where $\sin\theta \mathrm{d}\theta \mathrm{d}\phi \mathrm{d}\chi$ is the integration element for Euler angles. Equation 2.60 is analogous to that found for the translational partition function with the moment of inertia taking the place of mass. We thus make use of that result to arrive at

$$
\begin{aligned}
q_{\mathrm{rot}} &= \left(\frac{2\pi I_x k_B T}{h^2}\right)^{\frac{1}{2}} \left(\frac{2\pi I_y k_B T}{h^2}\right)^{\frac{1}{2}} \left(\frac{2\pi I_z k_B T}{h^2}\right)^{\frac{1}{2}} \int_0^{2\pi}\int_0^{2\pi}\int_0^{\pi} \sin\theta d\theta d\phi d\chi \\
&= \left(\frac{2\pi I_x k_B T}{h^2}\right)^{\frac{1}{2}} \left(\frac{2\pi I_y k_B T}{h^2}\right)^{\frac{1}{2}} \left(\frac{2\pi I_z k_B T}{h^2}\right)^{\frac{1}{2}} \frac{8\pi^2}{\sigma} \quad \text{(nonlinear molecules)}
\end{aligned}
\tag{2.61}
$$

Where σ is the symmetry number used to remove symmetrically equivalent rotations. This is the result for nonlinear molecules.

In the case of linear molecules, there will only be two (identical) principle moments of inertia ($I_x = I_y = I$), which correspond to, say, pitch and yaw. However, rotations involving the Euler angle corresponding to roll produce no new configurations. As a result, we lose one of the integrations over angular momentum as well as the integration over the Euler angle corresponding to roll (χ here).

$$
q_{\mathrm{rot}} = \left(\frac{2\pi I k_B T}{h^2}\right)\frac{4\pi}{\sigma} \quad \text{(linear molecules)} \tag{2.62}
$$

where we have combined the two terms corresponding to two identical moments of inertia and have lost a factor of 2π corresponding to the integration over roll (χ). All molecules with rotational degrees of freedom can be assessed according to either Equation 2.61 or 2.62 (symmetric tops for instance, simply have $I_x = I_y \neq I_z$).

We use the result for linear molecules here to assess thermodynamic functions for convenience:

$$
\begin{aligned}
F_{i,\mathrm{rot}} &= -k_B T \ln\left[\left(\frac{2\pi I k_B T}{h^2}\right)\frac{4\pi}{\sigma}\right]^{N_i} \\
\Rightarrow F_{i,\mathrm{rot}} &= N_i k_B T \ln\left[\left(\frac{2\pi I k_B T}{h^2}\right)\frac{4\pi}{\sigma}\right]
\end{aligned}
\tag{2.63}
$$

$$\mathrm{U_{i,rot}} = \mathrm{k_B T^2} \frac{\partial}{\partial \mathrm{T}} \ln \left[\left(\frac{2\pi I \mathrm{k_B T}}{\mathrm{h^2}} \right) \frac{4\pi}{\sigma} \right]^{\mathrm{N_i}} \tag{2.64}$$

$$\Rightarrow \mathrm{U_{i,rot}} = \mathrm{N_i k_B T}$$

$$\mathrm{S_{i,rot}} = \frac{\mathrm{U - F}}{T} \tag{2.65}$$

$$\mu_{\mathrm{i,rot}} = \mathrm{k_B T} \ln \left[\left(\frac{2\pi I \mathrm{k_B T}}{\mathrm{h^2}} \right) \frac{4\pi}{\sigma} \right] \tag{2.66}$$

$$\mathrm{G_{i,rot}} = \mathrm{F_{i,rot}} \tag{2.67}$$

$$\mathrm{H_{i,rot}} = \mathrm{U_{i,rot}} \tag{2.68}$$

Note that for a nonlinear molecule, $\mathrm{U_{i,rot}} = \frac{3}{2}\mathrm{N_i k_B T}$ in accordance with equipartition of energy.

2.2.6 The Configurational Contribution

The configurational contribution needed for distinguishable systems is not used directly in this dissertation but is included for completeness since lattice gas models are a main constituent herein. The configurational contribution is also important as it plays a similar role in systems of interacting distinguishable particles like lattice gases as the translational contribution does in systems of indistinguishable particles, and it is helpful to make explicit that they are in fact mutually exclusive. That is, a system’s configurational degrees of freedom are sampled via translations should they be present and are thus not needed in those cases while the presence of configurational degrees of freedom suggests a lack of translations.

We note that a closed form expression for the configurational contribution to the thermodynamics of a system is only possible for either ideal lattice gasses or otherwise heavily simplified systems. It is beyond the work of this dissertation, but we note that Monte Carlo, or stochastic, methods are typically employed to effectively sample the partition function of more complicated/real systems (i.e. those with significant lateral interactions leading to nonideal gas behavior). We thus concern ourselves here only with ideal (or otherwise, mean-field) systems and the ideal lattice gas in particular.

In an ideal lattice gas, particles do not interact beyond simple site-exclusion (i.e. no more than one particle can occupy a lattice point at a time). Due to this, all configurations of the same N distinguishable particles are energetically degenerate, so the system energy is fixed. In this case, we could actually operate in the microcanonical ensemble, which is the prototypical isolated system: the number of particles, N, the volume, $\mathcal{V}$, and the energy, E, are constant. However, particle adsorption releases energy, which we denote as V_0, and for a lattice gas, we assume that each adsorbate is trapped in a harmonic energy well which provides vibrational degrees of freedom whose energy levels are accessed as a function of temperature. The fact that we will eventually wish to include a vibrational component to the total thermodynamic description of this means that it is prudent to remain in the canonical ensemble. To help identify the configurational contribution, we rewrite the lattice gas partition function as

$$Q_{\mathrm{l.g.}}(\mathrm{N}, \mathcal{A}, \mathrm{T}) = \sum_{\mathrm{n}=0}^{\infty} \Omega e^{\beta[\mathrm{NV_0} + \mathrm{NE_{n,vib}}]} \tag{2.69}$$

where $\mathcal{A}$ is the total area of the surface (a spatial extent); $\mathrm{NV_0}$ is the total electronic energy of the system; $\mathrm{NE_{n,vib}}$ is the total vibrational energy of the system in its $\mathrm{n^{th}}$ vibrational excitation state, and Ω is the degeneracy of the $\mathrm{NV_0} + \mathrm{NE_{n,vib}}$ energy state and essentially corresponds to the total number of configurations the N adsorbates can take. As before, the vibrational energy contribution here is separable. The vibrational energy term can be collected with the summation as in section 2.2.4, and the (electronic) adsorption energy is constant. This gives

$$Q_{l.g.}(N, \mathcal{A}, T) = \Omega\, Q_{elec} Q_{vib} \tag{2.70}$$

By inspection, Equation 2.70, the configurational contribution to the canonical partition function is Ω. Even though Ω is *technically* a microcanonical partition function, for convenience and clarity, it is useful to insert it into Equation 2.70 as a configurational contribution to the canonical partition function (i.e. $Q_{config}(N, \mathcal{A}, T) = \Omega$) despite the lack of a temperature dependence. We now need only to determine Ω.

If each adsorbate occupies one site only ($n_S = 1$), the total number of configurations that N total particles on that surface can take becomes a simple counting problem given by

$$Q_{config}(N, N_S, T) = \Omega = \binom{N_S}{N} = \frac{N_S!}{N!\,(N_S - N)!} \tag{2.71}$$

where we've made the equivalence between area, $\mathcal{A}$, and surface sites, N_S, explicit. A more useful form of Equation 2.71 is found by invocation of Sterling's approximation and defining a coverage, $\theta = \frac{N}{N_S}$. After some slightly tedious algebra, we find that

$$Q_{config}(N, N_S, T) = \left[\frac{1}{\theta^{\theta}(1-\theta)^{(1-\theta)}}\right]^{N_S} \tag{2.72}$$

This gives the final result for the total ideal lattice gas partition function as

$$Q_{l.g.}(N, \mathcal{A}, T) = \left[\frac{1}{\theta^{\theta}(1-\theta)^{(1-\theta)}}\right]^{N_S} Q_{elec} Q_{vib} \tag{2.73}$$

It is important to note that Equation 2.72 is the result for the *entire* surface and not just a single particle. In this way, the configuration contribution is quite different from the other contribution where molecular partition functions were found and then raised to a power of N_i to form its total contribution to the total partition function. In practice, this seeming dependence on system size is avoided via some normalization of the thermodynamic quantity of interest to a “per site” or “per adsorbate” basis.

The resultant thermodynamic quantities are

$$\mathrm{F_{i,config}} = -\mathrm{k_B T} \ln\left(\left[\frac{1}{\theta^{\theta}(1-\theta)^{(1-\theta)}}\right]^{\mathrm{N_S}}\right)$$

$$\Rightarrow \mathrm{F_{i,config}} = \mathrm{k_B T N_S} \ln\left[\theta^{\theta}(1-\theta)^{(1-\theta)}\right] \tag{2.74}$$

$$\mathrm{U_{i,config}} = \mathrm{k_B T^2} \frac{\partial}{\partial \mathrm{T}} \ln\left(\left[\frac{1}{\theta^{\theta}(1-\theta)^{(1-\theta)}}\right]^{\mathrm{N_S}}\right)$$

$$\Rightarrow \mathrm{U_{i,config}} = 0 \tag{2.75}$$

$$\mathrm{S_{i,config}} = \frac{\mathrm{U-F}}{T} = -\mathrm{k_B T N_S} \ln\left[\theta^{\theta}(1-\theta)^{(1-\theta)}\right] \tag{2.76}$$

$$\mu_{\mathrm{i,config}} = \mathrm{k_B T} \ln\left[\frac{\theta}{1-\theta}\right] \tag{2.77}$$

$$\mathrm{G_{i,config}} = \mathrm{F_{i,config}} \tag{2.78}$$

$$\mathrm{H_{i,config}} = \mathrm{U_{i,config}} \tag{2.79}$$

2.3 Density Functional Theory

We turn now to detailing the underlying principles of density functional theory (DFT). As mentioned in *the electronic contribution* subsection above, DFT is used in this dissertation to find the (approximate) ground state electronic energy of all systems studied. With few exceptions, the electronic contribution makes up the vast majority of the total free or internal energies. It is thus imperative that these energies be computed as accurately as possible given the complexity of the systems we are interested in. However, while the time independent Schrödinger equation (Equation 2.2) can be solved for each of the other relevant contributions mentioned (though we opted to skip this step for most of the contributions), attempting to do so for our systems of interest, which have 100s of nuclei and potentially thousands of electrons, creates a wholly intractable problem.

2.3.1 Hartree-Fock as Prelude to DFT

To provide context for the remaining discussion on DFT, we consider the ideally full quantum mechanical treatment of a system with $\mathrm{N_n}$ nuclei and $\mathrm{N_e}$ electrons. In these systems, Coulomb's law is used to define the potential energy due to electrostatic attractions and repulsions between both nuclei and electrons, and Equation 2.2 becomes

$$\left(-\frac{\hbar^2}{2}\sum_{A}^{\mathrm{N_n}}\left[\frac{1}{\mathrm{m}_A}\nabla_A^2\right]-\frac{\hbar^2}{2\mathrm{m_e}}\sum_{\mathrm{i}}^{\mathrm{N_e}}\left[\nabla_{\mathrm{i}}^2\right]+\hat{\mathrm{V}}_{\mathrm{nn}}+\hat{\mathrm{V}}_{\mathrm{ne}}+\hat{\mathrm{V}}_{\mathrm{ee}}\right)\psi_{\mathrm{k}}(\{\mathbf{r}\})=\mathrm{E_k}\psi_{\mathrm{k}}(\{\mathbf{r}\}) \qquad (2.80.1)$$

where $\mathrm{m_A}$ is the mass of the $\mathrm{A^{th}}$ nucleus, $\mathrm{m_e}$ is the mass of an electron, and

$$\hat{\mathrm{V}}_{\mathrm{nn}}=\frac{1}{4\pi\epsilon_0}\sum_{\mathrm{A}}^{\mathrm{N_n}}\sum_{\mathrm{B}\neq A}^{\mathrm{N_n}}\frac{\mathrm{Z_A Z_B}e^2}{\|\mathbf{R}_{\mathrm{A}}-\mathbf{R}_{\mathrm{B}}\|} \qquad (2.80.2)$$

$$\hat{\mathrm{V}}_{\mathrm{ne}}=-\frac{1}{4\pi\epsilon_0}\sum_{\mathrm{A}}^{\mathrm{N_n}}\sum_{\mathrm{i}}^{\mathrm{N_e}}\frac{\mathrm{Z_A}e^2}{\|\mathbf{R}_{\mathrm{A}}-\mathbf{r}_{\mathrm{i}}\|} \qquad (2.80.3)$$

$$\hat{\mathrm{V}}_{\mathrm{ee}}=\frac{1}{4\pi\epsilon_0}\sum_{\mathrm{i}}^{\mathrm{N_e}}\sum_{\mathrm{j}\neq\mathrm{i}}^{\mathrm{N_e}}\frac{\mathrm{Z_A}e^2}{\|\mathbf{r}_{\mathrm{i}}-\mathbf{r}_{\mathrm{j}}\|} \qquad (2.80.4)$$

where ϵ_0 is the vacuum permittivity constant; $\mathrm{Z_A}$ and $\mathbf{R}_{\mathrm{A}}$ are the atomic number and (vector) position of the $\mathrm{A^{th}}$ nucleus, respectively, $\mathbf{r}_{\mathrm{i}}$ is the (vector) position of the $\mathrm{i^{th}}$ electron; e is the fundamental charge of an electron (which, when multiplied by $\mathrm{Z_A}$ gives the charge of the $\mathrm{A^{th}}$

nucleus); and the excitation states of the wavefunction are now indexed by k in Equation 2.80. The Born-Oppenheimer (BO) approximation allows some simplification of Equation 2.80 which removes the need to assess the kinetic energy of the nuclei directly thus turning Equation 2.80 into an electronic wavefunction. BO also allows Equation 2.80.2 to be determined exactly for any combination of nuclei positions, which is then added to the final electronic energy once it is determined. However, the $\mathbf{R}_\mathrm{A}$ nuclei positions in Equation 2.80.3 cannot be removed likewise. This means that the remaining electronic portion of the Hamiltonian and wavefunction are parameterized by the nuclei positions. The BO approximation gives the following form of Equation 2.80:

$$\left(\hat{\mathrm{T}}_\mathrm{e} + \hat{\mathrm{V}}_\mathrm{nn} + \hat{\mathrm{V}}_\mathrm{ne} + \hat{\mathrm{V}}_\mathrm{ee}\right)\psi_\mathrm{elec}(\mathbf{r};\boldsymbol{R}) = \mathrm{E}_\mathrm{elec}\psi_\mathrm{elec}(\mathbf{r};\boldsymbol{R}) \tag{2.81}$$

where we have made the wavefunction's parametric dependence on nuclei positions, $\boldsymbol{R}$, explicit. Also, while the potential energy operator corresponding to nuclear-nuclear repulsions ($\hat{\mathrm{V}}_\mathrm{nn}$) is not technically "electronic", it always integrates to a constant for a given set of $\boldsymbol{R}$ that we will consistently add to the electronic energy in Equation 2.81. Constantly setting this term aside as part of the "nuclear energy contribution" is overly pedantic for this author.

Despite the simplifications allowed by the BO approximation, the presence of more than two "bodies" in Equation 2.81 precludes its exact solution. In general, the number of terms that need to be solved increases on the order of N_e^2, though in practice the computational effort increases with a far greater exponent. Clearly approximations must be made.

The Hartree-Fock (HF) model is the starting point for most wavefunction-based models that attempt to approximate Equation 2.81. More importantly here, the HF model reveals the important terms and phenomena that must be considered when discussing the approximations made in DFT. The basic approach in the HF model is to approximate the total wavefunction as a product of 1-electron wavefunctions. This product is called a “Hartree product.” However, to account for spin (at least phenomenologically) and the antisymmetry of the total electronic wavefunction (as electrons are fermions), “spin functions” are appended onto the 1-electron spatial wavefunctions to create “spin-orbitals” which are then combined in linear combinations of Hartree products via the construction of a so-called Slater determinant. The Variational Principle can then be used to minimize the expectation value of the energy of the system, which is an approximation to the true eigenenergy in Equation 2.2. While the details of this process are beyond the needs of this section, the results form the basis upon which DFT is ultimately developed. Specifically, the HF equations give energies in the form of

$$\mathrm{E_{elec}} = \sum_{\mathrm{i}}^{\mathrm{N_e}} \mathrm{h_i} + \frac{1}{2}\sum_{\mathrm{i}}^{\mathrm{N_e}}\sum_{\mathrm{j}}^{\mathrm{N_e}}\left(\mathrm{J_{ij}} - \mathrm{K_{ij}}\right) + \mathrm{V_{nn}} \tag{2.82.1}$$

where

$$\mathrm{h_i} = \left\langle \phi_\mathrm{i}(\mathrm{i}) \middle| -\frac{\hbar^2}{2\mathrm{m_e}}\nabla_\mathrm{i}^2 - \frac{1}{4\pi\epsilon_0}\textstyle\sum_{\mathrm{A}}^{\mathrm{N_n}} \displaystyle\frac{\mathrm{Z_A e^2}}{\|\mathbf{R}_\mathrm{A} - \mathbf{r}_\mathrm{i}\|} \middle| \phi_\mathrm{i}(\mathrm{i}) \right\rangle \tag{2.82.2}$$

$$\mathrm{J_{ij}} = \left\langle \phi_\mathrm{i}(\mathrm{i})\phi_\mathrm{j}(\mathrm{j}) \middle| \frac{1}{4\pi\epsilon_0}\frac{\mathrm{Z_A e^2}}{\|\mathbf{r}_\mathrm{i} - \mathbf{r}_\mathrm{j}\|} \middle| \phi_\mathrm{i}(\mathrm{i})\phi_\mathrm{j}(\mathrm{j}) \right\rangle \tag{2.82.3}$$

$$\mathrm{K_{ij}} = \left\langle \phi_\mathrm{i}(\mathrm{i})\phi_\mathrm{j}(\mathrm{j}) \middle| \frac{1}{4\pi\epsilon_0}\frac{\mathrm{Z_A e^2}}{\|\mathbf{r}_\mathrm{i} - \mathbf{r}_\mathrm{j}\|} \middle| \phi_\mathrm{j}(\mathrm{i})\phi_\mathrm{i}(\mathrm{j}) \right\rangle \tag{2.82.4}$$

where bra-ket notation is used to symbolize the integration over the so-called “Hilbert Space” spanned by the wavefunctions. $|\phi_\mathrm{i}(\mathrm{i})\rangle$ is a “ket” representing the $\mathrm{i^{th}}$ spin orbital occupied by the $\mathrm{i^{th}}$ electron which will be operated on by its preceding operator. $\langle\phi_\mathrm{i}(\mathrm{i})|$ is a “bra” symbolizing the complex conjugate and/or “left multiplication” of that spin orbital. Combining a bra with a ket implies subsequent integration after any preceding operations. Equation 2.82.2 is the single electron integral, equivalent to the energy of a non-interacting electron moving through the mean potential field generated by all the nuclei in the system. Equation 2.82.3 is the "Coulomb integral" and provides the energetic effect of interacting classically with the other electrons, while Equation 2.82.4 is the "Exchange integral," which is similar except that the subscripts designating spin orbitals has been exchanged in its ket. This integral is an important contribution to the total energy as it has no classical analogue. It provides an energy penalty to electrons of the same spin, which is often referred to as "Pauli repulsion." Regardless of one's choice of interpretation, it is clear that an "exchange energy", i.e. "exact exchange", must be included in any system with more than one electron in it.

Another energetic contribution is missing from the HF approximated energy in Equation 2.82 that is a well-known deficiency: electron correlation. Because the HF model treats each electron as if it is moving through a mean-field, any correlations between electrons cannot be represented. For instance, as an electron approaches another electron, the trajectory or spin state of that electron should be affected in a dynamic way, i.e. their states and thus energetics should be correlated. Attempts to account for this are important and go well beyond the HF model. However, such methods were not used in this dissertation and will therefore not be discussed. For our purposes here, we have the ingredients needed to represent the energy of a molecular system in DFT. We know that the electronic energy of a system must ideally take the form

$$\mathrm{E_{elec}} = \mathrm{T_e} + \mathrm{V_{ne}} + \mathrm{J_{ee}} + \mathrm{K_{ee}} + \mathrm{V_{corr}} + \mathrm{V_{nn}} \qquad (2.83)$$

or otherwise account for the electron kinetic energy ($\mathrm{T_e}$), Coulombic attraction between the electrons and nuclei ($\mathrm{V_{ne}}$), Coulombic repulsion between electrons ($\mathrm{J_{ee}}$), correlation between electrons ($\mathrm{V_{corr}}$), and the (easily assessed) Coulombic repulsion between nuclei ($\mathrm{V_{nn}}$).

2.3.2 Basics of Density Functional Theory

DFT as it was originally designed, is completely free from the considerations of quantum mechanics and the wavefunctions it depends on. The first models sought to simply describe the energetics of atoms using the much simpler electron density, a function of three spatial variables alone. These early models could determine expressions (which implicitly include correlation) in terms of the electron density for $\mathrm{V_{ne}}$ and $\mathrm{J_{ee}}$ for any system and expressions for $\mathrm{T_e}$ and $\mathrm{K_{ee}}$ for a uniform electron gas. These are[7, 8]

$$\mathrm{V_{ne}}[\rho] = -\frac{e^2}{4\pi\epsilon_0}\sum_{\mathrm{A}}^{\mathrm{N_n}} \int\limits_{\text{all space}} \frac{\mathrm{Z_A}\rho(\mathbf{r})}{\|\mathbf{R}_\mathrm{A} - \mathbf{r}\|}\mathrm{d}\mathbf{r} \tag{2.84}$$

$$\mathrm{J_{ee}}[\rho] = -\frac{e^2}{4\pi\epsilon_0} \iint\limits_{\text{all space}} \frac{1}{2}\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|}\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \tag{2.85}$$

$$\mathrm{T_e}[\rho] = \frac{3}{10}(3\pi^2)^{\frac{2}{3}} \int\limits_{\text{all space}} [\rho(\mathbf{r})]^{\frac{5}{3}}\mathrm{d}\mathbf{r} \tag{2.86}$$

$$\mathrm{K_{ee}}[\rho] = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{\frac{1}{3}} \int\limits_{\text{all space}} [\rho(\mathbf{r})]^{\frac{4}{3}}\mathrm{d}\mathbf{r} \tag{2.87}$$

where we follow the convention of Jensen[8] that square brackets after a term (as in $\mathrm{V_{ne}}[\rho]$) represent a functional dependence while parentheses give a function (as in $\rho(\mathbf{r})$). Equations 2.84 – 2.87 provide a means of determining a unique ground state energy for any electron density which is a guarantee proven by Hohenberg and Kohn.[9] The most promising feature of these equations is that the number of variables (three spatial coordinates) does not increase with increasing system size. Unfortunately, these equations provide little accuracy and attempts to systematically improve this “orbital-free” DFT failed early on due to divergence in the Taylor expansions of Equations 2.86 and 2.87. The majority of the error can be traced back to the kinetic energy term, which is poorly described by Equation 2.86 in most systems.

Modern DFT stems from the work of Kohn and Sham[10] who addressed the inaccuracy of the previous equations by reintroducing orbitals, which are now called Kohn-Sham (KS) orbitals. They suggested that the system could be described by a system of 1-electron orbitals that provide the same electron density as the real system. Conceptually, this is very similar to the HF model, and so shares many of the same qualities, including the lack of correlation. The Kohn-Sham equation is very similar to the Fock equation of HF

$$\left(\frac{\hbar^2}{2\mathrm{m_e}}\sum_{\mathrm{i}}^{\mathrm{N_e}}\nabla_{\mathrm{i}}^2 + \mathrm{V_{ne}}[\rho] + \mathrm{J_{ee}}[\rho] + \mathrm{V_{XC}}[\rho]\right)\Psi_{\mathrm{KS}} = \mathrm{E_{KS}}\Psi_{\mathrm{KS}} \qquad (2.88)$$

where the kinetic energy term is replaced with the true kinetic energy operator, the largely exact expressions in Equations 2.84 and 2.85 for $\mathrm{V_{ne}}[\rho]$ and $\mathrm{J_{ee}}[\rho]$, respectively, are retained in their density functional form, and Ψ_{KS} is a Slater determinant of 1-electron spin orbitals or “KS orbitals”, which we will call χ_{i}. The exchange term is not used for the reasons mentioned previously, and instead, the exchange term and the missing correlation are rolled into one term, $\mathrm{V_{XC}}[\rho]$, typically called the exchange correlation functional. The electron-electron Coulombic interaction term, $\mathrm{J_{ee}}[\rho]$, includes the so-called self-interaction error just as it does in the Coulomb integral in Equation 2.82.3, $\mathrm{J_{ij}}$. However, unlike in the HF model, where this error is canceled out explicitly by $\mathrm{K_{ij}}$, the self-interaction error must be accounted for in $\mathrm{V_{XC}}[\rho]$ in Equation 2.88, as well. As a result, $\mathrm{V_{XC}}[\rho]$, if its form were known, would make Equation 2.88 in principle exact. In reality, however, the form of $\mathrm{V_{XC}}[\rho]$ is unknown, making exchange correlation functional development an important and active area of research. Exchange correlation functionals are developed based on certain theoretical considerations (typically starting from the known, exact form of $\mathrm{V_{XC}}[\rho]$ for a uniform electron gas[10-12]) and/or pragmatical considerations like reproducing experimental data for relevant systems[13-15]. It is this latter practice that earns DFT the ire of those who take issue with it being considered an "*ab initio*" method.

Equation 2.88 is solved iteratively, in a self-consistent field manner. The usual, (over)simplified process is to provide a guess for the electron density for the system, use this density to define the density functional terms in Equation 2.88, and then variationally determine the KS wavefunctions, Ψ_{KS}, and KS energy, E_{KS}, that results. A new electron density can then be found from

$$\rho(\mathbf{r}) = \sum_{\mathrm{i}}^{\mathrm{N_e}} \chi_{\mathrm{i}}^{*}(\mathbf{r})\chi_{\mathrm{i}}(\mathbf{r}) \tag{2.89}$$

which, along with the new Ψ_{KS}, can be inserted back into Equation 2.88. This iterative process is then repeated until the energy converges to within some tolerance.

2.3.3 DFT in Periodic Systems

To variationally solve the KS equation in Equation 2.88, a suitable basis set must be chosen to represent the KS orbitals. In molecular, isolated systems, the considerations involved are the same as with pure wavefunction based methods. However, in periodic systems like those studied in this dissertation, the periodicity of the potential energy imposes constraints on the wavefunction. This constraint is embodied in Bloch's theorem[9]

$$\psi_{\mathbf{k}}(\mathbf{r}) = \mathrm{e}^{i\mathbf{k}\cdot\mathbf{r}}u(\mathbf{r}) \tag{2.90}$$

where $\psi_{\mathbf{k}}(\mathbf{r})$ is a "Bloch wave" (i.e. the orbital of any periodic system), $\mathbf{k}$ is a wave vector, $\mathbf{r}$ is the position vector, and $u(\mathbf{r})$ is a wavefunction with the same periodicity as the crystal (i.e. $u(\mathbf{r} + \mathrm{A}\mathbf{m}) = u(\mathbf{r})$ for any $(\mathbf{m})$ integer translations of the (A) unit cell lattice vectors). The wave vector, **k**, spans the “first Brillouin zone” (FBZ) of the systems reciprocal, or “k”, space, and each value of **k** indexes an electronic state of the system.

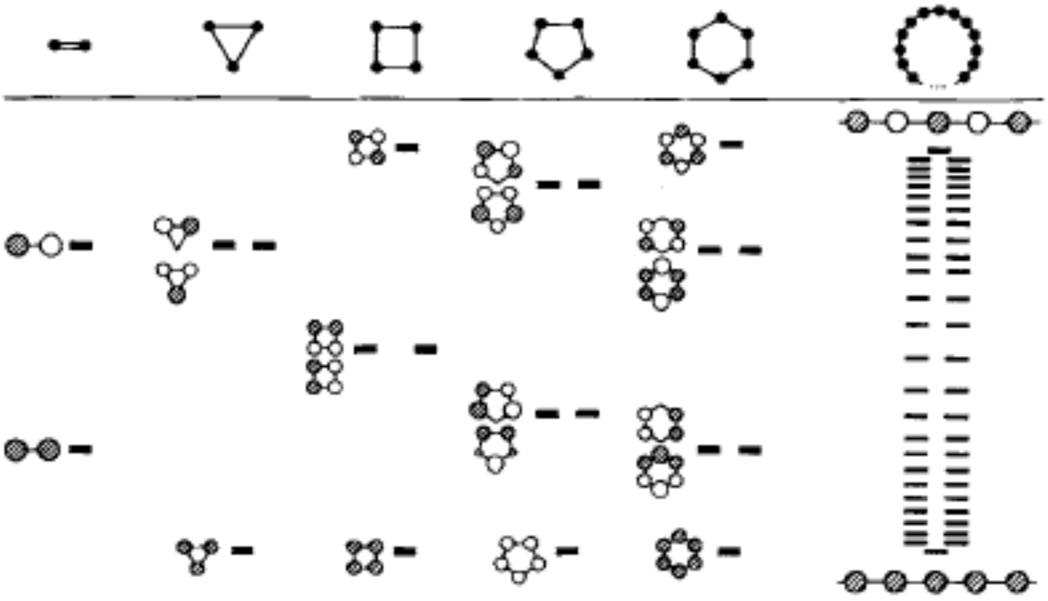


Figure 2.1 Illustration of 1-dimensional periodic molecular systems (above) and their corresponding molecular orbital diagrams (below). *Reproduced with permission from John Wiley and Sons.*

For finite periodic systems (think of benzene as a small 1-dimesional example), these electronic states are finite with discrete values of **k** corresponding to discrete molecular orbitals (see Figure 2.1 for an example from Hoffmann[16]). However, as systems become very large and the number of molecules (or unit cells) approach infinity (like in nanoparticles), **k** varies continuously within the FBZ and what would be an infinite number of molecular orbitals becomes a continuous band structure bounded by the maximally bonding and maximally antibonding molecular orbitals (again, see Figure 2.1). Describing this band accurately is necessary if the energies of the system are to be found, but this would necessitate solving the KS equations for an infinite number of values of **k.** This is of course impossible so, in practice, these equations are only solved for a discrete set of **k** within the FBZ, more commonly known as "k-points". Increasing the number of k-points naturally provides a better estimate of the band structure but also increases the associated computational cost. It is thus common practice to choose the minimum number of k-points that provides suitable convergence of the electronic energy.

Since 1-electron KS orbitals will ultimately be used, each can be constructed from Equation 2.90 to ensure each one complies with the constraints imposed by the periodicity of the potential energy. Each of these will form its own band structure, and so they are given a "band index," n, as

$$\psi_{\mathbf{k}}^{(n)}(\mathbf{r}) = \mathrm{e}^{i\mathbf{k}\cdot\mathbf{r}}\phi_{\mathbf{k}}^{(n)}(\mathbf{r}) \tag{2.91}$$

where $\psi_{\mathbf{k}}^{(n)}(\mathbf{r})$ is now the n^{th} 1-electron KS orbital and $\phi_{\mathbf{k}}^{(n)}(\mathbf{r})$ is its cell-periodic wavefunction which in reality carries all the chemical information of the system. $\phi_{\mathbf{k}}^{(n)}(\mathbf{r})$ is ultimately what must be described via choice of an appropriate basis set. Any basis set can be chosen but if its basis functions are not periodic (such as in atom centered basis sets), they must be made so, which typically makes $\phi_{\mathbf{k}}^{(n)}(\mathbf{r})$ a slightly more complex function of $\mathbf{k}$.

Atom centered basis sets are computationally expensive and even more so within the framework of Bloch's theorem. The simplest way to avoid this problem is to choose a basis set comprised of basis functions that are naturally periodic, such as a plane waves, which can be easily made to have the same periodicity as the underlying lattice. While $\phi_{\mathbf{k}}^{(n)}(\mathbf{r})$ could in principle be independent of the wavevector, $\mathbf{k}$, it is common (and potentially important) to introduce it as an index for plane wave coefficients. Explicitly, this is

$$\phi_{\mathbf{k}}^{(n)}(\mathbf{r}) = \sum_{\mathbf{G}}^{\mathbf{G}_{\max}} \mathrm{C}_{\mathbf{G},\mathbf{k}}^{(n)}\mathrm{e}^{i\mathbf{G}\cdot\mathbf{r}} \tag{2.92}$$

where $\mathbf{G}$ is an integer multiple of the reciprocal lattice vector and essentially determines the harmonic frequency of the plane wave which is cut off at some $\mathbf{G}_{\max}$; and $\mathrm{C}_{\mathbf{G},\mathbf{k}}^{(n)}$ is a plane wave expansion coefficient specific to the plane wave with frequency $\mathbf{G}$, the electronic energy state specified by the value of $\mathbf{k}$, and the band index, n. Inserting Equation 2.92 into 2.91 produces

$$\psi_{\mathbf{k}}^{(n)}(\mathbf{r}) = \sum_{\mathbf{G}}^{\mathbf{G}_{\max}} \mathrm{C}_{\mathbf{G},\mathbf{k}}^{(n)} e^{i(\mathbf{G}+\mathbf{k})\cdot\mathbf{r}} \tag{2.93}$$

whose form makes solution of the KS equation much simplified since the kinetic and potential energy parts are both algebraically tractable, precluding the need for storage and manipulation of large matrices.

The work in this dissertation solves the KS equation using a (Bloch-modified) plane wave basis set. The only downside is that very large numbers of plane waves must be used to get chemical fidelity. Like k-points, we choose the minimum number of plane waves needed to provide suitable energy convergence. Hundreds of thousands of plane waves are often needed. Nonetheless, plane waves are computationally so cheap that this is generally a non-issue.

2.3.4 A Short Digression into the Interpretability of KS Orbitals

In principle, the KS orbitals in Equation 2.88 and 2.89 are fictitious and non-physical; they were conceived of as a means to an end. This is certainly echoed in the choice of using a plane wave basis set, which is essentially as chemically nonsense as a basis set can get. However, the KS wavefunction can in fact be projected onto auxiliary basis sets of chemical significance with considerable success. This suggests, at least to this author, that the KS wavefunction is more physical in nature than given credit even when described with plane waves. In the context of this dissertation, though, this property is only utilized to reveal the s- p- and d-like bands to construct projected density of states (pDOS).

2.3.5 Core Electron Treatment in DFT

Many of the systems studied in this dissertation are made of elements with a very large number of electrons. Even with the assumptions made to arrive at the KS equation, its complexity still increases as the number of electrons increase in the system. Beyond requiring more KS orbitals and thus more variational equations to be solved, these added orbitals have the effect of increasing the number of radial nodes in the true wavefunctions of the valence electrons. Nodes introduce a constraint on the types of plane waves that can be used in the basis set (they all must have nodes at the location of the true wavefunction's nodes), and as a result of this, the amount of plane waves needed to approximate the wavefunction skyrockets to computationally prohibitive numbers. The number of nodes tends to increase nearer to the nucleus of an atom, and since chemical transformations rarely involve the core, or non-valence, electrons, one way to mitigate this is to assume that the core electrons are frozen. The potential energy of the remaining valence electrons near the nucleus can then be modified so as to produce a "true" wavefunction with fewer nodes, which means the size of the plane wave basis set can be brought down to reasonable values. This modified potential is called a "pseudopotential" and its associated true wavefunction a "pseudo-wavefunction."

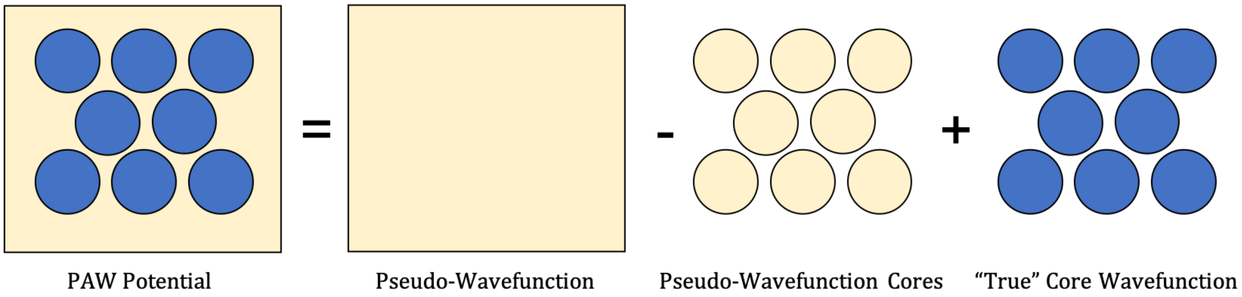


Figure 2.2. Illustration of the Projector Augmented Wave (PAW) approach used in this dissertation.

Pseudopotentials are developed by specifying a radial cutoff inside which the potential energy is modified to produce a smooth, nodeless wavefunction and outside which it matches the true potential. The pseudo-wavefunction is made to match up to the second derivative at the cutoff point. There are a number of approaches for dealing with the core region itself including simply not solving the wavefunction inside those regions (the so-called "scattering" approach) or designing the pseudopotential to replicate the charge distribution within the core (the "norm-conserving approach") so that then entire system can be treated. The approach taken for the DFT calculations within in this dissertation is the projector-augmented wave approach, which is illustrated in Figure 2.2 and, put simply, takes the pseudo-wavefunction, subtracts off the core region part, and then inserts the true wave function of that region in its place. An added benefit of using pseudopotentials is that the core electrons can be treated with exceptionally high accuracy, even including relativistic effects since these need only be done once (during pseudopotential development) and then the results tabulated for use in all subsequent calculations.

As stated previously, all DFT calculations in this dissertation were performed using the Vienna *ab initio* Simulation Package (VASP)[2-4] which implements periodic DFT using a plane wave basis set as described here. Chapters 3-8 all include a thorough methods subsection where further details of the individual calculations performed in those chapters are specified in addition to any theory not mentioned in this chapter.

REFERENCES

[1] A. Clark, *The theory of adsorption and catalysis*. Physical chemistry. 1970: Academic Press.

[2] D. A. McQuarrie, *Statistical thermodynamics*. 1st ed. Vol. 0. 1973: HarperCollins.

[3] R. A. Sack, Mol. Phys. 2 (1959) 8-22.

[4] G. Kresse, J. Hafner, Phys. Rev. B 47 (1993) 558.

[5] G. Kresse, J. Furthmüller, Comput. Mater. Sci. 6 (1996) 15-50.

[6] G. Kresse, J. Furthmüller, Phys. Rev. B 54 (1996) 11169.

[7] P. A. M. Dirac, Mathematical Proceedings of the Cambridge Philosophical Society 26 (1930) 376-385.

[8] F. Jensen, *Introduction to computational chemistry*. 2017: John wiley & sons.

[9] P. Hohenberg, W. Kohn, Phys. Rev. 136 (1964) B864-B871.

[10] W. Kohn, L. J. Sham, Phys. Rev. 140 (1965) A1133-A1138.

[11] J. C. Slater, Phys. Rev. 81 (1951) 385-390.

[12] J. P. Perdew, Phys. Rev. B 33 (1986) 8822-8824.

[13] J. Klimeš, D. R. Bowler, A. Michaelides, Phys. Rev. B 83 (2011) 195131.

[14] B. Hammer, L. B. Hansen, J. K. Nørskov, Phys. Rev. B 59 (1999) 7413-7421.

[15] Y. Zhang, W. Yang, Phys. Rev. Lett. 80 (1998) 890-890.

[16] R. Hoffmann, Angewandte Chemie International Edition in English 26 (1987) 846-878.

CHAPTER THREE:

CO-INDUCED INVERSION OF THE LAYER SEQUENCE OF A MODEL CoCu CATALYST

Gregory Collinge[a], *Yizhi Xiang*[a,], *Roland Barbosa*[a], *Jean-Sabin McEwen*[a,b,c], *and Norbert Kruse*[a,]*

[a] The Gene & Linda Voiland School of Chemical Engineering and Bioengineering, Washington State University, Pullman WA 99164

[b] Department of Physics and Astronomy, Washington State University, Pullman, WA 99164

[c] Department of Chemistry, Washington State University, Pullman, WA 99164

**Corresponding author:* Norbert Kruse; 509-335-6601 (phone), 509-335-4806 (fax), norbert.kruse@wsu.edu

Abstract

Experimental X-ray photoelectron spectroscopy (XPS) and theoretical density functional theory (DFT) calculations reveal the electronic and structural properties of CoCu catalysts before and after CO adsorption. DFT calculations show that, prior to CO adsorption, CoCu has a high tendency to self-assemble into a Co@Cu core-shell structure, which is in accordance with previous atom probe tomography (APT) results for CoCu based systems and the known mutually low miscibility of Co and Cu. We demonstrate that Co and Cu are *electronically* immiscible using a density of states (DOS) analysis wherein neither metal's electronic structure is greatly perturbed by the other in "mixed" CoCu. However, CO adsorption on Co is in fact weakened in CoCu compared to CO adsorption on pure Co despite being electronically unchanged in the alloy. Differential charge density analysis suggests that this is likely due to a lower electron density made available to Co by Cu. CO adsorption at coverages up to 1.00 ML are then

investigated on a Cu/Co(0001) model slab to demonstrate CO-induced segregation effects in CoCu. Accordingly, a large driving force for a Co surface enrichment is found. At high coverages, CO can completely invert the layer sequence of Co and Cu. This result is echoed by XPS evidence, which shows that the surface Co/Cu ratio of CoCu is much larger in the presence of CO than in H_2.

3.1. Introduction

Industrial research with syngas (CO/H_2) at the *Institut Français du Pétrole* (IFP) in the 1970's [1-3] resulted in the formation of short-chain alcohols (up to C_6). A number of catalyst formulations were developed on the basis of CoCu and others. The incentive for choosing these two materials was to design a modified methanol catalyst based on Cu by taking advantage of the chain lengthening properties of Co metal. The authors claimed that the homogeneity of catalyst precursors during the preparation is essential for the final catalyst performance. A modification of the metallic cobalt by alloying was also envisaged even though both metals show low solubility with respect to each other (9% at the most according to the thermodynamic phase diagrams [4]).

Recent studies in our group demonstrated ternary CoCuMn catalysts, prepared by oxalate coprecipitation, to exhibit core@shell structured nanoparticles [5]. In studies with Atom Probe Tomography (APT) Co atoms were shown to form the core in these nanoparticles while all three elements were present in an otherwise Cu dominated shell. Assuming a similar Co@Cu core-shell structure applies to binary CoCu catalysts, pronounced reconstruction was observed in combined TEM/XPS studies (Transmission Electron Microscopy/X-ray Photoelectron Spectroscopy). The surface composition of such catalyst was found to be strongly dependent on the activation procedure and the composition of the activating gas [6]. From a theoretical point of view, Ruban et al. and later Nilekar et al. produced density functional theory (DFT) evidence that Cu atom impurities in a cobalt host have a moderate-to-high segregation energy potential [7, 8]. This finding is in agreement with thermodynamic predictions; so Cu atoms would be expected to segregate away from Co in a CoCu catalyst.

The oxalate route to mixed-metal catalysts allows core@shell structured CoCu particles to self-assemble by stripping CO_2 molecules off the common oxalate framework [5, 6]. On the other hand, inverse Cu@Co structured particles can be produced on purpose using suitable experimental techniques [9, 10]. The Somorjai group has recently studied such catalysts in detail [9, 10]. While XPS data under vacuum conditions clearly indicated a Cu@Co core-shell structure, treatment under oxygen made Cu to segregate to the surface. This oxygen-induced segregation was shown to be irreversible: upon reexposure to H_2, the Cu remained on the surface rather than returning subsurface. The Somorjai group argued based on relative oxide formation energies that since CuO is less favorably formed over that of CoO then the driving force to create an oxide in the presence of O_2 could not account for the driving force to segregate Cu to the surface. Instead, they posited that kinetics or strain effects would have to be responsible for Cu segregation and that the permanency of its segregation is due to Cu's lower surface free energy, which itself stems from a lower bulk cohesive energy as compared to Co. Most recent work by the Somorjai group showed some Cu resegregation was in fact possible upon reexposure to H_2, but more Cu remained at the surface than was initially present in as-prepared inverse Cu@Co [9, 10].

Co/Cu based catalysts have also recently received attention from the experimental groups of Spivey and Salmeron [11-13]. The structure of on-purpose Cu@Co_3O_4 catalysts was elaborated upon by Subramanian et al. who, similar to the Somorjai group, showed that the Cu:Co surface ratio increased 5 times after high temperature oxidation [12]. Carenco et al. later showed that syngas exposure – but not H_2 or CO by themselves – actually depletes Cu@Co core-shell nanoparticles of Co, leading to hollow Cu-rich nanoparticles [13]. Both of these studies illustrate that the Cu@Co core-shell structure is very sensitive to adsorbates and thermal pretreatment, but why this is the case and the degree to which this can be expected is still unknown, and theoretical insights are necessary to further our understanding of CoCu-based catalysts.

Theoretical studies on bimetallic CoCu beyond the single atom Cu impurity work in a Co(0001) host [7, 8] are sparse. Most recent investigations of CoCu [14] assumed a CoCu structure with no justification for how the cobalt came to be on the surface, or, equivalently, how it came to be subsurface. To remedy this state of affairs, the present paper uses both experimental techniques and theoretical DFT calculations to elucidate the CoCu segregation behavior by examining the adsorption of a monolayer of Cu on a Co(0001) surface. We show that Cu has a large thermodynamic tendency to segregate to the surface of Cu/Co(0001) leading to Cu surface termination. We then provide experimental XPS evidence of Co segregation to the surface upon CO adsorption, and this is followed by theoretical electronic analyses suggesting how the surface termination of Cu/Co(0001) ultimately alters the adsorption strength of CO. We follow this with a clear, theoretically explored demonstration of the reversal of Cu/Co(0001) surface termination by CO adsorption via DFT calculations. This CO-induced reversal ultimately results in the inversion of the layer sequence of the CoCu bimetallic system. We conclude with a discussion of the implications of these results with regard to CO hydrogenation using bimetallic CoCu-based catalysts and an outlook to future work.

3.2. Methodology

3.2.1 Experimental

CoCu samples were prepared using the oxalate route of co-precipitation. Details of the preparation method were provided earlier [5]. Catalysts with a Co/Cu ratio of 2/1 and 3/1 were selected for characterization by X-ray Photoelectron Spectroscopy (XPS). CoCu mixed oxalate samples were activated by heating *in situ* to 400°C under atmospheric pressure in a flow of 30 mL min^{-1} of either hydrogen or carbon monoxide in a high-pressure reactor (base pressure 2x10^{-10} mbar) attached to the analytical XPS chamber via a fast sample transfer system. Samples were conditioned as pellets and heated resistively while exposing them to hydrogen and carbon monoxide. Samples were transferred into the analysis chamber (residual pressure 5 x 10^{-11} mbar) after cooling to room temperature and pumping off the gases. Details of the set-up were communicated earlier [15]. The X-ray source was operated with an acceleration voltage of 13 kV and an emission current of 10 mA. Non-monochromatized Mg Kα and Al Kα radiation were used for the analyses. High resolution scans were made for Co 2*p*, Cu 2*p*, C 1*s*, O 1*s* and Cu *LMM* employing a pass energy of 50 eV with a dwell time of 0.1 seconds and a step size of 0.05 eV. After subtraction of the Shirley-type background, the core-level spectra were decomposed into components with mixed Gaussian–Lorentzian (G/L) lines using a non-linear least-squares curve-fitting procedure. The C *1s* peak at 284.4 eV was used as reference energy for charge correction.

3.2.2 Theoretical

3.2.2.1 Computational Details

Ab-initio Density Functional Theory (DFT) calculations with periodic boundary conditions were performed using the Vienna Ab-initio Simulation Package (VASP) [16]. To accurately account for magnetic contributions, any systems containing Co were spin polarized, leaving Cu(111) the only system left in the closed shell approximation. The Perdew-Burke-Enzerhof (PBE) Generalized Gradient Approximation (GGA) [17] was used to describe the electron exchange and correlation functionals with core electrons accounted for by using Projector Augmented Wave (PAW) pseudopotentials [18] to solve the Kohn-Sham Equations [19]. The Brillouin Zone was sampled using a 5 × 5 × 1 Monkhorst-Pack k-point mesh, and plane waves were expanded to an energy cutoff of 400 eV. We used an electronic energy difference of 1.0×10^{-4} eV/atom and force tolerance of 3.0×10^{-2} eV/Å to establish self consistent field (SCF) and geometric optimization convergence criteria.

With the one exception of pure Cu, all systems in this study were hcp(0001) facets modeled using a four layer p(2 × 2) supercell with a ~15 Å vacuum layer. If the model catalyst was pure Cu, then a four layer p(2 × 2) fcc(111) facet with a ~15 Å vacuum layer was used instead. In all models, the bottom two layers were fixed in their bulk positions (with an optimized lattice constant of 2.498 Å) while the top two layers and any adsorbates, if present, were allowed to relax in all directions.

We calculated the adsorption energy according to:

$$E_{ads} = \frac{E^{adsorbates+surface} - E^{surface} - E^{adsorbate}}{N_{CO}} \tag{3.1}$$

for CO on Cu(111) and Co(0001) and compared the results with those where the Brillouin Zone was sampled using a 6 × 6 × 1 Monkhorst-Pack k-point mesh and the energy cutoff was increased to 450 eV. As can be seen from Table 3.1, the adsorption energies varied by only 0.05 eV at most.

Table 3.1. Comparison of DFT calculated adsorption energies with those reported in the literature. The Gajdoš et al**.** **[20]** DFT reference energies were calculated in VASP – as are those calculated here – and the Wellendorff et al. **[21]** DFT reference energies were calculated in Quantum ESPRESSO. For future reference in this paper, the CO adsorption energies on Cu and Co in perfectly segregated Cu/Co(0001) and 0.25 ML surface Co enriched Cu/Co(0001), respectively, are also shown.

Site	System/Set-up	PW-91	PBE	Ref. (PW-91) [a]	Ref. (PBE) [b]
top	CO/Co(0001) 400 eV, (5×5×1)	-1.73 eV	-1.68 eV	-1.65 eV	-1.53 eV
top	CO/Co(0001) 450 eV, (6×6×1)	-1.70 eV	-1.65 eV		
fcc	CO/Cu(111) 400 eV, (5×5×1)	-0.88 eV	-0.87 eV	-0.75 eV (top site)	-0.76 eV (top site)
fcc	CO/Cu(111) 450 eV, (6×6×1)	-0.92 eV	-0.91 eV		
top	CO/Cu(111) 400 eV, (5×5×1)	-0.73 eV	-0.72 eV		
top	CO/Cu(111) 450 eV, (6×6×1)	-0.78 eV	-0.76 eV		
Cu-fcc	CO/Cu/Co(0001) Fully Segregated 400 eV, (5×5×1)	-0.87 eV	-0.84 eV	-	-
Co-top	CO/Cu/Co(0001) 0.25 ML Surface Co 400 eV, (5×5×1)	-1.31 eV	-1.26 eV	-	-

[a] Gajdoš et al. [20]

[b] Wellendorff et al. [21]

Several previous studies have examined the adsorption of CO on Cu(111) and Co(0001) and as can be seen from Table 3.1, we get an agreement to within 0.05 eV when comparing our values with those obtained when using a similar computational setup to that of Gajdoš et al [20]. Recently, the adsorption energies for these systems were also provided by Wellendorff et al. [21] which show that DFT functionals provide calculated adsorption energies that are in error when compared against experimental values. Some of these errors are significant, but the PBE functional used in this study is one of the most accurate functionals. As a point of reference, our DFT calculated CO adsorption energies on pure Cu(111) and Co(0001) are also compared in Table 3.1 to the DFT calculated adsorption energies used in the Wellendorf et al. PBE calculations.

For the Cu/Co(0001) systems in Table 3.1, the adsorption energy was calculated according to:

$$E_{ads} = \frac{E^{total}(x, y) - E^{total}(x = 0,0) - N_{CO}E_{CO}}{N_{CO}} \qquad (3.2)$$

where $E^{total}(x, y)$ is the total DFT energy per supercell of a surface that has x ML equivalents of Co terminating the surface (henceforth, “x ML Co enrichment”) and y ML CO coverage. Further, N_{CO} is the number of CO molecules per unit cell and E_{CO} is the total DFT energy of a gas phase CO molecule.

To model Cu/Co(0001) systems, 4 of the top 8 atoms that make up the top two layers of a pure Co(0001) system were replaced with Cu. During permutations of the surface atoms, the Cu atoms were constrained to these top two layers; preliminary calculations showed no significant change in energy if Cu were placed in the third layer as opposed to the second. We note here that since the bottom two layers are meant to electronically represent the semi-infinite Co bulk, the resulting Cu/Co ratio of 1/3 is not reflective of any particular Cu/Co ratio used in experiments. Cu/Co(0001) atoms were permuted in every way possible so as to guarantee that all unique configurations of Cu and Co in the top two layers of the p(2 × 2) supercell were included in our calculations. This method ensured that the reported energy configurations correspond to minima.

3.2.2.2 Segregation Energy

Segregation energies ($E_{seg}(x,y)$) were defined similarly to Ma and Balbuena, [22] but always in reference to the total energy of the completely segregated Cu/Co(0001) surface having the same adsorbate (CO in this paper) coverage as the (anti)segregated surface. This can be written as:

$$E_{seg}(x,y) = \frac{E^{total}(x,y) - E^{total}(x=0,y)}{N_{Co}} \tag{3.3}$$

Where $E^{total}(x,y)$ is the same as defined previously, and N_{Co} is the number of Co atoms per supercell brought to the surface layer, which is equivalent to the number of Co-Cu "swaps" made per supercell. A negative E_{seg} implies that anti-segregating Cu (or equivalently, creating a surface alloy) is energetically favorable, while a positive E_{seg} implies that Cu segregation is more favorable. By comparing each system to an adsorbate coverage-equivalent surface (y) the energy lowering effect of adsorption is removed from the value and only segregation effects are left. As such, we interpret these segregation energies as effective driving forces for (anti)segregation.

3.2.2.3 Surface Energy Change

We calculate the surface energy for the case of CO on CoCu as:

$$\gamma(x,y) = \frac{E^{total}(x,y) - N_{Cu}E^{Cu}_{bulk} + N_{Co}E^{Co}_{bulk} - N_{CO}E_{CO}}{2A_{Cu/Co(0001)}} \tag{3.4}$$

where $E^{total}(x,y)$ is as defined above , E_{bulk}^{Cu} and E_{bulk}^{Co} are the total energy of the pure bulk Cu and Co, respectively, and E_{CO} is again the gas-phase energy of a single CO. N_{Cu}, N_{Co}, and N_{CO} are the number of Cu atoms, Co atoms, and CO molecules present per supercell, respectively, while $A_{Cu/Co(0001)}$ is the surface area of a single exposed surface in the supercell – there are two per supercell, hence the multiplication by 2.

We concern ourselves here only with the *change* in surface energy per CO as compared to the clean, perfectly segregated Cu/Co(0001) system and effectively subtract off the bulk terms:

$$\Delta\gamma(x,y) = \frac{\gamma(x,y) - \gamma(0,0)}{N_{CO}} = \frac{E_y^x - E_0^0 - N_{CO}E_{CO}}{2AN_{CO}} \qquad (3.5)$$

Division by N_{CO} in Eq. (5) ensures that systems with different surface coverages can be fairly compared. N_{CO} is omitted if no CO is present. We also change to a shorthand notation for surface and total energies of systems with x ML Co enrichment and y ML CO coverage. A negative value of $\Delta\gamma$ is associated with a lowering of surface energy and thus an increase in thermodynamic stability.

3.3.3. Results

3.3.1 XPS Surface Analysis

We start by determining the relative Co/Cu surface amounts of samples prepared by oxalate co-precipitation. The considerations will be limited to samples with 2/1 and 3/1 nominal

Co/Cu ratios. The XPS analysis was performed after *in-situ* decomposition of the oxalate precursors under hydrogen (H_2) or carbon monoxide (CO) at 400 ºC. Since we aim at determining the relative Co/Cu surface amounts we shall focus on an analysis of the Cu 2*p* and Co 2*p* spectra. According to Figure 3.1, the Cu 2*p* profiles for samples heated in either H_2 or CO are dominated by the metallic Cu^0 state (only minor amounts of Cu^{2+} appear on the high binding energy side). More specifically, the spectra are characterized by a doublet spin split of 19.8 ± 0.1 eV typical of metallic copper. Note that the main peak (932.3 ± 0.06 eV) is assigned to either Cu^0 or Cu^+. This is because both states have statistically similar binding energies and, therefore, the Auger LMM spectra of copper have been used in a qualitative manner to differentiate between the two. As shown elsewhere [6], these Auger spectra demonstrate all samples to contain Cu in the metallic state, except the one treated in the presence of CO. As to Co_2Cu_1[CO], a *negative* binding energy shift is observed for Cu^+, with Cu $2p_{3/2}$ and Cu $2p_{1/2}$ binding energies located at 930.7eV and 950.6 eV, respectively. This anomalous negative binding energy shift has been attributed to tetrahedral Cu^+ species in cubic spinel oxides [23].

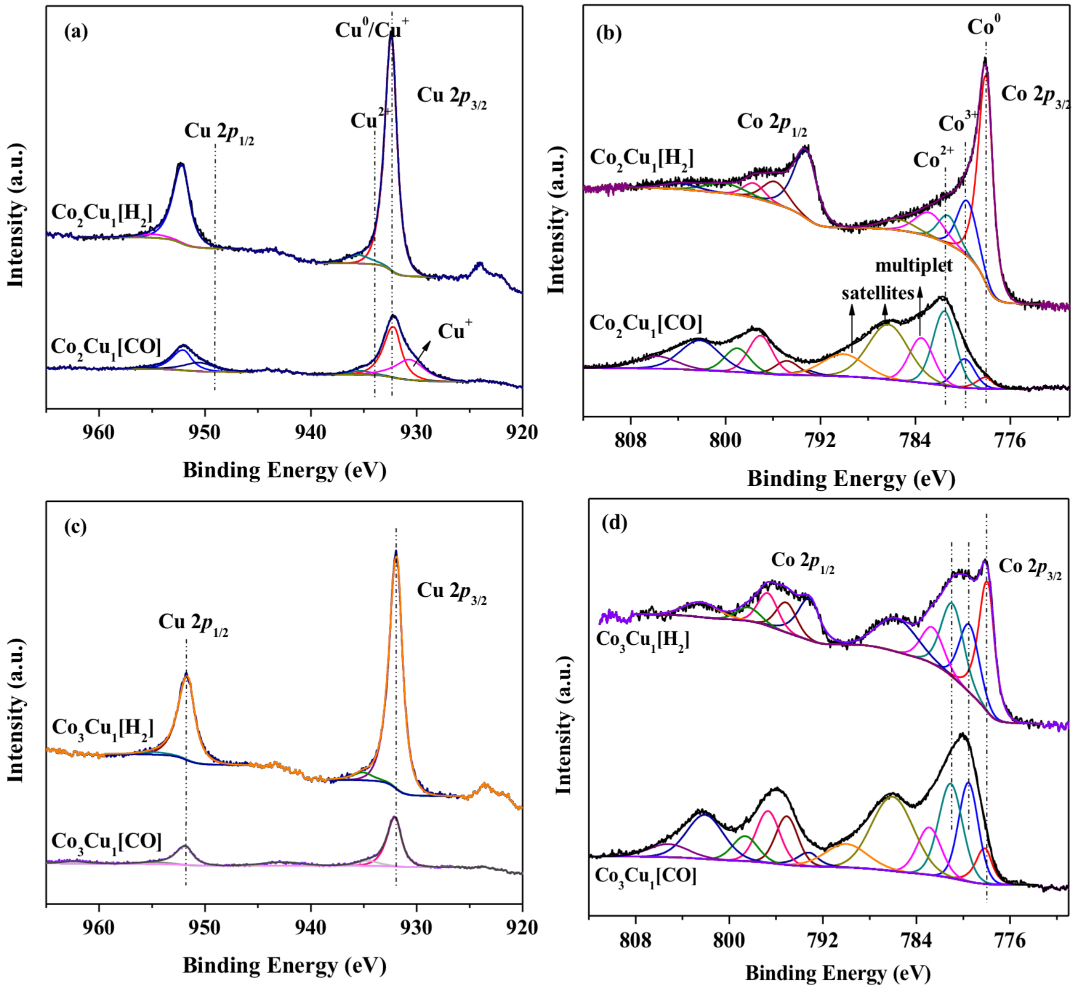


Figure 3.1. Cu 2*p* and Co 2*p* XPS spectra of Co_2Cu_1 and Co_3Cu_1 catalysts activated in-situ in H_2 and CO gas, respectively.

The presence of both Co^{2+} and Co^{3+} in the corresponding Co 2*p* spectra confirms the possible presence of a spinel phase. All of the Co 2*p* spectra contain Co^0, Co^{2+} and Co^{3+}, however, the relative intensities are varying. It is clear that the deconvolution of the Co $2p_{1/2}$, $2p_{3/2}$ excitations is rather involved due to the occurrence of satellite structures. While Co^0 dominates the surface of $Co_2Cu_1[H_2]$, it is the Co^{2+} species which dominates both CO-treated samples. As discussed elsewhere (by including an analysis of the C1s and O1s spectra) [6], Co-carbide formation in the CO-treated samples may be responsible for the occurrence of higher oxidation states of Co (and Cu) metal.

An important observation in relation with our XPS studies addresses the relative surface ratio of Co to Cu. The Co/Cu surface ratios (0.86 and 0.68 for Co_2Cu_1 and Co_3Cu_1) are clearly much lower than the bulk nominal ratios for the two samples. Similar Co/Cu surface ratios were also found for the samples activated under He. This is in qualitative accordance with the occurrence of a Co@Cu core-shell structure. Indeed, it has been shown that a Co rich core phase is encompassed by a Cu rich shell phase once the CoCu mixed oxalate has decomposed [5, 6]. While the Co/Cu ratio is lowest for activation in H_2 (or He) it increases to values of 1.7 and 4.7, respectively, for the CO treated samples, Thus, considerable Co segregation takes place under the influence of a CO gas phase. This chemical pumping is intensified because Cu enriched surface phases as present in Co@Cu core shell structures bind CO relatively weakly. Once the restructuring is consolidated, Co-rich surface phases can decompose adsorbed CO and possibly accumulate surface carbon.

3.3.2 Clean Surface CoCu Segregation

Previous atom probe tomography (APT) results have shown that a CoCu-based catalyst self-assembles into a core-shell structure wherein Cu predominates in the shell and Co predominates in the core [5]. To test the hypothesis that this is the result of a large thermodynamic segregation tendency for Cu in Co, the atoms of the top two layers of the model Cu/Co(0001) system were permuted and then segregation energies and surface energy changes were calculated. As can be seen in Figure 3.2, enriching the surface with Co causes the segregation energy to increase. This indicates that a mixed surface alloy has a very large driving force to segregate completely into a core-shell structure (movement from right to left in Figure 3.2(a)). This driving force is largest at 0.50 ML surface Co enrichment but is also large and positive for all surface Co concentrations. Figure 3.2(b) shows the corresponding values of Δγ for the clean surface as the surface is enriched with Co. As can be seen, Δγ steadily increases as more Co is brought to the surface (and likewise, Cu is pushed sub-surface). Thus, any degree of CoCu alloying would always favor perfect segregation. This phenomenon provides a convincing account of the CoCu core-shell structure found in our experimental APT results.

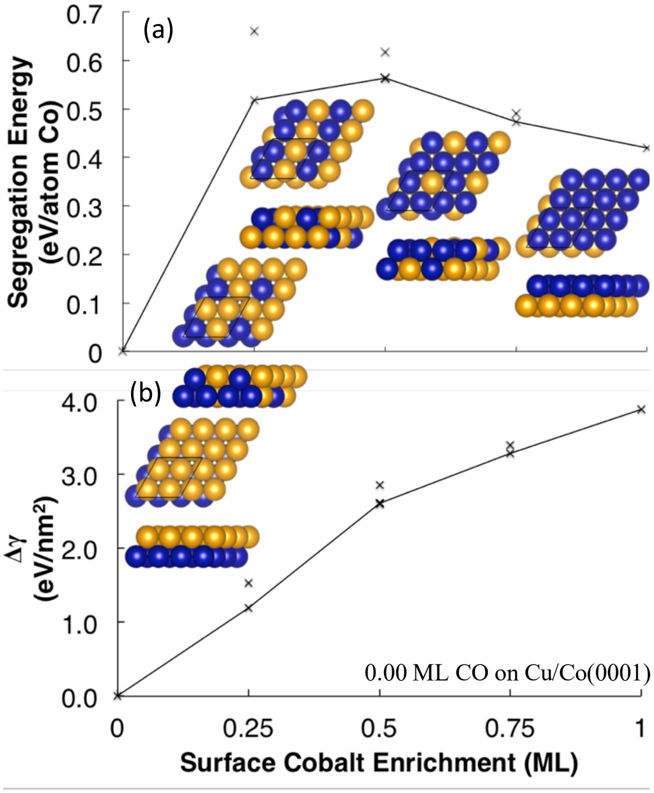


Figure 3.2. (a) Segregation energies, and (b) surface energy changes for Cu/Co(0001) in the absence of CO. Data points correspond to the different configurations that can be achieved through permutation of the Co and Cu in the top two layers of the surface. The solid lines connect the minimum energy configurations, for which top and side views are shown inset. The orange spheres are Cu atoms and the blue spheres are Co atoms.

3.3.3 Electronic Properties of CoCu and CO Adsorption

The CoCu core-shell structure was further studied through calculations of the projected density of states (pDOS) for Co and Cu in the top and second layers of Co(0001), Cu(111), perfectly segregated Cu/Co(0001), and Cu/Co(0001) at 0.25 ML surface Co enrichment (henceforth "$Cu_{0.75}Co_{0.25}$/Co(0001)"). The results can be seen in Figure 3.3 where the unique energetic behavior of each metal is represented. As usual, the features are dominated by the d-band since the sp-band is too diffuse. To place all systems on equal energetic footing, each system is referenced to its vacuum energy. Fermi levels are indicated with vertical lines.

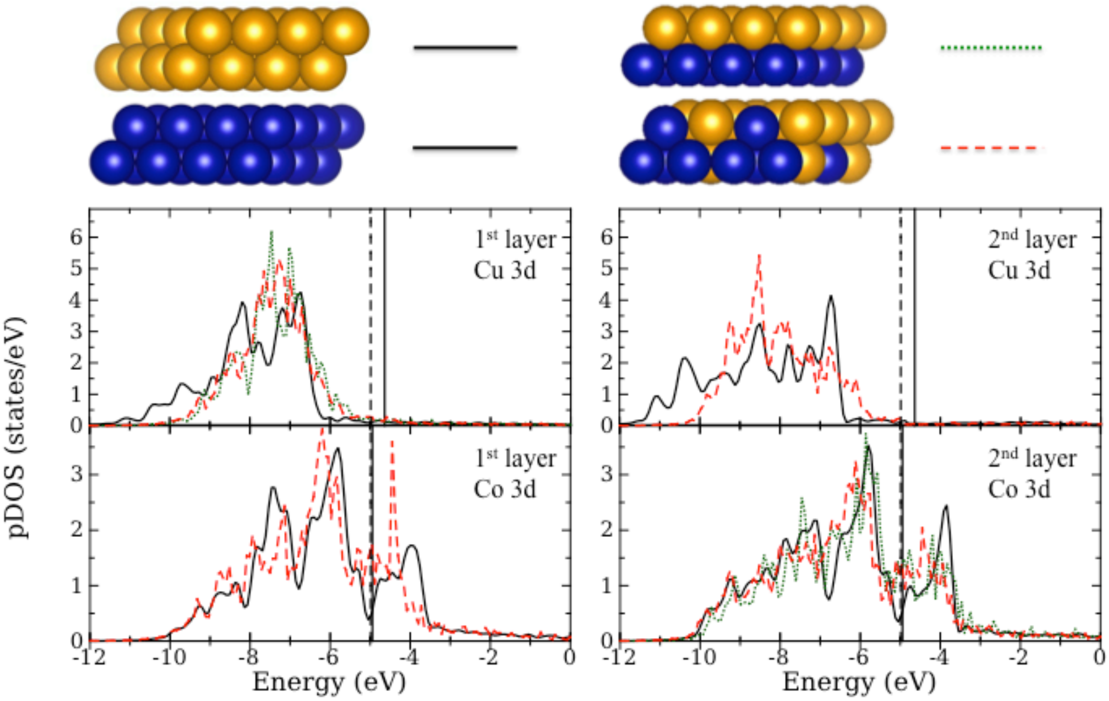


Figure 3.3. Projected density of states (pDOS) for Co(0001), Cu(111), perfectly segregated Cu/Co(0001), and 0.25 ML surface Co enriched Cu/Co(0001). The energies are referenced with respect to the vacuum energy. Solid black lines correspond to the pure metals, green dotted lines correspond to the perfectly segregated surface, and red dashed lines correspond to the 0.25 ML surface Co enriched Cu/Co(0001). Vertical solid and dashed lines indicate the Fermi levels for pure Co(0001) or Cu(111), and both alloyed CoCu systems, respectively. Fermi level shifts were too small to visually distinguish between the perfectly segregated and 0.25 ML surface Co enriched systems and therefore are represented by a single vertical dashed line.

Comparing the pDOS between the pure metals and the two configurations of the CoCu alloy, the most striking feature of the plots in Figure 3.3 is the overall *lack* of change as Co and

Cu systems are “alloyed”. There is some noticeable change in the shape of each band, indicating rehybridization occurs, but the energetic placement of most of the Cu and Co d-states is the same before and after alloying. The d-states of Co change the least, while the Cu d-states show a moderate shift in energy; the surface Cu d-band center moves toward the Fermi level by ~0.6 eV.

Table 3.2. Calculated d-band centers (in reference to the metal Fermi level) and work functions, ε_d-ε_F and ϕ, respectively, for Co and Cu in Co(0001), Cu(111), Cu/Co(0001) at perfect segregation (0.00 ML surface Co), and Cu/Co(0001) at 0.25 ML surface Co concentration. The table is set up to allow easy comparison between the pure metals and the Cu/Co(0001) surfaces, where deviations from the pure metal values can be viewed as deviation from electronic properties of the pure metal. ε_d^{surf} denotes the d-band center of the d-band projected onto a surface atom, while $\varepsilon_d^{subsurf}$ is d-band center projected similarly for those atoms in the subsurface (the second layer). The values in this table correspond to the pDOS shown in Figure 3.3.

	Φ	$\varepsilon_d^{surf} - \varepsilon_f$	$\varepsilon_d^{subsurf} - \varepsilon_f$
Cu in Cu(111)	4.66 eV	-3.11 eV	-3.60 eV
Cu in 0.00 ML surface Co Enriched Cu/Co(0001)	4.90 eV	-2.50 eV	no subsurface Cu
Cu in 0.25 ML surface Co Enriched Cu/Co(0001)	4.99 eV	-2.49 eV	-3.05 eV
Co in Co(0001)	4.95 eV	-1.99 eV	-1.99 eV
Co in 0.00 ML surface Co Enriched Cu/Co(0001)	4.90 eV	no surface Co	-2.07 eV
Co in 0.25 ML surface Co Enriched Cu/Co(0001)	4.99 eV	-1.98 eV	-2.17 eV

In likewise fashion, the Fermi levels of both metals barely move at all in relation to the vacuum energy. The Fermi level of Co moves insignificantly (the shift is ±0.04 eV), while that of the Cu shifts down by ~0.3 eV. From these shifts we can infer that Co is practically unaffected by the presence of Cu, while Cu is slightly activated by Co. However, in no case does the d-band of Cu become partially vacant; the Fermi level is always above the top edge of the d-band. The implication here is that these changes in the electronic structure of Cu are not enough to change Cu's nobility.

These aforementioned energetic changes are represented by d-band centers, ε_d-ε_F, and work functions, ϕ, and are shown in Table 3.2. Based on the previous qualitative analysis and the results presented in this table, we conclude that the Cu and Co in Cu/Co(0001) are electronically unaffected by their respective alloying.

To examine the effect of Cu on Co further, we compare the adsorption energies for CO on a Cu(111) surface, a Co(0001) surface, a Cu/Co(0001) surface, and a $Cu_{0.75}Co_{0.25}$/Co(0001) surface, which is given in Table 3.1. CO adsorption on Cu in the fully segregated Cu/Co(0001) is very similar to CO adsorption in a fcc hollow site of pure Cu(111) (-0.84 eV vs. -0.87 eV), which reflects Cu's persistent nobility even when in contact with Co. In contrast to CO adsorption on Cu, CO adsorption on Co in $Cu_{0.75}Co_{0.25}$/Co(0001) is markedly different than on Co in Co(0001) (-1.26 eV vs. -1.68 eV). Even though Co in $Cu_{0.75}Co_{0.25}$/Co(0001) is electronically very similar to Co in pure Co(0001), CO adsorbs on the surface Co in $Cu_{0.75}Co_{0.25}$/Co(0001) much more weakly. In the forgoing analysis, we make the tacit assumption that no surface rearrangement is occurring beyond that explicitly invoked in the model.

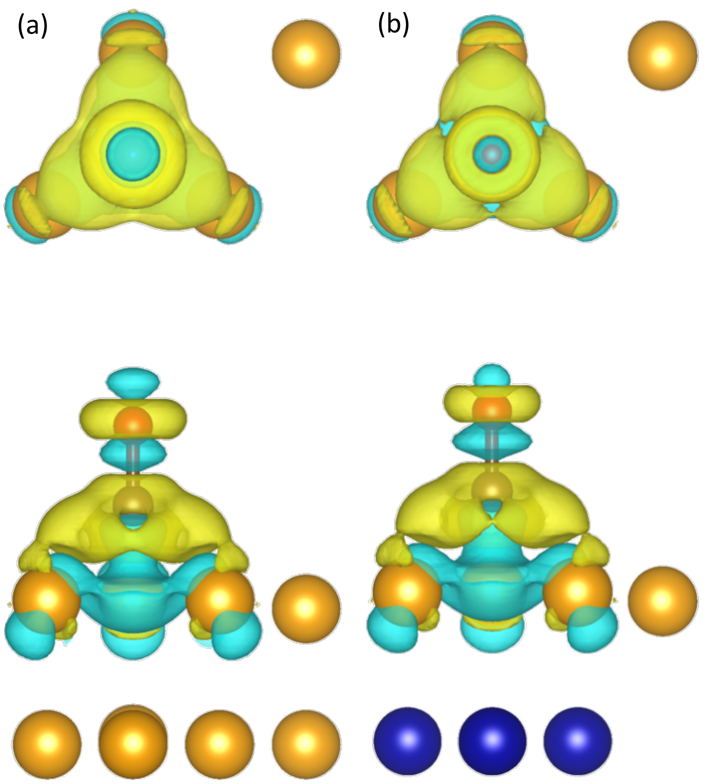


Figure 3.4. Top and side views of the differential charge density of CO adsorption on Cu's in (a) Cu(111) and (b) the fully segregated Cu/Co(0001). Blue shading indicates charge loss while yellow shading corresponds to charge gain. The lack of any significant change in the differential charge density between the two systems is posited to be responsible for the similarity of CO adsorption strength between the two systems. The color legend for the spheres is the same as Figure 3.1, except the red spheres are the oxygen atoms and the brown spheres are the carbon atoms. The isosurface level is set at 0.003 electrons/Bohr^3.

These results are correlated to the surrounding environment of the Co within the first layer: the surface Co in $Cu_{0.75}Co_{0.25}$/Co(0001) is surrounded by Cu, while in pure Co(0001), it is surrounded by Co. In particular, the electron density in the surface Cu of Cu/Co(0001) shows less change than the electron density of the surface Co of pure Co(0001). We see this in Figure 3.4, the differential charge density of CO adsorption in the fcc hollow sites of Cu in Cu(111) and the fully segregated Cu/Co(0001); and Figure 3.5, the differential charge density of CO adsorption on the top sites of Co in Co(0001) and $Cu_{0.75}Co_{0.25}$/Co(0001). In Figure 3.4, the atoms that surround the three Cu atoms involved in CO adsorption show equivalent change in charge density whether the surface is pure Cu(111) (Figure 3.4(a)) or fully segregated Cu/Co(0001) (Figure 3.4(b)). The differential charge elsewhere is practically equivalent, as well. This reflects very well the minor change in CO adsorption energy seen in Table 3.1. Conversely, the differential charge density for CO adsorption on Co(0001) seen in Figure 3.5 shows a distinct loss around the surrounding Co atoms (Figure 3.5(b)), which is not present around the surrounding Cu atoms in the $Cu_{0.75}Co_{0.25}$/Co(0001) system (Figure 3.5(a)). The only region of charge density gain is around the CO adsorption site for the partially segregated $Cu_{0.75}Co_{0.25}$/Co(0001) system. We therefore posit that the Cu in $Cu_{0.75}Co_{0.25}$/Co(0001) cannot similarly contribute to the chemisorption of CO and that this might account for the lowering of the CO adsorption energy seen in Table 3.1.

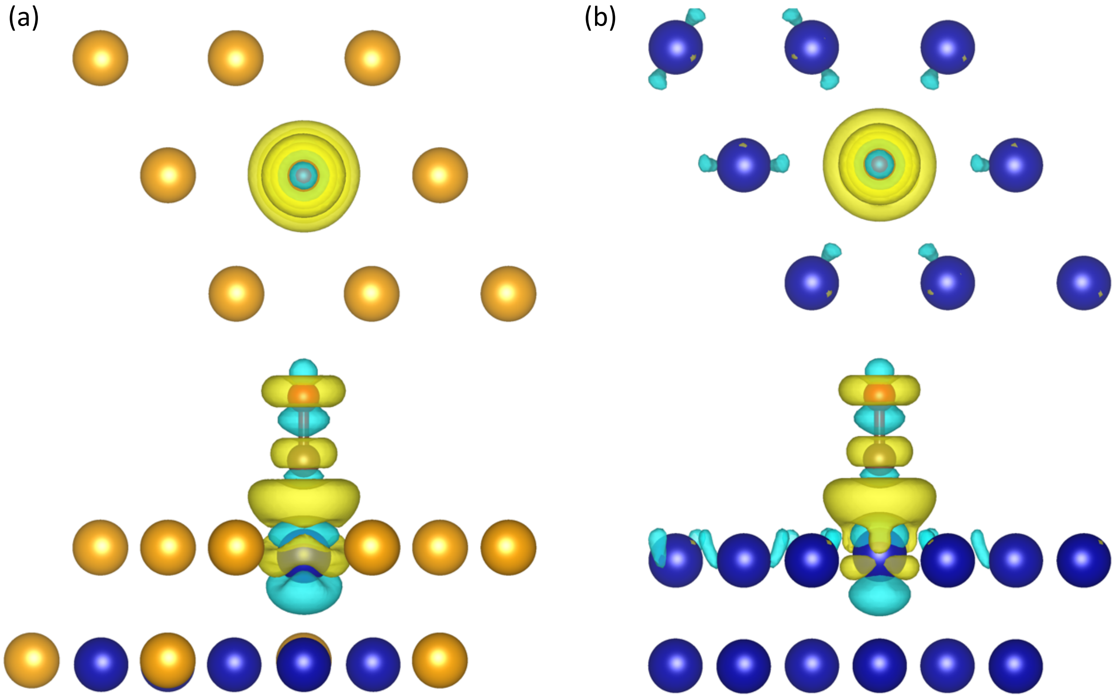


Figure 3.5. Top and side views of the differential charge density for CO adsorption on Co in (a) the fully segregated Cu/Co(0001), in which the surface Co adsorbent atom is surrounded by Cu, and (b) the pure Co(0001) system, where the adsorbent atom is surrounded by Co in the first layer. The charge density loss evident on the surrounding Co in the pure Co(0001) system that is not present in the Cu/Co(0001) system is posited to be the source of the higher adsorption energy of CO on Co in pure Co(0001) over that on Co in Cu/Co(0001). The isosurface level is set at 0.003 electrons/Bohr^3.

To examine this further, we also examined the differential charge density of the clean metal surfaces of Co(0001) and $Cu_{0.75}Co_{0.25}$/Co(0001), in which each surface Co is completely surrounded at the surface by either Co or Cu as shown in Figure 3.6. The metal atoms in Figure 3.6 are kept in the same position as those of the CO adsorption systems in Figure 3.5 and then the surface Co atom – which is eventually the Co adsorbent atom – is removed to create a vacancy in the surface. The charge densities of these defective systems $\left(\rho_{surface\ with\ vacancy}\right)$

and that of the lone Co atom ($\rho_{Co\ atom}$) that is removed are then calculated and compared to the charge densities of the metals without the vacancy $\left(\rho_{full\ surface}\right)$, which are the systems in Figure 3.4 in which the CO has been removed. The forgoing explanation can be written in equation form as:

$$\Delta\rho = \rho_{full\ surface} - \rho_{surface\ with\ vacancy} - \rho_{Co\ atom}$$

This differential charge density qualitatively shows the amount of charge that is being shared with the surface Co atom by the surrounding atoms. Thus, in effect, Figure 3.6 shows the extent to which the metal atoms surrounding the surface Co atom are able to provide/remove charge to/from the surface Co atom.

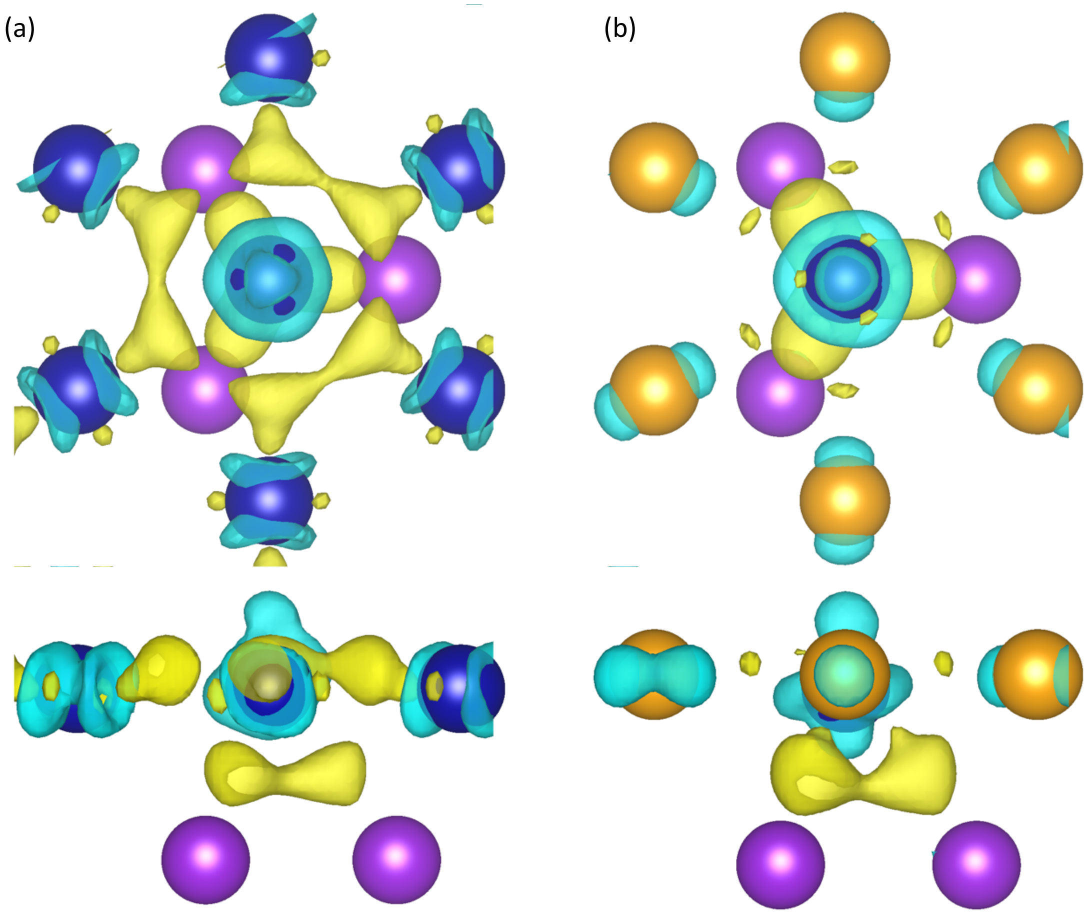


Figure 3.6. Top and side views of the differential charge density of Co adsorbent atoms in (a) the pure Co(0001) system and (b) the 25% surface Co enriched Cu/Co(0001) system. The blue and yellow shaded regions represent charge loss and charge gain, respectively. There is clearly more charge transfer between surface metal atoms in the Co(0001) system than in the Cu/Co(0001) system. This is posited to contribute to the ~0.4 eV discrepancy in CO adsorption energy between the two systems. The Co atoms in the top layer are blue spheres, the Co atoms in the second layer are purple spheres, and the Cu atoms are orange spheres. The isosurface level is set at 0.007 electrons/Bohr^3.

What we see in Figure 3.6 is a much smaller amount of charge transfer between the surface Co atom and its surrounding surface Cu atoms (Figure 3.6(b)) than between the surface Co atom and its surrounding surface Co atoms (Figure 3.6(a)). This shows that Cu makes its electrons much less available to the surface Co atom than does other Co. This is further highlighted when looking at the interactions between the first and the second layer in the

$Cu_{0.75}Co_{0.25}$/Co(0001) system shown in Figure 3.6(b): there is larger amount of charge transfer between the Co atom in the first layer and the Co atoms in the second layer as compared to the corresponding charge transfer between the Co atom in the first layer and the surrounding Cu atoms in the first layer. Therefore, even before CO adsorbs onto these systems there is already a discrepancy between the amount of charge made available by pure Co(0001) and the amount of charge made available by $Cu_{0.75}Co_{0.25}$/Co(0001). Such observations help explain the 0.42 eV drop in adsorption energy when the adsorbent Co is surrounded by Cu instead of other Co. Therefore, we speculate that the adsorption energy would be further lowered by the presence of even more adjacent Cu and would expect to see a similar dependence on systems containing noble metals due to their propensity to remain close shelled.

3.3.4 CO-Induced Co Antisegregation

X-ray photoelectron spectroscopy (XPS) experiments with CoCu catalysts have demonstrated a CO-induced increase in their surface Co/Cu ratios. This is strongly indicative of a surface restructuring during which Co is chemically "pumped" to the surface. To further support this experimental evidence the Cu/Co(0001) model surface was again subjected to surface permutations, but this time at various CO coverages. The $p(2\times2)$ supercell allows for four CO coverages: 0.25 ML, 0.50 ML, 0.75 ML, and 1.00 ML. With each degree of coverage present, the surface was permuted once more, similarly to those performed for the clean catalyst. However, with CO adsorbed on the surface, much of the degeneracy present for a clean catalyst is removed, and many more Cu/Co(0001) configurations exist. The segregation and surface energies for each configuration were calculated and the results are presented in Figure 3.7. Minimum energy configurations are shown as insets for each CO coverage.

An evident feature of these graphs concerns the segregation energies. A precipitous drop in segregation energy is associated with the first increase in surface Co concentration. This means that any amount of adsorbed CO strongly induces Co pumping where adsorbing CO provides a large driving force to reverse the segregation tendency of the clean CoCu surface. As can be seen in Figure 3.7(a), at 0.25 ML CO coverage this initial drop (-0.42 eV/Co), which is associated with 0.25 ML surface Co enrichment, is the only negative value of segregation energy; increasing surface Co concentration beyond this would require an input of energy. Δγ mirrors this result, and in this case the value of Δγ increases steadily as surface Co is enriched beyond 0.25 ML. At low CO coverages, only low surface Co concentrations are thermodynamically favorable.

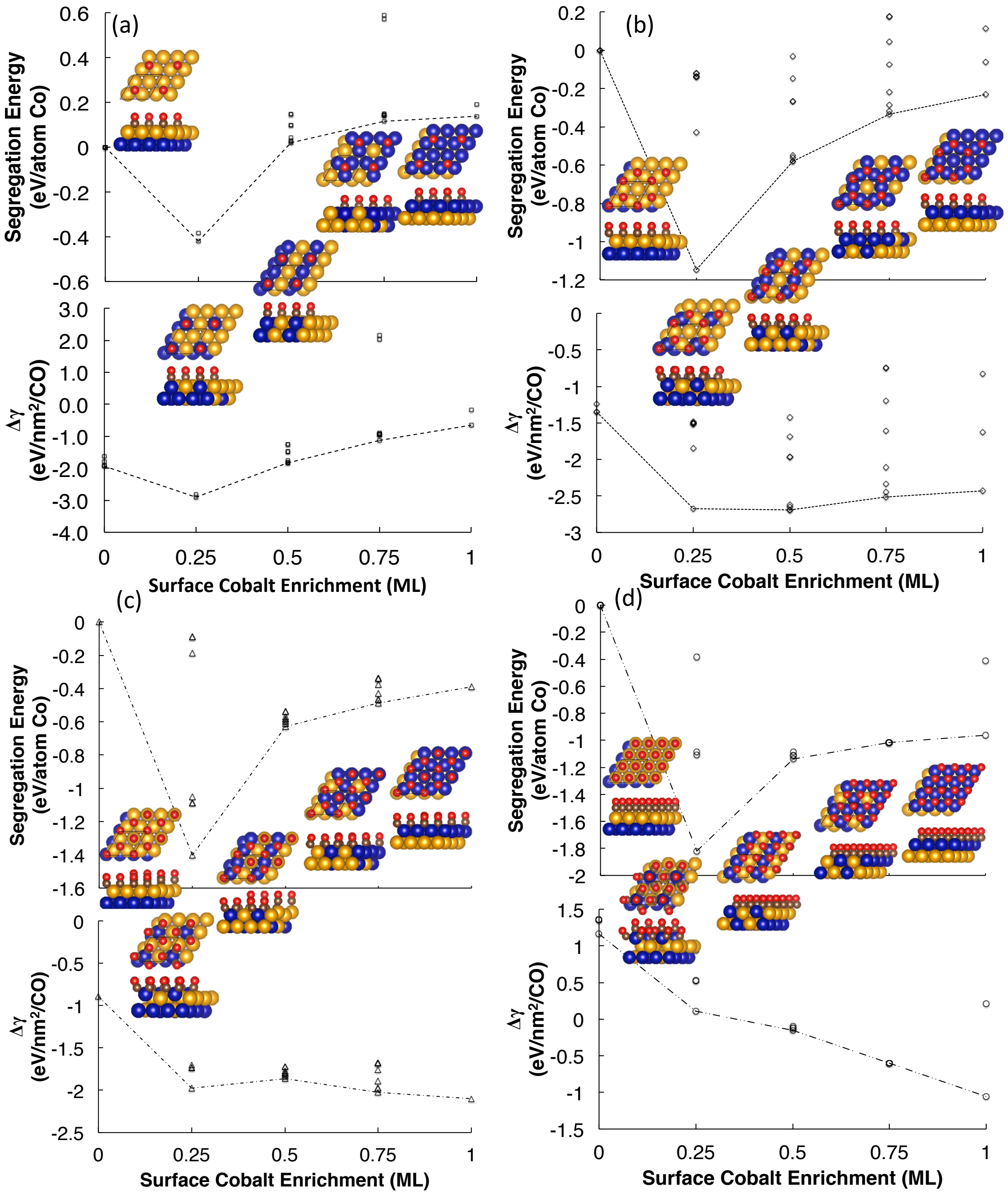


Figure 3.7. Segregation energies and surface energy changes for CO adsorbed on Cu/Co(0001) at (a) 0.25 ML CO, (b) 0.50 ML CO, (c) 0.75 ML CO, and (d) 1 ML monolayer of CO. For each CO coverage and Co enrichment, there are many possible configurations, which are represented by data points in each plot. Lines connect the minimum energy configurations for each CO coverage/Co enrichment system and their corresponding structures are shown as insets. Each abscissa range is set so as to best show the effect of enriching the surface with Co for that coverage and as such does not give a direct impression of the differences between coverages. These graphs are combined in Figure 3.4 to provide a full comparison of Co enrichment at each degree of CO coverage. The color legend for the spheres is the same as Figure 3.3.

At 0.50 ML CO coverage – Figure 3.7(b) – there is a similar large drop (-1.15 eV/Co) in segregation energy upon surface Co enrichment to 0.25 ML, but further enriching the surface with Co does not result in positive segregation energies like in the 0.25 ML CO coverage case. However, although Co concentrations past 0.25 ML have negative segregation energies, the values of Δγ reveal that 0.25 ML and 0.50 ML surface Co enrichment is overall more thermodynamically favorable than higher surface Co concentrations. Therefore, we conclude that once again only low surface Co concentrations are attainable.

A 0.75 ML CO coverage results in a similar segregation energy behavior as the 0.50 ML CO coverage, but the behavior of Δγ is quite different as surface Co enrichment is increased. We again have a large decrease (-1.40 eV/Co) in segregation energy at 0.25 ML surface Co enrichment, which is followed by much smaller, yet negative, segregation energies. Conversely, the plot of Δγ shows a local minimum at 0.25 ML Co enrichment, but an absolute minimum at 1.00 ML Co enrichment. Thus, the presence of a 0.75 ML CO coverage on Cu/Co(0001) will ultimately result in a complete *inversion* of the CoCu layer sequence; the topmost layer of the catalyst can become 1.00 ML enriched with surface Co.

The largest driving force (-1.82 eV/Co) for 0.25 ML Co enrichment in the surface is obtained at 1.00 ML CO coverage. However, by looking at the value of Δγ for this system, we can see that this is mostly due to the fact that a monolayer of adsorbed CO on a completely segregated Cu/Co(0001) surface is unstable (positive Δγ of +1.16 eV/nm^2/CO), and not due to any particularly high stability of the resulting $Cu_{0.75}Co_{0.25}$/Co(0001) system, which actually has a positive Δγ value of 0.11 eV/nm^2/CO. Even still, this full monolayer of CO does become more and more stable as Co is brought to the surface, and this progression results in a minimum energy configuration at 1.00 ML Co enrichment. This is something of a moot point, however, since 0.75

ML CO coverage is enough to induce the inversion of Cu and Co layers and since the values of Δγ at 1.00 ML CO coverage are never lower than the values of Δγ at 0.75 ML CO coverage regardless of the surface enrichment of Co, as can be seen by examining Figure 3.8.

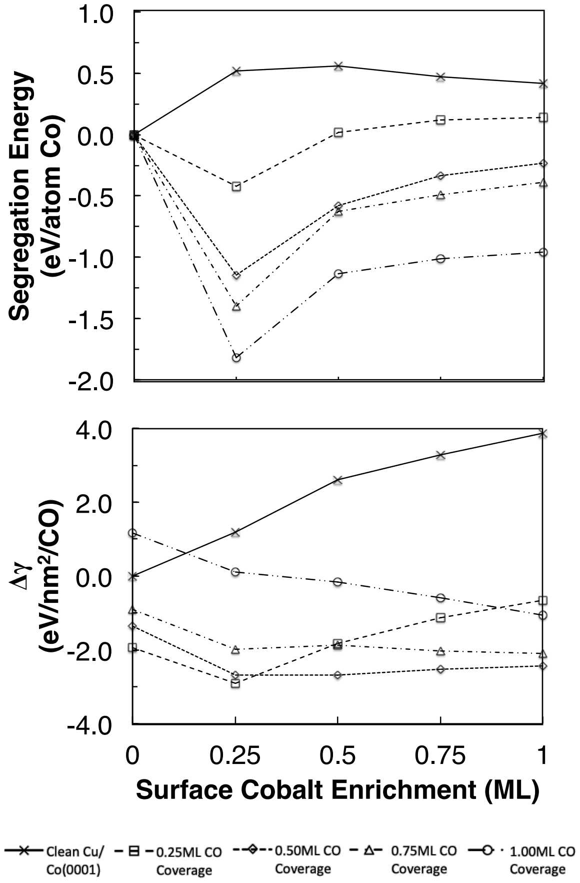


Figure 3.8. Summary plots of the energetic effects of each CO coverage on the surface Co enrichment of Cu/Co(0001). The data points and lines used are consistent with those presented in Figure 3.2 and Figure 3.7. It should be noted that for the clean surface, **Δγ** has units of eV/nm^2 and not eV/nm^2/CO.

By plotting all the segregation energies and Δγ data presented so far in Figure 3.8, we can see that segregation energies are highest for the clean surface and lowest for the 1.00 ML CO coverage system with a monotonic change as the CO coverage increases or decreases (movement up and down the plot instead of left and right). That is, with increasing CO coverage, the driving force for segregation of Co and Cu is gradually altered from favoring a Cu terminated surface to a Co terminated one.

Conversely, there is no monotonicity in the plot of Δγ as the CO coverage is increased up to 1.00 ML. In order to fully understand the implications of this plot, we must break down the trends for each CO coverage, and in this vain, we make the following observations:

- The lowest value of Δγ, and thus the most favorable configuration overall, is achieved at a 0.25 ML surface Co enrichment and a CO coverage of 0.25 ML. Thus, at even low CO coverages Co enrichment at the surface is thermodynamically favorable.
- The lowest overall values of Δγ at the remaining degrees of Co enrichment (0.50 ML -1.00 ML) are all achieved at a CO coverage of 0.50 ML. However, the highest concentrations of surface Co do not correspond to the absolute minimum energy configuration of 0.50 ML CO, which is achieved at 0.50 ML Co enrichment, and which is very closely followed in favorability by a 0.25 ML Co enrichment – a mere 0.02 eV/nm^2/CO higher than the value of Δγ at 0.50 ML enrichment, which is well within the error of DFT. The 0.50 ML CO coverage results in a slight increase in likelihood of pumping Co to 0.50 ML surface enrichment. It is also worth noting that the 0.75 ML and 1.00 ML surface Co enriched configurations are a mere 0.18 eV/nm^2/CO and 0.26 eV/nm^2/CO higher than the 0.50 ML Co enriched configuration.
- The next most favorable configuration at the two highest surface cobalt enrichments is achieved by 0.75 ML of CO, and for this coverage, complete surface Co enrichment *is* the absolute minimum energy configuration.
- Figure 3.8 also confirms what was noted previously, that 1.00 ML CO coverage is always unfavorable compared to lower coverages no matter what amount of Co is

pumped to the surface. This is due in part to the large nearest neighbor lateral interaction between the CO molecules, which may play a fundamental role in the underlying Fischer-Tropsch reaction mechanism on such catalysts [24].

3.4. Conclusion

We have shown here that Co and Cu have a very strong tendency to segregate into a Cu shell atop a Co core and that CO adsorption on this fully segregated surface is essentially very similar to that on pure Cu (provided that no surface rearrangement occurs). On the other hand, CO adsorption on Co in $Co_{0.25}Cu_{0.75}$/Co(0001)is markedly weaker than that on pure Co even though Co appears to be electronically unaffected by the presence of Cu according to our density of states analysis.

Whilst CoCu appears to exist as a Co@Cu core-shell structure we show that CO adsorption can induce an anti-segregation of Cu and Co in CoCu whereby Co is chemically “pumped” to the surface and is effectively exchanged for surface Cu. We illustrate this using an experimental XPS analysis, which shows a significant increase in the Co/Cu surface ratio upon interaction with CO gas, and using further DFT calculations on the various permutations of the Cu/Co(0001) surface. The DFT calculations show that the CO covered anti-segregated surface is thermodynamically favored over that of a CO covered fully segregated surface. If CO is present at high coverages, the surface can become 1.00 ML enriched in Co; the layer sequence of CoCu can become completely inverted.

To put the results of this paper into a more general context, we retain that our combined theoretical-experimental approach clearly demonstrates that major restructuring occurs with segregated Co@Cu core-shell catalyst particles as used for the CO hydrogenation to higher terminal alcohols. The next step will be to include CO dissociation because we anticipate that surface carbon and oxygen formed during this step are essential in the construction of the catalytically active phase [25]. Based on our density of states results on Co and Cu in Cu/Co(0001) and on the comparison of CO adsorption on pure Co(0001) and on Cu/Co(0001), we suspect CO dissociation to be site selective. With this in mind, Ge and Neurock have previously established that the activation energies for CO dissociation on pure Co flat surfaces are prohibitively high, and that CO dissociation is only feasible on stepped and kinked Co surfaces [26]. We would therefore not expect to see CO dissociation occurring on flat Cu/Co(0001), though facets with this orientation may well play a role in establishing stable particle morphologies. With this, we further conclude that future work into CO dissociation on CoCu will include stepped and kinked surfaces.

Acknowledgements

This work was supported by institutional funds provided to J.S.M. and N.K. from the Voiland School of Chemical Engineering and Bioengineering and the National Science Foundation under contract No. CBET-1438227. It was also partially funded by USDA/NIFA through Hatch Project #WNP00807 titled: “Fundamental and Applied Chemical and Biological Catalysts to Minimize Climate Change, Create a Sustainable Energy Future, and Provide a Safer Food Supply”. G.C., Seattle Chapter ARCS Fellow, gratefully acknowledges financial support from the Achievement Rewards for College Scientists foundation.

REFERENCES

[1] A. Sugier, E. Freund, US 4,122,110, (1978)

[2] B. C. P. Courty, D. Durand, C. Verdon, US 4,780,481, (1988)

[3] D. D. P. Courty, E. Freund, A. Sugier, J. Mol. Catal. 17 (1982) 241-254.

[4] T. Nishizawa, K. Ishida, Bull. Alloy Phase Diagr. 5 (1984) 161-165.

[5] Y. Xiang, V. Chitry, P. Liddicoat, P. Felfer, J. Cairney, S. Ringer, N. Kruse, J. Am. Chem. Soc. 135 (2013) 7114-7117.

[6] Y. Xiang, R. Barbosa, N. Kruse, ACS Catal. 4 (2014) 2792-2800.

[7] A. U. Nilekar, A. V. Ruban, M. Mavrikakis, Surf. Sci. 603 (2009) 91-96.

[8] A. V. Ruban, H. L. Skriver, J. K. Nørskov, Phys. Rev. B 59 (1999) 15990-16000.

[9] S. Alayoglu, S. K. Beaumont, G. Melaet, A. E. Lindeman, N. Musselwhite, C. J. Brooks, M. A. Marcus, J. Guo, Z. Liu, N. Kruse, G. A. Somorjai, J. Phys. Chem. C 117 (2013) 21803-21809.

[10] S. K. Beaumont, S. Alayoglu, V. V. Pushkarev, Z. Liu, N. Kruse, G. A. Somorjai, Farad. Discuss. 162 (2013) 31-44.

[11] M. L. Smith, N. Kumar, J. J. Spivey, J. Phys. Chem. C 116 (2012) 7931-7939.

[12] N. D. Subramanian, C. S. S. R. Kumar, K. Watanabe, P. Fischer, R. Tanaka, J. J. Spivey, Catal. Sci. Tech. 2 (2012) 621-631.

[13] S. Carenco, A. Tuxen, M. Chintapalli, E. Pach, C. Escudero, T. D. Ewers, P. Jiang, F. Borondics, G. Thornton, A. P. Alivisatos, H. Bluhm, J. Guo, M. Salmeron, J. Phys. Chem. C 117 (2013) 6259-6266.

[14] X.-C. Xu, J. Su, P. Tian, D. Fu, W. Dai, W. Mao, W.-K. Yuan, J. Xu, Y.-F. Han, J. Phys. Chem. C 119 (2015) 216-227.

[15] S. P. Chenakin, R. Prada Silvy, N. Kruse, J. Phys. Chem. B 109 (2005) 14611-14618.

[16] G. Kresse, J. Hafner, Phys. Rev. B 49 (1994) 14251-14269.

[17] J. P. Perdew, K. Burke, M. Ernzerhof, Phys. Rev. Lett. 77 (1996) 3865-3868.

[18] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, J. D. Joannopoulos, Rev. Mod. Phys. 64 (1992) 1045-1097.

[19] W. Kohn, L. J. Sham, Phys. Rev. 140 (1965) A1133-A1138.

[20] M. Gajdoš, A. Eichler, J. Hafner, J. Phys.: Condens. Matter 16 (2004) 1141–1164.

[21] J. Wellendorff, T. L. Silbaugh, D. Garcia-Pintos, J. K. Nørskov, T. Bligaard, F. Studt, C. T. Campbell, Surf. Sci., http://dx.doi.org/10.1016/j.susc.2015.03.023 (2015).

[22] Y. Ma, P. B. Balbuena, Surf. Sci. 602 (2008) 107-113.

[23] D. A. Kukuruznyak, J. G. Moyer, N. T. Nguyen, E. A. Stern, F. S. Ohuchi, J. Electron. Spectrosc. Relat. Phenom. 150 (2006) 275-281.

[24] M. Zhuo, A. Borgna, M. Saeys, J. Catal. 297 (2013) 217.

[25] J. Schweicher, A. Bundhoo, N. Kruse, J. Am. Chem. Soc. 134 (2012) 16135-16138.

[26] Q. Ge, M. Neurock, J. Phys. Chem. B 110 (2006) 15368-15380.

CHAPTER FOUR:

THE ROLE OF CARBON MONOXIDE IN CATALYST RECONSTRUCTION FOR CO HYDROGENATION: FIRST-PRINCIPLES STUDY OF THE COMPOSITION, STRUCTURE, AND STABILITY OF Cu/Co($10\bar{1}2$) AS A FUNCTION OF CO PRESSURE

Greg Collinge[a], Norbert Kruse[a,], and Jean-Sabin McEwen[a,b,c,]**

[a] The Gene & Linda Voiland School of Chemical Engineering and Bioengineering, Washington State University, Pullman WA 99164

[b] Department of Physics and Astronomy, Washington State University, Pullman, WA 99164

[c] Department of Chemistry, Washington State University, Pullman, WA 99164

* *Corresponding authors:*

Norbert Kruse; 509-335-6601 (phone), 509-335-4806 (fax), norbert.kruse@wsu.edu

Jean-Sabin McEwen; 509-335-8580 (phone), 509-335-4806 (fax), js.mcewen@wsu.edu

Abstract

CoCu-based catalysts are promising candidates for the large-scale application of CO hydrogenation to higher alcohols with varying hydrocarbon chain length. To mimic the Co@Cu core-shell structure of nanosized CoCu particles we choose a Cu/Co($10\bar{1}2$) oriented slab and find, in agreement with chemical imaging results, that the slab surface is always Cu-terminated – with Co underneath. Using DFT calculations we observe major surface atom exchange in the presence of adsorbed CO, with up to 50% of the Cu atoms being replaced by Co in the straight-chain steps of the slab surface. Co atom exchange beyond 50% is not observed. More specifically, this work is accomplished by scanning the configurational space of adsorbed CO, surface Co, and surface Cu, then identifying minimum energy surface configurations. Phase

diagrams are also constructed to determine the thermodynamic driving force imposed in the presence of adsorbed CO. The inclusion of surface phonon modes is shown to ensure the correctness of the calculations. Our results reveal that geminal di- and tricarbonyl Co are formed in steps at temperatures and CO pressures relevant to catalytic CO hydrogenation. Such subcarbonyl surface structures are reminiscent of Co-carbonyl complexes encountered in coordination chemistry.

4.1. Introduction

Alloying one metal into another usually gives rise to charge redistributions and alterations in the chemical reactivity. These electronic effects are thought to be relevant in many reactions of heterogeneous catalysis ever since the foundational work of Schwab, Eley and Dowden in the early 1950's[1-3]. One of the most illustrative examples of how alloying influences the catalytic reactivity was provided in the classical work of Sinfelt et al.[4]. Using Ni-Cu catalysts, the authors compared a structure sensitive with a structure-insensitive reaction (ethane hydrogenolysis vs. cyclohexane dehydrogenation) and found that despite Ni being the active metal in both reactions, their rates varied in a remarkably distinct manner when increasing the relative Cu amounts in the alloy. It soon became clear that geometrical ("ensemble") and surface enrichment effects have to be considered to understand the results[5-7]. From a present-day point of view it is clear that the surface composition of alloys can deviate substantially from the behavior implied by their bulk thermodynamics. Furthermore, not only can the surface composition (and configuration) of alloys deviate from that in the bulk, the surface can deviate from such behavior even further in the presence of adsorbates.

NiCu alloy catalysts still enjoy interest in the catalysis community and were more recently also studied in the CO hydrogenation to methanol[3, 8]. However, this alloy may be anticipated to suffer a strong CO-induced Ni enrichment in the surface, with the possibility of gaseous $Ni(CO)_4$ formation[9, 10]. Such catalysts are therefore deemed unstable under the conditions of high-pressure CO hydrogenation. Quite differently, CoCu bimetallics, despite showing rather limited miscibility as compared to NiCu, may be considered highly relevant to the Fischer-Tropsch (FT) synthesis, if the goal is oxygen functionalization of hydrocarbons to form higher alcohols and other oxygenates. Indeed, way back to the 1970's and 1980's[11, 12], the *Institut Francais du Petrol* (IFP) patented such catalysts for this purpose, but up to present times it is unclear how exactly CoCu induces hydrocarbon functionalization during CO hydrogenation.

While CO hydrogenation to alcohols has received fairly recent theoretical attention (e.g. the work of Medford et al.[13]) and an excellent review on bimetallic FT catalysts was recently published[14], theoretical CO hydrogenation work specific to CoCu appears to be somewhat sparse. Early theoretical studies on CoCu bimetallics were largely unrelated to the material's relevance to CO hydrogenation, and were instead focused on the electronic and magnetic modification of the two metals[15-21]. Some early work by Pedersen et al.[22] also investigated the phenomenon of Co island formation on Cu(111), showing that at very low temperatures and *away from equilibrium* Co trilayer islands can form imbedded within a Cu matrix. Congruent with our more recent findings[23], Cu was shown to have a tendency to engulf surface Co, though they show this process to be slow – on the order of 1 hr at room temperature[22]. A few groups provided theoretically calculated segregation energy databases for metal impurities in metal hosts, including Cu in Co (and vise versa)[24-26], but beyond this, full scale treatment of

CoCu and it's properties in the presence of adsorbates is lacking – a notable exception being recent theoretical work by Yin and Ge who investigated the role of Cu in CoCu/γ-Al_2O_3 as it relates to CO_2 activation and subsequent hydrogenation to alcohols[27]. Recent experimental work on CO hydrogenation over CoCu has suggested that Cu affects the selectivity to alcohols by controlling the "Co ensemble size," but relies on the tacit assumption of a carbide mechanism to inform this hypothesis[28] (A related DFT study[29] makes similar claims based on the same assumption, as well). The group of Spivey has also fairly recently shown in combined experimental/DFT studies that higher Cu concentration in CoCu catalysts increase selectivity to ethanol[30], but that overall alcohol yields are tied to higher surface Co concentration[31]. Furthermore, very recent scanning tunneling microscopy work done in the Sykes group on Co/Cu(111) has shown the tendency of adsorbed CO and H to segregate on the Co surface as well as the overall preference of CO adsorption over H adsorption − excess CO forces H to spillover onto the surrounding Cu[32, 33].

We have reported on CO adsorption onto CoCu previously[23] using Cu/Co(0001) as a model CoCu system, which we've shown also displays a core@shell structure even in the presence of Mn as a third metal[34]. In that work, the presence of CO dramatically changes the segregation behavior at the surface. Our low coverage results there echo recent work showing CO-induced Cu enrichment in PtCu[35], though the rationale for the enrichment is very different. At high enough coverages, our work showed that CO can induce a complete inversion of the Co/Cu layer sequence. However, our findings are at odds with the models used by Xu et al.[29], which assume Co surface patches exist in the presence of CO. We do not find any evidence of Co clustering in the presence of CO in this or our previous study[23]. While Co clustering is favorable for a clean surface, which was also shown to be favorable by Pedersen et al.[22], it is considerably more energetically favorable for the Co to remain dispersed in the presence of CO (see Figure A1 for our explicit calculation).

In order to probe the thermodynamic driving forces of a catalytically relevant system, the inclusion of temperature and pressure effects is required. The most thorough and accessible way to analyze these effects is through the construction of a phase diagram. As a relevant example of this, the group of Saeys has recently constructed theoretically sound arguments for the source of observed nano-island formation and CO surface coverage phase transitions by calculating phase diagrams[36, 37]. In general, surface phase diagrams are constructed through the calculation of the Gibbs free energy, and due to its explicit incorporation of entropy corrections from the vibration, translational, and rotational degrees of freedom (where applicable), the Gibbs free energy provides insights into the relative stability of ordered structures that would otherwise be only partially elucidated by electronic DFT calculations alone. However, the common practice is to assume that the surface metal phonon spectra are unchanged during reactions and that presumably entropy corrections are dominated only by the loss (or gain) of translational, rotational, and vibrational degrees of freedom from the gas phase and/or from the reacting adspecies adsorbed on the surface. The work of Scheffler and Schneider provide early examples of this assumption[38, 39]. As will be seen, this assumption is not valid for the work done here.

This paper concerns itself with Cu/Co($10\bar{1}2$) where a monolayer equivalent of Cu is adsorbed onto a Co slab of the same orientation containing single rows of steps along the [$\bar{1}2\bar{1}0$] direction separated by two-atom terraces in square arrangement. This surface was chosen based on two considerations: (1) the work of Neurock and Ge[40] showed that, amongst the Co surface facets they tested, direct CO dissociation was favorable both kinetically and thermodynamically only on Co($10\bar{1}2$), likely making CO adsorption on Cu/Co($10\bar{1}2$) catalytically relevant; and (2) the work of Prior et al.[41] showed that Co($10\bar{1}2$) was stable through the hcp ↔ fcc transition. We present changes to segregation and surface/adsorption energy as the surface layer of Cu/Co ($10\bar{1}2$) is enriched with Co in the presence and absence of variable CO coverages between 0.00 ML and 1.00 ML. In all instances, configuration space is sampled via the permutation of the atoms in the top layers such that permutation is consistent with the concentration of surface Co and CO coverage in question. Minimum energy configurations are noted and we discuss their coincidence with the formation of multiple bonding of several CO molecules to the same Co atom, similar to geminal di- and tricarbonyls observed in Ni and Ru in our previous surface science research[10, 42]. We go beyond our previous work in another significant way by, for the first time, generating phase diagrams for CO adsorption on a bimetallic surface. We do this for both the previously studied basal (0001) – shown in Appendix A – and the stepped ($10\bar{1}2$) surface taking the minimum DFT energy configurations and calculating the surface free energy across a range of pressures and temperatures.

4.2 Methodology

4.2.1 Computational Details

The Vienna *Ab-initio* Simulation Package (VASP)[43] was used to calculate total energies of all studied systems using the Perdew-Burke-Enzerhof (PBE) Generalized Gradient Approximation (GGA) functional[44]. All of the systems contain Co and are therefore spin polarized to account for the presence of a non-zero magnetic moment. Projector Augmented Wave (PAW) pseudopotentials[45] are implemented as part of VASP to solve the Kohn-Sham Equations[46]. A 4 × 5 × 1 Monkhorst-Pack k-point mesh was used to sample the Brillouin zone of the *p*(1×2) supercell. VASP employs a plane wave basis set, which was expanded to an energy cutoff of 400 eV. Self-consistent field (SCF) and geometric optimization criteria were set at 1.0 × 10^{-4} eV and 3.0 × 10^{-2} eV/Å, respectively, while vibrational calculations were performed using SCF optimization criteria of 1.0 ×10^{-8} eV and a central finite difference with atomic displacement of ±0.01 Å. For further details, we direct the reader to the SI.

The Cu/Co($10\bar{1}2$) surface was modeled using a *p*(1×2) supercell cut from a Co hcp crystal. The system is modeled as a slab containing 6 layers of atoms, 5 layers of Co and 1 layer of Cu. The bottom 4 layers are always Co, the bottom two of which are fixed in their bulk positions, and the top two layers are both Co and Cu and are used for permutation of the Co and Cu atoms during the study. The Cu atoms are placed in the top layer at positions (prior to relaxation) equivalent to the original Co. We utilize notation, as we've done previously, in the form of $Cu_{1-x}Co_x$ /Co($10\bar{1}2$) to denote a surface enriched with Co to x ML. Since the total concentration of Co and Cu is kept constant in the top two atomic layers, our notation does not include the concentration of Co and Cu in the second, or subsurface, layer – for $Cu_{1-x}Co_x$ in the top layer, the second layer will always be Cu_xCo_{1-x}. To eliminate z-direction interaction between slabs, a ~15 Å vacuum layer is imposed.

4.2.2 Adsorption, Segregation, and Surface Energy

Adsorption and segregation energies are defined as previously described[23]. The surface energy is more rigorously defined as the surface normalized difference in Gibbs free energy:

$$\gamma(\mathrm{N,T,p}) = \frac{\Delta G}{A}$$

$$= \frac{1}{A}(G(N = 1\, supercell, T, p) - N_{CO}\mu_{CO}(T, p) - N_{Co}\mu_{Co}(T, p)$$

$$- N_{Cu}\mu_{Cu}(T, p))$$

$$= \frac{1}{A}\Bigg[\left[E_{supercell}^{DFT} + \mu_{supercell}^{vib}(T)\right]$$

$$- N_{CO}\left(E_{CO}^{DFT} + \mu_{CO}^{vib,rot,trans}(T, p^0) + k_B T ln\left(\frac{p_{CO}}{p^0}\right)\right) \tag{4.1}$$

$$- N_{Co}\left(E_{Co}^{DFT} + \mu_{Co}^{vib}(T)\right) - N_{Cu}\left(E_{Cu}^{DFT} + \mu_{Cu}^{vib}(T)\right))\Bigg]$$

where A is the surface area of the supercell used; $G(N = 1\, supercell, T, p)$ is the total Gibbs free energy of the supercell at some temperature, T, and pressure, p; $\mu_{CO}(T, p)$, $\mu_{Co}(T, p)$, and $\mu_{Cu}(T, p)$ represent the total chemical potentials (at the same T and p) of gas phase CO, the Co, and the Cu, respectively; N_{CO}, N_{Co}, and N_{Cu} are the total number of molecules/atoms in the supercell of gas phase CO, the Co, and the Cu, respectively; $E_{supercell}^{DFT}$, E_{CO}^{DFT}, E_{Co}^{DFT}, and E_{Cu}^{DFT} represent the DFT energies at 0 K of the total supercell, gas phase CO, single atom Co in the bulk, and single atom Cu in the bulk, respectively; $\mu_{supercell}^{vib}$, $\mu_{CO}^{vib,rot,trans}(T, p^0)$, $\mu_{Co}^{vib}(T)$, and $\mu_{Cu}^{vib}(T)$ are the vibrational contribution to the chemical potential of the total supercell, the vibrational, rotational, and translational contributions to the chemical potential of gas phase CO, the vibrational contribution to the chemical potential of the Co, and the vibrational contribution to the chemical potential of the Cu, respectively; and p^0 is the standard state pressure of 1 bar.

We point out here that the dependences on temperature and/or pressure are assigned where appropriate, e.g. $\mu_{Co}^{vib}(T)$ depends on temperature alone while $\mu_{CO}^{vib,rot,trans}(T, p^0)$ depends on both temperature and the chosen value of p^0. A full derivation of the ΔG's used here can be found in Appendix A along with a detailed breakdown of our calculations.

As was illustrated by Getman et al.[39], by simultaneously adding and subtracting energy terms for the clean slab (corresponding to the minimum energy surface configuration of the clean surface), we can separate (1) into two terms: $\gamma_{Cu/Co(10\bar{1}2)}$, corresponding to the formation of the Cu/Co($10\bar{1}2$) surface from bulk materials; and $\Delta\gamma_{Y_{CO},X_{Co}}$, corresponding to the additional change associated with adsorption of some configuration, Y_{CO}, of CO and some surface configuration, X_{Co}, of Co:

$$\gamma_{Y_{CO},X_{Co}}(\mathrm{T}, p_{CO}) = \gamma_{Cu/Co(10\bar{1}2)}(T) + \Delta\gamma_{Y_{CO},X_{Co}}(T, p_{CO}) \tag{4.2}$$

where

$$\gamma_{Cu/Co(10\bar{1}2)}(T) = \frac{1}{A}\Big(\big[E_{clean}^{DFT} + \mu_{clean}^{vib}(T)\big] - N_{Co}\big(E_{Co}^{DFT} + E_{Co}^{vib}(T)\big) - N_{Cu}\big(E_{Cu}^{DFT} + \mu_{Cu}^{vib}(T)\big)\Big)$$

and

$$\begin{aligned}\Delta\gamma_{Y_{CO},X_{Co}}(T, p) &\\ = \frac{1}{A}\Bigg(&\big[E_{Y_{CO},X_{Co}}^{DFT} + \mu_{Y_{CO},X_{Co}}^{vib}(T)\big] \\ &- N_{CO}\left(E_{CO}^{DFT} + \mu_{CO}^{vib,rot,trans}(T, p^0) + kTln\left(\frac{p_{CO}}{p^0}\right)\right) - \big[E_{clean}^{DFT} + \mu_{clean}^{vib}(T)\big]\Bigg)\end{aligned}$$

Since the supercell and composition of the slab are kept constant in this study, $\gamma_{Cu/Co(10\bar{1}2)}$ will remain constant and will not be evaluated; $\Delta\gamma_{Y_{CO},X_{Co}}$ is the only thermodynamically relevant term here. $\Delta\gamma_{Y_{CO},X_{Co}}$ is further broken down and rearranged into its DFT energy contribution (which itself contains the DFT adsorption energy) and it's thermal and pressure corrections:

$$\Delta\gamma_{Y_{CO},X_{Co}}(T,p) = \Delta\gamma^{DFT}_{Y_{CO},X_{Co}} + \Delta\gamma^{corr}_{Y_{CO},X_{Co}}(T,p^0) - N_{CO}k_B T ln\left(\frac{p_{CO}}{p^0}\right) \tag{4.3}$$

where

$$\begin{aligned} \Delta\gamma^{DFT}_{Y_{CO},X_{Co}} &= \frac{1}{A}\left(E^{DFT}_{Y_{CO},X_{Co}} - E^{DFT}_{clean} - N_{CO}E^{DFT}_{CO}\right) \\ &= \left(\frac{N_s}{A}\right)\left(\frac{N_{CO}}{N_s}\right)\left(\frac{E^{DFT}_{Y_{CO},X_{Co}} - E^{DFT}_{clean} - N_{CO}E^{DFT}_{CO}}{N_{CO}}\right) \\ &= \sigma_{sites}\theta_{CO}E^{DFT}_{ads,Y_{CO},X_{Co}} \end{aligned} \tag{4.4}$$

and

$$\Delta\gamma^{corr}_{Y_{CO},X_{Co}}(T,p^0) = \frac{1}{A}\left(\mu^{vib}_{Y_{CO},X_{Co}}(T) - \mu^{vib}_{clean}(T) - N_{CO}\mu^{vib,rot,trans}_{CO}(T,p^0)\right) \tag{4.5}$$

where N_s is the total number of adsorption sites, σ_{sites} is the surface density of adsorption sites, defined by $\frac{N_s}{A}$, θ_{CO} is the CO surface coverage defined by $\frac{N_{CO}}{N_s}$, and $E^{DFT}_{ads,Y_{CO},X_{Co}}$ is the DFT adsorption energy as defined previously [23]. The remaining terms are as defined above. Briefly, the CO coverage of stepped surfaces is often split into terrace and step coverages, but we refrain from directly doing this here because the terrace size is rather small for the chosen $(10\bar{1}2)$ surface geometry. In fact, due to the small terrace size the work done here is more indicative of a generalized step. To allow for the evaluation of a large number of configurations, only (4) is initially calculated, but we present $E^{DFT}_{ads,Y_{CO},X_{Co}}$ here since this is a more commonly reported value as opposed to $\Delta\gamma^{DFT}_{Y_{CO},X_{Co}}$ when CO is present. However, phase diagrams are evaluated fully using (3). Free energy terms are determined from full frequency calculations as implemented in VASP and statistical mechanical techniques. We note here that we evaluate all free energy terms in (5) for calculation of (3) by fully calculating the vibrational modes of all surface atoms that were allowed to relax during geometric optimization, and that we have quadrupled the supercell size in these calculations to ensure that all relevant phonon modes were captured. More details can be found in the SI. To aid the reader, we comment that in this paper, we call $\Delta\gamma(T,p)$ "surface *free* energy" and $\mathit{\Delta}\gamma^{DFT}$ "DFT surface energy."

4.2.3 CO Adsorption Sites and Coverage

Prior to providing CO adsorption and surface reconfiguration results, we note here that the *p(*1×2) Cu/Co($10\bar{1}2$) supercell surface consists of 6 exposed surface metal atoms with variable coordination. In the absence of lateral interactions, this surface will accommodate 6 CO per supercell: one CO adsorbed on each exposed surface metal atom as on the (0001) surface. However, any attempt to optimize such a structure results in desorption of all but four CO. Similar results are obtained if five CO molecules per supercell are adsorbed. As such, the lateral interactions between the adspecies limits the number of CO molecules to four CO per supercell, which corresponds to a σ_{sites} of ~13.5 adsorption sites per nm^2 of stepped ($10\bar{1}2$) surface. This value is in contrast to the calculated ~18.5 adsorption sites per nm^2 for a flat (0001) surface.

As a direct result of the above considerations, we have a surface that can be antisegregated one of six surface metal atoms at a time (increments of 1/6 ML), but which can only accommodate CO adsorption on one of four possible sites at a time (increments of 1/4 ML if we define the saturation coverage to be 1 ML).

4.3. Results

4.3.1 Segregation of clean Cu/Co($10\bar{1}2$)

Similar to our previous finding for flat Cu/Co(0001)[24, 25], the stepped Cu/Co($10\bar{1}2$) surface is energetically most stable when the surface layer only contains Cu atoms. This is in agreement with experimental reports according to which Cu atoms dominate the surface region of Co@Cu core-shell structured CoCu-based catalysts[34]. The precise relationship between surface enrichment of Co and the surface's segregation energy as well as its change of surface energy relative to the fully Cu segregated surface can be seen in Figure 4.1. The segregation energy is always positive. In addition, the change of the surface energy is always positive and monotonically increasing as the surface is enriched in Co. Similarly to the flat surface, Cu/Co($10\bar{1}2$) reaches its maximum segregation energy (0.43 eV/Co) at 0.50 ML surface enrichment of Co – indicating that mixing Co and Cu at the surface in equal concentration is energetically most expensive *per Co exchange* – although it does not decrease substantially from there. We note here that as the surface is enriched with Co, the surface step sites only swap Cu with Co at 0.84 ML and 1.00 ML Co enrichment. Thus the low coordination Cu step sites are energetically the most reluctant to exchange for Co.

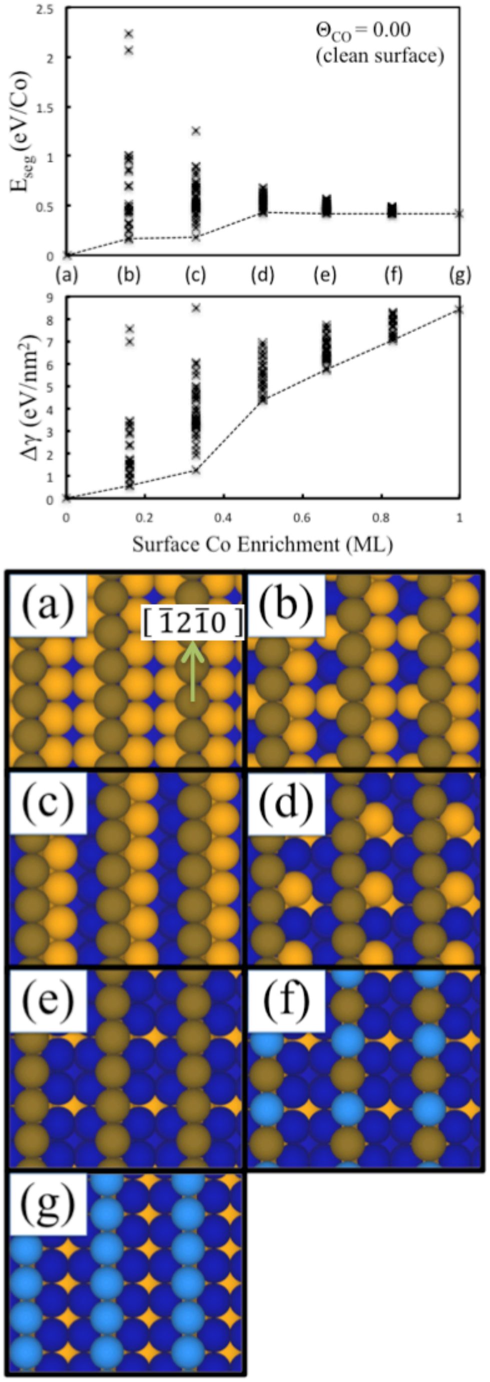


Figure 4.1. Segregation energy (top graph) and DFT surface energy (bottom graph) as a function of surface enrichment of Co in Cu/Co($\mathbf{10\bar{1}2}$). The data points correspond to energies associated with different configurations of Co and Cu. The line in each graph connects the minimum energy configurations, which are themselves indicated as (a)-(g) in the panels in the bottom of Figure 4.1. A top view of each minimum energy configuration is shown below the graphs. Orange spheres are Cu at terrace sites, dark orange spheres are Cu at step sites, dark-blue spheres are Co at terrace sites and light-blue spheres are Co at step sites. The perpendicular direction along which the steps run is shown inset (a).

4.3.3 CO-Induced Co surface enrichment of Cu/Co($10\bar{1}2$)

Various coverages of CO were examined in 0.25 ML increments to the stepped Cu/Co($10\bar{1}2$) surface prior to surface enrichment in Co and permutation of the top two layers, and here, segregation energies and CO adsorption energies were calculated.

At a CO coverage of 0.25 ML, as the surface is initially enriched in Co (to 0.16 ML, i.e. 1 Co-Cu swap out of 6 total swaps possible), the Cu/Co($10\bar{1}2$) surface becomes more stable. Specifically, the adsorption energy changes from -1.14 eV/CO for the perfectly segregated Cu/Co($10\bar{1}2$) to -1.35 eV/CO for the slightly enriched $Cu_{0.84}Co_{0.16}$/Co($10\bar{1}2$) surface, and the segregation energy becomes negative (-0.21 eV/Co). The driving force for antisegregation, i.e. Co surface enrichment, can be seen in Figure 4.2. However, any further Co enrichment results in positive segregation energies and increasing adsorption energy. Thus, with 0.25 ML CO, the stepped Cu/Co($10\bar{1}2$) surface can become only slightly Co-enriched, which is similar to what was seen for flat Cu/Co(0001) in our previous work [23]. As in that work, CO preferentially binds to the top of the newly exposed Co. Since the surface atoms of Cu/Co($10\bar{1}2$) are non-equivalent, it is also important to note that the newly exposed Co atoms actually prefer the upper terrace sites of the facet rather than the step. We will return to this point later on.

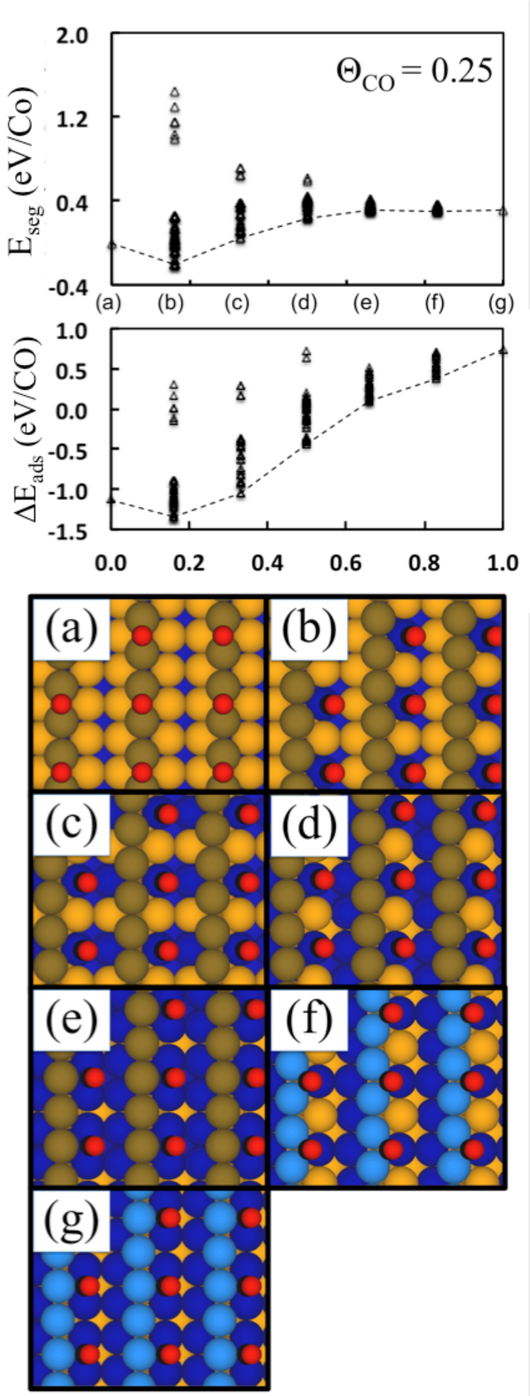


Figure 4.2. Segregation energy (top graph) and CO adsorption energy (bottom graph) as a function of surface enrichment of Co for the Cu/Co($\mathbf{10\bar{1}2}$) surface in the presence of 0.25 ML CO coverage. Data points and lines have the same meaning as in Figure 4.1. Sphere color scheme is identical to that used in Figure 4.1, as well. Additionally, black spheres are C and red spheres are O atoms in adsorbed CO.

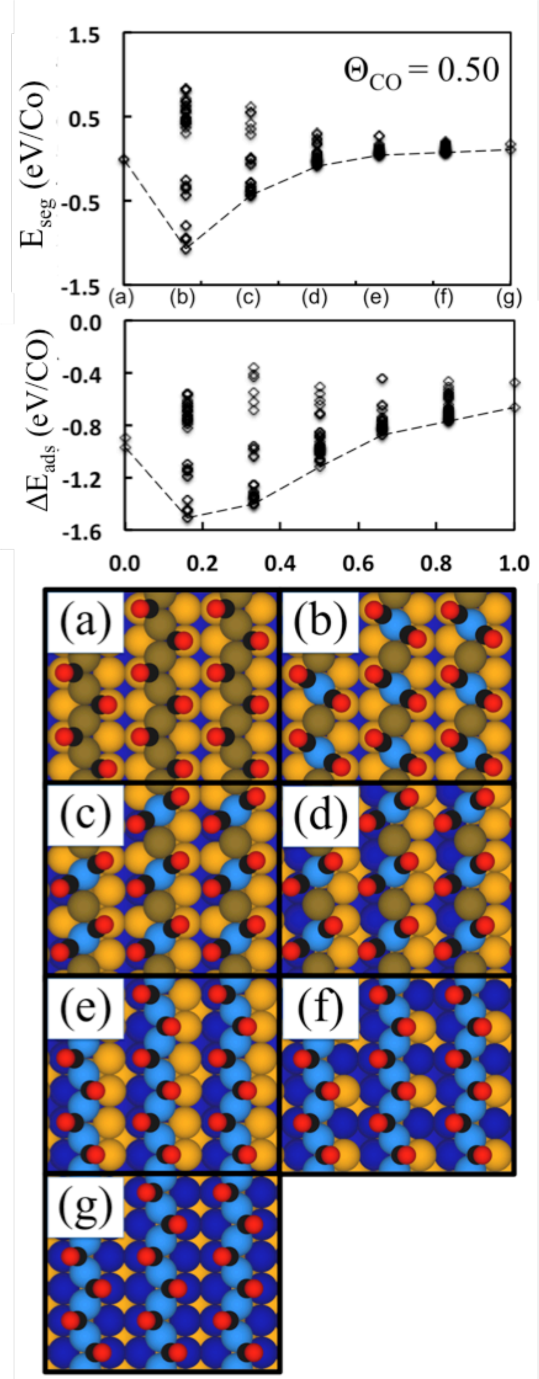


Figure 4.3. Segregation energy (top graph) and CO adsorption energy (bottom graph) as a function of surface enrichment of Co in the presence of 0.50 ML of CO. Data points and lines have the same meaning as in Figure 4.1. The sphere color scheme is identical to that used in Figure 4.1 and Figure 4.2, as well.

At 0.50 ML CO coverage, similar to that at 0.25 ML CO coverage, the surface enrichment in Co from Cu/Co($10\bar{1}2$) to $Cu_{0.84}Co_{0.16}$/Co($10\bar{1}2$) initially results in surface stabilization (adsorption energy changes from -0.97 eV/CO to -1.51 eV/CO), and segregation energies are correspondingly negative (-1.07 eV/Co). While segregation energies remain negative until the surface is enriched to $Cu_{0.33}Co_{0.67}$/Co($10\bar{1}2$), the CO adsorption energy decreases in magnitude as the surface is further enriched in Co. Thus, the surface enrichment in Co to 0.67 ML is accompanied by a lowering of the total energy when compared strictly to the fully segregated Cu/Co($10\bar{1}2$) surface, but the overall minimum energy configuration corresponds to a 0.16 ML surface enrichment in Co. It is interesting to note that the dramatic increase in the magnitude of the CO adsorption energy from 0.0 ML to 0.16 ML of Co enrichment is associated with binding modes in which two CO molecules share the newly exposed Co atom at the *step site*. This can be seen in Figure 4.3(b), and we will return to this point shortly, as well.

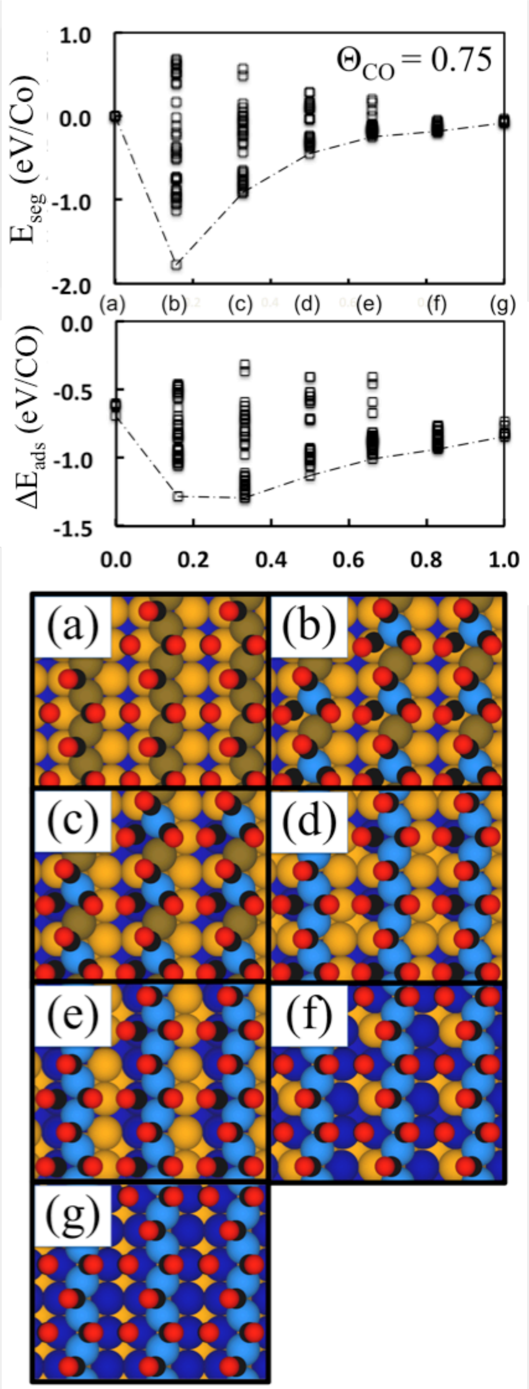


Figure 4.4. Segregation energy (top graph) and CO adsorption energy (bottom graph) as a function of surface enrichment of Co in the presence of 0.75 ML of CO. Data points and lines have the same meaning as in Figure 4.1. Sphere color scheme is identical to those used in Figure 4.1 and Figure 4.2, as well.

The 0.75 ML CO coverage scenario does little to change the antisegregation tendencies already seen for lower CO coverages. Again, as the surface is enriched in Co to $Cu_{0.84}Co_{0.16}/Co(10\bar{1}2)$, the magnitude of the CO adsorption energy increases (-0.69 eV/CO to -1.28 eV/CO). This is similar to the lower coverage case. However, it is practically thermo-neutral (-1.28 eV/CO to -1.30 eV/CO) to further enrich the surface in Co to $Cu_{0.67}Co_{0.33}/Co(10\bar{1}2)$. All segregation energies are negative as Co is enriched, indicating that all levels of Co enrichment are energy lowering compared to the perfectly segregated $Cu/Co(10\bar{1}2)$. However, any further enrichment in Co past 0.33 ML decreases the magnitude of the CO adsorption energy, and so leaves higher levels of enrichment energetically unlikely. The two minima in the adsorption energy are not reflected in the segregation energy because the segregation energy is expressed as energy change *per surface Co*, which increases as the surface is enriched with Co, lowering the value correspondingly. The two minimum energy configurations are associated with the formation of a surface layer with either three CO molecules sharing single Co step atoms as seen in Figure 4.4(b) or two CO molecules sharing such Co step atoms along with a third CO bridging two adjacent Co at the lower terrace site as seen in Figure 4.4(c).

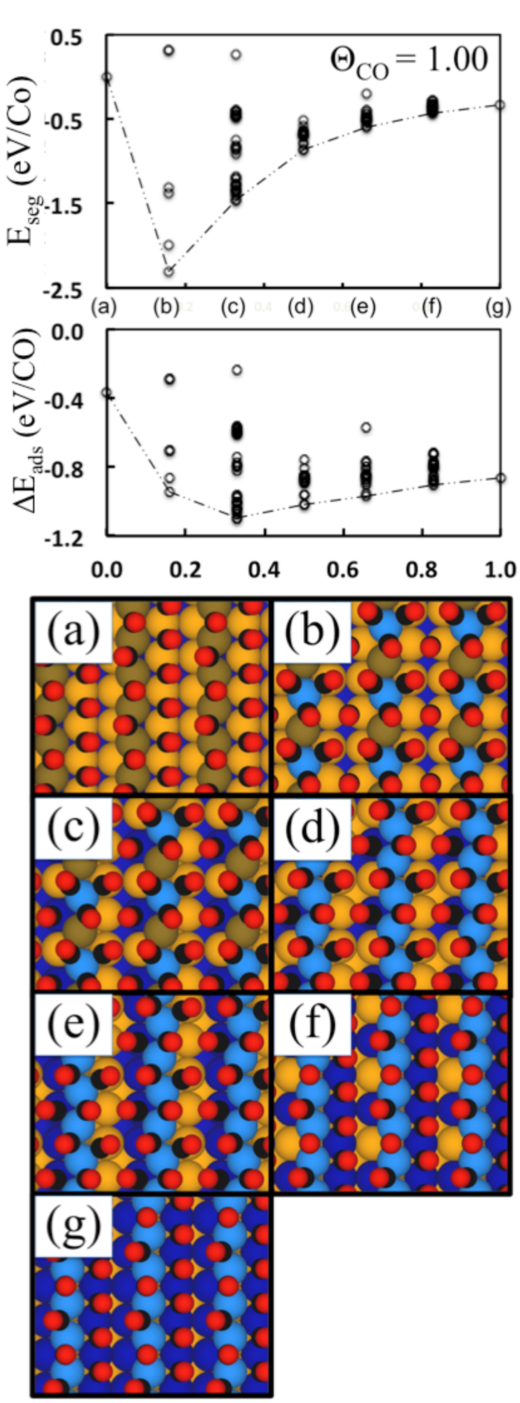


Figure 4.5. Segregation energy (top graph) and CO adsorption energy (bottom graph) as a function of surface enrichment of Co in the presence of 1.00 ML of CO. Data points and lines have the same meaning as in Figure 4.1. Sphere color scheme is identical to that used in Figure 4.1 and Figure 4.2, as well.

Finally, when proceeding to saturated surface layers with 1.00 ML of CO, see Figure 4.5, we find, much like for the 0.75 ML case, that the segregation energies are all negative showing that any amount of Co in the surface layer is energy lowering compared to no Co in the surface layer. The overall largest magnitude segregation energy (-2.30 eV/Co) amongst all CO coverages is achieved at 1.00 ML CO coverage for a surface 0.16 ML enriched with Co. However, the overall minimum energy configuration is reached on the $Cu_{0.67}Co_{0.33}/Co(10\bar{1}2)$ surface, corresponding to an adsorption energy increase (in magnitude) from -0.37 eV/CO on $Cu/Co(10\bar{1}2)$ to -0.94 eV/CO on $Cu_{0.84}Co_{0.16}/Co(10\bar{1}2)$ and then to -1.09 eV/CO on $Cu_{0.67}Co_{0.33}/Co(10\bar{1}2)$. This means that only 0.33 ML of the surface layer is favorably enriched with Co; a complete inversion of the layer sequence from Cu-only, as originally present in $CuCo(10\bar{1}2)$, to Co-only does not occur for whatever CO coverage is chosen. As for lower CO coverages, the minimum energy configuration for the 1 ML case, which can be seen in Figure 4.5(c), is associated with multiple CO sharing single Co atoms along the step ridge. In particular, three CO molecules per Co step atom are observed along with single CO on top of Co at the lower terrace site.

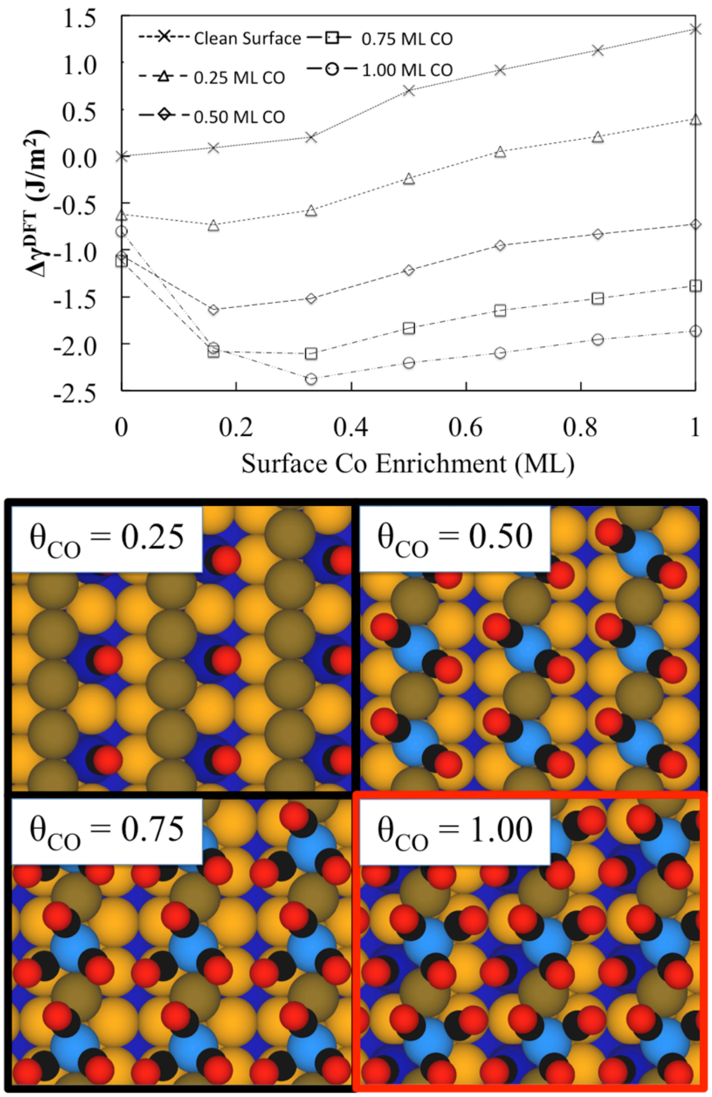


Figure 4.6. Summary of DFT surface energies as a function of Co surface enrichment for all four CO coverages investigated in this study. Dot-dash scheme is as indicated in the inset legend. Minimum energy configurations for each CO coverage are shown underneath the graph, and the global minimum energy configuration is indicated with a red border. Color scheme for the spheres is identical to that used in Figures 4.1-4.5.

In order to compare the (zero Kelvin) thermodynamic stability of the various CO coverages and surface configurations, the DFT surface energies of the minimum energy surface configurations are presented in Figure 4.6. The minimum energy configuration for each CO coverage is shown below the graph. The overall minimum surface energy is achieved with a full monolayer of CO adsorbed on $Cu_{0.67}Co_{0.33}$/Co($10\bar{1}2$), as shown in Figure 4.5(c) and described above. Different from the behavior of flat Cu/Co(0001), high coverages of CO do not result in the complete enrichment with Co on the stepped Cu/Co($10\bar{1}2$) surface. Instead, most of the surface Cu atoms resist an exchange despite the high concentration of CO adsorbates. Even half of the step sites remain Cu, which appears to be critical to the formation of multiple bonded CO configurations in which two or even three CO molecules bind to a single Co step atom. Fully swapping the Cu at step sites with Co results in zigzag-type adsorption of CO at the steps and a corresponding weakening of adsorption. This can be seen in Figure 4.3(e)-(g), Figure 4.4(d)-(g), and Figure 4.5(d)-(g). It should be noted that multiple bonded CO is reminiscent of Co-carbonyls as encountered in coordination chemistry and, more specifically, of the observation of geminal di- and tri- Co-carbonyls in atom-probe studies with pure Co nanosized particles conditioned as tips[47]. In these studies, field pulse desorption along with time-of-flight mass spectrometry were applied to rupture such subcarbonyl species as ions from the stepped surface of a Co tip during its interaction with CO or mixtures of CO and H_2. Since the ionic desorption of subcarbonyls involves Co step sites, a mechanism for the surface restructuring of the overall particle surface could be suggested in this work.

4.3.3.1 Surface Sites and Configurations of Cu/Co($10\bar{1}2$*)*

As the forgoing section indicated, the step sites of the clean Cu/Co($10\bar{1}2$) surface have a large preference to remain Cu. The addition of a small coverage of CO does not seem to change this finding. A single CO molecule coordinated to a Co at the step site is insufficient to stabilize it. A larger amount of electron density is needed to offset the loss Co experiences when placed at the lower coordinated step sites. However, once enough CO is available to form geminal carbonyls (0.50 ML CO coverage here), the Co *is* stabilized at the step site, and in fact corresponds to the strongest adsorption energy found here (-1.51 eV/CO). To confirm that geminal dicarbonyl formation is not favorably influenced by the choice of GGA functional, we have recalculated the adsorption energy of CO along with coverage-equivalent structures that are energetically competitive using revPBE with and without vdW-DF (inclusion of which was recently shown to give CO adsorption energies closer in agreement to experiment by the Saeys group[36, 37]), see Figure A2 in Appendix A. For all functionals, the geminal dicarbonyl is still predicted to be favored over these other configurations.

4.3.4 Phase Diagrams of Cu/Co($10\bar{1}2$) and Cu/Co(0001)

To provide an explicit connection to experiment, we now construct phase diagrams for the minimum DFT surface energy configuration of each coverage as well as those configurations that are only marginally less favorable (at zero Kelvin). This is done by full utilization of Equation 4.3, and the results can be seen in Figure 4.7. We have chosen to show the phase diagrams as a function of pressure and at two selected temperatures: 513 K, corresponding to a typical reaction temperature for CO hydrogenation studies; and 653 K, corresponding to the temperature when transforming suitable catalyst precursors – in our previous studies mixed CoCu-oxalates[34, 48, 49] - into Co@Cu core-shell structures.

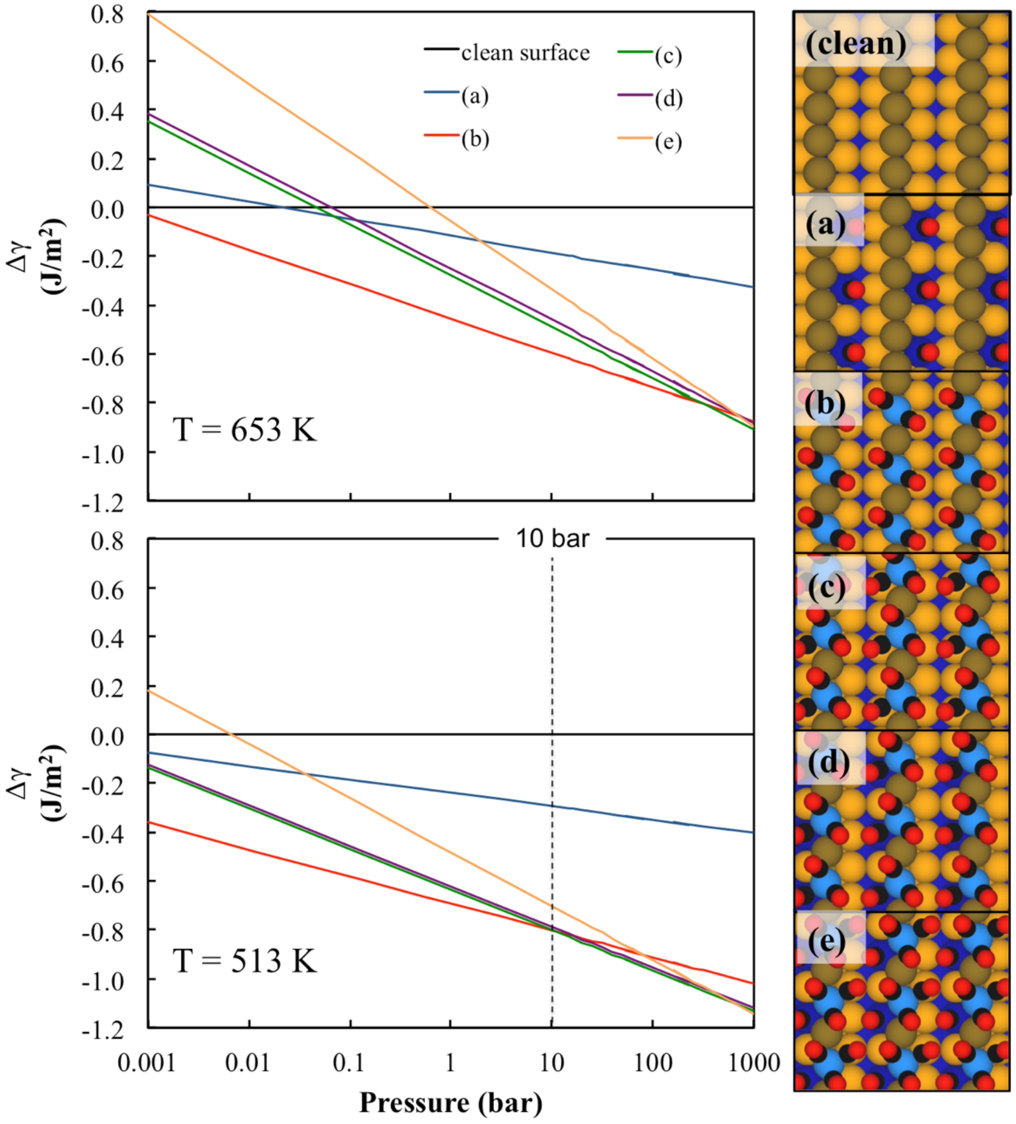


Figure 4.7. CO/Cu/Co $(10\bar{1}2)$ phase diagram of minimal DFT energy configurations for each CO coverage tested here at 513 K and 653 K. The vertical dotted line is placed at its marked pressure to delineate a phase transition. It should be noted that the green line corresponding to (c) is hidden behind the purple line corresponding to (d) in the bottommost graph.

As can be seen in Figure 4.7, at 513 K, the clean Cu-terminated surface represents the least favorable thermodynamic situation under all pressure conditions (except of course for very low pressures where the case of a surface fully covered with CO is the least stable). On the other hand, the geminal dicarbonyl configuration on the reconstructed surface for which every other

Cu atom along the step ridge is replaced by Co (Fig. 4.7(b)) is thermodynamically most favorable well beyond pressures where our assumption of ideal gas behavior breaks down (this occurs for these temperatures around CO's critical pressure of 35 bar). Therefore, no phase transformation is expected at 513 K – the geminal dicarbonyl Co at steps will persist. At 653 K, the clean surface is again not stable at pressures above 1 mbar. At all pressures examined at this temperature, the geminal dicarbonyl (7(b)) covered surface with 0.16 ML Co is the most stable up to 10 bar. At this pressure, a surface phase transition to the trigeminal carbonyl structure with a favorable surface enrichment of Co up to both 0.16 (7(c)) and 0.33 ML (7(d)) is observed – with the 0.16 ML Co surface being slightly more favorable than the 0.33 ML Co surface. These results imply that even in the presence of a small partial pressure of CO, the Cu-terminated surface will reconstruct and replace Cu for Co mainly along the step edges. The formation of geminal di-and tricarbonyls with a step coverage of 50% is the most characteristic structural feature, and the lower 0.25 ML monocarbonyl coordinated Co surface is never favorable, regardless of pressure.

4.3.5 The Importance of Surface Phonon Modes

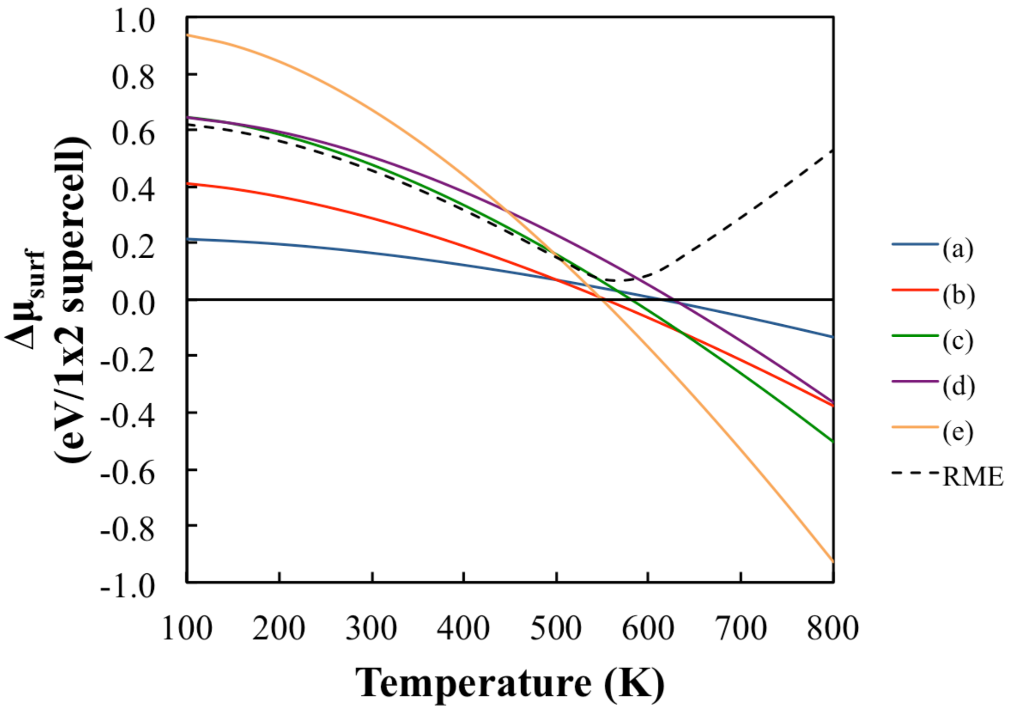


Figure. 4.8. The Cu/Co($10\bar{1}2$) surface chemical potential change (given in eV per p(1×2) supercell) for the minimum DFT energy Co/CO configurations studied here as a function of temperature. A common practice is to assume this value is zero or at least negligibly close to zero. The black dashed line shows the root mean error (RME), or root mean deviation from a value of zero, for all configurations. This error is as great as ~0.6 eV/p(1×2) supercell and as low as ~0.03 eV/p(1×2) supercell depending on the temperature and does not appear to be bounded.

We now justify our decision to include the surface phonon modes in our free energy calculations. It is common practice to assume that the surface phonon modes are unperturbed by adsorbates, which for our system would be essentially assuming that $\mu^{vib}_{Y_{CO},X_{Co}}(T) - \mu^{vib}_{clean}(T) = \Delta\mu_{surface}$ in eq. (5) is zero, and thus all entropy change is due to the loss of gas phase CO degrees of freedom. We have therefore calculated this difference explicitly for the minimum DFT energy Cu/Co($10\bar{1}2$) surface configurations across a wide range of temperatures and plotted these values in Figure 4.8. As can be seen, there are some very significant deviations from zero, especially at higher CO coverages and low temperatures. By calculating a root mean difference (or root mean error, RME) for all the configurations, we also find that the mean deviation from zero is considerable across all temperatures, with a minimum of ~0.03 eV/p(1x2) supercell around 550 K. We report this value on a "per p(1x2) supercell" basis (~0.30 nm^2) because computational work often uses cells of this size (a Cu/Co(0001) p(2x2) supercell is ~0.22 nm^2) and is thus intended to be more intuitively understood than normalization on a per unit area basis would be. Disconcerting as well, the RME does not appear to be bounded, monotonically increasing as the temperature increases. We have plotted this same value for the flat CO/Cu/Co(0001) system as well and the deviation from zero is even more marked (see Figure A4). These deviations thus lead to considerable error in the assumption that the surface phonon modes are negligible, and further, could possibly call into question their summary neglect in other systems as well. For now, we can only speculate that these profound deviations in $\Delta\mu_{surface}$ from zero are due to the complexity of our CO adsorbed systems (e.g. geminal di- and tri- carbonyl Co formation) and/or the CO-induced reconfiguration of the surface (i.e. Co chemical "pumping").

We direct the reader to section X of Appendix A for discussion concerning DFT error in these vibrational mode calculations, which has preliminarily been shown to result in Gibbs free energy errors that are within normal DFT error. Also, for reference, we provide our VASP-calculated vibrational modes in the form of a vibrational density of states (aka "spectral density function") in Appendix A (Figures A6 and A7) where the aforementioned deviations in the modes themselves can be clearly seen. Determining the source of these deviations is a point we will explicitly investigate in future work.

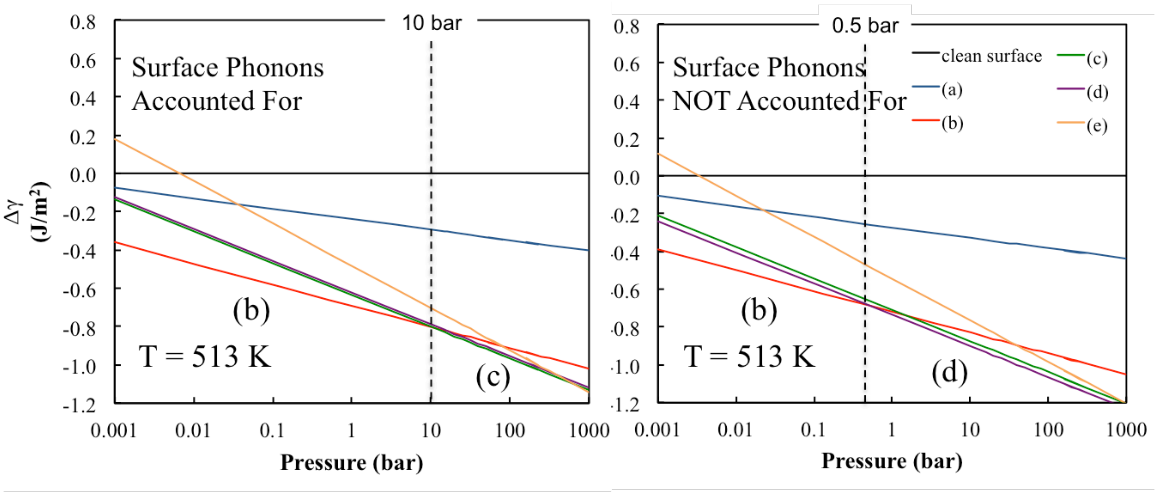


Figure 4.9. Comparison of phase diagrams constructed for the Cu/Co$(10\bar{1}2)$ surface that have had surface phonon modes accounted for vs. unaccounted for. To highlight the largest differences, the low energy surface phase (bold lettering in parenthesis) is shown in each region delineated by the vertical dashed lines. The legend corresponds to the configurations shown in Figure 4.7.

We illustrate the propagation of the aforementioned errors by constructing a phase diagram of the Cu/Co$(10\bar{1}2)$ surface wherein surface phonons have been neglected, and show this side-by-side with the phase diagram from Figure 4.7, which was constructed with their inclusion. This is presented in Figure 4.9. At 653 K, the relevant parts of the phase diagrams are by coincidence very similar and are not included here. However, comparing the phase diagrams at 513 K illustrates the potential risks in neglecting surface phonon modes rather well. We have placed within the phase regions, delineated by the vertical dashed lines, the minimum energy surface phase. At low pressure, both methods predict that configuration (b), the geminal dicarbonyl at the step site is the most favorable. However, the pressures at which the phase transitions occur are remarkably different (0.5 bar vs. 10 bar). Not only this, but different configurations are predicted after this transition: with phonon modes accounted for, the geminal tricarbonyl is more stable than the CO coverage equivalent with 0.50 ML surface Co, but if these modes are not included, the opposite is predicted, albeit within a fairly small margin (< 0.3 eV).

Further, the surface free energy values themselves are also impacted, reflecting the neglect of $\Delta\mu_{surface}$ in the calculation of the phase diagram. A similar comparison for the CO/Cu/Co(0001) surface can be found in Figure A5, where the juxtaposition is even more stark. As a result, neglecting the surface phonon modes in our free energy calculations would result in unacceptable errors.

4.4. Perspectives

Our theoretical calculations of the CO adsorption on the stepped Cu/Co($10\bar{1}2$) have brought forth conclusive results in line with experimental data for CoCu mixed metal catalysts as used in the CO hydrogenation to higher alcohols[30, 31, 34, 48, 49]. The ($10\bar{1}2$) crystallography of our slab model was chosen since early studies showed that pure Co($10\bar{1}2$) is a stable surface when repeatedly cycled through the hcp ↔ fcc phase transition[41] – it may therefore be regarded as an excellent model for a stepped plane at the surface of Co nanoparticles encountered in real Co-based catalysts. It is very interesting that the stepped slab Cu/Co model used in our study clearly favors a Cu terminated surface with the Co underneath just as our previously studied flat slab model did[23]. This is in agreement with the experimental finding that Co-Cu mixed metal oxalates – the preferred catalyst precursor in our laboratory - decompose thermally by self-assembling into active Co@Cu core-shell catalysts. The present theoretical study also shows that the Cu-terminated surface is not stable when CO gas adsorption occurs. Instead, Co atoms are swapped with Cu until one out of two step atoms are replaced by Co. CO molecules are adsorbed up to 0.50 ML at these Co atoms. Cu, on the other hand, acts as a non-adsorbing metal spacer enabling the formation of geminal Co carbonyls. It might be suspected that such geminal carbonyls lead to Co atom disruption similar to what time-dependent atom-probe studies with a pure Co nanoparticle surface suggested[47]. Our theoretical calculations for the stepped Cu/Co($10\bar{1}2$) surface with straight-chain step atoms indicate that such disruption is non-facile. This is not too surprising since step atoms along [$\bar{1}2\bar{1}0$] are sevenfold coordinated. It would be most interesting to check the thermodynamic feasibility of this process for a stepped surface with kink-step arrangements (coordination number of six) since the disruption is anticipated to be favored by decreasing both the steric constraints and the number of next-nearest Me-Me bonds to be broken. Clearly, Me-Me disruption with kink site liberation and diffusion of adsorbed $Co(CO)_{2,3}$ moieties would be a key step of a reconstruction process towards signature structures capable of dissociating the CO molecule as observed experimentally[48]. In this context, we should note that we have not been able to identify any direct CO dissociation on Cu/Co($10\bar{1}2$) so far. While CO dissociation must not necessarily lead to Fischer-Tropsch active surface carbon, it is believed that its occurrence helps form and stabilize the catalytically active surface phase enabling CO hydrogenation to hydrocarbons and oxygenates. Our future research efforts will focus on model studies with surface structures capable of breaking the C-O bond and providing energetically favorable pathways for the CO hydrogenation.

4.5. Summary

We have presented in this paper the effect that CO has on a stepped Cu/Co($10\bar{1}2$) surface configuration. We also present a case for the inclusion of surface phonon modes in free energy calculations. The major findings can be delineated as follows:

- Clean Cu/Co($10\bar{1}2$), much like the clean Cu/Co(0001) surface, is energetically driven to segregate completely into a Cu shell atop a Co slab (mimicking the observed Co@Cu core-shell structure in experiment, and congruent with the known thermodynamic miscibility of Cu in Co). Also, leaving Cu in the lower coordinated step sites is more energy conservative than placing Co in those same step sites.

- The Cu/Co($10\bar{1}2$) surface is enriched with Co to at most 0.33 ML as a result of CO adsorption. CO-induced chemical "pumping" preferentially places Co at the step sites with up to 50% of the straight-chain step Cu atoms replaced by Co atoms. In the most favorable configurations, the Cu persists at the terrace sites adjacent to the step sites of the Cu/Co($10\bar{1}2$) surface.
- Analysis of phase diagrams at 513 K and 653 K shows that the Cu/Co($10\bar{1}2$) surface will at most temperatures be dominated by geminal carbonyl Co at the steps, coincident with a Co surface concentration of 0.16 ML and 50% of the available steps sites, specifically. At higher pressures, we see the formation of a trigeminal carbonyl Co at the steps but Co enrichment is by and large stalled at 0.16 ML – with the available steps still enriched to only 50%. Enrichment of the step sites past 50% appears to be highly unfavorable.
- Surface phonon modes after adsorption are shown to be greatly perturbed in our system, and have thus been included in our free energy calculations. We show the range of the error in assuming these modes are negligible and also find that this error is unbounded. We further show how such errors impact our calculated phase diagrams.

Acknowledgments

This work was supported by institutional funds provided to J.S.M. from the Voiland School of Chemical Engineering and Bioengineering and the National Science Foundation under contract No. CBET-1438227. It was also partially funded by USDA/NIFA through Hatch Project #WNP00807 titled: "Fundamental and Applied Chemical and Biological Catalysts to Minimize

Climate Change, Create a Sustainable Energy Future, and Provide a Safer Food Supply". G.C., Seattle Chapter ARCS Fellow, gratefully acknowledges financial support from the Achievement Rewards for College Scientists foundation. We would also like to give our sincere thanks to Dr. Alyssa Hensley and Dr. Fanglin Che for productive discussions and helpful critiques.

REFERENCES

[1] G.-M. Schwab, Discuss. Faraday Soc. 8 (1950) 166-171.

[2] A. Couper, D. D. Eley, Discuss. Faraday Soc. 8 (1950) 172-184.

[3] D. A. Dowden, Journal of the Chemical Society (Resumed) (1950) 242-265.

[4] J. H. Sinfelt, J. L. Carter, D. J. C. Yates, J. Catal. 24 (1972) 283-296.

[5] W. M. H. Sachtler, P. Van Der Plank, Surf. Sci. 18 (1969) 62-79.

[6] P. van der Plank, W. M. H. Sachtler, J. Catal. 7 (1967) 300-303.

[7] J. H. Sinfelt, J. Catal. 29 (1973) 308-315.

[8] F. Studt, F. Abild-Pedersen, Q. Wu, A. D. Jensen, B. Temel, J.-D. Grunwaldt, J. K. Nørskov, J. Catal. 293 (2012) 51-60.

[9] D. B. Liang, G. Abend, J. H. Block, N. Kruse, Surf. Sci. 126 (1983) 392-396.

[10] V. K. Medvedev, R. Börner, N. Kruse, Surf. Sci. 401 (1998) L371-L374.

[11] D. D. P. Courty, E. Freund, A. Sugier, J. Mol. Catal. 17 (1982) 241-254.

[12] A. Sugier, E. Freund, US 4,122,110, (1978)

[13] A. J. Medford, A. C. Lausche, F. Abild-Pedersen, B. Temel, N. C. Schjødt, J. K. Nørskov, F. Studt, Top. Catal. 57 (2013) 135-142.

[14] V. R. Calderone, N. R. Shiju, D. C. Ferré, G. Rothenberg, Green Chem. 13 (2011) 1950.

[15] L. Gonzalez, R. Miranda, M. Salmerón, J. A. Vergés, F. Ynduráin, Phys. Rev. B 24 (1981) 3245-3254.

[16] P. Roubin, D. Chandesris, G. Rossi, J. Lecante, M. C. Desjonquères, G. Tréglia, Phys. Rev. Lett. 56 (1986) 1272-1275.

[17] K. Garrison, Y. Chang, P. D. Johnson, Phys. Rev. Lett. 71 (1993) 2801-2804.

[18] M. G. Samant, J. Stöhr, S. S. P. Parkin, G. A. Held, B. D. Hermsmeier, F. Herman, M. Van Schilfgaarde, L. C. Duda, D. C. Mancini, N. Wassdahl, R. Nakajima, Phys. Rev. Lett. 72 (1994) 1112-1115.

[19] X. Chuanyun, Y. Jinlong, D. Kaiming, W. Kelin, Phys. Rev. B 55 (1997) 3677-3682.

[20] R. Pentcheva, M. Scheffler, Phys. Rev. B 61 (2000) 2211-2220.

[21] J. Wang, G. Wang, X. Chen, W. Lu, J. Zhao, Phys. Rev. B 66 (2002) 014419.

[22] M. Ø. Pedersen, I. A. Bönicke, E. Laegsgaard, I. Stensgaard, A. Ruban, J. K. Nørskov, F. Besenbacher, Surf. Sci. 387 (1997) 86-101.

[23] G. Collinge, Y. Xiang, R. Barbosa, J.-S. McEwen, N. Kruse, Surf. Sci. 648 (2016) 74-83.

[24] A. V. Ruban, H. L. Skriver, J. K. Nørskov, Phys. Rev. B 59 (1999) 15990-16000.

[25] A. U. Nilekar, A. V. Ruban, M. Mavrikakis, Surf. Sci. 603 (2009) 91-96.

[26] A. Christensen, A. V. Ruban, P. Stoltze, K. W. Jacobsen, H. L. Skriver, J. K. Nørskov, F. Besenbacher, Phys. Rev. B 56 (1997) 5822-5834.

[27] S. Yin, T. Swift, Q. Ge, Catal. Today 165 (2011) 10-18.

[28] J. Su, Z. Zhang, D. Fu, D. Liu, X.-C. Xu, B. Shi, X. Wang, R. Si, Z. Jiang, J. Xu, Y.-F. Han, J. Catal. 336 (2016) 94-106.

[29] X.-C. Xu, J. Su, P. Tian, D. Fu, W. Dai, W. Mao, W.-K. Yuan, J. Xu, Y.-F. Han, J. Phys. Chem. C 119 (2015) 216-227.

[30] N. D. Subramanian, G. Balaji, C. S. S. R. Kumar, J. J. Spivey, Catal. Today 147 (2009) 100-106.

[31] G. Prieto, S. Beijer, M. L. Smith, M. He, Y. Au, Z. Wang, D. A. Bruce, K. P. de Jong, J. J. Spivey, P. E. de Jongh, Angew. Chem. Int. Ed. 53 (2014) 6397-6401.

[32] E. A. Lewis, D. Le, A. D. Jewell, C. J. Murphy, T. S. Rahman, E. C. Sykes, Chem. Commun. 50 (2014) 6537-9.

[33] E. A. Lewis, D. Le, A. D. Jewell, C. J. Murphy, T. S. Rahman, E. C. H. Sykes, ACS Nano 7 (2013) 4384-4392.

[34] Y. Xiang, V. Chitry, P. Liddicoat, P. Felfer, J. Cairney, S. Ringer, N. Kruse, J. Am. Chem. Soc. 135 (2013) 7114-7117.

[35] K. J. Andersson, F. Calle-Vallejo, J. Rossmeisl, I. Chorkendorff, J. Am. Chem. Soc. 131 (2009) 2404-7.

[36] G. T. K. K. Gunasooriya, A. P. van Bavel, H. P. C. E. Kuipers, M. Saeys, Surf. Sci. 642 (2015) L6-L10.

[37] A. Banerjee, A. P. van Bavel, H. P. C. E. Kuipers, M. Saeys, ACS Catal. 5 (2015) 4756-4760.

[38] K. Reuter, M. Scheffler, Phys. Rev. B 65 (2001).

[39] R. B. Getman, Y. Xu, W. F. Schneider, J. Phys. Chem. C 112 (2008) 9559-9572.

[40] Q. Ge, M. Neurock, J. Phys. Chem. B 110 (2006) 15368-15380.

[41] K. A. Prior, K. Schwaha, M. E. Bridge, R. M. Lambert, Chem. Phys. Lett. 65 (1979) 472-475.

[42] N. Kruse, Surf. Sci. 178 (1986) 820-830.

[43] J. Tang, L. Deng, S. Xiao, H. Deng, X. Zhang, W. Hu, J. Phys. Chem. C (2015).

[44] A. Lopes, G. Tréglia, C. Mottet, B. Legrand, Phys. Rev. B 91 (2015) 035407.

[45] G. Zvejnieks, A. Ibenskas, E. E. Tornau, J. Alloys Compd. 649 (2015) 313-319.

[46] B. C. Han, A. Van der Ven, G. Ceder, B. Hwang, Phys. Rev. B 72 (2005) 205409.

[47] N. Kruse, J. Schweicher, A. Bundhoo, A. Frennet, T. Visart de Bocarmé, Top. Catal. 48 (2008) 145-152.

[48] Y. Xiang, R. Barbosa, N. Kruse, ACS Catal. 4 (2014) 2792-2800.

[49] Y. Xiang, R. Barbosa, X. Li, N. Kruse, ACS Catal. 5 (2015) 2929-2934.

CHAPTER FIVE:

DISSOLUTION OF CoCu CATALYST STEP DEFECTS BY Co SUBCARBONYL FORMATION

Greg Collinge[a], Norbert Kruse[a,d,], and Jean-Sabin McEwen[a,b,c,d,]**

[a] The Gene & Linda Voiland School of Chemical Engineering and Bioengineering, Washington State University, Pullman WA 99164

[b] Department of Physics and Astronomy, Washington State University, Pullman, WA 99164

[c] Department of Chemistry, Washington State University, Pullman, WA 99164

[d] Institute for Integrated Catalysis, Pacific Northwest National Laboratory, Richland, WA, 99352

** Corresponding authors:*

Norbert Kruse; 509-335-6601 (phone), 509-335-4806 (fax), norbert.kruse@wsu.edu

Jean-Sabin McEwen; 509-335-8580 (phone), 509-335-4806 (fax), js.mcewen@wsu.edu

Abstract

In CoCu-based Fischer-Tropsch catalysis, the as-prepared nanoparticles, if allowed to self-assemble, exhibit a Co@Cu core-shell morphology that would render the catalyst inactive for CO hydrogenation. Therefore, a chemical reconstruction has to occur to create the catalytically active phase. While some of the thermodynamically-imposed driving forces for reconstruction have been identified and kinetic mechanisms experimentally probed, a thorough theoretical understanding on the molecular events has yet to be developed. Here, we employ a first-principles statistical mechanics approach to show that the reconstruction of CoCu in CO atmospheres is likely accomplished via subcarbonyl (multiple bonded CO) formation at the step and kink sites of CoCu catalysts. We find that the CO-induced antisegregation of subsurface Co

atoms to step sites and the subsequent rupturing of Co subcarbonyls from these sites is thermodynamically feasible under experimentally-relevant CO pressures and temperatures. The results suggest that Co tricarbonyl formation along with its rupturing and diffusion onto the terraces is responsible for reconstruction. These Co tricarbonyls are shown to favorably dimerize, suggesting a potential route for nanoisland formation and morphological changes. Our results illustrate a strong correlation to surface carbonyl and inorganic complex chemistry of Co metal.

5.1. Introduction

Surface reconstruction is a reaction phenomenon known to alter the activity and selectivity of catalyst particles. Seminally, Leidheiser and Gwathmey[1] and later L. D. Schmidt and coworkers[2-4] were among the first authors showing that metal surfaces may either suffer chemically-induced (chemical) or thermally-induced (mechanical) reconstruction. A heavily studied example that demonstrates the differences between the two cases is the "(1×2) missing row reconstruction" of a clean Pt(110) single-crystal surface[5, 6] in which every second row of atoms along the $[1\bar{1}0]$ direction is missing. Exposing a (1x2) Pt(110) crystal surface to CO inverts the reconstruction process and reestablishes the bulk-truncated (1x1) form through a homogeneous nucleation process in which atoms move over a few lattice sites[7-9]. Chemical processes were thereafter shown to play a large role in the reconstruction of many single-crystal systems and typically employed STM (Scanning Tunneling Microscopy) and LEED (Low Energy Electron Diffraction) methods to observe these phenomena[10-18]. In terms of nanoparticles, FIM (Field Ion Microscopy) was shown to be a viable methodological approach since the field emitter samples used in FIM largely resemble a single, hemispherically-shaped, nano-sized catalyst particle. Local reconstructions of small-size facets were observed using FIM and proved the validity of the approach[19, 20]. Importantly, a CO-induced morphological reconstruction of transition metal particles toward a cubo-octahedral shape was imaged with atomic resolution[21-23]. In the case of Ni and Rh field emitters, the morphological reshaping was correlated with the formation of $Ni(CO)_x$[24] and $Rh(CO)_x$ (x=1-3)[23, 25], respectively, using atom-probe mass spectroscopy. IR (Infrared) and/or Extended X-ray Absorption Fine Structure (EXAFS) measurements with supported Ni[26] and Rh[27-30] nano-sized particles provided additional information on the mobility of high-index subcarbonyl species. Due to their mobility, they may ultimately cause the dissolution of the nanoparticles. In other cases, including Fischer-Tropsch active Ru and Co metals, no such dissolution occurred despite subcarbonyls being detectable in considerable amounts[31, 32]. STM and atom probe mass spectrometry with either low-index Co single crystal surfaces[33] or Co field emitters[32], respectively, posited subcarbonyl species to be the source of surface reconstruction. Theoretically, however, these mechanistic propositions have yet to be directly supported and detailed to provide a sound picture of the reconstruction processes.

CO-induced nanoparticle reconstruction is a well-studied phenomenon in other metal systems with some notable recent reports on nanoparticle formation on Cu(111)[34] and Pt dimer stabilization on $Pt/Fe_3O_4(001)$[35] upon exposure to CO. We also note that the relevance of such reconstructions on bimetallic systems has been demonstrated both experimentally[36] and computationally[37].

We have been interested in Co and CoCu-based Fischer-Tropsch (FT) catalysts for some time. These metals have been experimentally shown to result in reconstruction and/or reconfiguration upon exposure to CO and H_2 under CO hydrogenation conditions[33, 38-42]. While some of the thermodynamic driving forces for these phenomena have been explored for both Co[43] and CoCu[39, 44], the processes responsible for reconstruction have yet to be theoretically elucidated. Based on the wealth of evidence presented in the aforementioned studies and the prediction of geminal di- and tri-carbonyls on CoCu surfaces[39, 44], we hypothesize that the formation of diffusive (or "mobile") Co subcarbonyls is the means by which reconstruction occurs on CoCu nanoparticles. The formation of subcarbonyls has been demonstrated, but the experimental evidence cannot always distinguish between immobile (highly metal-bound) and mobile (minimally metal-bound) metal subcarbonyls. This study aims at identifying, using density functional theory (DFT) calculations and statistical mechanical methods, the parameter space under which carbonyl formation and therefore reconstruction is possible on CoCu catalysts. Carbonyl formation has long been assumed to involve kink sites under high coverages of CO since metal-metal bonds can be repetitively ruptured in such sites and therefore lead to a reconstruction of the surface. However, this does not exclude step sites (layer edges), which can also bind multiple CO molecules. We therefore begin our study with a stepped CoCu surface: Cu/Co(755). This surface has a long 6-atom (111) terrace and a 1 atom high (100) step and is ideal for testing the feasibility of geminal Co di-, tri-, and tetracarbonyl rupturing from CoCu step sites. We provide our DFT parameters and methods in the Supplemental Information (SI) along with model justification.

5.2. Results

5.2.1 Adsorption Trends with CO Coverage

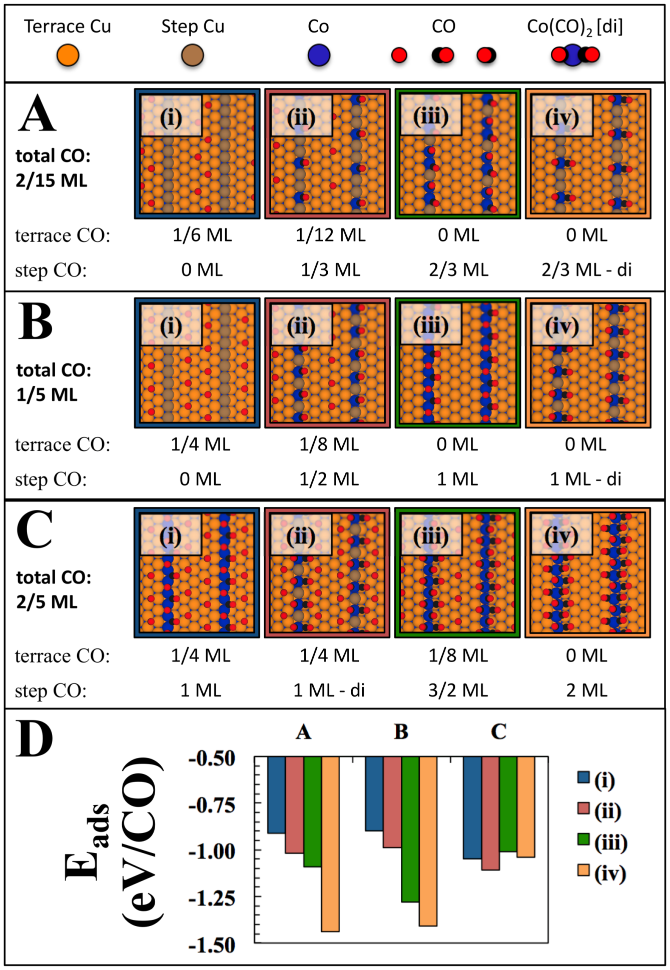


Figure 5.1. CO adsorption energy of the three highest coverage systems tested: A, 0.25 ML, B, 0.67 ML, and C, 1.00 ML. The associated terrace and step coverages are shown below each structure. Panel D provides the calculated CO adsorption energy (in eV/CO) for each system, with bar graph colors corresponding to the color of each structure's border. In the associated top-down structures, the blue spheres are Co, the orange sphere are terrace Cu, the brown/olive colored spheres are step Cu, the black spheres are C, and the red spheres are O. See Figure B2 and B3 for all systems tested.

The maximum stoichiometry of CO adsorption on terrace sites is presumed to be one CO per site while the maximum stoichiometry on step sites is four CO per site. Here, we separate step coverage (θ_{st}) from terrace coverage (θ_t) and relate these to the total coverage (θ) through their (nominal) relative proportions ($\frac{1}{5}\theta_{st} + \frac{4}{5}\theta_{t} = \theta$). With this in mind, we compute CO adsorption energies on Cu/Co(755) at various coverages and configurations wherein we keep total coverage constant and vary the step coverage from 0.00 ML to its associated maximum step coverage. We also consider converting a step Co monocarbonyl to step Co dicarbonyl where possible. Figure B2 and B3 show all six total coverages tested, which range from 0.050 ML to 0.400 ML (the maximum allowable for a final 2.00 ML step coverage with 0.00 ML terrace coverage). In all cases, we calculate the CO adsorption energy with CO in the gas phase and the clean, Cu-terminated Cu/Co(755) surface as reference. We note also that we treat the presence of CO at step sites as inducing Co/Cu antisegregation: nearby subsurface Co is swapped with the step site Cu of interest. No additional Co or Cu are added or removed in these systems, eliminating the need for a Co or Cu atom reference. This mimics experimental conditions where the Co and Cu nanoparticle concentrations are fixed after synthesis.

Remarkably, the energetic trends at each coverage are shown to be identical over all total coverages up to and including 0.200 ML (Figures B4, 5.1A, and 5.1B): as CO molecules are moved from the terrace to the step and then converted to dicarbonyls, the adsorption energy is steadily increased. In fact, Co dicarbonyl formation is shown to be significantly more favorable than dispersing an equivalent amount of CO over the steps as Co monocarbonyls, which can be seen in Figures 5.1A and 5.1B. At a step coverage of 2.00 ML (Figure 5.1C), forming a higher concentration of dicarbonyls (one dicarbonyl on each step site) is no longer most favorable. Instead, a configuration wherein a dicarbonyl forms at every other step site with excess CO spilling over to the terrace sites is calculated to be the most favorable (Figure 5.1-C(ii)). The CO adsorption energy for each dicarbonyl is also shown to be identical (-1.43 ± 0.02 eV) up to 0.200 ML (Figures B4, 5.1A, and 5.1B) indicating that lateral interactions between dicarbonyls are negligible up to this point. As such, if one assumes entropy contributions from the adsorbed CO are roughly equivalent, this result implies that a CO step coverage of at least 1.00 ML will be thermodynamically most favorable with a dicarbonyl configuration as shown in Figure 5.1B(iv). All results considered, this also shows that dicarbonyl formation saturates at a step coverage of 1.00 ML. However, because we wish to investigate the possibility of rupturing higher index subcarbonyls, which are predicted to exist at high CO partial pressures[44], we move forward with a system that nominally represents a 0.67 ML step coverage of CO (see Figure 5.1A(iv) as reference). This is an ideal model because the associated p(1×3) supercell is relatively small (computationally less burdensome) but still large enough to provide enough terrace space to more thoroughly test subcarbonyl formation at the step sites. We further note that because the energetics don't change as the supercell is deceased in its size along the direction parallel to the step edge until the supercell is only 2 atoms long in that direction (Figure 5.1C), we do not expect that increasing the supercell size in this direction will change the results of this study.

5.2.2 Direct Rupturing Processes

Direct rupturing processes can be seen in Figure 5.2A-5.2C. The left structures depict the optimized initial states (IS) and the right structures the optimized final states (FS). Their DFT-based energy differences (ΔE) are summarized as the green solid line in Figure 5.2G. As can be seen in Figure 5.2G, direct metal-metal bond rupturing due to di- and tricarbonyl formation is very unfavorable (+1.72 eV and +1.34 eV for the processes considered in Figure 5.2A and 5.2B, respectively). Direct rupturing of the tetracarbonyl (Figure 5.2C) is still found to be unfavorable (+0.47 eV) but is remarkably more favorable than the tricarbonyl rupturing process. This unfavorability, across all direct rupturing processes, is explained by further inspection of the FS of the direct rupturing processes (Figures 5.2A-5.2C): the process of rupturing creates an exposed Co atom, which we know is very unfavorable[39]. It is reasonable to assume this atom must be stabilized once exposed and this can be accomplished by re-formation of another geminal Co carbonyl which we will denote as the "kink-IS" (see the structures at the step edge of Figures 5.2D-5.2F). At this point, the kink-IS would be primed for another rupturing event. It is envisioned that kink site rupturing would be much more facile than the initial step site rupturing due to the lower number of metal bonds that must be broken to do so. Thus, step site geminal Co carbonyl rupturing would trigger a chain reaction of fast kink site rupturing events. This chain reaction of kink site rupturing events would continue until either the chemical potential of the subcarbonyls' ultimate FS reached that of the kink-IS, until another process like CO dissociate halted the process, or until newly-formed kink sites were exhausted. This latter possibility describes the effective dissolution of an entire step edge, which is necessary to explain the formation of cubo-octahedral particle shapes from spherical ones as observed earlier[21, 23].

The dissolution of step and kink sites hinges on the favorability of reforming a new geminal Co carbonyl at the kink site in conjunction with the rupturing of the step Co. These kink-IS's can be seen in Figure 5.2D-5.2F, where only the necessary number of CO molecules that are needed to create the kink-IS are added to each system. Their ΔE's are represented in Figure 5.2G as the dashed blue line. The geminal Co dicarbonyl (Figure 5.2D) rupturing with the formation of the kink-IS is 0.74 eV lower in energy than the same system without the formation of the kink-IS, but it is still endothermic overall suggesting that this particular process is still unlikely. However, the geminal Co tricarbonyl (Figure 5.2E) rupturing and kink-IS formation is essentially thermo-neutral (-0.08 eV) and the Co tetracarbonyl (Figure 5.2F) is energetically well below the typical threshold for deeming a process favorable (-0.74 eV). It's important to note that this system seems sensitive to the choice of DFT functional. As can be seen in Figure B3, the energy cost of rupturing appears to systematically raise if we switch to, for example, the vdW-DF functional. However, Table S2 shows that the vdW-DF functional increases the adsorption strength of CO on Cu/Co(755) despite the vdW-DF functional propensity to decrease it for Co(0001)[45, 46] and Cu(111) (we calculate E_{ads} with vdW-DF ~ -0.50 eV compared to E_{ads} with PBE ~ -0.80 eV). As such, the PBE functional was chosen in the present study because of the lower computed adsorption energy values on the Cu/Co(755) surface as compared to the corresponding vdW-DF functional values. For further information the reader is directed to the SI.

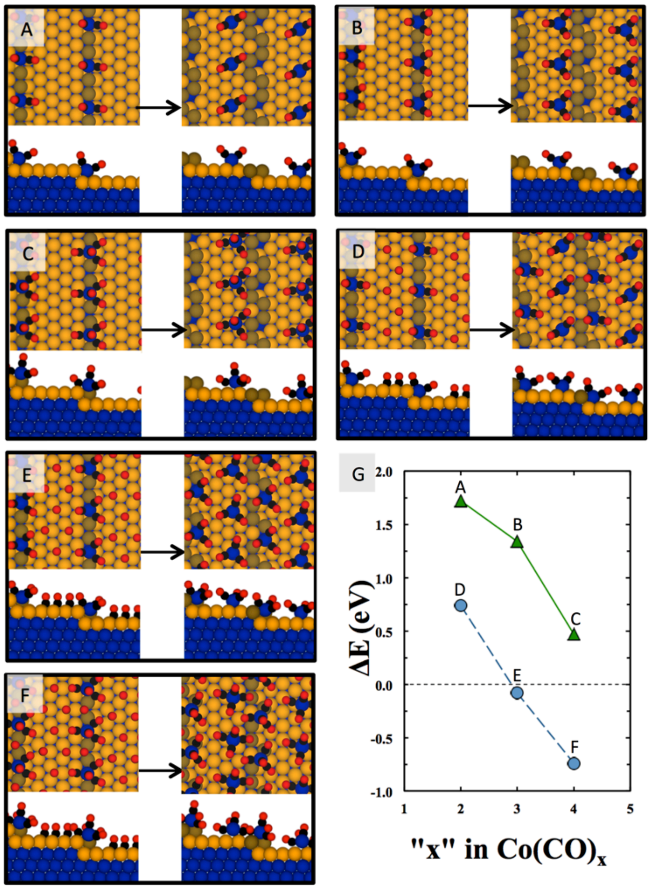


Figure 5.2. (A-F) The six rupturing processes tested: direct rupturing of (A) a geminal Co dicarbonyl, (B) a geminal tricarbonyl, and (C) a geminal Co tetracarbonyl; rupturing and reformation of (D) a geminal Co dicarbonyl, (E) a geminal tricarbonyl, and (F) a geminal Co tetracarbonyl. The graph (G) shows the energy change for each of the (A – F) processes. The color scheme used here is identical to that of Figure 5.1.

5.2.3 Thermodynamic Stability of Ruptured and Unruptured Co(CO)$_x$

Interestingly, the IS and FS structures corresponding to the highly exothermic Co tetracarbonyl rupturing process (Figure 5.2F: the brown lines in Figure 5.3) are some of the least favorable structures when temperature and pressure are accounted for. This highlights the need for caution and diligence when evaluating the favorability of chemical processes based solely on DFT energies alone—the energy lowering effect observed can easily be an artifact of a highly unfavorable IS.

The lowest free energy structures at sub-ambient and ambient pressures in Figure 5.3 are the step site geminal Co dicarbonyl (solid blue line) and step site geminal Co tricarbonyl (solid green line), both without nearby adsorbed CO. At the high pressures relevant to CO hydrogenation, Co tricarbonyl rupturing, represented by the red dashed line, becomes favorable only when the kink-IS is formed (Figure 5.2F: the FS).

By utilizing Equation B1, we calculate the equilibrium constant between the initial geminal Co tricarbonyl and the final ruptured Co tricarbonyl (with the kink-IS formed) to be between ~6,000 and ~50 in the 0.1−3.0 bar pressure ranges. This means that even given the errors inherent in the calculation of the exact free energies and thus propagated to the equilibrium constants (at 573 K, this error is roughly $\pm 6K_e$, or within a magnitude, given an approximate 0.2 eV free energy error), we can still expect there to be a non-negligible number of rupturing events at ambient pressures. Initial step-site rupturing events are expected to be the rate limiting step of the entire dissolution process as kink-IS's more easily rupture, and in this way facile restructuring of the CoCu catalyst would still be possible even at ambient pressures. It should be further noted that such subsequent kink site rupturing events implies that the vacancy left after Co tricarbonyl rupturing from the step will be quite short lived and thus unlikely relevant to the reactivity of the catalyst. For the interested experimentalist, relevant vibrational frequencies for the species listed in Figure 5.2 have been extracted and summarized in Table B3.

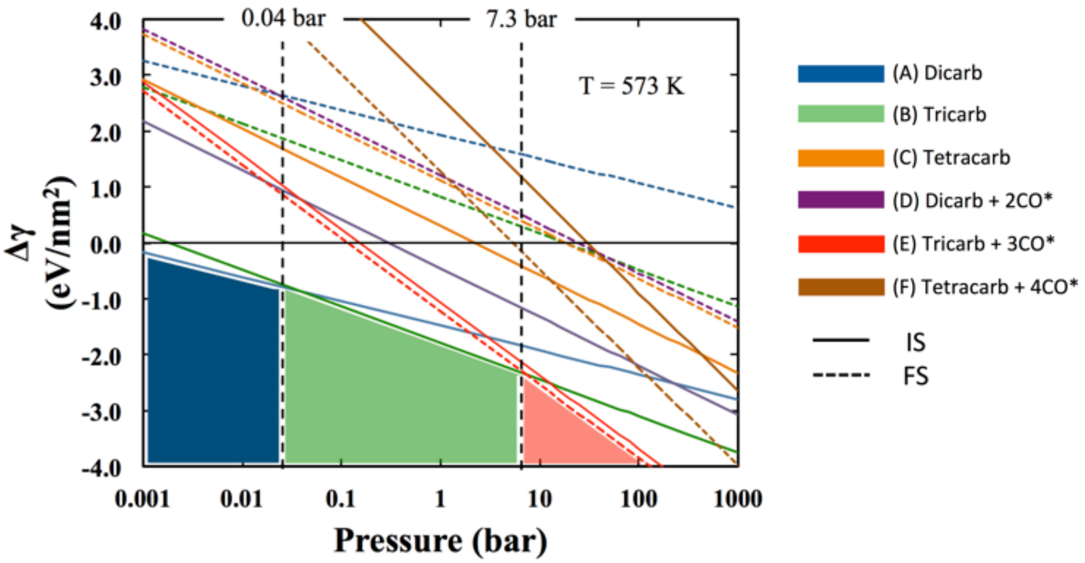


Figure 5.3. Surface phase diagram of all structures shown in Figure 5.2A-5.2F. The vertical dashed lines denote a "phase change" and the corresponding pressure of the change is shown above them. Shaded regions help denote the lowest free energy structures: (from left to right) the Co dicarbonyl IS without nearby adsorbed CO (Figure 5.2A: the left structure), the Co tricarbonyl IS without nearby adsorbed CO (Figure 5.2B: the left structure), and the Co tricarbonyl FS with the kink-IS formed (Figure 5.2E: the right structure).

5.2.4 Dimerization of $Co(CO)_3$: First Steps in Nanoisland Formation

We further investigate the possibility of Co tricarbonyl dimerization once ruptured Co tricarbonyls are formed. This process is depicted in Figure 5.4 where DFT energies relative to that of the initial state (Figure 5.4A) are shown in the accompanying graph. Two stable states (Figure 5.4A and 5.4B) are found as a newly ruptured Co tricarbonyl diffuses toward a previously ruptured Co tricarbonyl. This process is found to be essentially thermoneutral (ΔE_{rxn} ~ 0.00 eV). From here, three possible final dimerized states were tested: dicobalt subcarbonyl complexes with 5, 6, and 7 CO adsorbed; corresponding to the dicobalt penta-, hexa-, and heptacarbonyls seen in Figures 5.4C, 5.4D, and 5.4E, respectively. The dicobalt penta- and hexacarbonyl formations are uphill in energy by only 0.17 eV and 0.18 eV, respectively; while the dicobalt heptacarbonyl formation is uphill by 0.50 eV. Since dimerization will be accompanied by a translational/configurational entropy loss, it is not expected that the dicobalt hexacarbonyl will readily form. However, because a CO is kicked off when the dicobalt pentacarbonyl is formed, it is likely that there is an entropy gain as gas phase CO degrees of freedom are recovered. In fact, adding the DFT energy cost of CO desorption (~0.9 eV) to this process and accounting for the Gibbs free energy gain associated with gas-phase CO degrees of freedom from only rotations and translations (about -1.2 eV at 1 bar and 573 K), this process comes out to be roughly -0.1 eV exergonic. Being even more uphill in energy and having the exact opposite entropy argument as that of the dicobalt pentacarbonyl, the formation of the dicobalt heptacarbonyl is far too endergonic (roughly +0.8 eV) to be feasible.

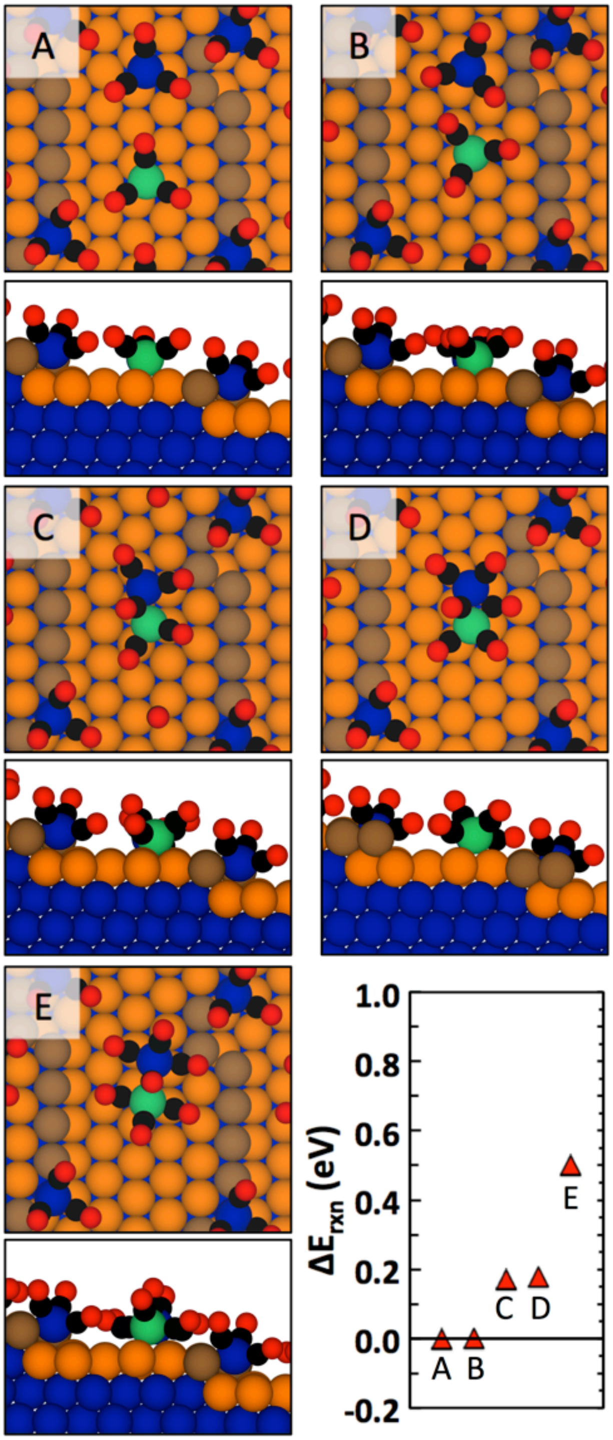


Figure 5.4. Structures (A-E) and relative DFT energy differences (lower right graph) of Co tricarbonyl dimerization reaction steps. All energy values are relative to structure (A). Two stable states, (A) and (B), are found as two Co tricarbonyl diffuse toward each other. Three possible final states are found depending on the number of CO adsorbed to the dicobalt complex: (C) pentacarbonyl, (D) hexacarbonyl, and (E) heptacarbonyl. The color scheme is identical to that used in Figure 5.1 and 5.2 except one Co has been colored green to aid the eye.

5.2.5 CO Adsorption Scan of the Cu/Co(755) Terrace

To examine how our choice to use a 6-atom long terrace in the (755) facet to approximate a (111) terrace might affect our results, we perform a scan of the CO adsorption energy across the terrace as a function of distance from the step edge and present the results in Figure 5.5. The central terrace sites of the Cu/Co(755) surface are shown to converge quite well to the approximate CO adsorption energy on Cu/Co(0001).[39] Also, we see that other than the step itself, CO adsorption at all other locations on the terrace is weaker than on the CuCo basal plane. Thus, we can assert that our results are at worst an underestimate of the thermodynamic stability of the ruptured adsorbates and increasing the length of the terrace would at best increase stability of adsorbed Co subcarbonyls and associated dimers, trimers, etc.

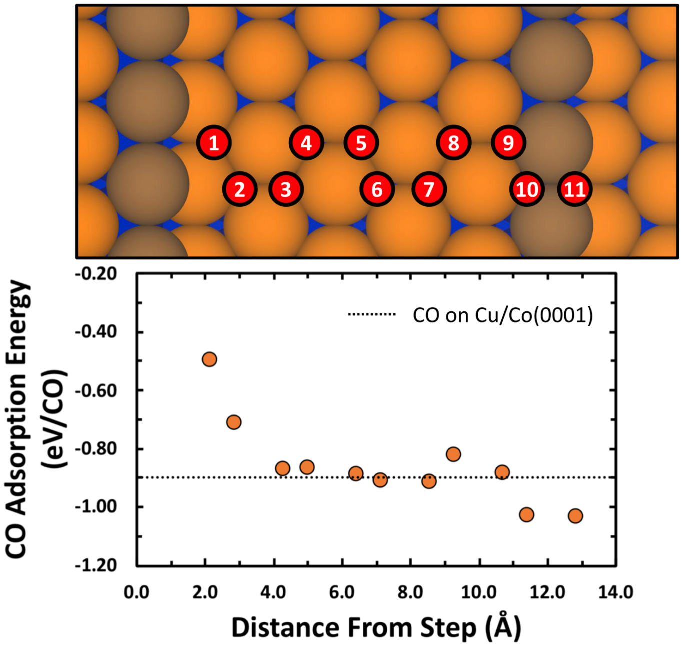


Figure 5.5. CO adsorption energy on Cu/Co(755) as a function of CO distance from the step edge of the cell. These calculations were performed in the p(1×3) supercell. An approximate average CO adsorption energy on the Cu/Co(0001) surface is shown in the graph as a dashed line, illustrating where the electronics of the Cu/Co(755) terrace converge approximately to the fcc-equivalent Cu/Co(0001) surface[39]. Above the graph, the corresponding CO adsorption sites are shown. The color scheme used is identical to that in Figure 5.2.

As can be seen in Figure 5.5, at roughly 4 Å from the edge (site 3), until roughly 9 Å from the edge (site 7), CO adsorption is essentially equivalent to adsorption on Cu/Co(0001). Terrace sites 1, 2, 8, and 9 are actually higher in energy (i.e. weaker adsorption energy) than these (0001)-equivalent sites. Sites 10 and 11 are formally step sites. In this study, all molecules adsorbed on the terrace have been kept as closely as possible to the (0001)-equivalent sites. However, because of the size of the Co subcarbonyls it is possible that these higher energy terrace sites could affect the results by destabilizing the subcarbonyls. In this way, the results are at worst an underestimate of these species stability and thus would be even more likely to exist in real CoCu catalysts. We note also that these results compare interestingly to similar lattice gas modeling and temperature programmed desorption work done by Payne and Kreuzer who implicitly assumed that all terrace sites of stepped surfaces, like the (755) surface, would be equivalent[47]. The results in Figure 5.5 show that this is clearly not always the case.

5.2.6 Diffusion of Co(CO)$_3$

To explicitly show that diffusion on this surface is indeed facile, we present the minimum energy pathway (MEP) for diffusion from (hcp) hollow site to adjacent (fcc) hollow site on the terrace in Figure 5.6. The calculation was carried out on a p(1×6) supercell model where a Co(CO)$_3$ is ruptured and its corresponding kink-IS (geminal Co(CO)$_3$) reformed. The relevant diffusion pathway is set such that the initial state (IS in Figure 5.6) and final state (FS in Figure 5.6) stay as close as possible to the Cu/Co(0001)-equivalent terrace sites. As can be seen, the diffusion MEP is incredibly shallow and a very low energy transition state (TS in Figure 5.6) with an activation barrier of ~0.018 eV is found. The somewhat endothermic reaction energy (~0.015 eV) is due to either the fcc hollow site being less stable by this amount or due to the fcc hollow site interacting with the weaker (with respect to CO adsorption) terrace sites that are closer to the step edge. Since activation energy tends to lower with lowering reaction energy, we can say that this is likely an over-estimated activation barrier, as well. Not only does this demonstrate facile diffusion of these species, but it further justifies the use of a 2-D free translator partition function for translational motion. We note that an intermediate bridge site (Br in Figure 5.6) is found to be stable to within the force tolerances used (0.01 eV/Å) as well.

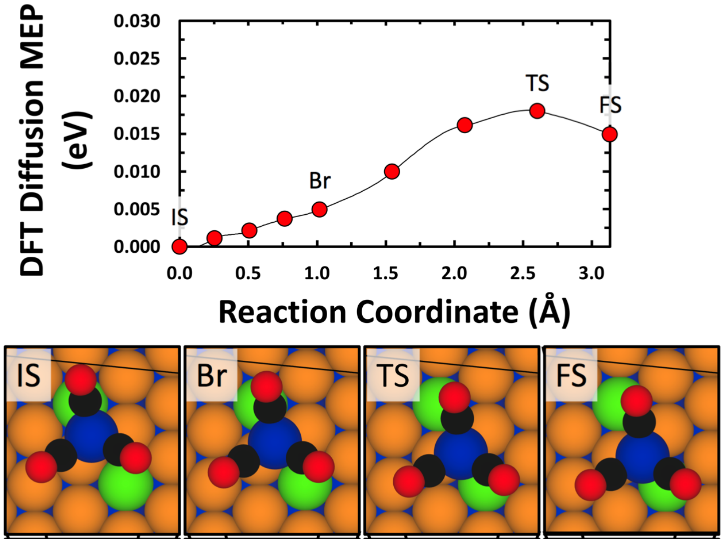


Figure 5.6. Minimum energy pathway for Co tricarbonyl diffusion across the central terrace of the Cu/Co(755) system; the corresponding structures for the labeled images are shown below the graph where "IS," "Br," "TS," and "FS" signify the initial state (an hcp hollow site), an intermediate bridge site, the transition state, and the final state (an fcc hollow site), respectively. The color scheme used here is identical to that used in Figure 5.2 except two Cu atoms have been colored green to help aid the eye along the diffusion pathway.

3. Summary and Conclusions

With all this information in hand, we can propose a scheme for step and kink site dissolution, as shown in Figure 5.7. In this snapshot, we can see that a rupturing event occurred originally at point A and subsequently induced the dissolution process that is still ongoing at point B, where the most recently ruptured kink-IS can be seen at point C. The dissolution process creates mobile Co subcarbonyls, as shown at point D (amongst others). Some of these Co subcarbonyls can dimerize, as seen at point E, or, as we envision it, even trimerize as posited at point F, and begin forming Co nanoislands with the liberation and subsequent entropy gain of additional gas phase CO driving each Co tricarbonyl addition. In the meantime, another geminal Co tricarbonyl could independently form at another location on the same step as at point G or on the new step as at point H, where a new geminal Co tricarbonyl ruptures and starts another dissolution process. While the exact order and structure of the catalyst shown here is speculative, we assert that the individual details that lead to this picture are well grounded in the results that were presented. In the aforementioned way, we have gained deep insights into the reconstruction mechanism experienced by a CoCu catalyst during the transient build-up of the catalytically active surface during CO hydrogenation reactions.

In conclusion, we have shown here that the formation of geminal Co tricarbonyls at step sites, predicted to occur at pressures and temperatures relevant to CO hydrogenation over CoCu catalysts, will induce the dissolution of the associated step edge. The formation of stable Co tricarbonyls is predicted to be the driving force for this process. Dicobalt subcarbonyl complex formation was shown to be feasible and suggest a route toward Co nanoisland formation: structures studied extensively in recent work[48-50]. While the ultimate fate of the Co subcarbonyls are not precisely known, the reconstruction of the catalyst is an unavoidable consequence of the process presented here as the diffusive Co tricarbonyls have opportunity to find a stable final configuration. Such changes in aggregate would result in morphological changes of nano-sized metal particles. While the exact details involved will differ for other systems, we regard this result as a proof-of-concept for a general atomistic picture of the underlying reaction mechanism for well-known chemically-induced nanoparticle reconstruction:

through low-coordinated metal site dissolution brought about by formation and diffusion of ligand-stabilized metal complexes. Given the experimental observation of similar carbonyl complexes on Ni, Ru, Rh and Co field emitter tips,[23, 24, 31, 32] and similar thiol complexes on Au surfaces,[51-53] the type of process reported here may turn out to be universal across systems.

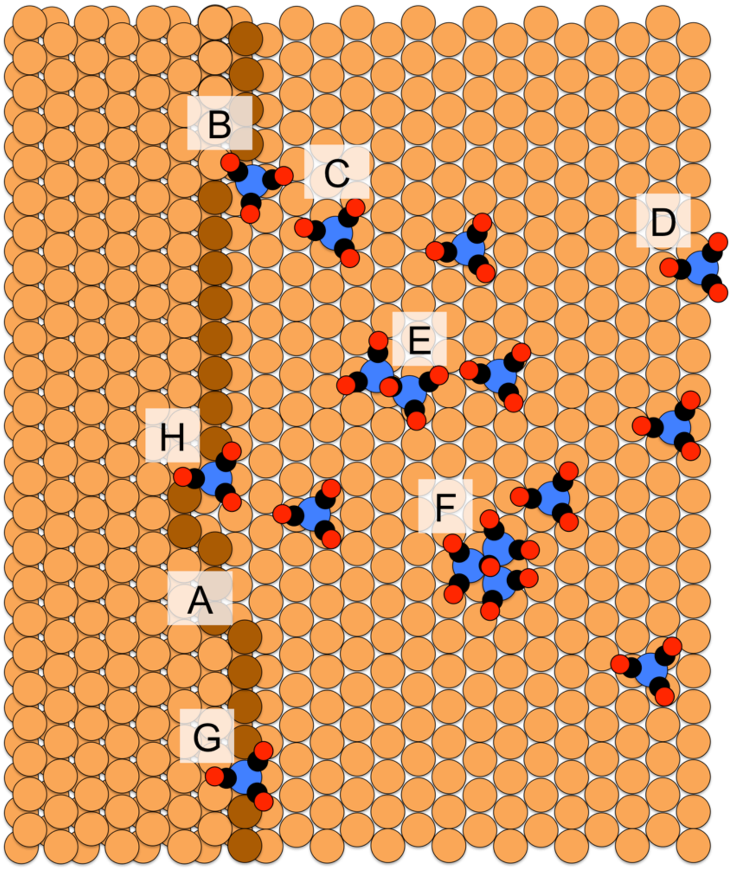


Figure 5.7. A proposed scheme of the step dissolution process based on the results presented. The inset letters, A-H, are to guide the reader through the explanation of this picture in the main text. The color scheme used is identical to that used in Figure 5.1 and 5.2.

Acknowledgements

This work was supported by the National Science Foundation under contract No. CBET-1438227 and by the National Science Foundation Graduate Research Fellowship Program under

Grant No. 1347973. G.C., Seattle Chapter ARCS Fellow and NSF Graduate Research Fellow, gratefully acknowledges financial support from the Achievement Rewards for College Scientists foundation as well as the National Science Foundation Graduate Research Fellowship Program. A portion of the computer time for the computational work was performed using EMSL, a national scientific user facility sponsored by the Department of Energy's office of Biological and Environmental Research and located at PNNL. PNNL is a multi-program national laboratory operated for the US DOE by Battelle. We would also like to give our sincere thanks to A. J. R. Hensley, C. J. Weststrate, and Catherine Stampfl for fruitful discussions and opportunities.

REFERENCES

[1] H. Leidheiser, A. T. Gwathmey, J. Am. Chem. Soc. 70 (1948) 1200-1206.

[2] R. W. McCabe, T. Pignet, L. D. Schmidt, J. Catal. 32 (1974) 114-126.

[3] M. Flytzani-Stephanopoulos, S. Wong, L. D. Schmidt, J. Catal. 49 (1977) 51-82.

[4] M. Flytzani-Stephanopoulos, L. D. Schmidt, Prog. Surf. Sci. 9 (1979) 83-111.

[5] P. Fenter, T. Gustafsson, Phys. Rev. B 38 (1988) 10197-10204.

[6] P. Fery, W. Moritz, D. Wolf, Phys. Rev. B 38 (1988) 7275-7286.

[7] R. Imbihl, S. Ladas, G. Ertl, Surf. Sci. 206 (1988) L903-L912.

[8] T. Gritsch, D. Coulman, R. J. Behm, G. Ertl, Phys. Rev. Lett. 63 (1989) 1086-1089.

[9] T. Gritsch, D. Coulman, R. J. Behm, G. Ertl, Appl. Phys. A 49 (1989) 403-406.

[10] J. D. Batteas, J. C. Dunphy, G. A. Somorjai, M. Salmeron, Phys. Rev. Lett. 77 (1996) 534-537.

[11] D. J. Coulman, J. Wintterlin, R. J. Behm, G. Ertl, Phys. Rev. Lett. 64 (1990) 1761-1764.

[12] T. Gritsch, D. Coulman, R. J. Behm, G. Ertl, Surf. Sci. 257 (1991) 297-306.

[13] M. Kiskinova, Chem. Rev. 96 (1996) 1431-1448.

[14] A. Baraldi, S. Lizzit, D. Cocco, G. Comelli, G. Paolucci, R. Rosei, M. Kiskinova, Surf. Sci. 385 (1997) 376-385.

[15] G. Somorjai, Annual Reviews of Physical Chemistry 45 (1994) 721-751.

[16] B. E. Hayden, K. C. Prince, P. J. Davie, G. Paolucci, A. M. Bradshaw, Solid State Commun. 48 (1983) 325-328.

[17] S. Zou, R. Gómez, M. J. Weaver, Surf. Sci. 399 (1998) 270-283.

[18] G. Rupprechter, T. Dellwig, H. Unterhalt, H. J. Freund, J. Phys. Chem. B 105 (2001) 3797-3802.

[19] A. Gaussmann, N. Kruse, Surf. Sci. 266 (1992) 46-50.

[20] A. Gaussmann, N. Kruse, Surf. Sci. 279 (1992) 319-327.

[21] W. A. Schmidt, J. H. Block, K. A. Becker, Surf. Sci. 122 (1982) 409-421.

[22] A. Gaussmann, N. Kruse, Catal. Lett. 10 (1991) 305-315.

[23] N. Kruse, A. Gaussmann, J. Catal. 144 (1993) 525-543.

[24] D. B. Liang, G. Abend, J. H. Block, N. Kruse, Surf. Sci. 126 (1983) 392-396.

[25] N. Kruse, G. Abend, J. H. Block, Surf. Sci. 211 (1989) 1038-1043.

[26] M. Mihaylov, K. Hadjiivanov, H. Knözinger, Catal. Lett. 76 (2001) 59-63.

[27] J. T. Yates, T. M. Duncan, S. D. Worley, R. W. Vaughan, The Journal of Chemical Physics 70 (1979) 1219-1224.

[28] F. Solymosi, M. Pasztor, J. Phys. Chem. 89 (1985) 4789-4793.

[29] M. Frank, R. Kühnemuth, M. Bäumer, H. J. Freund, Surf. Sci. 427-428 (1999) 288-293.

[30] H. F. J. Van't Blik, J. B. A. D. Van Zon, T. Huizinga, J. C. Vis, D. C. Koningsberger, R. Prins, J. Phys. Chem. 87 (1983) 2264-2267.

[31] N. Kruse, Surf. Sci. 178 (1986) 820-830.

[32] N. Kruse, J. Schweicher, A. Bundhoo, A. Frennet, T. Visart de Bocarme, Top. Catal. 48 (2008) 145-152.

[33] J. Wilson, C. de Groot, J. Phys. Chem. 99 (1995) 7860-7866.

[34] B. Eren, D. Zherebetskyy, L. L. Patera, C. H. Wu, H. Bluhm, C. Africh, L.-W. Wang, G. A. Somorjai, M. Salmeron, Science 351 (2016) 475.

[35] R. Bliem, J. E. S. van der Hoeven, J. Hulva, J. Pavelec, O. Gamba, P. E. de Jongh, M. Schmid, P. Blaha, U. Diebold, G. S. Parkinson, Proceedings of the National Academy of Sciences 113 (2016) 8921.

[36] J. Y. Park, Y. Zhang, M. Grass, T. Zhang, G. A. Somorjai, Nano Lett. 8 (2008) 673-677.

[37] C. Weilach, S. M. Kozlov, H. H. Holzapfel, K. Föttinger, K. M. Neyman, G. Rupprechter, J. Phys. Chem. C 116 (2012) 18768-18778.

[38] B. Böller, M. Ehrensperger, J. Wintterlin, ACS Catal. 5 (2015) 6802-6806.

[39] G. Collinge, Y. Xiang, R. Barbosa, J.-S. McEwen, N. Kruse, Surf. Sci. 648 (2016) 74-83.

[40] S. Carenco, A. Tuxen, M. Chintapalli, E. Pach, C. Escudero, T. D. Ewers, P. Jiang, F. Borondics, G. Thornton, A. P. Alivisatos, H. Bluhm, J. Guo, M. Salmeron, J. Phys. Chem. C 117 (2013) 6259-6266.

[41] S. K. Beaumont, S. Alayoglu, V. V. Pushkarev, Z. Liu, N. Kruse, G. A. Somorjai, Farad. Discuss. 162 (2013) 31-44.

[42] S. Alayoglu, S. K. Beaumont, G. Melaet, A. E. Lindeman, N. Musselwhite, C. J. Brooks, M. A. Marcus, J. Guo, Z. Liu, N. Kruse, G. A. Somorjai, J. Phys. Chem. C 117 (2013) 21803-21809.

[43] A. Banerjee, A. P. van Bavel, H. P. C. E. Kuipers, M. Saeys, ACS Catal. 5 (2015) 4756-4760.

[44] G. Collinge, N. Kruse, J.-S. McEwen, J. Phys. Chem. C 121 (2017) 2181-2191.

[45] J. Wellendorff, T. L. Silbaugh, D. Garcia-Pintos, J. K. Nørskov, T. Bligaard, F. Studt, C. T. Campbell, Surf. Sci. 640 (2015) 36-44.

[46] G. T. K. K. Gunasooriya, A. P. van Bavel, H. P. C. E. Kuipers, M. Saeys, Surf. Sci. 642 (2015) L6-L10.

[47] S. H. Payne, H. J. Kreuzer, Surf. Sci. 399 (1998) 135-159.

[48] E. A. Lewis, D. Le, A. D. Jewell, C. J. Murphy, T. S. Rahman, E. C. H. Sykes, ACS Nano 7 (2013) 4384-4392.

[49] E. A. Lewis, D. Le, A. D. Jewell, C. J. Murphy, T. S. Rahman, E. C. Sykes, Chem. Commun. 50 (2014) 6537-9.

[50] E. A. Lewis, M. D. Marcinkowski, C. J. Murphy, M. L. Liriano, E. C. Sykes, J. Phys. Chem. Lett. 5 (2014) 3380-5.

[51] G. E. Poirier, Chem. Rev. 97 (1997) 1117-1128.

[52] G. E. Poirier, Langmuir 13 (1997) 2019-2026.

[53] P. Maksymovych, D. C. Sorescu, J. T. Yates, Phys. Rev. Lett. 97 (2006) 146103.

CHAPTER SIX:

FORMULATION OF MULTICOMPONENT LATTICE GAS MODEL CLUSTER EXPANSIONS PARAMETERIZED ON AB INITIO DATA: AN INTRODUCTION TO THE AB INITIO MEAN-FIELD AUMGENTED LATTICE GAS MODELING (AMALGM) CODE

Greg Collinge[a]*, Kyle Groden*[a]*, Catherine Stampfl*[b]*, and Jean-Sabin McEwen*[a,c,d,e,f]***

[a] *The Gene & Linda Voiland School of Chemical Engineering and Bioengineering, Washington State University, Pullman WA 99164*

[b] *Department of Physics, University of Sydney, Sydney, NSW 2006, Australia*

[c] *Department of Physics and Astronomy, Washington State University, Pullman, WA 99164*

[d] *Department of Chemistry, Washington State University, Pullman, WA 99164*

[e] *Department of Biological Systems Engineering, Washington State University, Pullman, Washington 99164, USA*

[f] *Institute for Integrated Catalysis, Pacific Northwest National Laboratory, Richland, WA, 99352*

** Corresponding authors:*

Jean-Sabin McEwen; 509-335-8580 (phone), 509-335-4806 (fax), js.mcewen@wsu.edu

Abstract

A formalism is presented for the construction of multi-component lattice gas models parameterized with *ab initio* (typically density functional theory) data. The Leave-Multiple-Out (LMO) and the Leave-One-Out (LOO) Cross Validation (CV) Score are showcased and practical algorithms are developed and implemented in a new code called the *Ab initio* Mean-field Augmented Lattice Gas Modelling (AMALGM) code. The functionality of these algorithms are demonstrated with a fully worked out example using the O/Fe(100) system. AMALGM and the

formalism on which it is based is envisioned as a surface-oriented lattice gas alternative to other cluster expansion codes that are typically geared toward bulk systems whose lateral interactions between components are parameterized using an Ising model. While the formalism is created within the context of surfaces, it is equally applicable to bulk materials.

6.1. Introduction

The parameterization of *ab initio* electronic energies has become a serious and urgent goal for computational chemists and physicists alike over the last few decades due to the inherent limitations of quantum chemical calculations on modern computing infrastructure. Density functional theory (DFT) cannot be expected to handle systems comprised of more than a few 10^3–10^4 atoms while more accurate wavefunction-based *ab initio* theories feasibly handle an even smaller number than that. Even with the rapid increase in computing power, moving to systems of sizes that are statistically relevant (i.e. where ensemble average thermodynamic and kinetic properties of meso- and macroscale systems can be statistically assessed) within the framework of computational chemistry algorithms is likely out of reach for the considerable future. The challenge, therefore, is to retain the accuracy and chemical nuance of these *ab initio* calculations while avoiding their computational burden when system sizes are increased to statistical significance.

The issue with the solution of large-scale system energetics and properties via *ab initio* techniques is ultimately the time needed to numerically assess the Schrödinger (for wavefunction-based methods) or Kohn-Sham[1] (for DFT) equations. This is because the solution of these equations requires the evaluation of a remarkable number of real- or Fourier-space integrals followed by a series of iterative matrix diagonalizations. The most promising strategy for overcoming this issue is to design some explicitly evaluable mathematical function that takes in the same inputs (system geometry) and yields the same outputs (system energetics or other properties) in a fraction of the time needed for the same *ab initio* calculation. Since such a function is unlikely to be based on the same quantum physics as the Schrödinger equation, it can in principle be of any convenient form. However, in order to ensure the function chosen replicates *ab initio* data, it is necessary to parameterize this function via fitting to representative/example *ab initio* data. Clearly, to have utility, the function must be finite and convergent such that any extension in system size does not necessitate further *ab initio* calculations. Thankfully, interactions affecting the chemical behavior of an atom or molecule fall off as a function of distance and we can be assured a finite, convergent function can be found (i.e. new terms won't be needed to account for larger system sizes). With such a function defined, statistical sampling can be performed using stochastic methods like Monte Carlo simulations. Choosing a mathematical form for this function is a matter of providing parameter flexibility and desired chemical interpretability, often with any particular choice being a tradeoff between the two.

In terms of flexibility, a neural network (NN) is an attractive choice for the parameterization of the *ab initio* data. This parameterization method relies on a purely mathematical, black-box fit of energies using geometries and chemical identities as input. Use of NNs has shown to be a powerful method of extending *ab initio* DFT calculations to large-scale homogeneous[2-8] and heterogeneous[9-13] systems, as well as nanoparticles[14-18]. Due to their basically infinite flexibility, NNs essentially eschew any underlying physics and chemistry in favor of accurately reproducing DFT energies, which is an excellent tradeoff for applications such as materials screening.

When knowledge of the underlying physics or chemistry resulting in observed meso- and macroscopic properties *is* desired (i.e. where chemical interpretability is of interest), NNs are clearly not suitable and parameterization onto cluster expansions (CEs) in either the Ising or lattice gas (LG) paradigms is more appropriate. This is because CEs have a well-defined mathematical form based on the principle of cluster interactions, which are both physically motivated and, as formalized by Sanchez,[19-22] shown to form a complete orthonormal basis set capable of perfectly representing a system property. For a given system, its total electronic energy (or other *ab initio* output) is viewed as the sum of the various 1-, 2-, 3-,…, and higher-body interactions that exist between its constituent atoms and/or molecules. The amount of energy contributed by each interaction term is then termed its effective cluster interaction (ECI). Thus, when a system is calculated to have a certain energy and we wish to know why, we can point to the fact that it had X number of particular 2-body (pairwise) ECIs, Y number of a certain 3-body interactions, Z number 4-body interactions, and so on. The physical or chemical characteristics of these interactions can be assessed, and from there it can then be posited how changing the number, strength, or nature (attractive vs. repulsive) of these interactions might affect the system at large. Small (yet often very mathematically sophisticated) CEs and specifically the Ising model have been very thoroughly tested in this manner by the solid state physics and alloy communities, where ECIs have been fundamentally linked via Monte Carlo simulations to order-disorder phase transitions and critical temperatures (the Curie Temperature in ferromagnetic materials, for example).[23, 24] However, these early CE models tended to be either defined for academic purposes (i.e. for determining how changes in ECIs manifest as changes in critical temperatures and other phase phenomena) or derived from fitting to experimental data such as temperature programmed desorption spectra and/or thermodynamic phase diagrams, a practice of clear utility which by no means has been discontinued.[25-31]

Fitting CEs to *ab initio* data started in 1983 with the seminal work of Connolly and Williams[32] but is still a relatively new endeavor in the surface science community. In 1999, Stampfl et al.[33] advocated the use of LG CEs (rather than Ising-type CEs) specifically for surfaces. Stampfl et al. fitted their DFT data on the O/Ru(0001) system to 7 pre-supposed LG ECIs, which ultimately showed modest agreement with experiment when the underlying LG model is used to simulate the corresponding phase diagram.[34] Similar to Connolly and Williams only the number of ordered structure DFT-based energies needed to solve for these ECIs were calculated. With so few terms, parameter estimation is highly uncertain with no guarantee that the energetics of new structures would be accurately predicted. Thus, while these seminal studies represented a critical first step in the application of CEs, their application method is obviously not ideal. The more data that is available, the more accurately the *ab initio* data can be fit (via a least-squares or similar minimization procedure). This, of course, assumes that the quantum chemical theory employed is an accurate depiction of the system, as a CE can only describe a system's energetics as well as the fundamental physics upon which it is built. Importantly, at the time, there did not exist any computational tools for the parsing and deconvolution of a system's configuration into its constituent CE interactions and the counting of these interactions had to be done entirely by hand. Nonetheless, Connolly-Williams, using the Ising paradigm, and Stampfl et al., using the LG paradigm, pioneered the most widely utilized versions of the CE formalism to date.

The dichotomy of CE paradigms has continued since 1999, where alloy and materials science researchers typically follow the Connolly-William approach and use the Ising CE model and surface science and heterogenous catalysis researchers typically follow the Stampfl et al. approach and utilize LG CE models. We are interested in CEs for surfaces here, and since the original work of Stampfl et al. in 1999, efforts to fit CEs to DFT data for surfaces have continued steadily with a large number of groups fitting their *ab initio* data on surface systems to LG (or in some cases, Ising) CEs.[35-63] Very recently, a general review of the CE method has been published[64] which well encapsulates the progress of the discipline in general, showcasing the utility and enthusiasm for the CE formalism.

However, we would be remiss to not note the seeming issues raised by various groups[65-67] and Sanchez himself,[21, 22, 68] who have alluded to/concluded that the usual CE models used are fundamentally flawed due to their lack of concentration-dependent ECIs. Indeed, the mathematical derivation of the CE method explicitly results in ECIs that should indirectly depend on the concentration of constituent species. The use of concentration-independent ECIs has therefore been stated to be a lower-order approximation.[22] Concentration-dependent ECIs stem from primarily the ECI's dependence on the unit cell volume and lattice parameters, which are typically relaxed during optimization in alloy and materials research. The variation in unit cell volume, at minimum, correlates to variation in bulk concentration due to differing sizes of atomic radii, and thus concentration-dependent ECIs should indirectly capture this effect. Luckily, unit cell volume and lattice parameters *are fixed* from their optimized bulk values in surface science and heterogeneous catalysis models, and thus, the use of concentration-independent ECIs are more easily justified. Internal relaxations are typically much greater, admittedly, but we see no reason to assume these relaxations correlate consistently with the concentration or coverage of a surface. Indeed, coverage-dependent LG

ECIs would require a complete and, we think, unnecessary shift in the current LG ECI interpretation.

Constructing CEs requires that all cluster interaction terms be defined and counted for every system making up the *ab initio* data. Constructing CEs by hand is cumbersome and introduces the possibility of human error; and some degree of automation is highly beneficial. Alloy and materials science researchers have been the beneficiaries of remarkably useful computational tools for automating the development of Ising CE models for nearly 20 years, mainly using the Alloy Theoretic Automated Toolkit (ATAT)[69-72] and the UNiversal CLuster Expansion (UNCLE)[73] codes. Prior to this, all CE fitting was (seemingly) done by hand or with in-house codes and the simplicity of the models reflect this. The application of ATAT to surface science and heterogeneous catalysis research has been relatively recent[59, 63, 74-79] and largely ad hoc as ATAT has gradually added surface-specific features (e.g. the latest versions include a tag to restrict structure creation to two dimensions, but it is clear this was added as an afterthought since the surface orientation is often flipped in certain structures; and this two dimensional restriction has yet to be applied to **k**-point generation). UNCLE has been comparatively less utilized for surfaces,[80, 81] but appears to allow for a treatment of surfaces based on the idea of correcting against the bulk values of a system. Nonetheless, we should note that both ATAT and UNCLE are indeed capable of handling surfaces—this in and of itself is not a deficiency of these codes. In fact, while more mathematically complex in origin and in terms of surfaces, UNCLE's strategy may prove to be essential as researchers move toward modeling full nanoparticles.

In terms of determining which clusters to include in their respective CEs, ATAT utilizes a Leave-One-Out Cross Validation (LOO CV) score, which is an objective function criticized by a number of statisticians;[82-84] while UNCLE utilizes a Leave-Multiple-Out Cross Validation (LMO CV) score,[84] which is an asymptotically consistent improvement over LOO CV. It is worthwhile to note that the criticisms leveled on LOO CV are valid only for parametric models. The CE problem could be considered non-parametric since, in principle, there could be an infinite number of terms in the CE. However, the CE formalism developed by Sanchez[19-22] is just as valid for a finite number of sites as it is for an infinite number of sites. Since real surfaces are finite, we can restrict ourselves to this viewpoint, which leads to finite possible clusters in the CE and thus a parametric model. Even for an infinite number of sites, the correlation spheres of surface sites will be finite leading again to a finite number of possible clusters and a parametric problem—the LOO CV is thus less desirable as described by the aforementioned statisticians[82-84]. We therefore choose to use LMO CV in our work here, and importantly, we directly test the validity of the two objective functions side-by-side on the O/Fe(100) system in Section 6.5. Nonetheless, regardless of their choice of cross validation objective functions, both codes still necessitate using the Ising-type variable paradigm and no software has been published specifically for LG models.

The primary difficulty that arises in the Ising paradigm occurs when multibody interactions are present between the adspecies, where the transformation between the LG paradigm and the Ising paradigm is not straightforward. Indeed, pairwise lateral interactions in the Ising paradigm, when translated in the LG paradigm, will depend on the multibody interactions within the LG CE, as was explicitly shown by Binder and Landau for a relatively simple LG CE model.[23] Also, from a practical standpoint, Monte Carlo simulations can be performed equally well within either the LG or Ising paradigms (as is done within the ATAT code) and sticking to the Ising paradigm would be fine if we simply wanted to get the equilibrium distribution of the particles on the surface at a given temperature and applied magnetic field. While there is a relationship between the magnetic field and the chemical potential, it is not intuitive and will depend on the aforementioned lateral interactions between adspecies[23]. This is important since when one simulates, for example, temperature programmed desorption spectra using the method of Widom[85, 86] one needs to calculate the chemical potential at every temperature increment. Doing this in the Ising language is simply not practical.

It is important to note that the difference between the Ising and LG paradigms is formally only mathematical, with the Ising model using site occupation (or "spin flip") variables (σ) that can take on values of +1 and -1 for binary systems, and the LG model using site occupation variables (n) that can take on values of 1 and 0 for binary systems. Mathematically, this is simply a change of variables ($\sigma = 2n - 1$) and so the validity and strength of the CE formalism established by Sanchez, which is formulated within this Ising variable paradigm, remains unchanged. However, the extension of the CE formalism to multicomponent systems[19] leads to wholly confusing site variable definitions and chemically awkward ECIs in the LG paradigm. In Sanchez's formulation each of the *m* component in a multicomponent system is assigned a site variable of $\pm m$, $\pm$(m-1), …, ±1, (and 0 for odd numbers of components). However, if we perform the substitution of variables to get the equivalent LG site variables, a 3-component system has half-integer values. The physical meaning of a half-integer occupation is not easily interpreted in a LG model. To date, we do not know of any groups that have attempted to use such definitions, and as a result, we are not aware of any multicomponent CE modeling that doesn't stick to the Ising-type variable definitions. The work here establishes a solution to this problem in a physically intuitive way.

To allow for the creation of generalized multicomponent LG models parameterized from *ab initio* data and provide surface, materials, and heterogeneous catalysis researchers the tools necessary to create them without resorting to the Ising-type paradigm, we present here the theoretical underpinning of the *Ab-Initio* Mean-Field Augmented Lattice Gas Modeling (AMALGM) code. A formulation of the multicomponent LG site variables is proposed based on the principles established—in the Ising convention—for coupled cluster expansions[87, 88] along with the ideas hinted at by, respectively, the original Stampfl et al. 1999 paper then explicitly by McEwen et al. in 2003[38] for "multi-site" LG models. These are combined with sublattice concepts to construct generalized site variables recast as the more intuitive terms, "site numbers" and "site types," whose flexibility and comprehensibility allow for the swift conceptualization and implementation of the LG model corresponding to the system of interest. We further present the algorithms used to construct these models from first-principles data and note what is required from the user. AMALGM processes *ab initio* data to create an optimized LG model using LMO CV[84] and is designed to utilize an external CV score to ultimately validate the models produced. While not currently implemented, a final software package containing AMALGM is envisioned to provide the tools necessary to automatically generate structures and directly interface with *ab initio* codes, such as the Vienna *Ab Initio* Simulation Package (VASP).[89-92] At present, AMALGM is capable of processing VASP outputs to create all necessary AMALGM inputs, but how this data is generated is currently up to the user. Future versions of AMALGM may also include the capability of processing outputs from other *ab initio* codes. Finally, we are actively working on simulation software to interface with AMALGM as part of this software package, but this is beyond the scope of the current work outlined here.

AMALGM will be released in C++ to take advantage of parallelization schemes applicable to AMALGM's algorithms and to facilitate code maintenance.

6.2. Theoretical Reformulation of the Multicomponent LG Model

As stated above, due to its intuitive nature in the context of surface science, we utilize here the CE formalism within the LG paradigm. Since we are interested specifically in parameterizations of surfaces and 2-D interfaces, we restrict ourselves to that perspective, though the formulation herein is equally generalized to 3-D systems if truly desired. We should note that the basic mathematical tools needed here have already been established for multicomponent CEs, as for example in the UNCLE code[73] where the underlying lattice is extended via a transformation from $\mathbb{N}^3$ to $\mathbb{N}^4$ to express Ising spin states. Our goal here therefore is to reformulate the underlying mathematics in a way that is more intuitive from within the LG paradigm and the perspective of surfaces and interfaces. In its simplest form, the LG CE gives the electronic energy, $E(\boldsymbol{n})$**,** of a surface as

$$E(\boldsymbol{n}) = \sum_{i}^{1-body} V_i n_i + \sum_{i>j}^{2-body} V_{ij} n_i n_j + \sum_{i>j>k}^{3-body} V_{ijk} n_i n_j n_k + \dots \tag{6.1}$$

where n_i, n_j, n_k, are "occupation variables" that take on a value of "1" when its associated site is occupied and a value of "0" when its associated site is unoccupied. In this way, $\boldsymbol{n} = \{n_1, n_2, n_3, \dots, n_{N_S}\}$, a microstate of occupation variables, uniquely describing any configuration of species on or in a surface with N_S adsorption sites. V_i is the electronic energy associated with the occupation of each isolated site (e.g. the adsorption energy when this site is that of an adsorbed species); and the V_{ij} and V_{ijk} are "effective cluster interactions" (ECIs) corresponding to 2-body and 3-body clusters of species, respectively. We should note that, here, the indices of the occupation variables do not contain information about what *type* of site is occupied and V_i can in principle be unique to every site. The effect of multiplying occupation variables in Equation 6.1, as in "$n_i n_j$", is to contribute one corresponding ECI to the total energy only when all of the constituent sites are occupied. Thus, each "$\prod n_i$" term is an effective occupation variable for its associated cluster.

The symmetries (i.e. rotation, mirror) of crystalline surfaces means there will be a number of sites and clusters that are of the same type (thus having the same V_i and ECI) and this means that combined with these effective cluster occupation variables, the summations in Equation 6.1 essentially "count" the number of symmetrically distinct occupied sites and clusters present in a given microstate, $\boldsymbol{n}$. This motivates defining unique "site types" and a recasting of Equation 6.1.

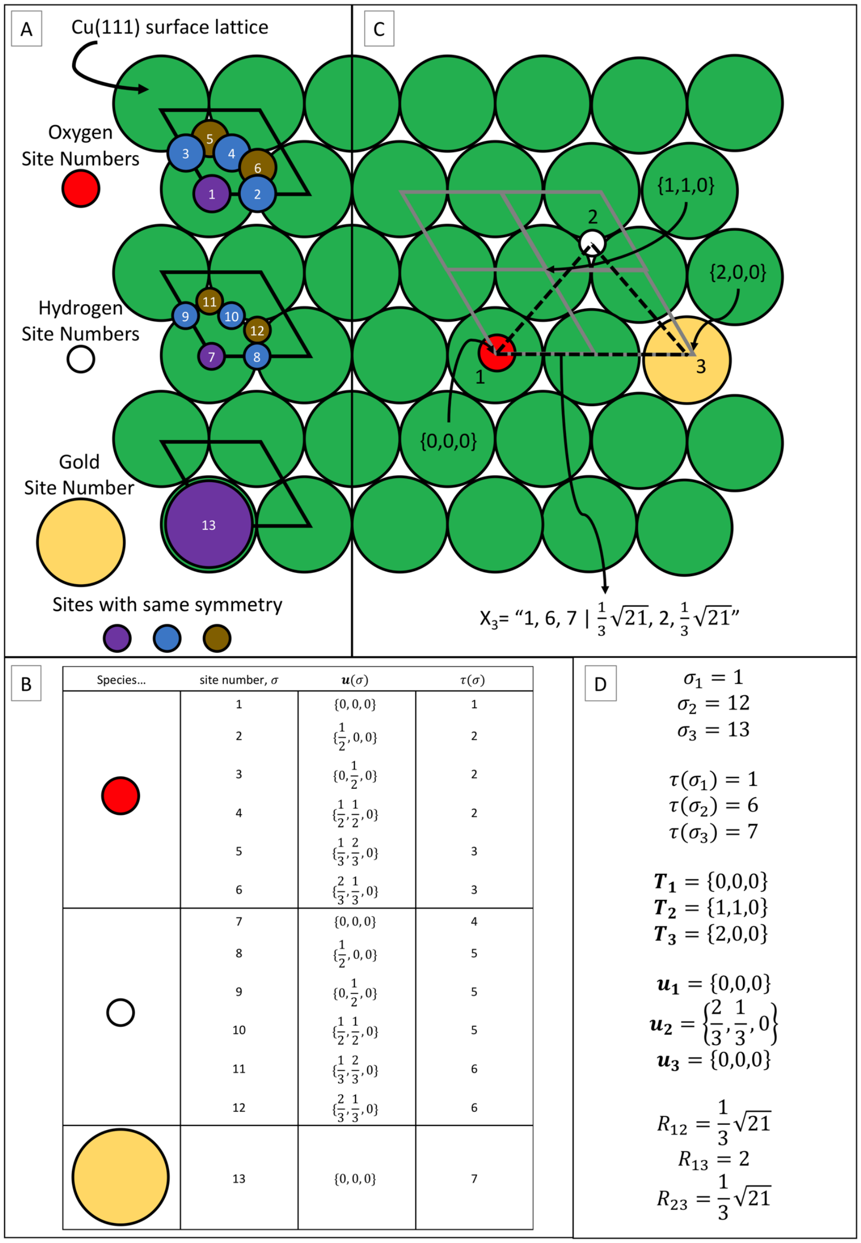
We introduce the concept of "site types," which we will label with Greek lowercase letters $(\alpha, \beta, \gamma, \dots)$, that can uniquely describe any situation of site occupation: defining equally a single species at symmetrically distinct sites and different species at a symmetrically equivalent site, for instance. A purposely complex example is shown in Figure 6.1 for oxygen and hydrogen adsorption on a gold-doped Cu(111) surface.

As can be seen in Figure 6.1A, "site numbers" are merely a sequential labeling of the chemically and/or spatially unique sites within the host lattice unit cell. Any location within the unit cell can be potentially deemed a site. In this example, it is only our *a priori* knowledge of the typical stable adsorption sites of an FCC(111) surface that allows us to limit it to the 6 sites shown in Figure 6.1A. In any system, this information is indeed needed *a priori* as these define the problem in the first place. After assigning site numbers, site types can be assigned based on symmetry and chemical identity, which is again illustrated in Figure 6.1A and worked out completely in the table shown in Figure 6.1B. With any k-body cluster, like the 3-body cluster shown in Figure 6.1C, it is relatively then straightforward to identify which site types are involved. This is step number one in terms of defining these clusters.

Since each k-body cluster is uniquely defined by the $\binom{k}{2}$ combination of pairwise distances between occupied sites, we can label each unique ECI by its particular combination of site types, identified as above, and its pairwise distances. This will be denoted by "X_k". These are defined as

$$
\begin{gathered}
X_1 \equiv "\alpha" \\
X_2 \equiv "\alpha, \beta \mid R_{\alpha\beta}" \\
X_3 \equiv \alpha,\beta,\gamma \mid R_{\alpha\beta},R_{\alpha\gamma},R_{\beta\gamma} \\
X_4 \equiv \alpha,\beta,\gamma,\delta \mid R_{\alpha\beta},R_{\alpha\gamma},R_{\beta\gamma},R_{\alpha\delta},R_{\beta\delta},R_{\gamma\delta} \\
etc \ldots
\end{gathered}
\tag{6.2}
$$

and so on for higher k-body clusters. Each pairwise distance, $R_{\alpha\beta}$, specifies which site type pair is involved via its subscript, $\alpha\beta$, and by keeping the ordering of site types and pairwise distances constant, the label is unambiguous. We note here that while only $2k-3$ pairwise distances are technically needed to define a cluster of $k>1$, it is computationally expedient to use all $\binom{k}{2}$ distances as above because the set of $2k-3$ distances is not unique for $k>3$ while the set of $\binom{k}{2}$ distances is.



Species...	site number, σ	$\boldsymbol{u}(\sigma)$	$\tau(\sigma)$
🔴	1	$\{0,0,0\}$	1
	2	$\{\frac{1}{2},0,0\}$	2
	3	$\{0,\frac{1}{2},0\}$	2
	4	$\{\frac{1}{2},\frac{1}{2},0\}$	2
	5	$\{\frac{1}{3},\frac{2}{3},0\}$	3
	6	$\{\frac{2}{3},\frac{1}{3},0\}$	3
○	7	$\{0,0,0\}$	4
	8	$\{\frac{1}{2},0,0\}$	5
	9	$\{0,\frac{1}{2},0\}$	5
	10	$\{\frac{1}{2},\frac{1}{2},0\}$	5
	11	$\{\frac{1}{3},\frac{2}{3},0\}$	6
	12	$\{\frac{2}{3},\frac{1}{3},0\}$	6
🟡	13	$\{0,0,0\}$	7

$$\sigma_1 = 1$$
$$\sigma_2 = 12$$
$$\sigma_3 = 13$$

$$\tau(\sigma_1) = 1$$
$$\tau(\sigma_2) = 6$$
$$\tau(\sigma_3) = 7$$

$$\boldsymbol{T_1} = \{0,0,0\}$$
$$\boldsymbol{T_2} = \{1,1,0\}$$
$$\boldsymbol{T_3} = \{2,0,0\}$$

$$\boldsymbol{u_1} = \{0,0,0\}$$
$$\boldsymbol{u_2} = \left\{\frac{2}{3},\frac{1}{3},0\right\}$$
$$\boldsymbol{u_3} = \{0,0,0\}$$

$$R_{12} = \frac{1}{3}\sqrt{21}$$
$$R_{13} = 2$$
$$R_{23} = \frac{1}{3}\sqrt{21}$$

Figure 6.1. How lattice gas k-body cluster labels, X_k, are defined for the example system O/H/Au/Cu(111). In this example, the Cu(111) surface is the "empty" lattice, and occupied oxygen or hydrogen sites are *additions* to the Cu lattice while Au occupied sites represent *replacement* of the Cu atoms. (A) Each site within the Cu(111) unit cell (delineated with a black outline) is assigned a unique value (its "site number"); the symmetrically equivalent sites are represented with a different color: purple ("top site"), blue ("bridge site"), and brown ("hollow site") which determine each site number's "site type." Note that the hollow sites are treated as symmetrically equivalent and thus energetically equivalent here. (B) Completed table of site numbers, their shift vectors, $\boldsymbol{u}(\sigma)$, and site types, $\tau(\sigma)$, for this system. (C) Example O-H-Au 3-body cluster where, for purposes of subscripting, body #1 is O, body #2 is H and body #3 is Au. The underlying lattice is shown in grey to illustrate how translation vectors, $\boldsymbol{T}$, are identified. The final cluster label, X_3, for this cluster is shown below the lattice. (D) All information collected from the table and analysis of the cluster in order to determine the cluster label, X_3.

Since there are discrete sets of X_k labels, which uniquely defines each possible k-body cluster, we can rewrite Equation 6.1 as

$$E(\boldsymbol{m}) = \sum_{X_1}^{\infty} V_{X_1} m_{X_1} + \sum_{X_2}^{\infty} V_{X_2} m_{X_2} + \sum_{X_3}^{\infty} V_{X_3} m_{X_3} + \dots \tag{6.3}$$

where $V_{X_1}, V_{X_2}, V_{X_3}$ are each a unique k-body ECI labeled by its particular X_k. $m_{X_1}, m_{X_2}, m_{X_3}$ are the corresponding number of each unique k-body cluster present in a given configuration now defined by the vector of these values, $\boldsymbol{m}$**.** There is an implied relationship between occupation variables, n_i, and this vector of "counts," $\boldsymbol{m}$. This can be made explicit and in so doing show that there is a general method for defining occupation sites in a way that incorporates all relevant details needed to define the labels in Equation 6.2 and subsequently the LG CE in Equation 6.3.

First, we recognize (as has been recognized before) that each site, regardless of its uniqueness, must conform to the translational symmetry of the underlying lattice. That is, site number 1 in Figure 6.1A will be repeated in each surface lattice unit cell. Thus, all sites at any point on the entire surface can be defined by the location of its surface lattice unit cell relative to some arbitrary original (as in "being at the origin") unit cell, plus some vector defining where exactly that site is within the original unit cell. Due to rotational and mirror symmetry, some of these sites might be energetically equivalent, but each site must retain a unique identity in order to conform to the lattice's translational symmetry. Therefore, the term "site number" will generally be used to delineate the location of each possible site within and relative to the original surface unit cell, and a translation vector, $\boldsymbol{T}$, will incorporate the information needed to find which integer-valued translations of the unit cell are needed to locate a specific site on the surface. Each site number can then be assigned its "site type" for the purposes of defining k-body clusters as in Equation 6.2. It is therefore required that these site numbers and site types be defined before any k-body cluster definitions can be made. We reiterate that site types are used in the labels of Equation 6.2 instead of site numbers because they are specifically defined to be chemically unique while site numbers can have chemical degeneracy, which will be discussed more shortly. Thus, we will now rewrite the occupation vector as a four-dimensional "occupation matrix," $\mathbf{\Phi}$, as

$$\mathbf{\Phi} = \mathbf{\Phi}(T_1, T_2, T_3, \sigma) = \mathbf{\Phi}(\boldsymbol{T}\,, \sigma) \tag{6.4}$$

where $\boldsymbol{T} = \{T_1, T_2, T_3\}$ is the aforementioned translation vector of integer values that identifies in which repeated unit cell of the surface lattice one will find a site, which is then finally specified completely by its integer coordinate σ (i.e. site number). As before, the occupation matrix entry specified by $\mathbf{\Phi}$ takes on a value of "0" or "1" depending on whether that specific surface site is occupied or unoccupied. Thus, in this formulation, the occupation matrix remains an array of Boolean type entries each requiring merely 1 bit of allocated memory.

Each site type, τ, is a function of which site number is chosen:

$$\tau = \tau(\sigma) \tag{6.5}$$

where the degeneracy of site types is determined ahead of time after analysis of the underlying lattice symmetry. That is, after assigning all sites a site number, if for example sites numbers "2", "3", and "4" are all site type "2," (e.g. oxygen adatoms at bridge sites in Figure 6.1A) that information is encoded in Equation 6.5 as $\tau(2) = \tau(3) = \tau(4) = 2$. This can be seen completely worked out for our example in Figure 6.1B.

In addition to site types, we also need Cartesian distances to uniquely specify a label in Equation 6.2. To remain consistent with the formalism so far employed, we want to find these distances by using the same information utilized by the occupation matrix, namely the 4-D vector, $\{\boldsymbol{T}, \sigma\}$. To do this, we first define a "shift vector," which must be specified for each site number, σ, (again, *a priori*) that defines the real-space position of the site relative to the unit cell, i.e. in "fractional" or "direct" coordinates, as

$$\boldsymbol{u}(\sigma) = \{u_1, u_2, u_3\} \tag{6.6}$$

This shift vector can then be added to the translation vector, $\boldsymbol{T}$, to find the fractional coordinate of a site anywhere on the underlying lattice:

$$\boldsymbol{r}(T_1, T_2, T_3, \sigma) = \boldsymbol{r}(\boldsymbol{T}, \sigma) = \boldsymbol{T} + \boldsymbol{u}(\sigma) = \{r_1, r_2, r_3\} \tag{6.7}$$

Since the unit cell of a surface is specified by three lattice (column) vectors, $\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}$, forming a matrix, $A = [\boldsymbol{a_1} \;\; \boldsymbol{a_2} \;\; \boldsymbol{a_3}]$, we can find the pairwise Cartesian distances, $\mathrm{R_i}$, between occupied sites by finding all vector differences between their individual $\boldsymbol{r}$ vectors (defined by Equation 6.7), and calculating those differences' vector norms. The Cartesian coordinate of a site is given simply as the matrix multiplication of A and $\boldsymbol{r}$:

$$\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{T}, \sigma, A) = A\boldsymbol{r}(\boldsymbol{T}, \sigma) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{6.8}$$

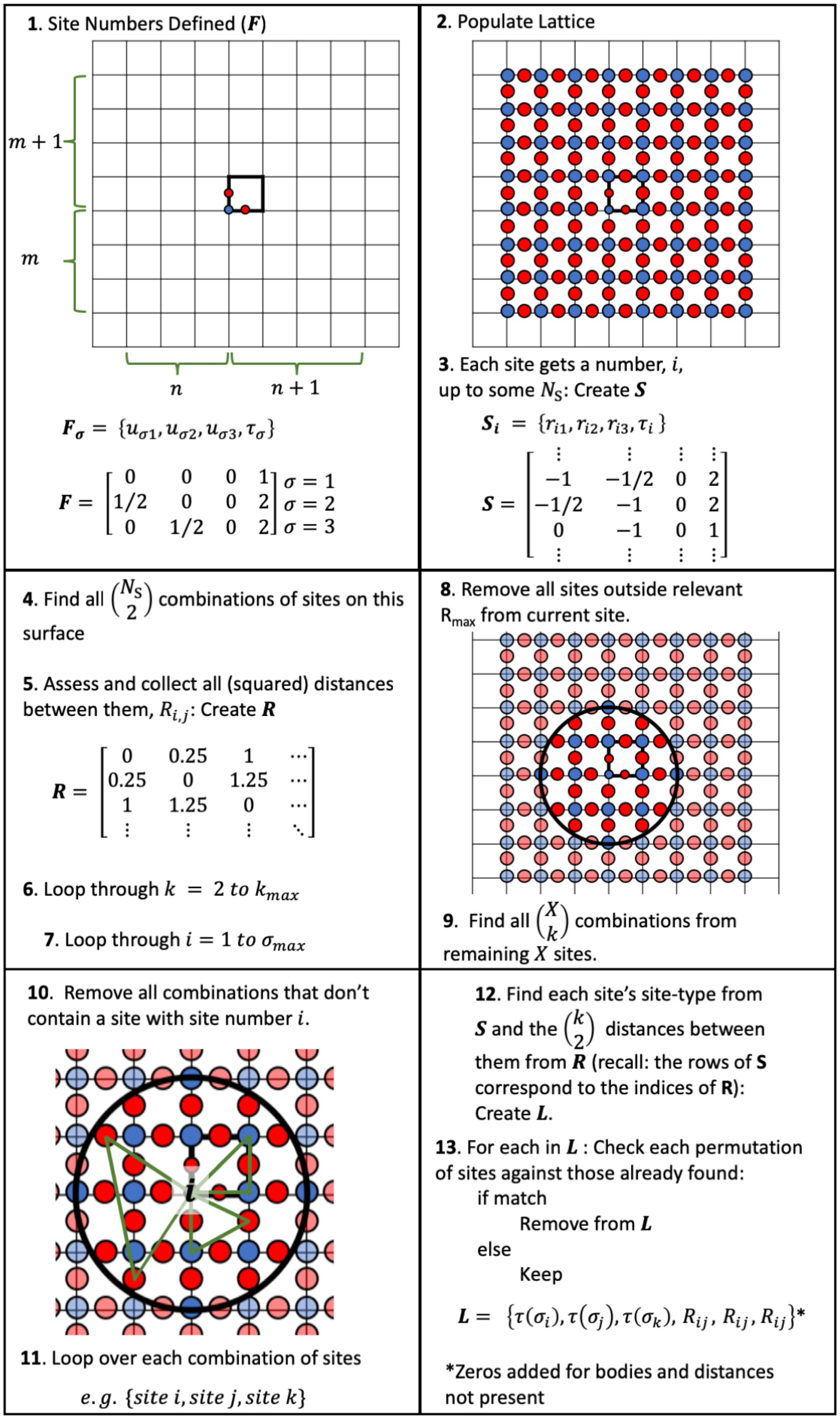
and the pairwise distance between two sites at $\boldsymbol{r_i}(\boldsymbol{T_i}, \sigma_i)$ and $\boldsymbol{r_j}(\boldsymbol{T_j}, \sigma_j)$, is then

$$R_{\tau(\sigma_i)\tau(\sigma_j)}\left(\boldsymbol{r_i}(\boldsymbol{T_i}, \sigma_i), \boldsymbol{r_j}(\boldsymbol{T_j}, \sigma_j)\right) = \|\boldsymbol{c_i} - \boldsymbol{c_j}\| = \sqrt{[\boldsymbol{r_i} - \boldsymbol{r_j}]^T (A^T A)[\boldsymbol{r_i} - \boldsymbol{r_j}]} \tag{6.9}$$

where superscript "T" indicates the matrix or vector transpose. "$R_{\tau(\sigma_i)\tau(\sigma_j)}$" in Equation 6.9 is simply the functional representation of "$R_{\alpha\beta}$" in Equation 6.2.

Since these distances (Equation 6.9) and occupation variable site types (Equation 6.5) are represented uniquely in the label, X_k (Equation 6.2), the functional relationship between occupation variables, n_i, and the number of unique k-body clusters, m_{X_k}, is established. The pertinent data that needs to be collected/determined within this formalism for our O/H/Au/Cu(111) example is shown in Figure 6.1D.

The practical implementation of this formalism within AMALGM is as follows:



1. Site Numbers Defined ($\boldsymbol{F}$)

$$\boldsymbol{F_\sigma} = \{u_{\sigma 1}, u_{\sigma 2}, u_{\sigma 3}, \tau_\sigma\}$$

$$\boldsymbol{F} = \begin{array}{c}\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 2 \\ 0 & 1/2 & 0 & 2 \end{bmatrix}\end{array}\begin{array}{l}\sigma = 1 \\ \sigma = 2 \\ \sigma = 3\end{array}$$

2. Populate Lattice

3. Each site gets a number, i, up to some N_{S}: Create $\boldsymbol{S}$

$$\boldsymbol{S_i} = \{r_{i1}, r_{i2}, r_{i3}, \tau_i\}$$

$$\boldsymbol{S} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -1 & -1/2 & 0 & 2 \\ -1/2 & -1 & 0 & 2 \\ 0 & -1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

4. Find all $\binom{N_{\mathrm{S}}}{2}$ combinations of sites on this surface

5. Assess and collect all (squared) distances between them, $R_{i,j}$: Create $\boldsymbol{R}$

$$\boldsymbol{R} = \begin{bmatrix} 0 & 0.25 & 1 & \cdots \\ 0.25 & 0 & 1.25 & \cdots \\ 1 & 1.25 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

6. Loop through $k = 2\ to\ k_{max}$

7. Loop through $i = 1\ to\ \sigma_{max}$

8. Remove all sites outside relevant R$_{\mathrm{max}}$ from current site.

9. Find all $\binom{X}{k}$ combinations from remaining X sites.

10. Remove all combinations that don't contain a site with site number i.

11. Loop over each combination of sites

$e.g.\ \{site\ i, site\ j, site\ k\}$

12. Find each site's site-type from $\boldsymbol{S}$ and the $\binom{k}{2}$ distances between them from $\boldsymbol{R}$ (recall: the rows of **S** correspond to the indices of **R**): Create $\boldsymbol{L}$.

13. For each in $\boldsymbol{L}$: Check each permutation of sites against those already found:

```
if match
    Remove from L
else
    Keep
```

$$\boldsymbol{L} = \{\tau(\sigma_i), \tau(\sigma_j), \tau(\sigma_k), R_{ij}, R_{ij}, R_{ij}\}^*$$

*Zeros added for bodies and distances not present

Figure 6.2. Illustrated algorithm used to find all unique clusters in a system with inputted unit cell, A, and site definitions (matrix $\boldsymbol{F}$, above)

1. Determine all possible *k*-body clusters by finding all unique labels, X_k. This is illustrated in Figure 6.5.2.
    a. This is done by first specifying the surface unit cell, A, then the site definitions (matrix $\boldsymbol{F}$ in Figure 6.2) containing each site number's shift vector, $\boldsymbol{u}(\sigma)$, and site type, $\tau(\sigma)$, of each site number in the unit cell of the surface of interest (step 1 in Figure 6.2)
    b. The user must also specify a *k*-body cutoff and a radial cutoff, $\mathrm{R_{cut}}$, for each *k*-body cluster—or in principle for each subset of site types. There are an infinite number of distances and clusters, so this is required to make the process finite. It is important to input $\mathrm{R_{cut}}$ in the same units as used to define surface unit cell, A. As will be seen in Section 6.4, we advocate factoring out the lattice constant (or site-to-site distance) from A. Thus, our $\mathrm{R_{cut}}$ is defined in units of lattice constants.
    c. Populate the surface with all possible translations of the sites defined by the shift vectors of the specified site numbers (Step 2 in Figure 6.2).
        i. Loop over each site number, σ, specified. These are all assumed to be associated with translation vector {0,0,0}.
        ii. Loop over both the T_1 (aka "*x*") and T_2 (aka "*y*") components of all potentially relevant translation vectors, $\boldsymbol{T}$, starting from $\{-n, -m, 0\}$ and ending at $\{n+1, m+1, 0\}$where$n = ceiling\left(\frac{\max(\mathrm{Rcut})}{\|\boldsymbol{a_1}\| sin\theta}\right)$, $m = ceiling\left(\frac{\max(\mathrm{Rcut})}{\|\boldsymbol{a_2}\| sin\theta}\right)$, and θ is the angle made by $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ (0° < θ < 180°). Since we assume that a surface is of interest, we do not need to loop over the T_3 component: the assumed direction perpendicular to the surface. In

other words, it is the z-direction and its translations are irrelevant to the surface as that is the direction a vacuum is imposed. Populate a matrix of vectors $\boldsymbol{S} = \{\boldsymbol{r}(\boldsymbol{T}, \sigma), \tau(\sigma)\}$ in each loop (step 3 in Figure 6.2). Note which sites in $\boldsymbol{S}$ are the site numbers inside the unit cell.

d. Find all possible pairs of now populated surface sites (step 4 in Figure 6.2).

e. Use Equation 6.9 to find their pairwise distances (in practice it is better to compute the square of the distance to avoid numerical round-off errors). Store this information in a matrix whose indices map to the indices in the matrix of vectors $\boldsymbol{S}$. This creates a (large) symmetric matrix,$\boldsymbol{R}$, of all pairwise distances for all potentially relevant sites on the surface (step 5 in Figure 6.2).

f. Find all possible *k*-body clusters and determine their Equation 6.2 labels.

    i. Loop over the *k* possible *k*-body clusters up to the specified cutoff (step 6 in Figure 6.2).

    ii. Loop over each site number (we are only interested in clusters that connect to the sites in the unit cell) (step 7 in Figure 6.2).

        1. Determine which of the other sites in $\boldsymbol{S}$ are outside the relevant radius cutoff. Eliminate these from consideration (step 8 in Figure 6.2).

        2. For all *x* sites left in $\boldsymbol{S}$, find all *k*-body combinations, $\binom{x}{k}$ of those sites (step 9 in Figure 6.2).

        3. Remove all clusters of sites that don’t contain the original site numbers (step 10 in Figure 6.2).

            4. Loop over each remaining combination of sites (step 11 in Figure 6.2).

                a. Determine the site types and pairwise distances between all constituent bodies in each of those clusters via a lookup of $\boldsymbol{S}$ and $\boldsymbol{R}$**,** respectively. Create a matrix, $\boldsymbol{L}$, of vectors corresponding to the labels defined by Equation 6.2: each cluster's constituent site types and, in a consistent ordering, their pairwise distances (step 12 in Figure 6.2).

        iii. Continue adding until all loops above have been exhausted.

    g. Find all unique cluster labels in $\boldsymbol{L}$ being sure to account for all possible permutations of each label (step 13 in Figure 6.2).

2. Given a surface configuration, $\boldsymbol{\Phi}$, find its representation in terms of number of "counts," $\boldsymbol{m}$.

    a. Since most configurations are found from a DFT calculation with periodic boundary conditions, input each surface configuration as a combination of their supercell vectors in the unit cell "direct" coordinates, and the position of each occupied site in unit cell "direct" coordinates (i.e. using Equation 6.7). Append which site number each occupied site within the supercell corresponds to.

    b. Rerun the process described in step 1(d) and 1(f) above. However, instead of finding all unique values, match each *k*-body cluster label to the labels found in step 1, again accounting for potential permutations. Each match is a "count" for that cluster. Continuously adding to the counts already found produces the vector, $\boldsymbol{m}$**.**

There are in principle an infinite number of terms both with respect to the summation of each k-body cluster (continuing to infinity by systematically increasing pairwise distances between its k bodies) and with respect to the number of bodies in clusters (continuing to infinity by adding higher-body cluster summation terms) in Equations 6.1 and 6.3. In practice, however, these CEs must be truncated in some manner. The first step in this is to specify, as described above, a maximum k-body cluster size (e.g. truncate at 4-body cluster terms, X_4) and maximum pairwise distance or radius cutoff. This cutoff can be specified overall, for each k-body cluster, or for each combination of site types. However, given reasonable truncations, this can still leave intact easily over 100 X_k terms in Equation 6.3 and potentially well over 1000 terms even if the system has only a couple of site numbers defined. Since the CE needs to ultimately *predict* new DFT electronic energies accurately, the problem of overfitting the data with so many terms is a non-trivial one; the set of terms that provides the most predictiveness is desired. This is an optimization problem and requires an objective function and optimization algorithm to solve it.

6.3. Finding the Most Predictive Lattice Gas Cluster Expansion

6.3.1 The Leave Multiple Out Cross Validation Score as an Objective Function

The typical objective function for the purposes of variable selection is the Leave-One-Out (LOO) Cross Validation (CV) score. Its strength appears to lie in its simplicity. However, as was determined by Baumann in 2009, the LOO-CV score is not the best objective function for variable selection.[84] Baumann supported the use of the Leave-Multiple-Out (LMO) CV score with a fraction “left out” to range from 0.4 to 0.6. The LOO-CV score was explored and criticized previously[82, 83] as well but, as noted by Baumann, this work appears to have been largely ignored. In Section 6.4 we compare the LOO-CV score to the LMO-CV score for a prototypical LG model. Figure C1 shows the associated algorithm for the LOO-CV score for reference.

The LMO-CV score calculation algorithm is shown in Figure 6.5.3 along with the algorithm for evaluating the root mean squared residual (RMSR) of the fit for comparison. The algorithm used to produce the LMO-CV score will be referenced to as process #1 (P1) in subsequent figures. Due to the fact that the LMO-CV score uses all known data at some point in its evaluation, it is an “internal CV score,” contrasting with an “external CV score,” which is defined by prediction errors of data that lie outside of the dataset that is used for fitting. We will come back to this and their juxtaposition in a moment.

As shown in Figure 6.3, we assume we start with a set of structures whose electronic energies have been calculated and whose occupied sites and clusters have been counted via the formalism developed in the previous section. We further require the input of some subset of the known cluster terms, i.e. of the known X_k labels. Assuming there are N structures and n cluster terms with their associated counts, we begin with an $(N \times n)$ matrix and an $(N \times 1)$ vector of N energies. In practice, we can manipulate the matrix of counts and vector of energies in some way to produce adsorption, formation, or surface energies, but we assume some generalized energy here as this does not affect the algorithm as shown. In Figure 6.3, the matrix of counts is represented with black line-filled rectangles and the vector of energies with colored line-filled rectangles; this comprises the “data set”. To calculate the RMSR, a least-squares fit of the entire data set is used to produce a model (shown as a red box in Figure 6.3) of ECI values, Figure 6.3(A). These are then used to predict, Figure 6.3(B), the energies of the entire data set—the same data used to do the fit—and the root mean squared deviation from the actual DFT energies is the RMSR, Figure 6.3(C). It is the RMSR that is minimized in the least-squares fit.

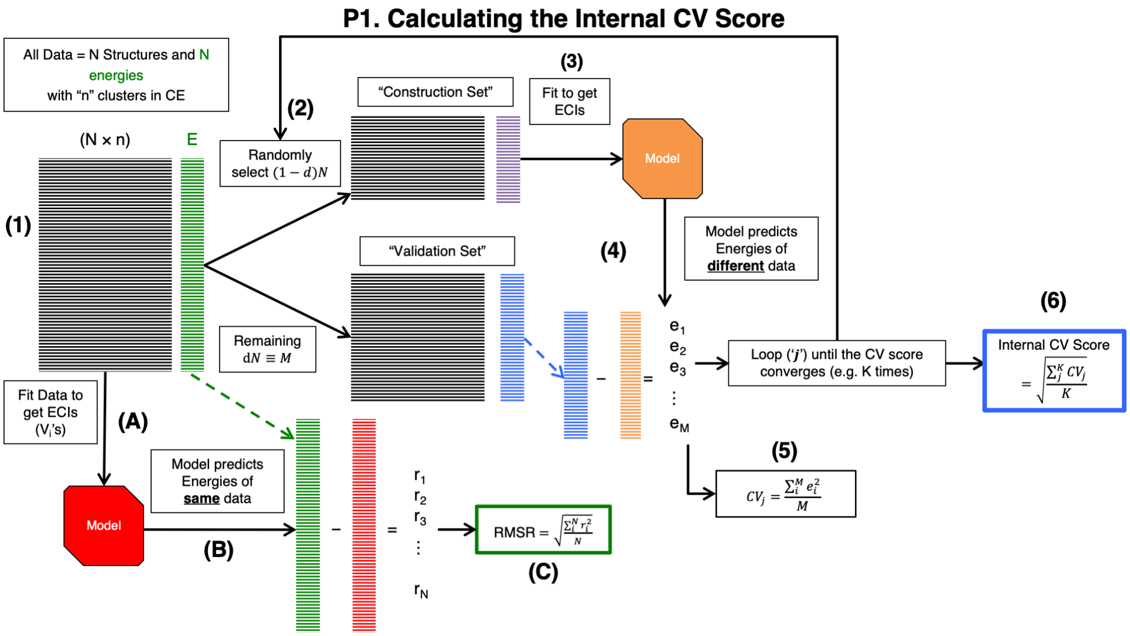

P1. Calculating the Internal CV Score
All Data = N Structures and N energies with "n" clusters in CE
(N × n)
E
(1)
(2)
Randomly select $(1-d)N$
"Construction Set"
(3)
Fit to get ECIs
Model
"Validation Set"
(4)
Model predicts Energies of different data
Remaining $dN \equiv M$
e_1
e_2
e_3
e_M
Loop ('j') until the CV score converges (e.g. K times)
(6)
Internal CV Score
$= \sqrt{\frac{\sum_j^K CV_j}{K}}$
(5)
$CV_j = \frac{\sum_i^M e_i^2}{M}$
Fit Data to get ECIs (V_i's)
(A)
Model predicts Energies of same data
Model
(B)
r_1
r_2
r_3
r_N
RMSR = $\sqrt{\frac{\sum_i^N r_i^2}{N}}$
(C)

Figure 6.3. Flow diagram for the algorithm used to calculate the LMO-CV (represented by labels 1 to 6) and RMSR (represented by labels A to C).

To calculate the LMO-CV score, the data set shown at Figure 6.3(1) is split, Figure 6.3(2), into a random subset of some fraction, $(1-d)$, of the total N data to create the "construction set". Following the recommendation from Baumann, we set d to 60% of the dataset in our work but this can be arbitrarily set by the user. The remaining data then constitutes the "validation set." A set of ECIs is then found, Figure 6.3(3), by performing a least-squares fit of the construction set data to create a model. These ECIs are used to predict, Figure 6.3(4), the energies of the validation set (dot product of vector of ECIs with vector of counts for that structure). The mean squared deviation from the actual DFT energies, Figure 6.3(5), is stored as CV_j for that particular random cut of the data. This process is then repeated from step 2 with another random cut of the data. The LMO-CV score is the root mean CV_j value and is evaluated at the end of every loop. This process ends when the LMO-CV score converges to some tolerance. Another stopping criteria, not shown in Figure 6.3 explicitly, is that each structure must, in the end, be represented in the validation set an equal number of times as every other structure to avoid biasing the value toward any particular structure(s). A count bias manifests as an unstable (non-deterministic) final value upon repeated evaluations of the LMO-CV score.

6.3.2 Optimizing a Cluster Expansion: Steepest Descent with the LMO-CV Score

The value of the LMO-CV score is a function of the subset of clusters used to define the CE. However, the choice of which clusters one should use is an entirely separate issue. Our goal is to find the subset of clusters that gives the *lowest* possible LMO-CV score. To do this, we implement a steepest descent type algorithm shown in the flow diagram in Figure 6.4 which will be referred to as process #2 (P2).

1. Some starting CE, comprised of a subset of the clusters remaining after the truncation mentioned is Section 6.1, is inputted. We will let the number of total clusters available be ‘X’.
2. We then use P1 to find the LMO-CV score of that CE to establish the “old CV score” as a point of comparison for subsequently generated CV scores.
3. An X-length array, ‘A’, is created before heading into the ‘Calculating the Gradient’ loop labeled in *orange* in Figure 6.4.
4. To calculate the gradient, we loop over all X clusters (the loop index is ‘j’ in Figure 6.4), adding these *to the starting CE* if it isn’t part of the starting CE, and removing *from the starting CE* if it is. In this way the starting CE is never changed by more than 1 cluster at a time. This is the finite-space equivalent of the usual gradient.
5. For each perturbed CE, its CV score is calculated using P1. This result is added to the array, A, at its position, j.

6. After looping over all X, the array, A, is sorted in ascending order—such that the first entry is the lowest calculated CV-score.
7. We then enter the steepest descent portion of the algorithm, labeled in *blue* in Figure 6.3. Here, we loop over the clusters corresponding to the newly sorted A array (the loop index is ‘k’ in Figure 6.4).
8. Starting with the first entry in A, if the cluster in question is part of the CE, it is added, and if it is not, it is removed. This becomes the ‘new CE’.
9. P1 is used to calculate the CV score once more. However, this time, if the CV score lowers compared to the “old CV score” (first established in step 2), then cluster addition or removal is made permanent and the CE and target CV score are updated, becoming the “old CE” and “old CV score,” respectively. A “lowering flag” is also activated if the CV score lowers. If the CV score does not lower, then the CE and target CV score are not updated, and a “count” is incremented.
10. Each time a new CV score does not lower, the “count” is evaluated against some value Z, here we allow the CV score to not lower 3 times (Z = 3) before determining that we’ve reached the minimum along the gradient. This is the default behavior but is user-optional and a strict steepest descent algorithm would exit the loop as soon as the CV score stopped lowering.
11. When the “count” threshold is reached, the “lowering flag” is checked. If it was activated, the newly updated CE is sent back to the top of the algorithm for further optimization starting at step 2 and the lowering flag is then deactivated for use in the next cycle. If the lowering flag stayed inactive, then no new additions or removals to the CE score resulted in the lowering of the CV score and the CE is declared optimized.

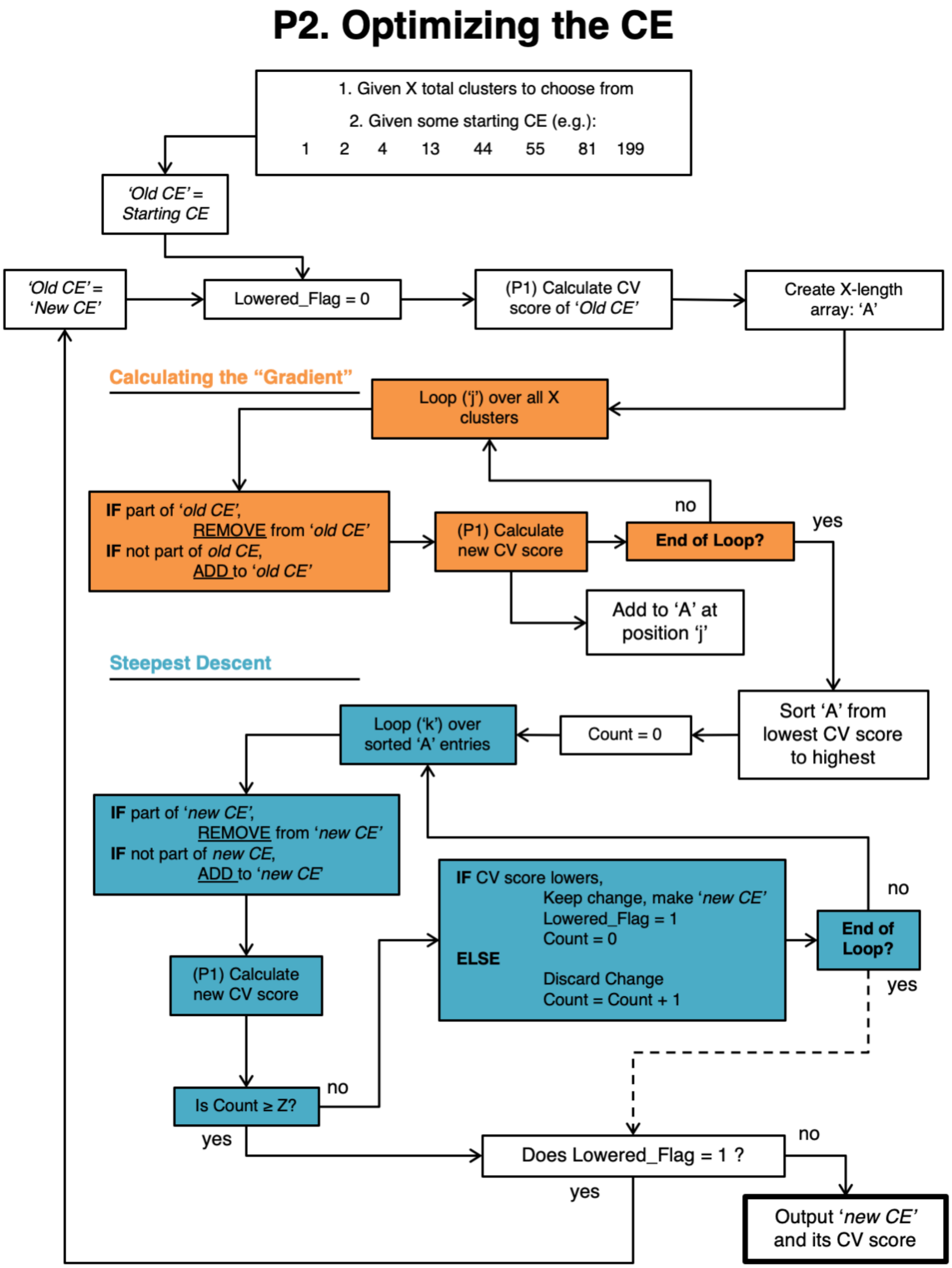

P2. Optimizing the CE
1. Given X total clusters to choose from
2. Given some starting CE (e.g.):
1 2 4 13 44 55 81 199
'Old CE' = Starting CE
'Old CE' = 'New CE'
Lowered_Flag = 0
(P1) Calculate CV score of 'Old CE'
Create X-length array: 'A'
Calculating the "Gradient"
Loop ('j') over all X clusters
IF part of 'old CE', REMOVE from 'old CE'
IF not part of old CE, ADD to 'old CE'
(P1) Calculate new CV score
End of Loop?
no
yes
Add to 'A' at position 'j'
Steepest Descent
Sort 'A' from lowest CV score to highest
Count = 0
Loop ('k') over sorted 'A' entries
IF part of 'new CE', REMOVE from 'new CE'
IF not part of new CE, ADD to 'new CE'
(P1) Calculate new CV score
Is Count ≥ Z?
no
yes
IF CV score lowers, Keep change, make 'new CE' Lowered_Flag = 1 Count = 0
ELSE Discard Change Count = Count + 1
End of Loop?
no
yes
Does Lowered_Flag = 1 ?
no
yes
Output 'new CE' and its CV score

Figure 6.4. Flow diagram for the steepest descent algorithm (Process #2, P2) described here. The color coding is to help the reader identify each distinct part of the algorithm.

6.3.3 Optimizing the Cluster Expansion: Global Minimum Search and the External CV Score.

With a parameter space of X potential clusters, it can be anticipated that a large number of local minima exist within the optimization space. To find at least an approximate global minimum, a large number of randomly (or partial randomly) generated CEs must be generated and optimized via P2. This process, Process #3 or P3, is described in Figure 6.5.

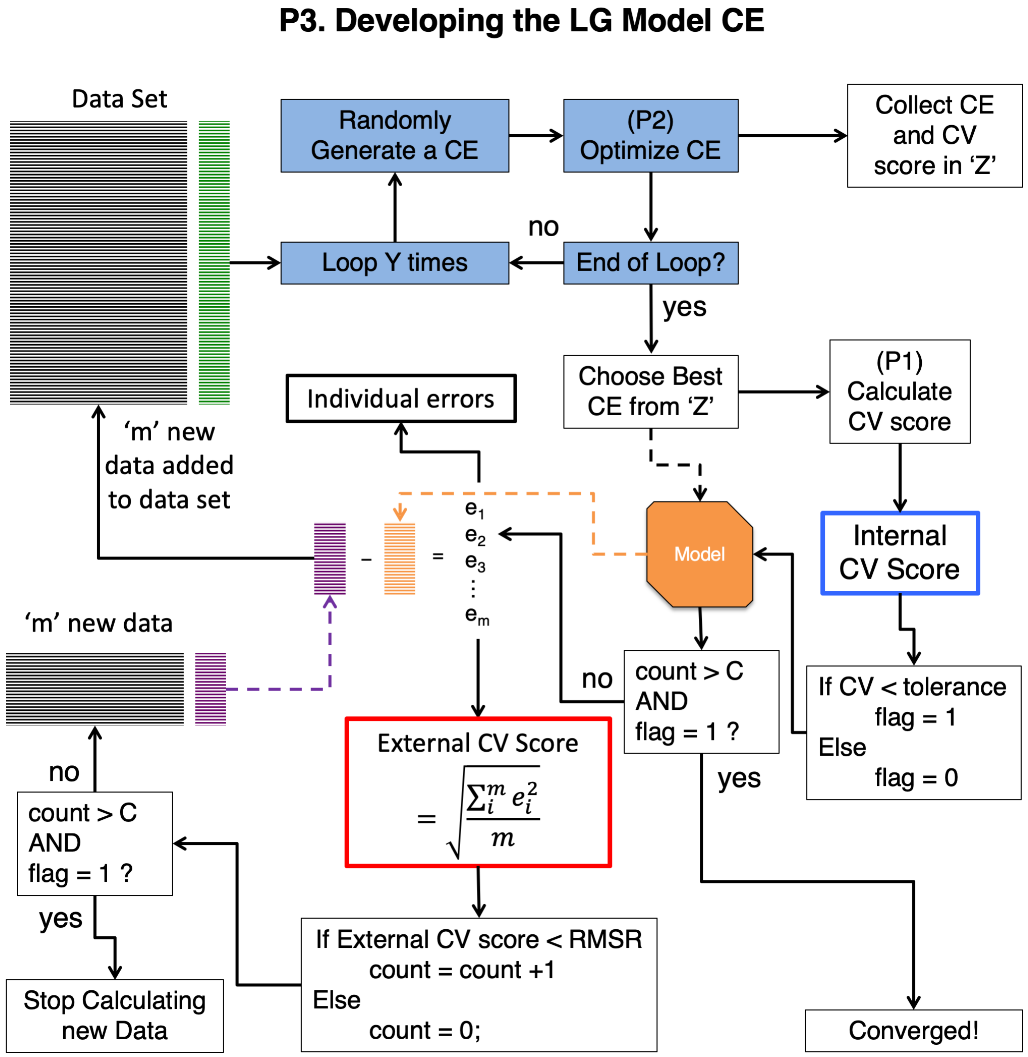


Figure 6.5. Overall process for developing a LG CE as new data is added to the dataset. The values in thick black, blue, and red boxes are collected during each loop along with the RMSR for graphic representation. Full arrow connectors represent logical progression while dashed lines represent a flow of information.

As can be seen in Figure 6.5, the data set is used to optimize a large number of randomly generated CEs (we have generally used no less than 30, often 100s). After all of these CEs have been optimized, the one with the best CV score is chosen and can be reported for this data set. However, it is helpful to have an external validation of the predictiveness of the LG CE beyond the LMO-CV score. In principle, the LG CE should be capable of predicting the energies of *new* data, structures that are outside the data set and therefore never seen by the algorithm. This introduces the idea of an external CV score, which is determined by calculating new data with DFT and then seeing how well these new energies are predicted with the current LG CE. This is then contrasted by the *internal* CV score, which can be the LOO-CV score, LMO-CV score, or any other metric of cross validation that is similarly devised that uses in some fashion all of the data within the data set. This external CV score is very sensitive to the characteristics of the new data, and as such, is not as suitable as the internal CV score (the LMO-CV score, here) as an optimization function. Nonetheless, it is a useful check since a truly robust LG CE should be able to predict the energies for entirely new data to within at least the same accuracy with which the CE describes data within the dataset, i.e. it should fall below the RMSR. After this, the new data are added to the dataset set to help refine the LG CE. If the external CV score does fall below the RMSR, we have instituted an idealized heuristic for convergence stating that the external CV score must fall below the RMSR and stay there over C = 4 subsequent loops of P3. (We would consider falling to within twice the RMSR the lowest bound on convergence). However, this choice is based on our general experience and is thus more or less arbitrary; C can therefore be chosen by the user as well if desired. This can be seen in Figure 6.5. This describes a general cycle that then continues until both the internal and external CV scores reduce to their respective acceptable tolerances and the CE is declared complete.

6.3.4 Augmentation with a Mean Field

AMALGM has been designed to allow for the optional implementation of pre-processing energy data using a mean field (MF) model. The rationale behind this is that by describing the largest energy changes in a mean field manner, the job of describing the remaining energy fluctuations should be more straightforward. Such a procedure was adopted in the past in a phenomenological multi-site lattice gas model of CO/Pt(111)[38] and, more recently, in a lattice gas model of O/Fe(100) where the lateral interactions between the oxygen adatoms where determined from first principles.[91] With a set of *ab initio* data, the "coverage determining sites" can be specified and a MF model constructed from a user-specified n^th^-order polynomial expansion of those sites. The program can then be instructed to use the residuals of the least squares fitting procedure as the input energies for the LG model fitting algorithm.

The MF augmented LG Hamiltonian is shown in Equation 6.10, where we explicitly convert total energy to surface energy, γ, via a division by N_s:

$$\begin{gathered}\gamma = \frac{E(\boldsymbol{n})}{N_\mathrm{s}} = V_0\theta + c\sum_{n=1} \frac{V_n^{mf}}{n+1}\theta^{n+1} \\ + \frac{1}{N_\mathrm{s}}\left(\sum_{X_1}^{\infty} V_{X_1} m_{X_1} + \sum_{X_2}^{\infty} V_{X_2} m_{X_2} + \sum_{X_3}^{\infty} V_{X_3} m_{X_3} + \dots\right)\end{gathered} \tag{6.10}$$

where the terms in parentheses are the LG contribution from Eq. 3 and the first two terms are the MF model. Here, θ is coverage of the "coverage-determining-sites," which can be specified as any or all of the site numbers defined by the user; V_0 is the MF adsorption energy of the coverage-determining-sites at the limit of zero coverage; V_n^{mf} are the MF coefficients corresponding to $n+1$ body interactions (or, in terms of its formulation, the interaction of 1 site with n other bodies[93]); and c is the nearest neighbor coordination number of the surface (e.g. 4 for square lattices, 6 for hexagonal).

The degree of polynomial to be used in the MF model and how often to refit it to the data are entirely up to the user and should be chosen with some care. As with any polynomial fitting, the higher the degree of polynomial, the better the fit will be. However, there is risk of overfitting here as well. Even though the LG model is meant to "fill the gaps" in the MF model, severe overfitting can be difficult to compensate for. It is also unnecessary and ill-advised to continually refit the MF model as new data is added to the dataset, which can make convergence of the LG model difficult due to the fluctuations in the residuals of the MF model that the LG model is trying to fit to. While the polynomial degree could be systematically changed along with the clusters in the LG model to ultimately find the most predictive overall model, automating the fitting schedule of the MF model is not a straightforward matter. As a result, MF augmentation is seen as a preprocessing step for the dataset and the user must take care to implement it optimally. Nonetheless, as shown by Bray et al.,[91] augmentation with a MF model can have excellent utility and yield highly predictive results when applied properly.

6.4. Example: O/Fe(100)

To demonstrate the algorithms and utility of the AMALGM code, we construct here a LG CE of O/Fe(100) using the raw data from Bray et al.[91] and fitting using the procedures outlined in this paper. In the work of Bray et al., the MF option was invoked and the interested reader is directed to that work for an example of this option. Here, we choose to instead use our implemented option of pre-defining the adsorption energy of isolated O on Fe(100) and also the first nearest neighbor (1NN) interaction energy. These two quantities are calculated as:

$$E_{\mathrm{ads}}^{0} = E_{\mathrm{O/Fe(100)}} - E_{\mathrm{Fe(100)}} - \frac{1}{2}E_{\mathrm{O_2(g)}} \tag{6.11}$$

$$E_{\mathrm{1NN}} = E_{\mathrm{2O/Fe(100)}} - E_{\mathrm{Fe(100)}} - E_{\mathrm{O_2(g)}} - 2E_{\mathrm{ads}}^{0} \tag{6.12}$$

Where E_{ads}^{0} is the adsorption energy of O on Fe(100) with superscript "0" indicating the adsorption energy is taken at the zero coverage limit, $E_{\mathrm{O/Fe(100)}}$ is the total DFT energy of the system with O adsorbed on the Fe(100) surface, $E_{\mathrm{Fe(100)}}$ is the total DFT energy of the clean Fe(100) surface, $E_{\mathrm{O_2(g)}}$ is the DFT energy of gas-phase O_2, E_{1NN} indicates the 1NN interaction energy and $E_{\mathrm{2O/Fe(100)}}$ is the DFT energy of two O atoms on Fe(100) placed at the 1NN distance from each other but otherwise isolated. These quantities, E_{ads}^{0} and E_{1NN}, were found to be -3.08 eV/O and +0.229 eV/interaction, respectively.

The surface unit cell was inputted as

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5.335 \end{pmatrix} \tag{6.13}$$

which is a simplified Fe(100) surface unit cell where the lattice constant of Fe has been factored out. Doing this allows us to define lengths in "units" of lattice constants (i.e. the distance from one top site to the next is "1" instead of 2.868 Å). Unless the z-component of the user-defined shift vectors have non-constant values, the final column in Equation 6.13 has no real consequence. This is the case here but may not be the case for other systems thus necessitating that a 3×3 matrix generally be defined.

Only a single site, the 4-fold hollow site is defined in the data and was provided as input to the AMALGM code with shift vector {0,0,0} and site type "1". The radial cutoff for 2-body and 3-body clusters was set to "4" (i.e. 11.472 Å) while the radial cutoff for 4-body clusters was set to "3" (i.e. 8.604 Å). Higher-body clusters beyond this were not considered. In total, 77 *k*-

body clusters were found using this set-up: one 1-body cluster, 9 2-body clusters, 35 3-body clusters, and 32 4- body clusters. The corresponding labels, X_k, outputted by AMALGM are shown in Table 6.1. While it is by no means a necessity, the benefit of factoring out the lattice constant is more apparent here: distances are kept integer in value. The cluster labels shown in Table 6.1 are depicted in Figures C2 and C3.

Table 6.1. Cluster labels, X_k, for the O/Fe(100) system as outputted by AMALGM. The first column is the cluster ID # that is used to reference the label with the number of such clusters in a particular configuration. Columns labeled "B_1 – B_4" show the site type(s) involved in the *k*-body cluster, where a "0" indicates the absence of a body. The remaining columns are the (squared) distances between pairwise bodies indicated by the subscripts.

Cluster ID #	B_1	B_2	B_3	B_4	R_{12}	R_{13}	R_{23}	R_{14}	R_{24}	R_{34}
1	1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	1	0	0	0	0	0
3	1	1	0	0	2	0	0	0	0	0
4	1	1	0	0	4	0	0	0	0	0
5	1	1	0	0	5	0	0	0	0	0
6	1	1	0	0	8	0	0	0	0	0
7	1	1	0	0	9	0	0	0	0	0
8	1	1	0	0	10	0	0	0	0	0
9	1	1	0	0	13	0	0	0	0	0
10	1	1	0	0	16	0	0	0	0	0
11	1	1	1	0	1	1	2	0	0	0
12	1	1	1	0	1	4	1	0	0	0
13	1	1	1	0	1	2	5	0	0	0
14	1	1	1	0	4	2	2	0	0	0
15	1	1	1	0	1	4	5	0	0	0
16	1	1	1	0	2	8	2	0	0	0
17	1	1	1	0	5	5	2	0	0	0
18	1	1	1	0	1	5	8	0	0	0
19	1	1	1	0	1	9	4	0	0	0
20	1	1	1	0	4	5	5	0	0	0
21	1	1	1	0	1	5	10	0	0	0
22	1	1	1	0	4	2	10	0	0	0
23	1	1	1	0	4	4	8	0	0	0
24	1	1	1	0	9	2	5	0	0	0
25	1	1	1	0	1	9	10	0	0	0
26	1	1	1	0	5	13	2	0	0	0
27	1	1	1	0	10	5	5	0	0	0
28	1	1	1	0	10	8	2	0	0	0
29	1	1	1	0	1	8	13	0	0	0
30	1	1	1	0	4	5	13	0	0	0
31	1	1	1	0	9	5	8	0	0	0
32	1	1	1	0	1	10	13	0	0	0
33	1	1	1	0	4	10	10	0	0	0

34	1	1	1	0	4	16	4	0	0	0
35	1	1	1	0	1	16	9	0	0	0
36	1	1	1	0	4	9	13	0	0	0
37	1	1	1	0	16	5	5	0	0	0
38	1	1	1	0	2	13	13	0	0	0
39	1	1	1	0	10	10	8	0	0	0
40	1	1	1	0	10	13	5	0	0	0
41	1	1	1	0	16	2	10	0	0	0
42	1	1	1	0	9	10	13	0	0	0
43	1	1	1	0	16	8	8	0	0	0
44	1	1	1	0	16	5	13	0	0	0
45	1	1	1	0	16	13	13	0	0	0
46	1	1	1	1	1	1	2	2	1	1
47	1	1	1	1	1	4	1	2	1	2
48	1	1	1	1	1	2	5	1	2	1
49	1	1	1	1	1	4	1	1	2	5
50	1	1	1	1	1	2	5	2	1	4
51	1	1	1	1	2	2	4	4	2	2
52	1	1	1	1	5	2	1	5	2	1
53	1	1	1	1	1	1	2	5	8	2
54	1	1	1	1	1	2	5	4	5	2
55	1	1	1	1	4	2	2	5	5	1
56	1	1	1	1	1	4	1	5	4	5
57	1	1	1	1	1	4	1	9	4	1
58	1	1	1	1	1	4	5	5	4	1
59	1	1	1	1	1	1	2	8	5	5
60	1	1	1	1	1	2	5	5	4	5
61	1	1	1	1	1	2	5	5	8	1
62	1	1	1	1	1	9	4	2	1	5
63	1	1	1	1	4	2	2	4	8	2
64	1	1	1	1	1	4	1	4	5	8
65	1	1	1	1	1	9	4	5	2	2
66	1	1	1	1	1	2	5	5	2	9
67	1	1	1	1	1	5	8	4	5	1
68	1	1	1	1	1	5	8	5	4	4
69	1	1	1	1	5	2	9	4	5	2
70	1	1	1	1	5	2	9	5	2	5
71	1	1	1	1	9	2	5	5	8	1
72	1	1	1	1	1	9	4	5	4	8
73	1	1	1	1	9	2	5	8	5	2
74	1	1	1	1	1	9	4	8	5	5
75	1	1	1	1	4	4	8	8	4	4
76	1	1	1	1	1	5	8	8	5	9
77	1	1	1	1	5	2	9	9	8	5

We should note that the number of potential clusters increases dramatically as site-types are added and that careful consideration of the *k*-body cut-off radii may be necessary to avoid prohibitively large numbers of potential clusters. Parallelization of AMALGM is being considered to extend its applicability to larger/more complicated systems without having to impose egregious cut-off radii.

With the potential clusters found, 271 structures with their associated DFT energies were inputted from Bray et al.'s work. AMALGM found that only 204 of these were unique. Here, uniqueness is determined simply by comparing the number of clusters per site and if all these are the same between two or more structures, all but the lowest energy structure is removed from consideration.

Typically, one would not know how much data is necessary to produce a predictive, converged LG CE, i.e. one wouldn't start with 204 structures already in the dataset. Instead, one would preferably start with a much smaller data set and only perform *ab initio* calculations for new structures if necessary. To simulate this, we begin by adding 17 structures and then running the "P3" algorithm shown in Figure 6.5 where at least 30 of the inner loops are made to populate the "Z" array (in practice it was often more than this as the only cost to add more is the time needed to loop through the inner loop of P3). Each inner loop of P3 invokes the "P2" algorithm shown in Figure 6.4. We show sample output from this algorithm using AMALGM in Figure C4 Based on the considerations detailed previously and illustrated in Figures 6.3-6.5, the final CE determined by our algorithms is shown in Figure 6.6.

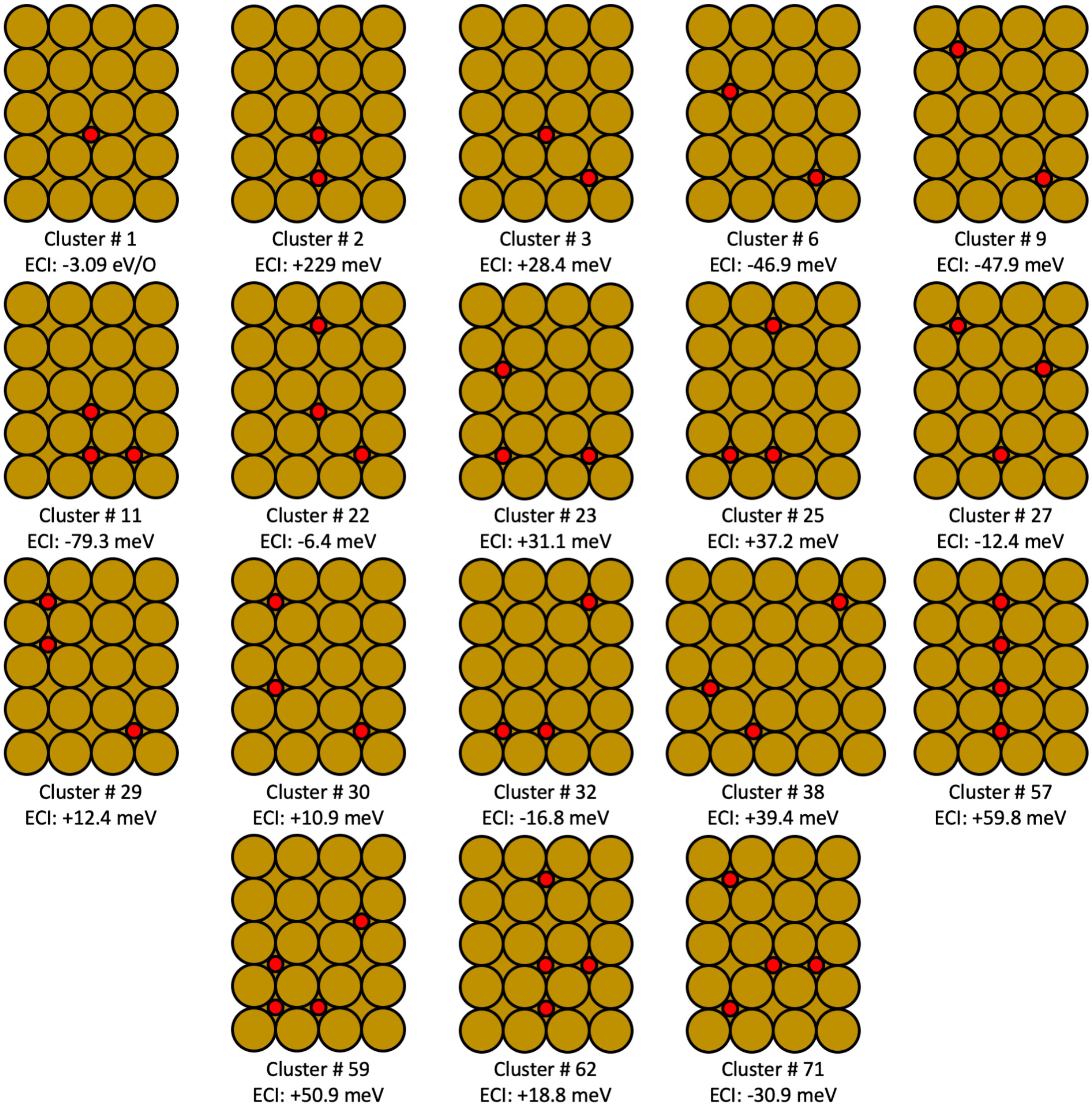


Figure 6.6. Final LG CE for O/Fe(100) and its associated ECIs using the LMO-CV score as objective function. The red circles are oxygen atoms and the gold spheres are Fe atoms.

We can construct a parity plot between the DFT-calculated surface energies and those predicted by the LG CE shown in Figure 6.6. This is shown in Figure 6.7. The plot shows nearly perfect parity with barely visible deviations. This reflects the low RMSR of 9.9 meV/site. Because the LMO-CV score is also low (11.5 meV/site) and the external CV score appears well converged (final four values: 9.7, 4.5, 9.8, and 6.2 meV/site), we can declare this a highly predictive and converged LG CE model. This demonstrates the practicality and power of the AMALGM code.

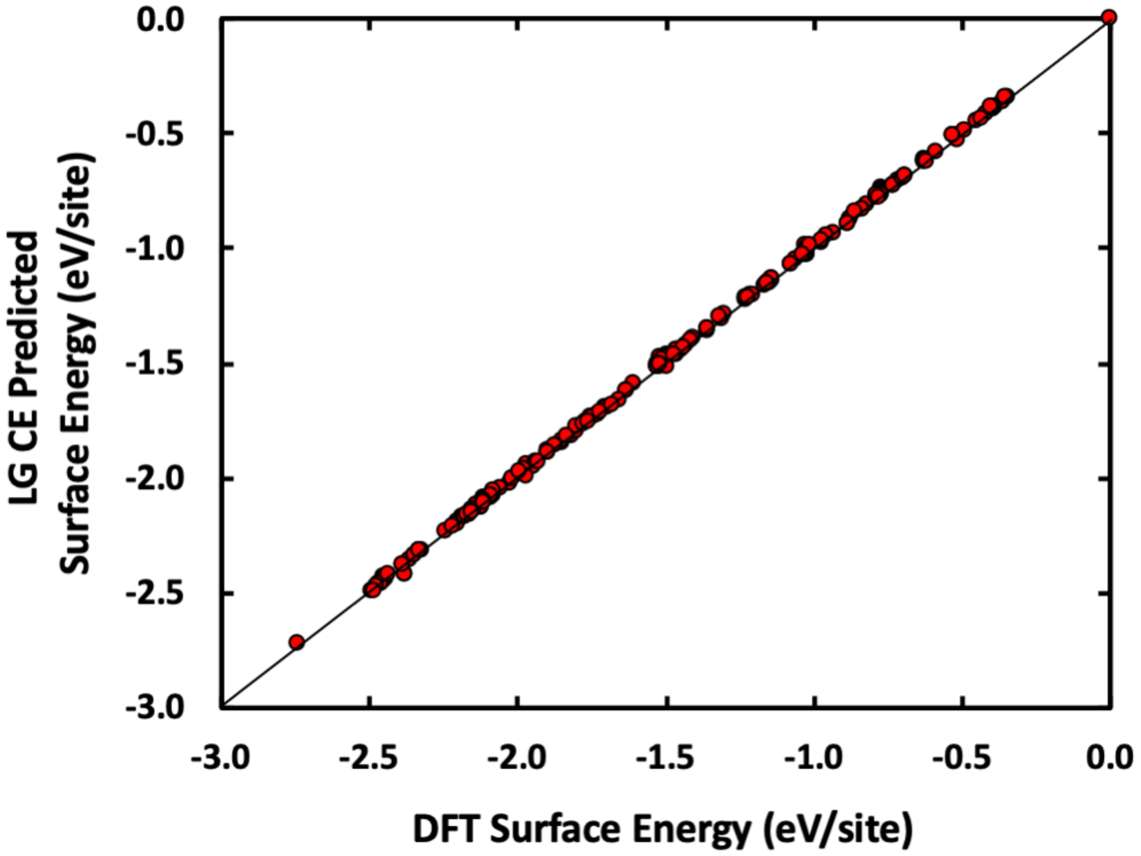


Figure 6.7. Parity plot of DFT vs LG-predicted energies using the LG CE found using AMALGM and the LMO-CV score.

6.5. Leave-Multiple-Out vs. Leave-One-Out Cross-Validation

To illustrate the practical difference between the LOO-CV score and LMO-CV score (and confirm that LMO-CV is indeed preferable), we ran two simultaneous "simulations" using both LOO and LMO for the "P1" algorithms in Figure C1 and Figure 6.3, respectively. To begin, 17 DFT-optimized structures were added to make the data set. P3 was then ran to determine the CE that minimizes the internal CV score, once with the LMO-CV score and once with the LOO-

CV score. The results were quite striking even at this early stage: the best LOO-CV score was 0.002 meV/site (effectively 0 meV/site) and corresponded to a CE containing 13 clusters {1, 2, 13, 17, 20, 22, 26, 30, 31, 52, 53, 72, 76} (see Figures C2 and C3 to cross reference the cluster structures), while the best LMO-CV score was 16.0 meV/site corresponding to a CE containing 4 clusters {1, 2, 13, 53}. This illustrates a general tendency of the LOO-CV score: it tends to choose larger CEs than the LMO-CV score. These internal CV scores are plotted in blue in Figure 6.8A and 6.8C and the number of clusters in each CE is shown in purple in Figure 6.8B and 8D for the LMO- and LOO-CV scores, respectively. It should be noted that the resulting LG CE model corresponding to the above CEs is constructed by fitting the ECIs to *all* 17 data points; the root mean squared residual (RMSR) of this fit is plotted in green in Figure 6.8A and 6.8C. These CEs and their corresponding ECIs constitutes the LG CE model associated with the number of unique structures in the dataset; here, 17.

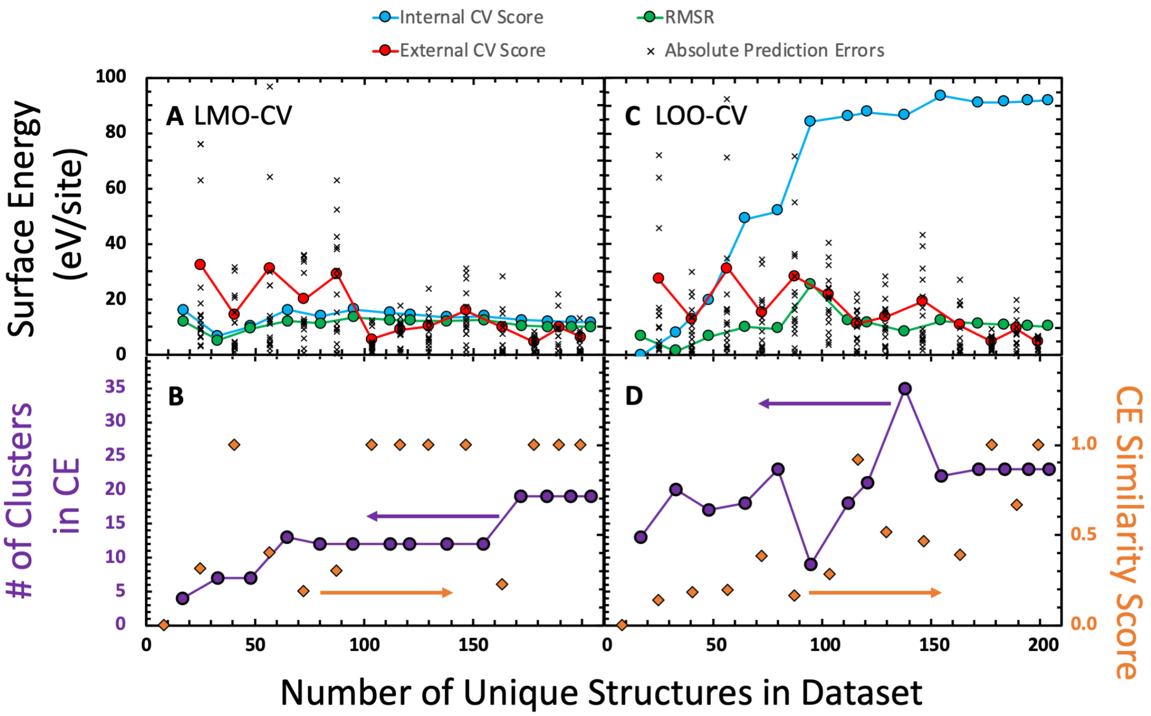


Figure 6.8. Simulated progressions of CE optimization using the P1 – P3 algorithms shown in Figures 6.3-6.5. Panels (A-B) Progression in the LMO-CV score objective function (blue) for choosing the best CE as the data included in the dataset increases is compared to the (B & D) LOO-CV score.

While the LOO-CV score is lower than the LMO-CV score, we find that this does not translate to higher predictiveness. The DFT surface energies of 17 new structures (i.e. never included in the evaluation of the internal CV scores) were predicted using each LG CE model (final step in P3, seen in Figure 6.5). The absolute errors on these predictions are shown forward-offset as black X’s in Figure 6.8A and 6.8C, and the external CV score is determined from these and shown in red. These first external CV scores are 27.4 meV/site and 32.2 meV/site for the LOO- and LMO-determined LG CE models, respectively. While the LOO-determined LG CE model gives a slightly better external CV score than LMO-CV score here, neither is very good. We thus add these new structures to the dataset and begin the process of determining the best CE again. The best CEs are then used to predict the energies of another set of structures as before. This process is continued for all 204 structures in the data set and tracked in Figure 6.8(A-B) for the LMO-CV score and Figure 6.8(C-D) for the LOO-CV score. To avoid frivolous changes to the CE, we impose a rule that the CE is only allowed to change if the new CE lowers the internal CV score by at least 1 meV/site as compared to what the current CE provides for the new dataset.

Another metric of interest is the convergence of the CE itself: ideally the clusters found to provide the most predictiveness should not change if the CE is converged or converging. Clearly, the size of the data set reflects this, but if every one of the clusters changes as data is added but the overall number of clusters doesn't, this won't be reflected. A "CE similarity score" (CESS) is devised here to directly assess this and is defined as the dot product of a normalized vector representing the CE, $\boldsymbol{CE_i}$ with the normalized vector representing the previous CE, $\boldsymbol{CE_{i-1}}$:

$$\boldsymbol{CESS} = \left(\frac{\boldsymbol{CE_i}}{\|\boldsymbol{CE_i}\|}\right) \cdot \left(\frac{\boldsymbol{CE_{i-1}}}{\|\boldsymbol{CE_{i-1}}\|}\right) \tag{6.14}$$

where $\boldsymbol{CE_i}$ is an occupation vector of 1's and 0's with as many components as there are potential clusters, each of which corresponds to the cluster ID numbers in Table 6.1 and Figures C2 and C3. If a cluster is present in the CE, its associated component is "1". Otherwise, the component is "0". As an example, if there were 10 potential clusters and the current CE was {1,2, 5, 7, 8}, the CE vector would be $\boldsymbol{CE_i} = \langle 1, 1, 0, 0, 1, 0, 1, 1, 0, 0 \rangle$. The CESS is thus defined between 0, (no similarity) and 1 (perfect similarity). This value is shown left-offset in orange in Figure 6.8B and 6.8D for the LMO- and LOO-determined CEs, respectively (note: offsetting to the left effectively places each CESS data point between the two CEs it compares). It should be noted that this value contains no information about the relative ECIs associated with the clusters. While a similarity score can be defined that includes this information (replacing the 1's in the vector with the clusters' ECI values), such a definition is sensitive to the relative magnitudes of ECIs and will be biased toward the clusters with largest ECIs. Furthermore, the dot product would be defined from -1 to 1 due to potential sign changes and whether the dot product is greater than or less than 0 would be determined by the relative magnitude of same-sign and opposite-sign ECIs. For certain applications this definition could be useful, but for our purposes here, the CESS as defined in Equation 6.14 is preferable as it treats the presence or absence of any one cluster as equivalent to the presence or absence of any other cluster.

With the terms defined, we can now discuss the final results of these "simulations" shown in Figure 6.8. The first and most striking difference between the two internal CV methods is that the internal LOO-CV score trends upward and increases dramatically as the data set grows but that the internal LMO-CV score trends mostly downward and remains similar in magnitude to the RMSR. This means that the internal LMO-CV score reflects a more consistent fit to the data; the ECIs found from fits to the validation sets must be much more similar to the ECIs found from the final fit of the data. This is not the case for internal LOO-CV score. In fact, if the external CV score were not also being tracked, the internal LOO-CV score would indicate that more data was needed because, as an estimate of error, an internal LOO-CV score of nearly 100 meV/site is far from acceptable (see the far-right values of the blue data series in Figure 6.8C). On the other hand, the final internal LMO-CV score is 11.5 meV/site, which is a far more acceptable value (see the far-right values of the blue data series in Figure 6.8A).

If we assume the external CV score and CESS *are* always tracked in a practical setting, these values would show that convergence occurred or is occurring after 172 structures were added to the dataset for the LOO-determined CEs. However, the CESS is not perfectly stable after 172 structures, a change to the CE occurred at 195 structures (see the last three CESSs, orange data points, in Figure 6.8D) and would likely indicate that more data was required to finally declare convergence. The CESS does not show a single value of 1 up to this point, as well, showing that the CE is constantly changing as new data is added, often radically. On the other hand, the LMO-CV score fell below the RMSR after only 95 structures were added to the dataset and stayed there from then on with only one data point going slightly above it (with a value of ~1.33*RMSRs). The CESS from 95 to 155 structures in the dataset was 1, indicating the CE was perfectly stable throughout, as well. The CE did change after 155 but again remained stable thereafter.

Interestingly, if the prior stable CE corresponding to the flat CESS in Figure 6.8B (between 95 – 155 structures) in the dataset is kept instead, we find that the external CV score, while slightly higher (no more than 2.3 meV/site), does not change drastically enough to affect the overall conclusion that the CE has converged. This is shown in Figure C5. Assuming that we'd declare the CE converged after 4 times of the CE not changing, the root mean difference in the ECI values from this point (138 structures in the dataset) to the final CE (204 structures in the dataset) is only 6 meV/interaction. These two CEs and their ECIs are shown in Table 6.2.

Table 6.2. ECIs (in meV) for the first stable CE in Figure 6.8A based on fitting to the first 138 structures in the dataset and the fit to the total 204 structures. Because they are inputted and don't change, cluster number 1 and 2 (the adsorption energy and first nearest neighbor interaction, as defined in Equation 6.10 and Equation 6.11) are not shown.

		Structures in Data set	
		138	204
Clusters in Data set	4	-50.9	-51.8
	11	-49.8	-48.0
	14	34.0	24.1
	16	-33.2	-21.4
	18	-41.9	-40.9
	23	17.6	22.9
	37	38.8	34.0
	56	71.3	63.8
	59	90.9	87.6
	60	-35.8	-31.8

Finally, we note that we add structures in "batches" because this more accurately simulates the process by which one would typically construct a LG CE model. Typically, one would calculate the energies of new structures in a roughly continuous way and, as the dataset grows, wish to check if the CE is converged enough to stop new calculations. Unfortunately, finding the best CE given either the LOO- or LMO- CV scores is not instantaneous, and it is more pragmatic to determine the best CE for the current data set and allow a buffer of new calculations to build up in the meantime. The external CV score would ideally be determined from the prediction errors on this buffer of data. As such, while not a formal statement, it is reasonable to expect that a converged CE should have the external CV score fall below 1 – 2 times the RMSR and remain there for at least 4 separate evaluations of the external CV score based on at least 10 new structures per evaluation.

6.6. Conclusions

We have presented here new computational and theoretical schemes and algorithms aimed at providing researchers with a tool for creating robust and arbitrarily complex multicomponent lattice gas cluster expansions based on *ab initio* data for surfaces and interfaces. These algorithms have been implemented in the *Ab initio* Mean-field Augmented Lattice Gas Model code (AMALGM). AMALGM is in its beta stage of development and can be made available to researchers upon request in the near future. The algorithms described herein are used on the O/Fe(100) system to illustrate the performance of AMALGM. This same system is used to demonstrate the desirability of the leave-multiple-out cross validation score over the popular leave-one-out cross validation score when used as the objective function for lattice gas cluster expansion optimization. The final release of AMALGM is expected to be written in a high-level, non-proprietary language (i.e. C++) that can be compiled on standard UNIX infrastructures with optional parallelization schemes.

Acknowledgements

J.-S.M. was primarily funded by the U.S. Department of Energy (DOE), Office of Basic Energy Sciences, Division of Chemical Sciences, Biosciences and Geosciences under Award Number DE-SC0014560. G.C. was supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. 1347973 and the NSF East Asia and Pacific Summer Institutes (EAPSI) fellowship under grant number 1613890. G.C., Seattle Chapter ARCS Fellow, NSF EAPSI Fellow, and NSF Graduate Research Fellow, also gratefully acknowledges financial support from the Achievement Rewards for College Scientists foundation. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. DOE.

REFERENCES

[1] W. Kohn, L. J. Sham, Phys. Rev. 140 (1965) A1133-A1138.

[2] X. Ma, Z. Li, L. E. K. Achenie, H. Xin, J. Phys. Chem. Lett. 6 (2015) 3528-3533.

[3] T. Morawietz, J. Behler, J. Phys. Chem. A 117 (2013) 7356-7366.

[4] H. T. T. Nguyen, H. M. Le, J. Phys. Chem. A 116 (2012) 4629-4638.

[5] J. Behler, R. Martoňák, D. Donadio, M. Parrinello, Phys. Rev. Lett. 100 (2008) 185501.

[6] M. Gastegger, P. Marquetand, Journal of Chemical Theory and Computation 11 (2015) 2187-2198.

[7] J. N. Wei, D. Duvenaud, A. Aspuru-Guzik, ACS Central Science 2 (2016) 725-732.

[8] R. Z. Khaliullin, H. Eshet, T. D. Kühne, J. Behler, M. Parrinello, Phys. Rev. B 81 (2010) 100103.

[9] S. Lorenz, A. Groß, M. Scheffler, Chem. Phys. Lett. 395 (2004) 210-215.

[10] B. Kolb, X. Luo, X. Zhou, B. Jiang, H. Guo, J. Phys. Chem. Lett. 8 (2017) 666-672.

[11] S. K. Natarajan, J. Behler, Phys. Chem. Chem. Phys. 18 (2016) 28704-28725.

[12] T. B. Blank, S. D. Brown, A. W. Calhoun, D. J. Doren, The Journal of Chemical Physics 103 (1995) 4129-4137.

[13] S. Kondati Natarajan, J. Behler, J. Phys. Chem. C 121 (2017) 4368-4383.

[14] R. Jinnouchi, R. Asahi, J Phys Chem Lett (2017) 4279-4283.

[15] R. Jinnouchi, H. Hirata, R. Asahi, J. Phys. Chem. C 121 (2017) 26397-26405.

[16] N. Artrith, A. M. Kolpak, Nano Lett. 14 (2014) 2670-2676.

[17] N. Artrith, A. M. Kolpak, Comput. Mater. Sci. 110 (2015) 20-28.

[18] B. Sun, M. Fernandez, A. S. Barnard, Journal of Chemical Information and Modeling 57 (2017) 2413-2423.

[19] J. M. Sanchez, F. Ducastelle, D. Gratias, Physica A: Statistical Mechanics and its Applications 128 (1984) 334-350.

[20] J. M. Sanchez, Phys. Rev. B 48 (1993) 14013-14015.

[21] J. M. Sanchez, Phys. Rev. B 81 (2010) 224202.

[22] J. M. Sanchez, Journal of Phase Equilibria and Diffusion 38 (2017) 238-251.

[23] K. Binder, D. P. Landau, Surf. Sci. 108 (1981) 503-525.

[24] K. Binder, Rep. Prog. Phys. 50 (1987) 783.

[25] S. H. Payne, J. M. LeDue, J. C. Michael, H. J. Kreuzer, R. Wagner, K. Christmann, Surf. Sci. 512 (2002) 151-164.

[26] A. P. J. Jansen, Phys. Rev. B 69 (2004) 035414.

[27] D.-J. Liu, J. W. Evans, Surf. Sci. 563 (2004) 13-26.

[28] D.-J. Liu, The Journal of Chemical Physics 121 (2004) 4352-4357.

[29] G. Zvejnieks, V. N. Kuzovkov, V. Petrauskas, E. E. Tornau, Appl. Surf. Sci. 252 (2006) 5395-5398.

[30] D.-J. Liu, J. W. Evans, The Journal of Chemical Physics 124 (2006) 154705.

[31] D.-J. Liu, J. Phys. Chem. C 111 (2007) 14698-14706.

[32] J. W. D. Connolly, A. R. Williams, Phys. Rev. B 27 (1983) 5169-5172.

[33] C. Stampfl, H. J. Kreuzer, S. H. Payne, H. Pfnür, M. Scheffler, Phys. Rev. Lett. 83 (1999) 2993-2996.

[34] J.-S. McEwen, S. H. Payne, C. Stampfl, Chem. Phys. Lett. 361 (2002) 317-320.

[35] K. A. Fichthorn, M. Scheffler, Phys. Rev. Lett. 84 (2000) 5371-5374.

[36] K. Honkala, P. Pirilä, K. Laasonen, Phys. Rev. Lett. 86 (2001) 5942-5945.

[37] E. Schröder, Comput. Mater. Sci. 24 (2002) 105-110.

[38] J.-S. McEwen, S. H. Payne, H. J. Kreuzer, M. Kinne, R. Denecke, H. P. Steinrück, Surf. Sci. 545 (2003) 47-69.

[39] C. G. M. Hermse, F. Frechard, A. P. van Bavel, J. J. Lukkien, J. W. Niemantsverdriet, R. A. van Santen, A. P. J. Jansen, The Journal of Chemical Physics 118 (2003) 7081-7089.

[40] H. Tang, A. Van Der Ven, B. L. Trout, Mol. Phys. 102 (2004) 273-279.

[41] H. Tang, A. Van der Ven, B. L. Trout, Phys. Rev. B 70 (2004) 045420.

[42] G. S. Karlberg, G. Wahnström, Phys. Rev. Lett. 92 (2004) 136103.

[43] R. Singer, R. Drautz, M. Fähnle, Surf. Sci. 559 (2004) 241-248.

[44] S. Ovesson, B. I. Lundqvist, W. F. Schneider, A. Bogicevic, Phys. Rev. B 71 (2005) 115406.

[45] S. H. Payne, J. S. McEwen, H. J. Kreuzer, D. Menzel, Surf. Sci. 594 (2005) 240-262.

[46] R. Drautz, A. Díaz-Ortiz, Phys. Rev. B 73 (2006) 224207.

[47] J. Wintterlin, J. Trost, R. Schuster, A. Eichler, J. S. McEwen, Phys. Rev. Lett. 96 (2006) 166102.

[48] Y. Zhang, V. Blum, K. Reuter, Phys. Rev. B 75 (2007) 235406.

[49] J.-S. McEwen, A. Eichler, J. Chem. Phys. 126 (2007) 094701.

[50] M. Nagasaka, H. Kondoh, I. Nakai, T. Ohta, The Journal of Chemical Physics 126 (2007) 044704.

[51] M. Borg, C. Stampfl, A. Mikkelsen, J. Gustafson, E. Lundgren, M. Scheffler, J. N. Andersen, ChemPhysChem 6 (2005) 1923-1928.

[52] J. Rogal, K. Reuter, M. Scheffler, Phys. Rev. B 77 (2008) 155410.

[53] C. Lazo, F. J. Keil, Phys. Rev. B 79 (2009) 245418.

[54] R. Sathiyanarayanan, T. L. Einstein, Surf. Sci. 603 (2009) 2387-2392.

[55] S. Piccinin, C. Stampfl, Phys. Rev. B 81 (2010) 155427.

[56] V. I. Tokar, H. Dreyssé, Phys. Rev. B 82 (2010) 115446.

[57] T. Franz, F. Mittendorfer, The Journal of Chemical Physics 132 (2010) 194701.

[58] S. Piskunov, G. Zvejnieks, Y. F. Zhukovskii, S. Bellucci, Thin Solid Films 519 (2011) 3745-3751.

[59] C. Wu, D. J. Schmidt, C. Wolverton, W. F. Schneider, J. Catal. 286 (2012) 88-94.

[60] Y. Xia, W. Wang, Z. Li, H. J. Kreuzer, Surf. Sci. 617 (2013) 131-135.

[61] P. N. Abufager, G. Zampieri, K. Reuter, M. L. Martiarena, H. F. Busnengo, J. Phys. Chem. C 118 (2014) 290-297.

[62] G. Zvejnieks, A. Ibenskas, E. E. Tornau, Surf. Coat. Technol. 255 (2014) 15-21.

[63] J. M. Bray, J. L. Smith, W. F. Schneider, Top. Catal. 57 (2014) 89-105.

[64] L. Cao, C. Li, T. Mueller, Journal of Chemical Information and Modeling (2018).

[65] M. Asta, C. Wolverton, D. de Fontaine, H. Dreyssé, Phys. Rev. B 44 (1991) 4907-4913.

[66] C. Wolverton, M. Asta, H. Dreyssé, D. de Fontaine, Phys. Rev. B 44 (1991) 4914-4924.

[67] A. H. Nguyen, C. W. Rosenbrock, C. S. Reese, G. L. W. Hart, Phys. Rev. B 96 (2017) 014107.

[68] J. M. Sanchez, Phys. Rev. B 95 (2017) 216202.

[69] A. van de Walle, G. Ceder, Journal of Phase Equilibria 23 (2002) 348.

[70] A. van de Walle, M. Asta, G. Ceder, Calphad 26 (2002) 539-553.

[71] A. v. d. Walle, M. Asta, Modell. Simul. Mater. Sci. Eng. 10 (2002) 521.

[72] A. van de Walle, Calphad 33 (2009) 266-278.

[73] D. Lerch, O. Wieckhorst, G. L. W. Hart, R. W. Forcade, S. Müller, Modell. Simul. Mater. Sci. Eng. 17 (2009) 055003.

[74] A. P. J. Jansen, C. Popa, Phys. Rev. B 78 (2008) 085404.

[75] L. M. Herder, J. M. Bray, W. F. Schneider, Surf. Sci. 640 (2015) 104-111.

[76] W. Chen, P. Dalach, W. F. Schneider, C. Wolverton, Langmuir 28 (2012) 4683-4693.

[77] A. Bajpai, K. Frey, W. F. Schneider, J. Phys. Chem. C 121 (2017) 7344-7354.

[78] D. J. Schmidt, W. Chen, C. Wolverton, W. F. Schneider, Journal of Chemical Theory and Computation 8 (2012) 264-273.

[79] W. Chen, D. Schmidt, W. F. Schneider, C. Wolverton, J. Phys. Chem. C 115 (2011) 17915-17924.

[80] T. C. Kerscher, W. Landgraf, R. Podloucky, S. Müller, Phys. Rev. B 86 (2012) 195420.

[81] P. Welker, O. Wieckhorst, T. C. Kerscher, S. Müller, J. Phys.: Condens. Matter 22 (2010) 384203.

[82] P. Zhang, Ann. Statist. 21 (1993) 299-313.

[83] J. Shao, Journal of the American Statistical Association 88 (1993) 486-494.

[84] K. Baumann, Trends Anal. Chem. 22 (2003) 395-406.

[85] B. Widom, J. Phys. Chem. 86 (1982) 869-872.

[86] B. Widom, J. Stat. Phys. 19 (1978) 563-574.

[87] P. D. Tepesch, G. D. Garbulsky, G. Ceder, Phys. Rev. Lett. 74 (1995) 2272-2275.

[88] B. C. Han, A. Van der Ven, G. Ceder, B. Hwang, Phys. Rev. B 72 (2005) 205409.

[89] G. Kresse, J. Hafner, Phys. Rev. B 47 (1993) 558.

[90] G. Kresse, J. Furthmüller, Comput. Mater. Sci. 6 (1996) 15-50.

[91] G. Kresse, J. Furthmüller, Phys. Rev. B 54 (1996) 11169.

[92] G. Kresse, J. Furthmüller, Vienna: Vienna University (2001).

CHAPTER SEVEN:
QUANTIFYING ERRORS IN THE EFFECTIVE CLUSTER INTERACTIONS OF LATTICE GAS CLUSTER EXPANSIONS

Greg Collinge[a], *Alyssa Hensley*[a], *and Jean-Sabin McEwen*[a,b,c,d,e]*

[a] The Gene & Linda Voiland School of Chemical Engineering and Bioengineering, Washington State University, Pullman WA 99164, USA

[b] Department of Physics and Astronomy, Washington State University, Pullman, WA 99164, USA

[c] Department of Chemistry, Washington State University, Pullman, WA 99164, USA

[d] Institute for Integrated Catalysis, Pacific Northwest National Laboratory, Richland, WA, 99352, USA

[e] Department of Biological Systems Engineering, Washington State University, Pullman, WA 99164, USA

* Corresponding author:
Jean-Sabin McEwen; 509-335-8580 (phone), 509-335-4806 (fax), js.mcewen@wsu.edu

Abstract

The promise of lattice gas (LG) cluster expansions (CEs) is that they can describe any system property to any level of accuracy since the orthogonal “cluster basis functions” have been shown to span the complete space of configurations available to any arbitrarily large surface of finite sites. Unfortunately, this is only true for the case of an ideal, fixed lattice decorated with components at precise lattice points (the lattice “sites”) with no distortions or relaxations subsequently allowed. Since most systems, and surfaces specifically, do not conform to such an ideal set of constraints, errors in the LG CEs must be expected or CE convergence severely

hampered. Beyond this, numerical errors in the provided data can complicate the proper construction of a truly predictive and/or physically significant CE. We show here how reliance on typical statistical tools like confidence intervals cannot be expected to provide an accurate representation of the uncertainty of the effective cluster interactions (ECIs) in the CE due to the nature of the target *ab initio* data and the nature of CEs themselves. We develop a method for estimating these errors that does not rely on statistical assumptions about the model or data. We then use these ECI errors to quantify the fundamental consequences on the uncertainty of ECIs in CEs built from O/Fe(100) data whose surface and adsorbates have been allowed to relax in the typical manner and from O/Fe(100) data whose surface and adsorbates are fixed in ideal lattice positions. We also quantify the effect of using a different density functional theory exchange correlation functional, using these ECI errors to assess the significance in any deviations. In both cases, our method is shown to have incredible utility in the quantification of errors in the ECIs of CEs. While we stick to the lattice gas convention in this work, the method is in principle equally applicable to the Ising convention.

7.1. Introduction

Fitting cluster expansions (CEs)[1-4] to *ab initio* data—generally from density functional theory (DFT) calculations—is a well-established tool for the characterization and rationalization of alloy and surface behavior. The theory (and to an extent, the validity) of CEs relies on the treatment of systems as a configurational problem: that of decorating an ideal, fixed lattice with substitutional components. Early on in the development of CEs, the effect of geometric relaxations in the underlying structural data used to fit the CE was recognized as an issue[5-12] requiring special attention and having considerable consequence on the energetic behavior of the constituent system including (but not limited to) lowered miscibility-gap temperature,[5] incorrect predicted order structures,[8] and shifted density of states.[7] More recently, it has been suggested that these relaxations are in direct contradiction to the theory of CEs and that some supercell volume/concentration-dependency needs to be included in the effective cluster interactions (ECIs) of CEs or else the CE will fail.[4, 13] The idea behind concentration-dependent ECIs is that in, e.g., substitutional alloys the higher concentration of one component over another will cause the unit cell to shrink or expand to accommodate the extra atoms (depending on the relative covalent radii of the components). If the unit cell of an alloy shrinks or expands, the distance between nearest neighbors shrinks or expands accordingly and the same energetic contribution, or ECIs, should not be expected from clusters made up of these nearest neighbors across unit cells of varying size. Interestingly, volume-dependent ECIs were indeed used long before this in the early 1990s,[14, 15] but they appear to have fallen out of favor since then. Nonetheless, the consequences of geometric relaxations are still of active interest within the community of CE developers/users,[4, 16] but a method for directly quantifying their effect on the CEs themselves is still missing.

Here, we are specifically interested in the lattice gas (LG) model[17, 18] and how internal geometric relaxations of surface/adsorbate system calculations affect fitted LG CEs. In such calculations, the supercells used are typically not allowed to relax, meaning the volume is constant and concentration-dependent ECIs, as proposed by Sanchez,[4] will not capture the effect of relaxations in our LG CEs. However, internal relaxations of the supercell's constituent atoms are allowed. Due to the presence of a vacuum layer and often significant surface void space, these internal relaxations can be quite severe—perhaps more so than in substitutional alloy systems. Despite this, concentration-*in*dependent ECIs are also important to the interpretation of ECIs in the LG paradigm, and so we are inclined to assume they will remain a mainstay of LG CE development. In this case, our aim is to quantify the uncertainty in these ECIs due to relaxations. As the ECIs within LG CEs are typically fit to *ab initio* data, another important factor in the ECI uncertainty energy variations induced by the choice of exchange correlation function, so we take the opportunity to also assess the uncertainty of ECIs due to this variable.

In order to quantify the extent to which geometric relaxations (away from ideal lattice positions) and DFT functional affect the quality of CEs, our goal is to assess, quantitatively, the uncertainty in CE's ECIs. Since CEs are generally constructed via a linear regression procedure, it is tempting to appeal to the statistics of linear regression for this purpose since Analysis of Variance (ANOVA) and subsequent confidence intervals (CIs) are often used for precisely this purpose. Unfortunately, the legitimacy of these statistics is based on three basic assumptions[19] that *ab initio* data and CEs do not uphold: (1) normality—that the residuals are normally (or at least likewise) distributed with a mean value of zero, (2) homoscedasticity—that the model independent variables (the ECIs) are distributed equivalently (e.g. same variance) about their mean value, which is taken to be the true underlying value, and (3) independence—that the independent variables (again, the ECIs) are uncorrelated. Because the *ab initio* data we wish to use in fitting CEs have errors unrelated to the kind of random sampling error seen in experimental data, it is not difficult to see that the first assumption, normality, is very likely violated. More crucially however, the second and third assumptions, homoscedasticity and independence, are violated by the very nature of CEs themselves. This is because higher-body ECIs are expected to have sequentially lesser effect on the final energies or else truncating the

CE would not be possible (violation of assumption 2). Also, since higher-body clusters are built up from lower-body clusters (e.g. the removal of any of the 3 two-body clusters making up a three-body cluster will *always* result in the removal of that three-body cluster) the ECIs will absolutely be correlated (violation of assumption 3). Thus, we should not at all expect CIs (or other ANOVA-derived statistics) to be representative of the CE's true uncertainty.

Here, we will present a method for determining "ECI errors" that avoids statistical assumptions in their entirety. Instead we rely on a direct sampling procedure that takes advantage of our convergent implementation[20] of the leave-multiple-out cross-validation (LMO CV) score[21]. These ECI errors are then used to allow a direct comparison of four similar systems wherein atoms are either fixed or allowed to relax to their simultaneous energetic minimum; and wherein a standard generalized gradient approximation (GGA) functional and a van der Waals containing functional are used.

7.2. Theory and Formulation

7.2.1. Defining ECI errors

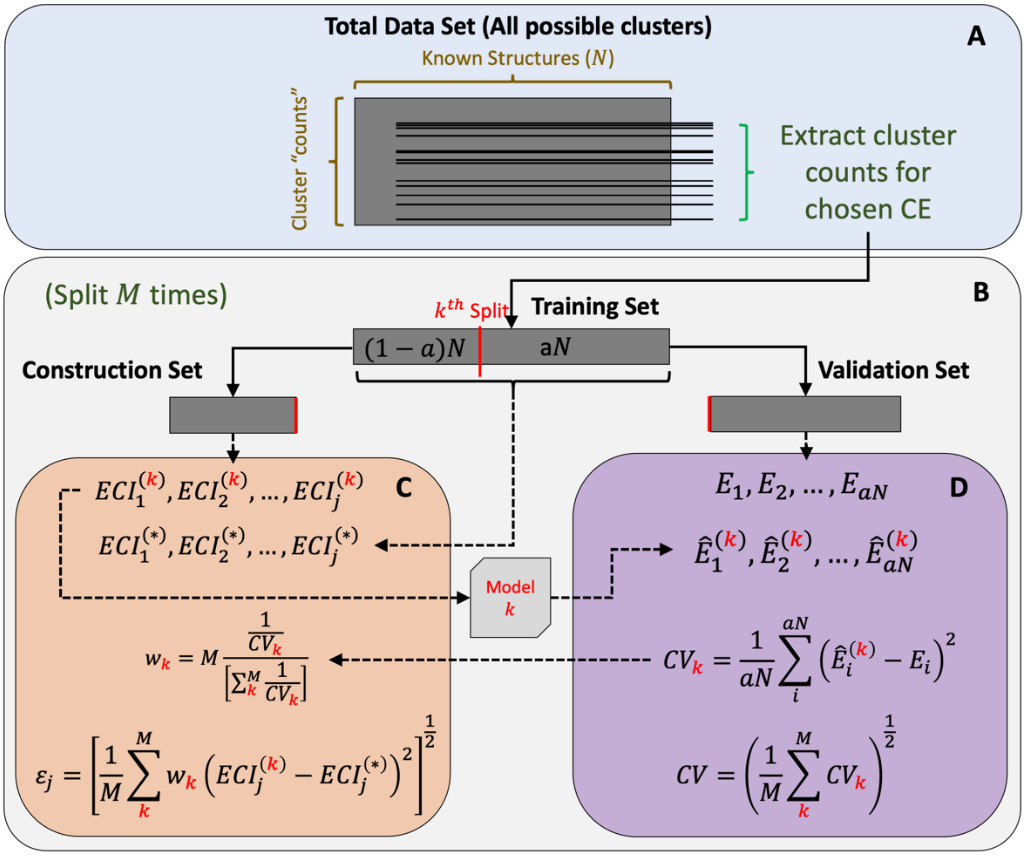

Total Data Set (All possible clusters)
Known Structures (N)
Cluster "counts"
Extract cluster counts for chosen CE
A
(Split M times)
k^{th} Split
Training Set
B
$(1-a)N$
aN
Construction Set
Validation Set
$ECI_1^{(k)}, ECI_2^{(k)}, \dots, ECI_j^{(k)}$
C
$ECI_1^{(*)}, ECI_2^{(*)}, \dots, ECI_j^{(*)}$
Model k
$w_k = M \frac{\frac{1}{CV_k}}{\left[\sum_k^M \frac{1}{CV_k}\right]}$
$\varepsilon_j = \left[\frac{1}{M}\sum_k^M w_k \left(ECI_j^{(k)} - ECI_j^{(*)}\right)^2\right]^{\frac{1}{2}}$
$E_1, E_2, \dots, E_{aN}$
D
$\hat{E}_1^{(k)}, \hat{E}_2^{(k)}, \dots, \hat{E}_{aN}^{(k)}$
$CV_k = \frac{1}{aN}\sum_i^{aN}\left(\hat{E}_i^{(k)} - E_i\right)^2$
$CV = \left(\frac{1}{M}\sum_k^M CV_k\right)^{\frac{1}{2}}$

Figure 7.1. Illustration of the pertinent information extracted during the calculation of the LMO CV score starting from (A) the complete set of data comprised of $\boldsymbol{N}$ structures where all specified possible cluster 'counts' are included. A subset of clusters corresponding to the chosen CE is extracted from all structures becoming the training set used for (B) evaluation of the LMO CV score. The data (known structures) are split into various (ideally) random subsets of size $(\mathbf{1}-\boldsymbol{a})\boldsymbol{N}$ (the construction set) and $\boldsymbol{aN}$ (the validation set) where $\boldsymbol{a}$ is the "fraction left out." The data needed to evaluate the ECI errors are sequestered and shown in (C), while the data needed for the evaluation of the LMO CV score is sequestered and shown in (D).

Our procedure for determining ECI errors is shown in Figure 7.1. Once a complete data set of N *ab initio* structural data (which contains a CE "fingerprint" for each structure along with their computed *ab initio* energies) has been collected (Figure 7.1A), the "training set" can be extracted, which is simply the complete data set with a subset of cluster "counts" selected to form a potential CE (thus implicitly setting all other possible cluster ECIs to zero). As shown in Figure 7.1B, this training set is then split into a "construction set" (CS) and a "validation set" (VS). The split is ideally made by selecting a random subset of aN structures (where a is the "fraction left out") to be the VS and then using the remaining $(1-a)N$ structures as the CS (we use $a = 0.6$, based on the recommendations of Baumann[21] and considerations described elsewhere[22, 23]). We denote each split and corresponding CS/VS subset of structures with index k and the ECIs are then fit to the CS data to provide a set of ECIs specific to that k^{th} subset of CS structures: $\left\{ECI_j^{(k)}\right\}$, where j demarks the j^{th} cluster in the chosen CE and superscript (k) denotes from which CS subset the ECI was fit. From this newly created CE model, the energies of the structures in the VS (which were not used in the previous fitting) are then predicted ($\hat{E}_i^{(k)}$ for the i^{th} structure; a "hat" denoting prediction) and compared against their actual computed energy values (E_i). A root mean squared prediction error is then assessed for that k^{th} split as

$$CV_k = \left[\frac{1}{aN}\sum_{i}^{aN}\left(\hat{E}_i^{(k)} - E_i\right)^2\right]^{\frac{1}{2}} \tag{7.1}$$

The LMO CV score is calculated as the root mean squared CV_k across some M random splits of the training set data (making it the root mean squared error of all predictions):

$$CV = \left(\frac{1}{M}\sum_{k}^{M} CV_k^2\right)^{\frac{1}{2}} \tag{7.2}$$

The number of cuts required (M) is chosen so that the LMO CV score converges to some tolerance (more on this value in section 7.3.5). The optimum CE for a data set is that which produces the lowest LMO CV score and the final model ECIs are assessed by fitting to the *entire* training set, $\left\{ECI_j^{(*)}\right\}$, where superscript $(*)$ denotes the final model "ultimate" ECI .

To directly assess the "ECI errors," the data shown in Figure 7.1C is collected alongside the data needed for the evaluation of the LMO CV score in Figure 7.1D as described above. As different training set splits are made, a different $ECI_j^{(k)}$ is produced, unique to the subset of structures in the CS. Since the CS has fewer structures than the total training set, we can expect these values to be "better" solutions for those structures and the deviation of these values from the final ECI ($ECI_j^{(*)}$), a good indication of the potential variance in that ECI. However, at the same time, if the model produced with $ECI_j^{(k)}$ gives a poor prediction error in the VS (i.e. a high CV_k or low $1/CV_k$) we lower the significance of that deviation (whether initially large or small). We do this by weighting each deviation with a value proportional to $1/CV_k$. These considerations result in the following equation:

$$\varepsilon_j = \left[\frac{1}{M}\sum_k^M w_k\left(ECI_j^{(k)} - ECI_j^{(*)}\right)^2\right]^{\frac{1}{2}} = \left[\frac{1}{M}\sum_k^M \left\{\sqrt{w_k}\left(ECI_j^{(k)} - ECI_j^{(*)}\right)\right\}^2\right]^{\frac{1}{2}} \qquad (7.3)$$

where ε_j is the "ECI error" and w_k is a weighting factor that accounts for the aforementioned poor prediction behavior:

$$w_k = M\frac{\frac{1}{CV_k^2}}{\left[\sum_k^M \frac{1}{CV_k^2}\right]} \qquad (7.4)$$

which uses the square of the LMO CV score so that the weight on the absolute deviation is in fact proportional to $1/CV_k$, and which is then normalized such that the sum of all w_k is the number of training sets cuts performed, M, and designed to give greater weight to $ECI_j^{(k)}$ deviations that produce better predictions in the VS. An unweighted version of Equation 7.3 would have w_k set to 1.

A difficulty arises in the evaluation of the LMO CV score that deserves mentioning here. Due to the random nature of the training set splits, some structures can become over (or under) represented in the various M number of VS's. This is illustrated in Figure 7.2A – 7.2C. The bias can be quantified by defining a "VS residence" vector, $\mathbf{v}$, with N components $\{\mathrm{v_i}\}$, one for each structure in the training set, initially starting at 0 (Figure 7.2A). The bias is then defined as

$$Bias = \max[\mathbf{v}] - \min[\mathbf{v}] \tag{7.5}$$

As structures are randomly assigned to the VS, their presence is marked by an increment in its VS residence (Figure 7.2B) and the build-up of bias becomes apparent after a subsequent split of the training set, where due to the random nature of the cuts, some structures are added twice while others are left out twice (Figure 7.2C). A CV score based on these two training set splits would be biased toward the structures overrepresented in the VS's. To fix this, the underrepresented structures are constantly identified and added to the next validation set, with any remainder being chosen at random. This is shown in Figure 7.2D. If this is done continuously, the degree of non-randomness is kept to a minimum and the bias can be kept to below 2, which at large enough M is for all practical purposes completely unbiased.

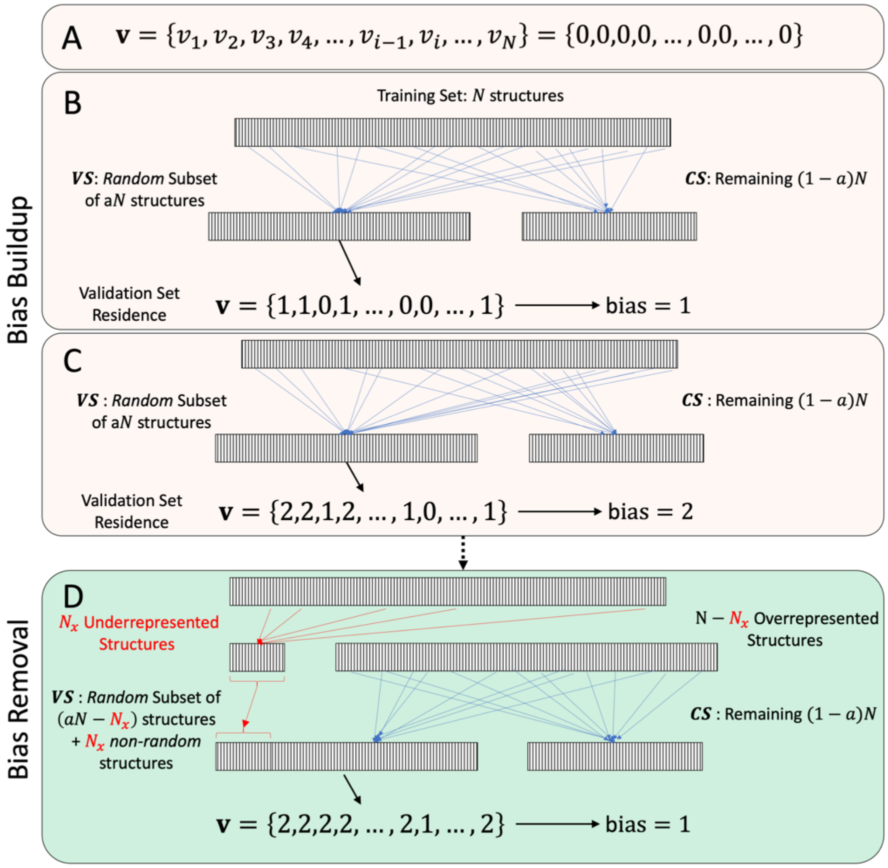

A
$\mathbf{v} = \{v_1, v_2, v_3, v_4, \dots, v_{i-1}, v_i, \dots, v_N\} = \{0,0,0,0, \dots, 0,0, \dots, 0\}$
Bias Buildup
B
Training Set: N structures
VS: *Random* Subset of aN structures
CS: Remaining $(1-a)N$
Validation Set Residence
$\mathbf{v} = \{1,1,0,1, \dots, 0,0, \dots, 1\}$ → bias = 1
C
VS : *Random* Subset of aN structures
CS : Remaining $(1-a)N$
Validation Set Residence
$\mathbf{v} = \{2,2,1,2, \dots, 1,0, \dots, 1\}$ → bias = 2
Bias Removal
D
N_x Underrepresented Structures
N − N_x Overrepresented Structures
VS : *Random* Subset of $(aN - N_x)$ structures + N_x *non-random* structures
CS : Remaining $(1-a)N$
$\mathbf{v} = \{2,2,2,2, \dots, 2,1, \dots, 2\}$ → bias = 1

Figure 7.2. Demonstration of bias build up (orange shaded regions) and bias removal (green shaded region). (A) The initialization of the "VS residence" vector defined in the main text before Equation 7.5. (B) The training set is split into a validation set (VS) and construction set (VS) choosing $\boldsymbol{aN}$ structures at random in order to form the VS. The VS residence vector indices corresponding to the $\boldsymbol{aN}$ structures chosen are incremented by 1. (C) The same process in (B) is repeated and the formation of a bias in the structures chosen to be in the VS becomes apparent. (D) To remove the bias, those $(\boldsymbol{N_x})$ structures that have been underrepresented are identified (red font and arrows) and added to the VS before the remaining $(\boldsymbol{aN} - \boldsymbol{N_x})$ VS structures are chosen at random to finish forming the VS. In so doing, the bias is reduced.

In order for all these procedures and formulae to be visualized, an LMO CV score calculation is shown in Figure 7.3 where the incremental CV score is allowed to converge *without* bias removal in order to demonstrate bias build-up and the resultant behavior in the evaluations of CV_k and w_k before and after bias removal is implemented (as shown in Figure 7.2). As can be seen in Figure 7.3A and 7.3B, the random sampling of the data is largely unaffected by the bias removal, and importantly, the incremental CV score starts converging to a completely different result after the bias is removed, showing in this case that the sampling was biased towards structures that were better predicted (since it was converging to a *lower* CV score) than other structures in the data set. The w_k values distribute about a value of 1 in a way consistent with the development of Equation 7.4. The point at which bias removal begins is apparent in Figure 7.3D, where a bias of nearly 75 had accumulated shortly before 1000 training set splits were made where it is evident that the bias removal algorithm shown in Figure 7.2 performs very quickly and efficiently. It is important to note that, in practice, the bias removal algorithm is always implemented from the start of the LMO CV score calculation with negligible effect on the final converged CV score.

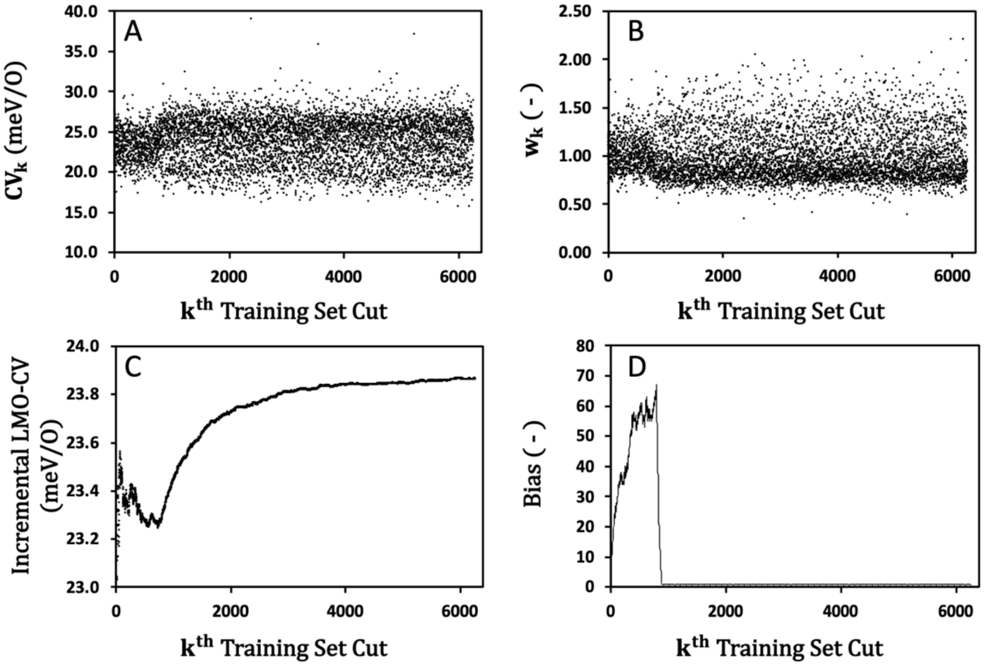


Figure 7.3. Visualization of various quantities calculated during the evaluation of the LMO CV score and corresponding ECI errors: (A) $\boldsymbol{CV_k}$ as defined in Equation 7.1 and Figure 7.1 for each $\boldsymbol{k^{th}}$ training set cut as also illustrated in Figure 7.1; (B) the weight, $\boldsymbol{w_k}$, applied to each squared ECI deviation of Equation 7.3; (C) the incremental LMO CV score as calculated up to the current $\boldsymbol{k^{th}}$ training set cut (i.e. the root mean squared value of the preceding data in (A)); and (D) the bias as defined in Equation 7.5.

7.2.2. Data Sets and Methodology

Our primary motivation for developing the procedure for determining ECI errors is to be able to quantify the uncertainty in CEs brought about by geometric relaxations and choice of exchange correlation functional. To do this (and also demonstrate the implementation of the procedure outlined in the previous section), we have developed a LG CE for four different systems: (1) the relaxed-O/Fe(100) RPBE system, using the raw data from Bray et al.,[24] (2) for the fixed-O/Fe(100) RPBE system using data produced for this work, (3) for the fixed-O/Fe(100) optB88-vdW system, using data again produced for this work, and (4) relaxed-H/Fe(100) system using the raw data from Henlsey et al.[25]. In the O/Fe(100) systems, O adatoms are confined to the 4-fold hollow sites of the Fe(100) surface (which is the most favorable site[24]), while in the H/Fe(100) system both 4-fold hollow and bridge sites are included due to their similar adsorption energies; a pseudo 3-fold hollow site was also included in the work of Hensley et al.[25] due to massive lattice relaxations during their optimizations (hence why this system is of interest here). In the "relaxed" systems, the top two layers of the Fe(100) model slab and adatoms were allowed to relax completely during geometric optimization, while in the fixed systems, the atoms of the Fe(100) slab and the O adatoms are kept fixed in their "ideal" positions (i.e. the clean surface geometry and the O in the direct center of the 4-fold hollow site at its ideal height for an isolated O adatom on that fixed surface). The data produced for this work, i.e. "fixed" O/Fe(100) systems, were generated in an automated manner using the Alloy Theoretic Automated Toolkit (ATAT)[26-28] while the lattice gas models were found by fitting to the adsorption energies of all data and optimizing using our recently developed *Ab initio* Mean-field Augmented Lattice

Gas Modeling (AMALGM) code[29]. Computational methodology for the *ab initio* calculations is given in Appendix D. The *ab initio* adsorption energies for all systems along with their LG CE predicted adsorption energies can be seen in Figure 7.4 with their final LMO CV scores and root means squared residuals (RMSRs). The corresponding clusters for these systems are shown in Figure 7.5. Note that while the clusters shown for the O/Fe(100) systems (Figure 7.5A – 7.5C) are exhaustive, only the clusters with ECI magnitudes greater than 10 meV are shown for the H/Fe(100) system (Figure 7.5D) because this system has more than double the number of clusters than the O/Fe(100) systems.

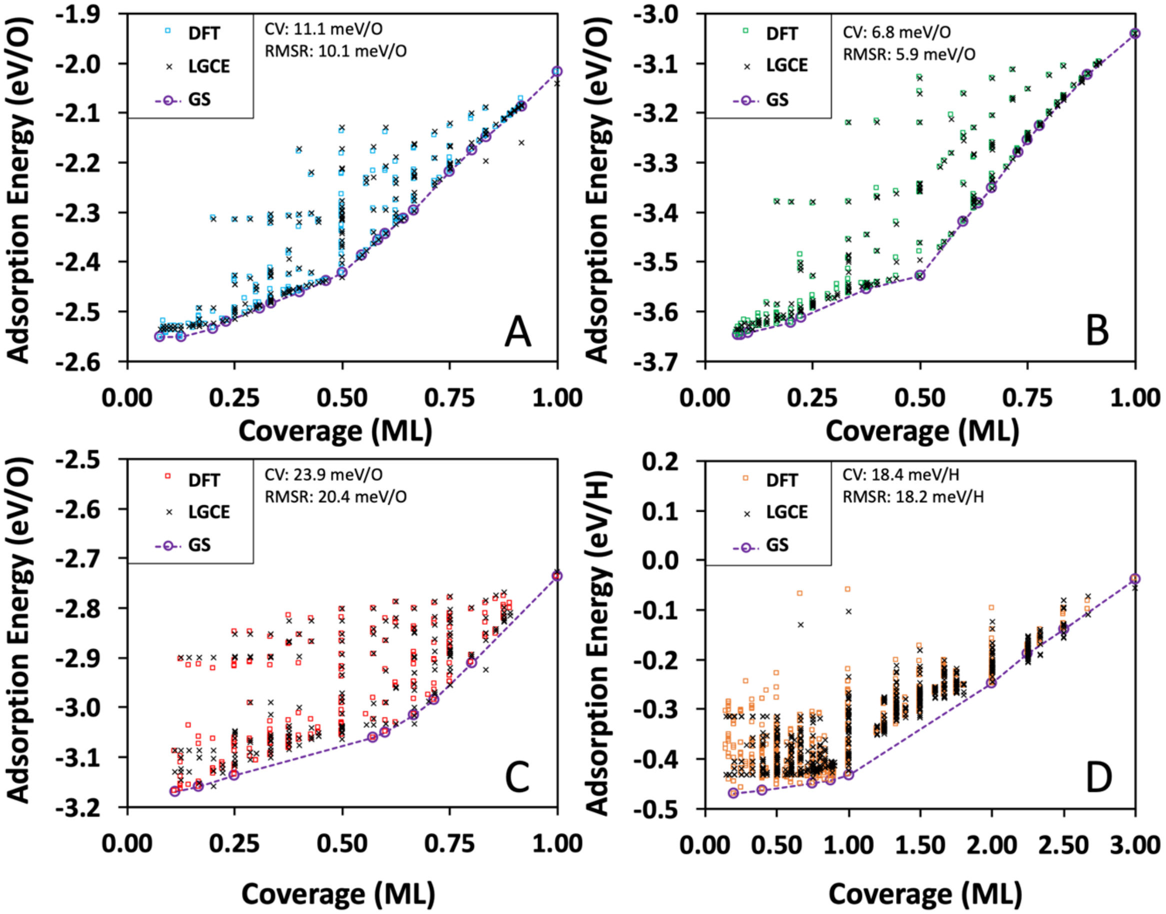


Figure 7.4. DFT-calculated and LG-predicted adsorption energies for (A-C) the O/Fe(100) system and (D) the H/Fe(100) system. In (A) and (B), the Fe lattice is fixed in its bulk position and the O is fixed in

the ideal position for an isolated O atom on this fixed lattice. In (C) and (D), both adsorbate and top two layers of the lattice are allowed to fully relax. The RPBE functional was used in (A) and (C), while the optB88-vdW functional was used in (B) and (D). The predicted ground states are denoted GS in the legends. Note that the x-axis in (D) ends at 3.0 ML instead of 1.0 ML due to the greater complexity and configurational space spanned in the H/Fe(100) system (see Ref. [25]).

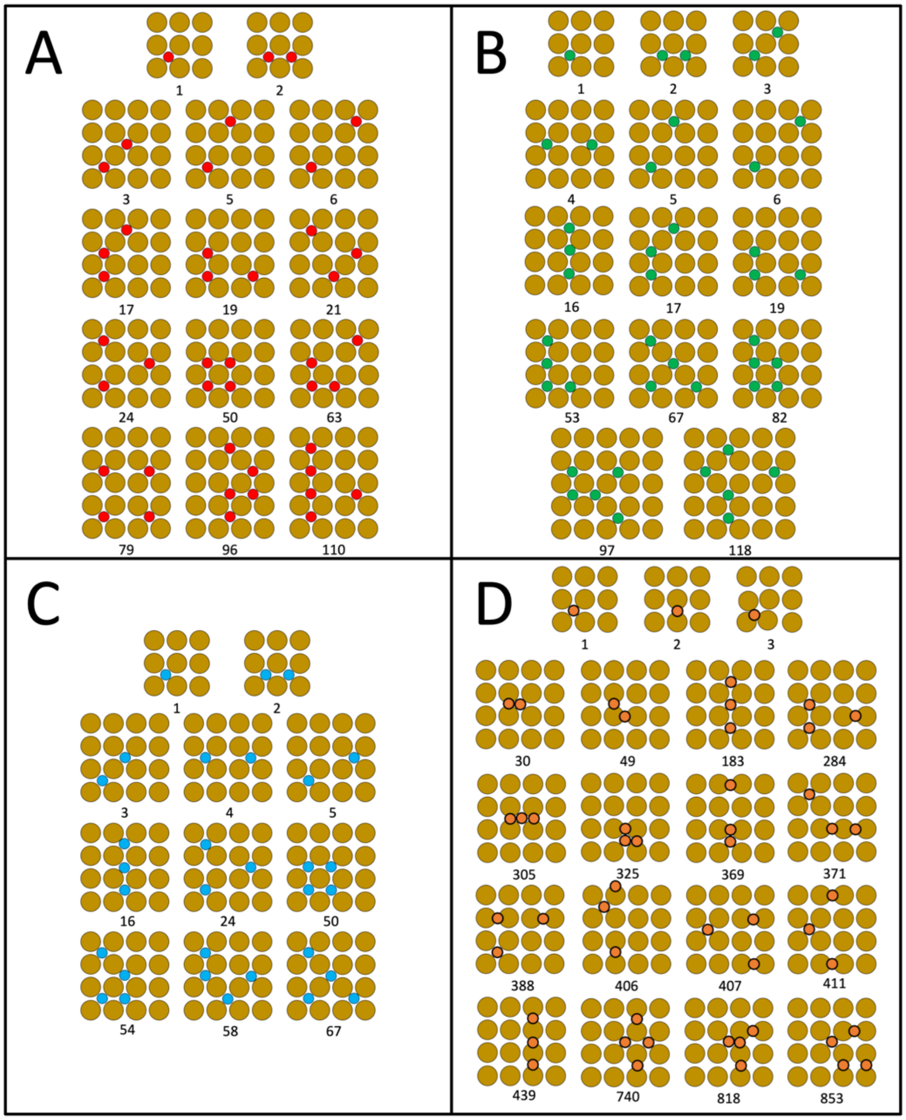


Figure 7.5. Clusters and their cluster IDs corresponding to the optimum LG CEs found for the systems shown in Figure 7.4: (A) relaxed-O/Fe(100) RPBE, (B) fixed-O/Fe(100) optB88-vdW, (C) fixed-O/Fe(100) RPBE, and (D) relaxed-H/Fe(100) optB88-vdW. The color of the adatoms corresponds to the color coding established in Figure 7.4, with red circles, green circles, and blue circles denoting oxygen atoms in their respective system; and orange circles denoting hydrogen atoms. The gold circles denote iron atoms.

7.3. Results and Discussion

7.3.1. Visualization of the ECI errors

Before analyzing and comparing the ECI errors found for the optimum LG CEs produced for the four systems shown in Figure 7.4, we wish to explicitly allow the reader to visualize the direct sampling being made in the evaluation of the ECI errors as defined in Equation 7.3 as the interpretation of ECI errors is contingent upon the behavior of the ECI deviations from which they are derived. Figure 7.6 shows the quantity in curly brackets in Equation 7.3 collected over all training set splits needed to converge the respective LMO CV scores for the relaxed-O/Fe(100) RPBE (Figure 7.6A – 7.6B) and fixed-O/Fe(100) RPBE systems (Figure 7.6C – 7.6D) using both the optimum LG CEs for the relaxed system (Figure 7.6A and 7.6C) and the fixed system (Figure 7.6B and 7.6D). It is immediately apparent when comparing the relaxed-O/Fe(100) to the fixed-O/Fe(100) system that the ECI deviations in the relaxed system have an overall greater variation (~175% greater) than in the fixed system regardless of what CE is used, suggesting greater uncertainty in the relaxed system ECIs that is fundamentally unavoidable. This is apparent because, while the ECI deviations are greater in both systems when using their non-optimum LG CEs, the non-optimal CE for the fixed system (Figure 7.6C) still displays less variation (relaxed system displays ~70% greater overall variance) in its ECI deviations than the optimum CE for the relaxed system (Figure 7.6A).

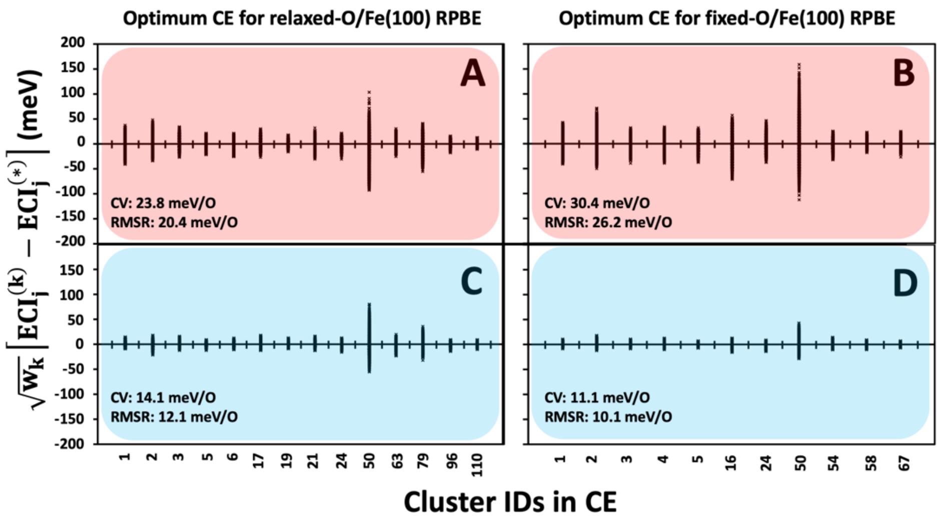


Figure 7.6. Illustration of how the weighted ECI deviations ultimately distribute about zero for each respective cluster in the displayed CE. Panels (A – B) show these results for the relaxed-O/Fe(100) RPBE system while panels (C – D) show results for the fixed-O/Fe(100) RPBE system. To compare the two on a one-to-one basis, the optimized CE for the relaxed-O/Fe(100) RPBE system is used in A and C while the optimum CE for the fixed-O/Fe(100) RPBE system is used in B and D. The LMO CV scores and RMSRs for these systems are shown inset for context.

It is important to comment, for the interpretation of the ECI errors we wish to report, that the values in Figure 7.6 distribute nearly symmetrically about 0 meV for all of the clusters. This suggests that their root mean squared value (i.e. ε_j from Equation 7.3) can indeed be used as a $\pm$ "standard deviation" of the distribution for the final ECIs. A "two sigma" value (i.e. two times the root mean squared value) accurately captures the height of these data and we will therefore use $\pm 2\varepsilon_j$ as the reported ECI error and to determine ECI significance. For comparison, the unweighted version of these distributions of ECI deviations are shown in Figure D2. We should note that, in this work, the ECI error values derived from the unweighted version is negligibly different (<1 meV) from the weighted version.

7.3.2. Quantifying Uncertainty Due to Relaxations and DFT Functional

We first compare ECIs and their ECI errors for the O/Fe(100) systems to illustrate the *quantitative* errors induced in CEs by geometric relaxations and choice of exchange correlation functional in the underlying *ab initio* data. Figure 7.7 shows the ECIs (colored columns) and their associated "two sigma" ECI errors (black error bars) for these systems. The relaxed-O/Fe(100) RPBE system is compared directly to the fixed-O/Fe(100) RPBE system using their respective optimum CEs in Figures 7.7A and 7.7B; the ECI errors for these systems are derived from the data shown in Figure 7.6. Based on the LMO CV scores for these two systems when using the optimum LG CE for the relaxed-O/Fe(100) RPBE system (Figure 7.7A), the fixed system (in blue) is actually quite amenable to using a non-optimal CE (its LMO CV score increases to 14.1 meV/O from its optimum value of 11.1 meV/O—only a 27% increase) and is still more predictive than the optimum value for the relaxed system (23.8 meV/O). We are therefore very confident in the direct comparison between these two systems. The relaxed system LMO CV score increases by a similar amount (28%) when using the optimum LG CE for the fixed-O/Fe(100) RPBE system (Figure 7.7B), but the disparity between the two CV scores is great enough that we are less confident in their direct comparison. We therefore turn to Figure 7.7A for our analysis first.

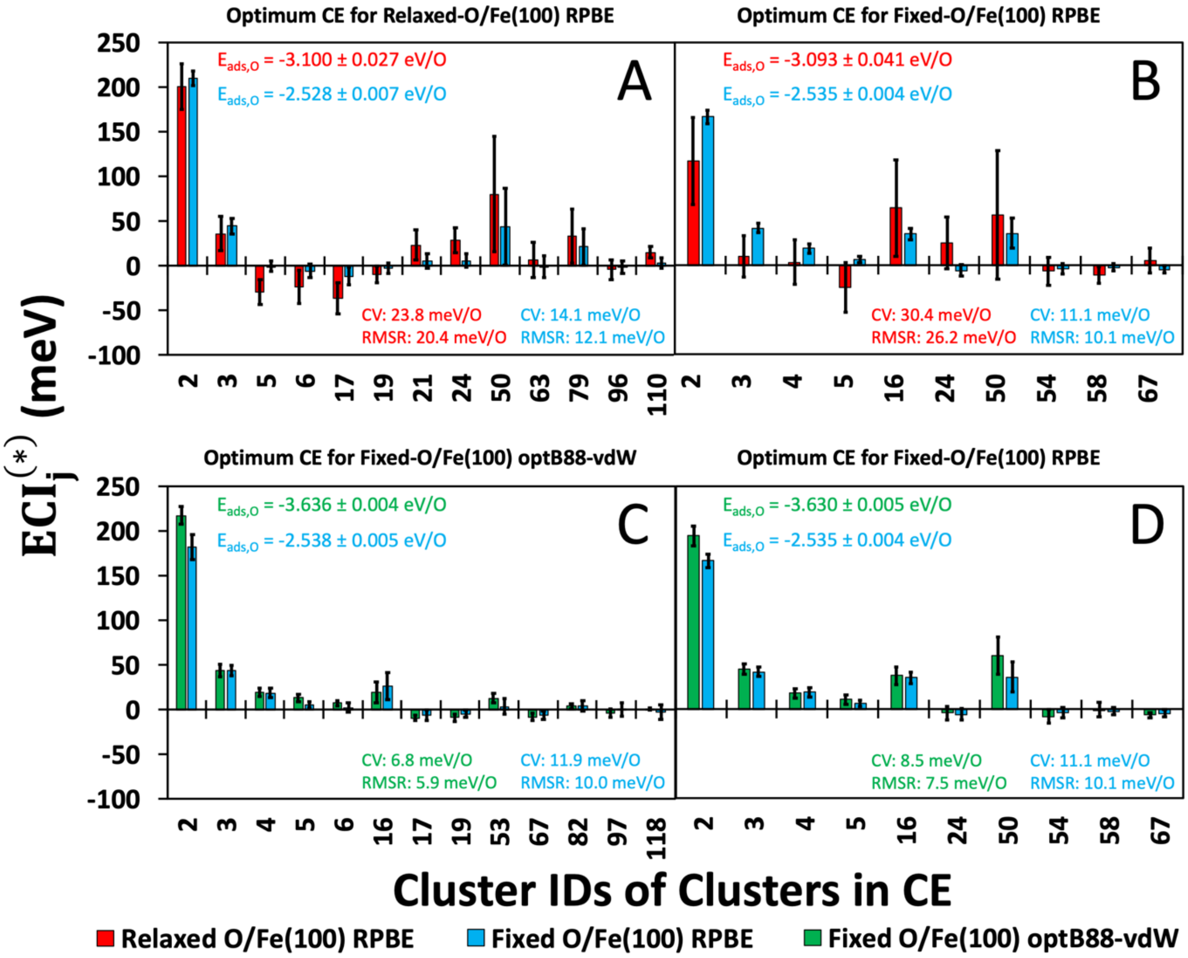


Figure 7.7. Comparison of the (A – B) relaxed-O/Fe(100) RPBE (red bars and text) vs. fixed-O/Fe(100) RPBE (blue bars and text) systems; and comparison of the (C – D) fixed-O/Fe(100) optB88-vdW (green bars and text) vs. fixed-O/Fe(100) RPBE (blue bars and text) systems. The optimum CE for each data set is used to allow direct comparison of ECIs and ECI errors. The ECI errors are shown as black error bars. Due to magnitude discrepancy, the isolated O/Fe(100) adsorption energy, $E_{ads,O}$ (corresponding to cluster ID #1), is shown in the inset in each panel. The LMO CV score and RMSR are also shown as insets with the appropriate color coding established in Figure 7.4.

An interesting feature of the ECIs in Figure 7.7A themselves is that their character (i.e. repulsive vs. attractive) is essentially the same between the fixed vs. relaxed systems. Only cluster #63 shows contrary character in the two systems, but only barely (-1 meV vs +7 meV in the fixed and relaxed systems respectively). This suggests that relaxations do not necessarily serve to reverse the most basic physics of the underlying system. However, this is not necessarily true if the ECI errors are taken into account, which indicates that for many structures in the data, ECI values could in fact be repulsive, attractive or simply 0 meV. It is for this reason that we must deem ECI values with magnitudes less that their ECI errors to be insignificant, and in Figure 7.7A, this is true of clusters 5, 6, 19, 21, 24, 63, 96, and 110 for the fixed system. For the fixed system in Figure 7.7A, such an effect is mostly due to the fact that the ECI magnitudes are small to begin with (recall that this is a non-optimal LG CE for this system). In the relaxed system, only clusters 63 and 96 are insignificant and again primarily because their ECI magnitudes are small. However, comparing between the two systems we can see that for all clusters except 24 and 110, the fixed and relaxed systems' ECI error bars overlap. The relaxed system ECI errors are on average 10 meV greater than then those in the fixed system suggesting that the overlap can be primarily blamed on the increased uncertainty in the relaxed system ECIs. This suggests that there are structures within the relaxed data set that could be better predicted using the fixed system ECIs.

It is important to clarify, however, that this apparent equivalence does not mean that the final fixed system LG CE can be used in place of the relaxed system LG CE. On the one hand, there is no way to know *a priori* what this "essentially same" LG CE for the fixed system is as it is not its optimum CE. On the other hand, it was a *set* of ECIs that provided better agreement to the energies, not just a single perturbation to a single ECI value—meaning, as we've stated previously, that the errors are *correlated*—so we cannot expect to get better fits from just any set of ECIs that fall within the error bars. Nonetheless, this does mean that we are highly *uncertain* that the final ECIs in the relaxed system will accurately predict the energetic consequences in the relaxed structure data that are outside the training set much better than if those data were based on fixed structures. While this does not mean that relaxations cannot be captured by CEs, it does mean that it is incredibly important to be able to quantify the uncertainties inherent in their inclusion, and our method provides exactly that.

The results shown in Figure 7.7C and 7.7D, comparing the fixed-O/Fe(100) RPBE system (blue bars) to the fixed-O/Fe(100) optB88-vdW system (green bars), demonstrate the consequences brought about by a change in the DFT functional. Here, the LMO CV scores are both small and comparable allowing for the direct comparison of both CEs used. Strikingly, the error bars again overlap for virtually every cluster in either Figure 7.7C or 7.7D; only the adsorption energy and 1st nearest neighbor (NN) (Cluster #2) are shown to be significantly different. However, this time, there is only a 2 meV discrepancy in average ECI errors, so this is due to the ECIs themselves being practically equivalent for all clusters (again except for the adsorption energy and 1st NN). Thus, in this case, we have shown that the primary consequence of using a different DFT functional is in the adsorption energy and 1st NN, but in practically no other terms. Even discounting the error bars, there is only one cluster (#50 in Figures 7.7D) that is large enough and different enough to be consequentially different (a 24 meV difference). This corresponds to a fairly standard "usual suspect": the most compressed square 4-body interaction (Cluster #50 in Figure 7.5A). This means that, in principle, one need not actually collect an entire dataset of structures using a new functional, but instead simply calculate a handful of interactions corresponding to the "usual suspects" and replace these in an already optimized CE. This of course is only potentially true in these fixed systems, but given the uncertainty shown in the relaxed system, this result may well be universal.

7.3.3. Uncertainty in the H/Fe(100) system

Here, we expand our consideration of ECI error quantification of relaxations in a more complex system, namely H/Fe(100). In this system, both 4-fold hollow and bridge sites were included, but due to massive geometric relaxations of the underlying Fe(100) lattice that are a function of hydrogen coverage, a pseudo 3-fold hollow had to be added to the site definitions in the LG CE.[25] This added complexity along with the underlying severity of relaxation makes for an ideal test case for the analyses already laid out in Section 7.3.2.

As can be seen in Figure 7.8, the ECI errors assessed for the H/Fe(100) system are significant. This leads to high uncertainty in the resulting ECIs. Despite this, none of the insignificant ECIs are greater than 10 meV, meaning that the insignificant ECIs would likely have little effect on the chemical behavior for the LG CE. Thus, despite the massive geometric relaxations present in this system, the LG CE has captured the most important physical interactions even with a quantitatively larger degree of uncertainty in their exact values than those for the O/Fe(100) systems. This is also reflected in the reasonably low LMO CV score of 18.4 meV/H (Hensley et al. [30] fit their LG CE to the data's surface energies, providing a value of 11.5 meV/unit cell, which is difficult to compare to our values fitted to adsorption energies due to different normalization). It is also worth noting that the majority of the most significant ECIs (e.g. Clusters 30, 49, 284, 305, and 325) involve both 4-fold hollow and bridge sites showing that the increased complexity of this system is being captured. Nonetheless, these interactions have non-negligible ECI errors and thus high uncertainty, making it difficult to determine if one source of uncertainty is not the increased complexity itself. Based on the results from section 7.3.2, however, we can definitely say that a very major component contributing to these uncertainties is geometric relaxation. Also, it is clear from these results that even weakly bound adatoms like hydrogen can produce quantitatively high ECI uncertainty.

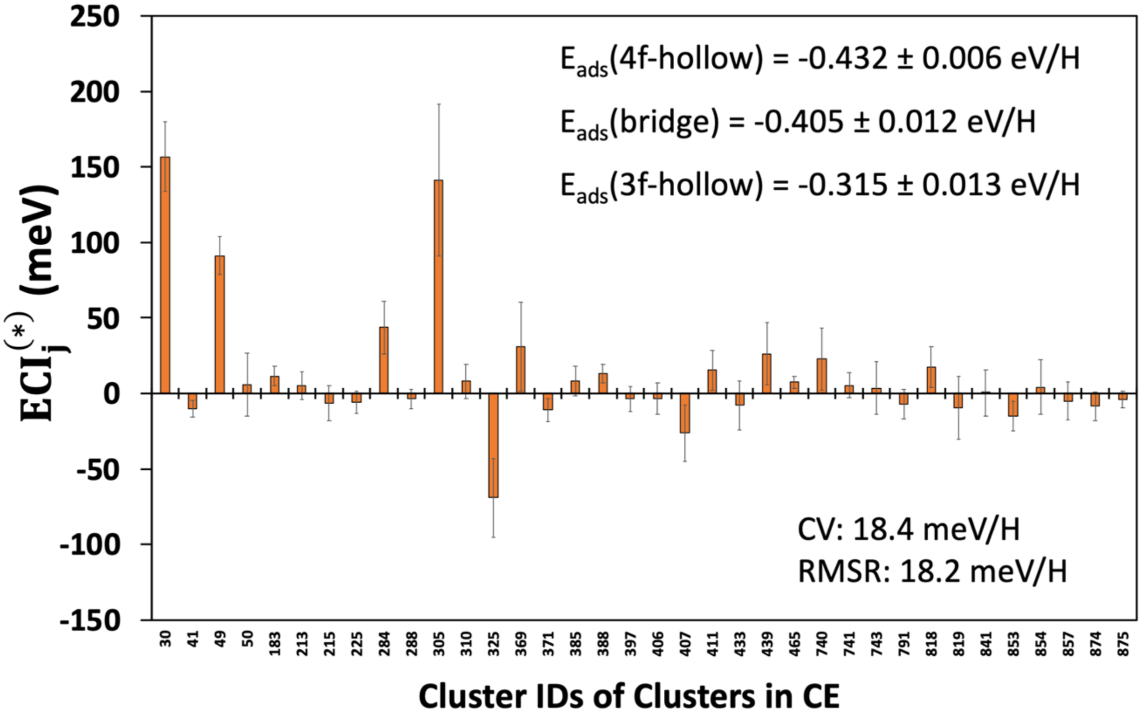


Figure 7.8. ECIs with their corresponding ECI errors for the optimized CE of the H/Fe(100) optB88-vdW system. Due to magnitude discrepancy, the isolated H/Fe(100) adsorption energies, E_{ads}, (corresponding to clusters #1, #2, and #3) are given in the inset.

3.4. ECI errors vs. Confidence Intervals

To explicitly illustrate the difference between ECI errors and the more customary Student t-test based 95% CIs, we present both side by side for the relaxed-O/Fe(100) RPBE and fixed-O/Fe(100) RPBE systems in Figure 7.9. The 95% CI are constructed from Equation 7.6:

$$95\%\ CI_j = t^{-1}(N - x; 0.95)\frac{RMSR}{\sigma_j} \qquad (7.6)$$

where $t^{-1}(N - x; 0.95)$ is the inverse cumulative Student-t distribution with $N - x$ degrees of freedom and a bound of 0.95, N is the number of structures in the training set, x is the number of clusters specified in the CE, and σ_j is the square root of the variance, σ_j^2, in cluster j which is found from

$$\sigma_j^2 = [(A^T A)^{-1}]_{jj} \qquad (7)$$

where A is the $(N \times x)$ design matrix used in regression, $(A^T A)^{-1}$ is the variance-covariance matrix[31], and $[(A^T A)^{-1}]_{jj}$ is the j^{th} diagonal of the variance-covariance matrix.

Figures 7.9A – 7.9D show immediately that the ECI errors are not equivalent to their cluster's corresponding 95% CIs and that not even trends are the same between the two. For example, the 95% CIs for the fixed system (Figure 7.9A – 7.9B) are indeed smaller than for the relaxed system (Figure 7.9C – 7.9D), which is qualitatively similar to the ECI errors, but this is most likely due to the better fit to the data set (i.e. lower RMSR) as will be explained shortly. While the 95% CI appear to capture this basic property, it is important to remember that the form of Equation 7.6 is based on the (violated) assumptions presented in section 7.2.1. Thus, we should still not expect these values to be accurate representations of the uncertainty in these systems. This leaves only the question of whether or not our method for determining ECI errors are simply capturing the fixed system's better fit to the CEs as this is the largest factor controlling the CIs.

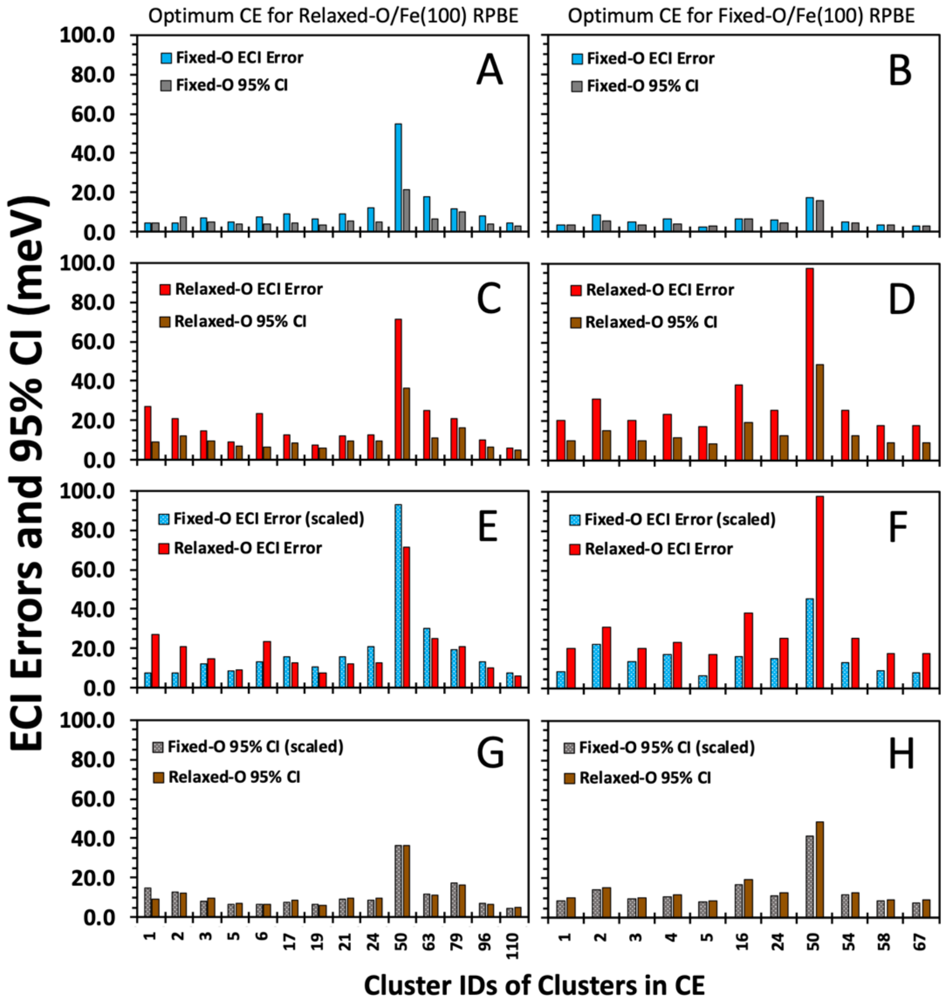


Figure 7.9. Comparison of ECI errors against 95% CIs for the fixed and relaxed O/Fe(100) RPBE systems using the optimized CE for the relaxed system (left column) and the optimized CE for the fixed system (right column).

As shown in Equation 7.6, there are three factors that control the value of the CI: the inverse cumulative student-t distribution, the data variance, and the RMSR. With a large enough degree of freedom, $t^{-1}(N - x; 0.95)$ approaches the inverse cumulative normal distribution which produces the same output regardless of data set size, and our data set sizes are large enough that this is the case here. The variance-covariance matrix from which the variance is derived is a function of the structural fingerprints alone, meaning that regardless of the energies of the structures in the dataset, any two data sets with the same structures (say one where the atoms are allowed to relax and one where they are fixed) will output the same variance. The O/Fe(100) data sets here are not exact replicas of each other, but the number of dissimilar structures is quite low. The RMSR, it would therefore seem, is the main controller of the 95% CIs here. We can then ask if this dependence on the RMSRs is also true of our ECI errors or if they in fact capture something more.

To answer this question, we scale up the fixed system ECI errors such that its RMSR matches that of the relaxed system and present the results in Figure 7.9E – 7.9F. This is also done to the 95% CIs and presented in Figure 7.9G – 7.9H to show the degree of control of the RMSR. As expected, the fixed system 95% CI is quite nearly a perfect match to the relaxed system 95% CI after being scaled up, especially when the relaxed system optimum LG CE is used (Figure 7.9G). There is clearly more influence from the variance when the optimum LG CE for the fixed system is used (Figure 7.9H), but the match is still quite significant. This is not true at all for the ECI errors in Figure 7.9E – 7.9F. In fact, even despite the RMSR scaling, the nearest neighbor ECI errors (clusters 2 – 6) for the fixed system are still significantly smaller than for the relaxed system, showing that the ECI errors capture the fundamental ideality in the fixed system that is being lost in the relaxed system. The scaled ECI errors for the higher body clusters (clusters 17 – 110) in the fixed system (Figure 7.9E) are generally significantly larger than the ECI errors in the relaxed system when the fixed system is fit to a non-optimal CE (i.e. the optimum CE for the relaxed system). This suggests that the better fit to the data (i.e. the lower RMSR) of the fixed system is what is primarily being reflected in the overall lower ECI errors in the higher body clusters seen in the fixed system CEs when a non-optimal CE is used. However, when the optimum CE for the fixed system is used, *every* scaled ECI error is smaller than for the relaxed system. This suggests that there is some connection between the minimizing of the LMO CV score (leading to the optimum CE) and what property of the system is being primarily captured by the ECI errors. This is perhaps unsurprising since the ECI errors are found from the CS's used in the determination of the LMO CV score.

7.3.5 Practical Considerations: Effect of Convergence Criteria

In the preceding sections, we have presented data taken from the numerous calculations of LMO CV scores (and associated ECI errors) using an excessively stringent convergence criteria ("CV tolerance") of 10^{-8} eV to avoid any potential issues of convergence. However, this is a burdensome tolerance criterion resulting in computationally onerous time requirements and choosing a lower tolerance would increase the practicality of our method. A lower tolerance in effect lowers the number of training set splits needed to converge the LMO CV score ("M" in Figure 7.1) and thus the amount of time required to arrive at a suitable value. Furthermore, we would expect that the stochastic nature of the training set splits will result in some uncertainty in the final overall value of the CV score and ECI errors but that increased splits will reduce this uncertainty. This deserves explicit testing. To do this, we specify a simple, non-optimal CE containing clusters 1 through 10 for the relaxed-H/Fe(100) optB88-vdW system and then calculate the LMO CV score and ECI errors 50 times for each of a range of CV tolerances. The results are shown in Figures 7.10 and 7.11.

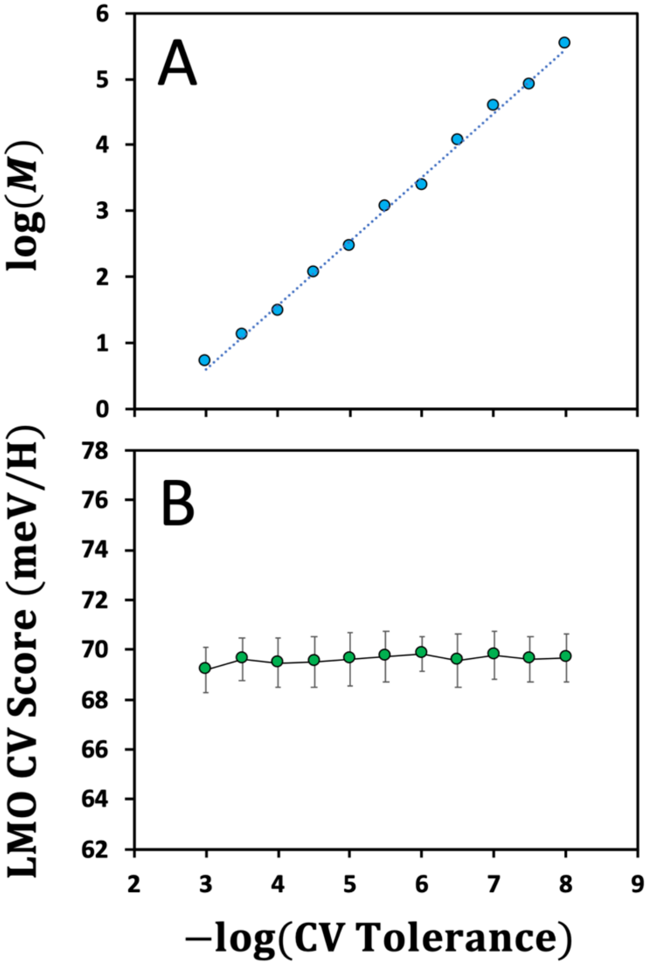


Figure 7.10. (A) The average number of training set cuts required to converge the LMO CV score of a set CE for the relaxed-H/Fe(100) optB88-vdW system to the tolerance specified on the x-axis (units in eV) over 50 separate runs of the algorithm presented in Figure 7.1, and (B) the corresponding average LMO CV Score with two standard deviations of the 50 runs shown as error bars.

Figure 7.10A shows the relationship between the CV tolerance (in eV) and the average number of splits required to reach that tolerance, M, and Figure 7.10B shows the corresponding average LMO CV score with error bars indicating two standard deviations of the 50 LMO CV scores calculated at each CV tolerance. There is a very linear positive dependence between the logarithm of the (average) number of splits and the negative logarithm of the CV tolerance selected demonstrating that a magnitude decrease in the CV tolerance results in a magnitude increase in the number of splits (and thus time) required. Remarkably, the calculated LMO CV scores are largely stable across all convergence criteria and the uncertainty (error bars) remains effectively the same with no appreciable reduction as the CV tolerance is decreased from 10^{-3} eV to 10^{-8} eV (left to right in both Figures 7.10 and 7.11). This LMO CV score uncertainty is approximately ±1 meV/H, which means our method of calculating the LMO CV score is sufficiently reliable even at the least stringent convergence criteria for this system. However, this is not true for the ECI errors.

In Figure 7.11, the ECI errors for each cluster in the CE used, clusters 1 – 10, as a function of the set CV tolerance for LMO CV score convergence. The ECI error values are not as stable across the range of CV tolerances as was the case for the LMO CV score, and as the CV tolerance is decreased (made more stringent), the maximum uncertainty in their values decreases significantly, appearing to converge to less than ±1 meV by a CV tolerance of $10^{-5.5}$ eV and less than ±0.5 meV by 10^{-7} meV. For the purpose of determining an ECIs significance, a ±1 meV uncertainty is more than sufficient, and since the certainty in the LMO CV score is ±1 meV/H regardless of the CV tolerance chosen, we would regard a CV tolerance of $10^{-5.5}$ eV a reasonable default value—assuming that the H/Fe(100) system is a representative, worst case scenario for ECI errors. Importantly, at a CV tolerance of $10^{-5.5}$ eV, only ~1000 training set splits are typically required, and in our lab, this takes less than a couple of seconds to achieve. While this is encouraging, this type of convergence analysis should be performed as a general rule for the individual system of interest.

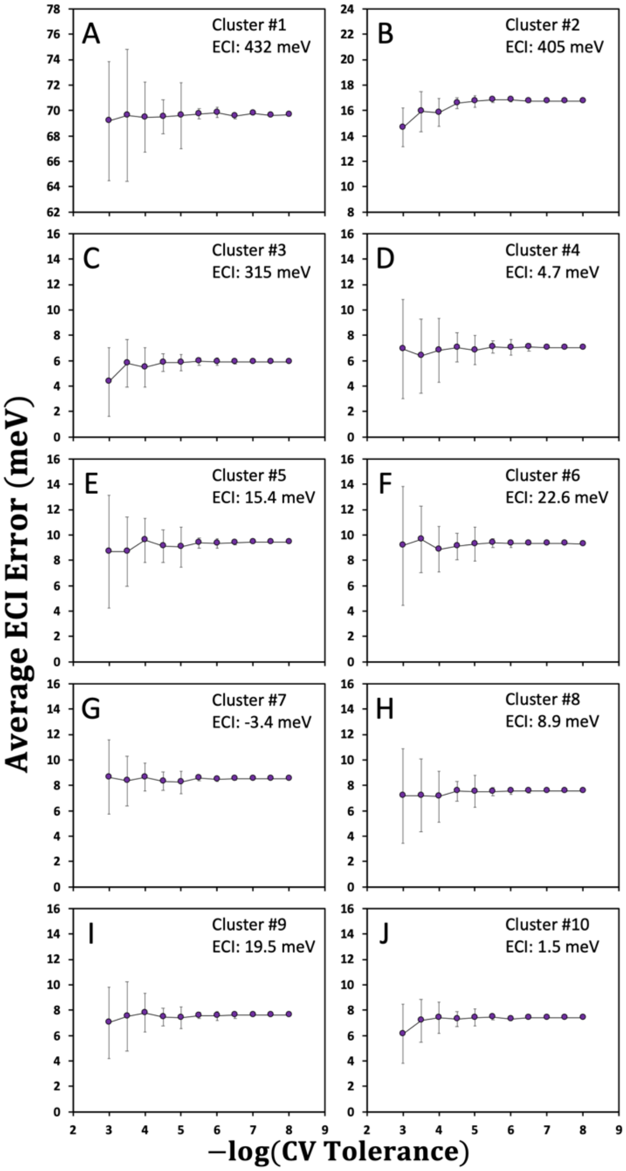


Figure 7.11. Average ECI errors for clusters within a set CE for the relaxed-H/Fe(100) optB88-vdW system found over 50 separate calculations of the LMO CV score at the specified tolerance. Two standard deviations of the 50 runs are shown as error bars demonstrating at which tolerance the ECI errors converge. The corresponding ECI values for these clusters are given in the inset. Note that while the bounds change, the y-axes in panels A – J are the same size (16 meV).

7.4. Conclusions

We have demonstrated here a method for quantifying the uncertainty in CE ECIs free from major statistical assumptions. We provide the expression and process needed to calculate these “ECI errors” and have demonstrated their use in specifically quantifying the effect of geometric relaxations and functional used in the underlying structural data. As has been described previously, geometric relaxations do indeed cause greater degrees of uncertainty in the quality of the ECIs, and our method shows quantitively that this corresponds to a general doubling of the ECI errors in CEs built from data where relaxations were allowed as compared to a model where atoms were fixed in ideal lattice positions. This result does not mean that the energetic consequences of relaxations cannot be captured in CEs but does point to the need for being able to quantify the degree of uncertainty that results from them. ECI errors were also able to reveal that the effect of DFT functional is primarily to change the adsorption energy and 1st nearest neighbor ECIs, potentially precluding the need to calculate whole new data sets to build CEs for systems when only the functional is changed. This would allow for the quick screening of functionals when attempting to build CEs.

ECI errors are calculated as part of our recently implemented convergent version of the LMO CV score, and the worst case uncertainty in the LMO CV score due to the stochastic nature of the method was shown to be ±1 meV/adsorbate for the H/Fe(100) system while the uncertainty in the ECI errors can be brought to below ±1 meV using suitable convergence criteria. We correspondingly advise that these convergence criteria be tested for any new system of interest. While a LMO CV score uncertainty of ±1 meV/adsorbate is very sufficient for determining the predictiveness of the vast majority of CEs, it should be noted that we have found that a nonnegligible subset of CEs have LMO CV scores well within 1 meV/adsorbate of the optimum LMO CV score found when using AMALGM. ECI errors present an opportunity to fully differentiate this subset of structures by filtering out CEs with insignificant clusters or otherwise give weight to those with the smallest ECI errors. We are actively considering this possibility in our own work.

Overall, the ability to calculate ECI errors as shown here permits researchers to go beyond qualitative descriptions of CE suitability. With this tool, as was done here, direct comparisons between systems can be performed and conclusions made about their CE's reliability. We posit that such analyses will guide development of CEs with minimal uncertainty and maximal transparency.

Acknowledgements

Work at WSU was primarily funded by the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Chemical Sciences, Biosciences and Geosciences (DE-SC0014560), and by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1347973. G.C., Seattle Chapter ARCS Fellow and NSF Graduate Research Fellow, gratefully acknowledges financial support from the Achievement Rewards for College Scientists foundation as well as the National Science Foundation Graduate Research Fellowship Program. This research also used resources from the Center for Institutional Research Computing at Washington State University. PNNL is a multi-program national laboratory operated for the U.S. DOE by Battelle.

REFERENCES

[1] J. M. Sanchez, F. Ducastelle, D. Gratias, Physica A: Statistical Mechanics and its Applications 128 (1984) 334-350.

[2] J. M. Sanchez, Phys. Rev. B 48 (1993) 14013-14015.

[3] J. M. Sanchez, Phys. Rev. B 81 (2010) 224202.

[4] J. M. Sanchez, Journal of Phase Equilibria and Diffusion 38 (2017) 238-251.

[5] S. H. Wei, L. G. Ferreira, A. Zunger, Phys. Rev. B 41 (1990) 8240-8269.

[6] S. de Gironcoli, P. Giannozzi, S. Baroni, Phys. Rev. Lett. 66 (1991) 2116-2119.

[7] Z. W. Lu, S. H. Wei, A. Zunger, S. Frota-Pessoa, L. G. Ferreira, Phys. Rev. B 44 (1991) 512-544.

[8] Z. W. Lu, S. H. Wei, A. Zunger, Phys. Rev. Lett. 68 (1992) 1961-1961.

[9] D. B. Laks, L. G. Ferreira, S. Froyen, A. Zunger, Phys. Rev. B 46 (1992) 12587-12605.

[10] M. Asta, R. McCormack, D. de Fontaine, Phys. Rev. B 48 (1993) 748-766.

[11] A. Silverman, A. Zunger, R. Kalish, J. Adler, J. Phys.: Condens. Matter 7 (1995) 1167.

[12] A. G. Khachaturyan, *Theory of structural transformations in solids*. 2013: Courier Corporation.

[13] J. M. Sanchez, Phys. Rev. B 95 (2017) 216202.

[14] C. Wolverton, D. de Fontaine, H. Dreyssé, Phys. Rev. B 48 (1993) 5766-5778.

[15] V. Ozoliņš, J. Häglund, Phys. Rev. B 48 (1993) 5069-5076.

[16] A. H. Nguyen, C. W. Rosenbrock, C. S. Reese, G. L. W. Hart, Phys. Rev. B 96 (2017) 014107.

[17] K. Binder, D. P. Landau, Surf. Sci. 108 (1981) 503-525.

[18] K. Binder, Rep. Prog. Phys. 50 (1987) 783.

[19] A. J. Dobson, A. Barnett, *An introduction to generalized linear models*. 2008: Chapman and Hall/CRC.

[20] G. Collinge, C. Stampfl, J.-S. McEwen, Journal of Chemical Theory and Computation (2018) (Submitted).

[21] K. Baumann, Trends Anal. Chem. 22 (2003) 395-406.

[22] J. Shao, Journal of the American Statistical Association 88 (1993) 486-494.

[23] P. Zhang, Ann. Statist. 21 (1993) 299-313.

[24] J. Bray, G. Collinge, C. Stampfl, Y. Wang, J.-S. McEwen, Top. Catal. 61 (2018) 763-775.

[25] A. J. R. Hensley, G. Collinge, C. Stampfl, J.-S. McEwen, J. Phys. Chem. C (2018) (in Preparation).

[26] A. van de Walle, M. Asta, G. Ceder, Calphad 26 (2002) 539-553.

[27] A. van de Walle, G. Ceder, Journal of Phase Equilibria 23 (2002) 348.

[28] A. van de Walle, Calphad 33 (2009) 266-278.

[29] G. Collinge, K. Groden, C. Stampfl, J.-S. McEwen, Journal of Chemical Theory and Computation (2018) (Submitted).

[30] A. J. R. Hensley, G. Collinge, C. Stampfl, J.-S. McEwen, J. Phys. Chem. C (2018) (in Preparation).

[31] H. Toutenburg, Biometrische Zeitschrift 12 (1970) 195-195.

CHAPTER EIGHT:

EFFECT OF CRYSTOLLOGRAPHIC MORPHOLOGY ON CO ADSORPTION OVER COBALT CATALYSTS

Greg Collinge[a]*, and Jean-Sabin McEwen*[a,b,c,d,e]*

[a] The Gene & Linda Voiland School of Chemical Engineering and Bioengineering, Washington State University, Pullman WA 99164, USA

[b] Department of Physics and Astronomy, Washington State University, Pullman, WA 99164, USA

[c] Department of Chemistry, Washington State University, Pullman, WA 99164, USA

[d] Institute for Integrated Catalysis, Pacific Northwest National Laboratory, Richland, WA, 99352, USA

[e] Department of Biological Systems Engineering, Washington State University, Pullman, WA 99164, USA

* Corresponding author:

Jean-Sabin McEwen; 509-335-8580 (phone), 509-335-4806 (fax), js.mcewen@wsu.edu

Abstract

Cobalt undergoes a phase change from its native (low-temperature) hexagonal close packed (hcp) to face centered cubic (fcc) for nanoparticles smaller than ~10 nm in diameter. These smaller fcc nanoparticles are known to exhibit lower activity in Fischer-Tropsch synthesis than their larger hcp counterparts, and this could be for one or a few different reasons. Here, we explore the possibility that the reduced activity is due to an effective ~2% compressive strain experienced in the fcc surface due to the smaller effective bulk lattice constant of the Co fcc unit cell. We construct two lattice gas (LG) cluster expansions (CEs) to assess the underlying energetics, decomposed into effective cluster interactions (ECIs), of the CO/Co(111) fcc and the CO/Co(0001) hcp systems. To ensure the most meaningful LG CEs are found, we utilize our

recently developed method for determining ECI errors to effectively filter out insignificant clusters during the optimization procedure. The final optimum LG CEs show that the primary and most meaningful energetic difference between the two systems is simply the first nearest neighbor interaction energy, which is nearly ~100 meV less repulsive in the hcp system as compared to the fcc system. Since proximal CO can destabilize or stabilize key FT intermediates, that proximal CO are more difficult to induce on fcc nanoparticles could be a contributing factor into their reduced activity.

8.1. Introduction

CO adsorption on Co surfaces has historically and consistently maintained the interest of the heterogenous catalysis community due to its relevance to the Fischer-Tropsch (FT) reaction[1]. While bulk Co is stable in its hcp phase at most temperatures relevant to FT (at least ~420ºC at any pressure[2]), it has long been known that nanoparticles of Co have been observed to transition to the fcc phase if they are of sufficiently small size.[3-7] Kitakami and coworkers reported that this transition point occurs around a particle diameter of ~10 nm and that the fcc phase is indeed stable against temperature changes, indicating that the fcc phase is not merely metastable.[8-10] These results are of particular consequence because it has been reported that hcp nanoparticles are more active in CO hydrogenation than their fcc counterparts.[11-13] Combined with the important observation by the group of de Jong that Co nanoparticles show a pronounced and sudden size-insensitivity as the diameter of a Co nanoparticle increases past 6-8 nm in diameter[14], these studies suggest that the crystallographic morphology of cobalt may be of serious consequence to FT.

The origin of the reduced activity in Co FT catalysts for small nanoparticles was investigated by Liu et al.[15] In their work, they argued that the increased activity in hcp nanoparticles is likely due to the increase in stepped facets observed in its Co in-vacuum Wulff construction as compared to that of the fcc Wulff construction. This seems to be corroborated to some degree experimentally by the work of den Breejan et al.[16] Because this increase in stepped facets is imposed by the crystal symmetry of the hcp unit cell (i.e. there are no basal planes in directions perpendicular to the <0001> axis), the massive reconstructions observed in Co FT catalysts[17, 18] are unlikely to offset this disparity without also inducing a phase change. Thus, it seems likely that the relative preponderance of stepped facets in hcp nanoparticles will persist even in the presence of reactants. While this phenomenon may indeed be an underlying contributor to the observed difference in activity between fcc and hcp nanoparticles, there are potentially two other complementary or competing sources: (1) the chemical dissimilarity of coordination-similar fcc and hcp surfaces (the simplest example of "coordination-similar" facets being the Co(0001) and Co(111) surfaces), and (2) the surface atom strain brought about by the change in lattice parameters. The first potential source, chemical dissimilarity, is easiest to dismiss for the two basal planes, Co(0001) and Co(111), since one must go three layers deep before the chemical dissimilarity is evident. In other facets, this source may play a larger role, but since the coordination number of Co in the two crystals remains the same, comparable facets seem unlikely to exhibit major chemical dissimilarity. However, the second potential source, surface lattice strain, cannot be so easily dismissed.

Surface lattice strain has been shown to have a profound impact on the catalytic performance of many catalysts[19-23]. The typical (and importantly, ideally tunable) method for imparting strain is via a lattice mismatch between a lattice-fixing core and a lattice-matching shell. However, and relevant to our work here, in supported cobalt catalysts (which don't exhibit core-shell morphology), strain effects can be brought about by other sources such as via carbon adatom penetration[24] — unfortunately this does not seem likely to be a tunable parameter. While the surface of an fcc cobalt nanoparticle cannot be said to be strained in the traditional sense, the lattice constant of fcc cobalt is indeed smaller as compared to that of the hcp nanoparticle, corresponding to a roughly 2% compressive strain at the surface (see Figure 8.1). Since the top two layers of the basal planes of fcc and hcp Co are identical with likely similar chemical properties, this "strain" may indeed be of considerable consequence to the CO overlayers which form on the catalysts surface in the early stages of the FT reaction.

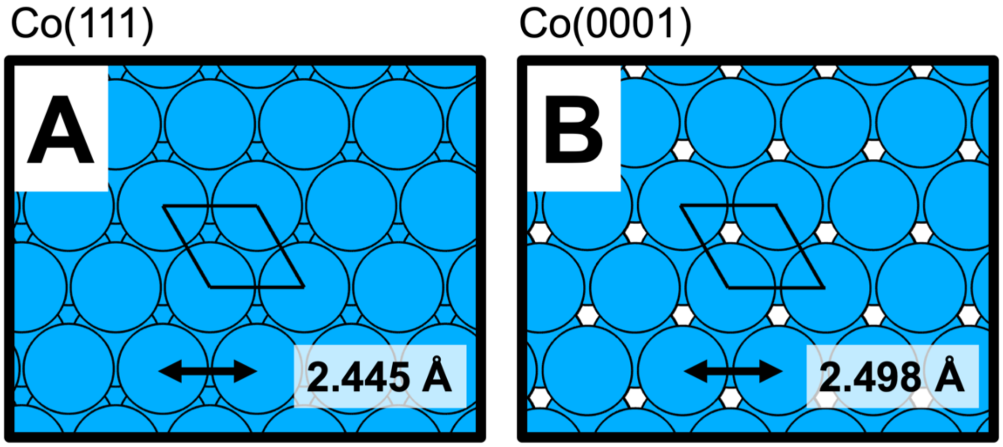


Figure 8.1. Comparison of the surface lattice constants of (A) fcc Co(111) and (B) hcp Co(0001).

To systematically investigate this potential strain effect on CO adsorption and configurations, we construct two separate lattice gas (LG) cluster expansions (CEs) of the CO/Co(0001) and CO/Co(111) systems. We use these to determine the fundamental chemical consequences of catalyst crystallographic morphology on CO adsorption via direct comparison of the LG effective cluster interactions (ECIs). We incorporate also our newly developed method of estimating ECI errors to effectively "filter" clusters with insignificant ECIs out of consideration during the CE optimization procedure. The end result is a relatively compact LG CE for each of the CO/Co(0001) and CO/Co(111) systems with minimal loss in predictiveness.

8.2. Methods

The Vienna *ab initio* Simulation Package (VASP)[25-27] was used to perform electronic structure and geometric relaxation calculations on the CO/Co(0001) and CO/Co(111) systems. VASP employs periodic boundary conditions and a plane wave basis set (which we expand to an energy cutoff of 450 eV) and projector augmented wave[28, 29] (PAW) pseudopotentials (version: VASP2012/VASP) to solve the Kohn-Sham[30] equations. Self-consistent field electronic ground states were solved to within an energy tolerance of 1×10^{-4} eV while forces were minimized to within 0.03 eV/Å to perform geometric relaxations. Here, the Perdew-Burke-Enzerhof[31] (PBE) exchange-correlation functional was used to be consistent with our previous work.[32-34] All calculations were spin polarized and dipole corrections were applied to account for the fictitious dipole created by our asymmetric slab model.

302 unique CO/Co(0001) and 332 unique CO/Co(111) configurations were generated in an automated fashion using the Alloy Theoretic Automated Toolkit (ATAT).[35-37] Each structure consists of a 4-layer Co(0001) or Co(111) slab with a ~15Å vacuum imposed. The bottom two layers are fixed in bulk positions (calculated as 2.494 Å (1.618 c/a ratio) and 3.457 Å for hcp Co and fcc Co, respectively) while the top two layers are allowed to completely relax during geometric optimization. CO molecules are adsorbed at top sites only in this study (a simplification to make the problem more tractable) and the xy-coordinates of the carbon atoms are fixed to avoid relaxation to bridge and hollow sites at high coverages. The oxygen atoms are, however, allowed to fully relax to permit potential “tilting” of the CO molecules off

perpendicular, a phenomenon shown to be important in the CO/Ru(0001) system.[38] ATAT generates structures of increasing supercell size to avoid (presumably) unnecessary computational burden, preferentially creating the smallest possible supercell to represent the periodicity of a given configuration. When this is done, the reciprocal space dimensions of the supercells change considerably, and k-point sampling cannot be performed in a precisely equivalent manner from one supercell to the next. As a result, ATAT sets the k-point sampling on a "k-points per reciprocal atom" (KPPRA) basis, which, at least in alloy systems, exhibit a near one-to-one equivalence to k-points per reciprocal angstrom. We set our KPPRA to a rather high value of 2400 as an attempt to hopefully overcompensate for the aforementioned imprecision and issues related to imposing a vacuum whose lattice vector has a tendency to act like a k-point sink within ATAT.

Lattice gas (LG) cluster expansion (CE) models were created using the Ab initio Mean-field Augmented Lattice Gas Modeling (AMALGM)[39] code. To create the library of potential clusters from which LG CE could be constructed, up to 5-body clusters were considered with a 2-body cutoff radius of 5 surface lattice constants (s.l.c.'s), 3- and 4-body cutoff radii of 3.7 s.l.c.'s, and a 5-body radius cutoff of 3 s.l.c.'s (these correspond to radial cutoffs of 12.47/12.22 Å, 9.23/9.04 Å, and 7.48/7.33 Å cutoffs for Co(0001) and Co(111), respectively). We fit the LG CE to the adsorption energies found for the structures in our two data sets. Here, we further implement our recently devised method for determining effective cluster interaction (ECI) errors[40] which are reported in the "two sigma" form that work suggests is appropriate for capturing their full variation within the data.

Two LG CEs were constructed for both of the systems considered here, one wherein the optimal LG CE was determined as that which minimizes the leave-multiple-out cross-validation

(LMO-CV) score as implemented within AMALGM regardless of the ECI errors found and another where the optimal LG CE was determined by minimizing the LMO-CV score but allowing only for LG CEs whose ECIs have magnitudes greater than their ECI errors by at least 1 meV (i.e. where $|\mathrm{ECI_j}| - (\mathrm{ECI\ Error})_\mathrm{j} > 1\ \mathrm{meV}$). Practically speaking, this was done by assigning a very large number (e.g. 10,000 eV/CO) to the LMO-CV score of LG CEs containing clusters that violate this rule during optimization. AMALGM implements a convergent version of the LMO-CV score calculation; we set the fraction-left-out to 0.6 and the "CV tolerance" to $10^{-7.5}$ eV/CO, which was determined to provide LMO CV scores accurate to within 1 meV/CO upon rerunning.

8.3. Results and Discussion

We first wish to compare the effects that ECI error "filtering" has on the LG CE results for both the CO/Co(111) and CO/Co(0001) systems, which we will refer to as the "fcc" and "hcp" systems for simplicity. The motivation for ECI error filtering is illustrated in Figure 8.2 where the fcc system results are shown in blue and the hcp system results are shown in orange. The LG CE used in Figure 8.2A is that optimized for the fcc system (i.e. the LG CE giving the minimum LMO-CV score for the fcc system data set) with the blue bars showing the resultant ECIs; the orange bars correspond to the ECIs for the hcp system using the same clusters shown for comparison. The LMO-CV score for the optimized fcc system LG CE is 28 meV/CO, however, its root mean squared residual (RMSR) is only slightly lower at 27 meV/CO. This similarity in values is worth elaborating on as it indicates that we can expect the predictiveness to only be barely worse than the fit to the data set itself. In other words, similar LMO-CV scores and RMSRs indicate that if new data were added to the dataset from the same theoretical distribution (i.e. the new data contains the same relevant energetic information as that captured by the current data set), the LG CE would predict their energies to within the same accuracy that it currently does the known data. That the LG CE fit is as large as it is regardless of this parity could indicate that the dataset itself contains significant deviations from ideality (whether from numerical errors/inconsistencies or relaxations). The error bars shown in Figure 8.2A bear this out to a degree since there are many ECIs with errors greater than the ECI values themselves. When this is the case, we know there are many subsets of the data set whose predicted CO adsorption energies would be better predicted if these ECIs were modified well away from the final fit value shown.

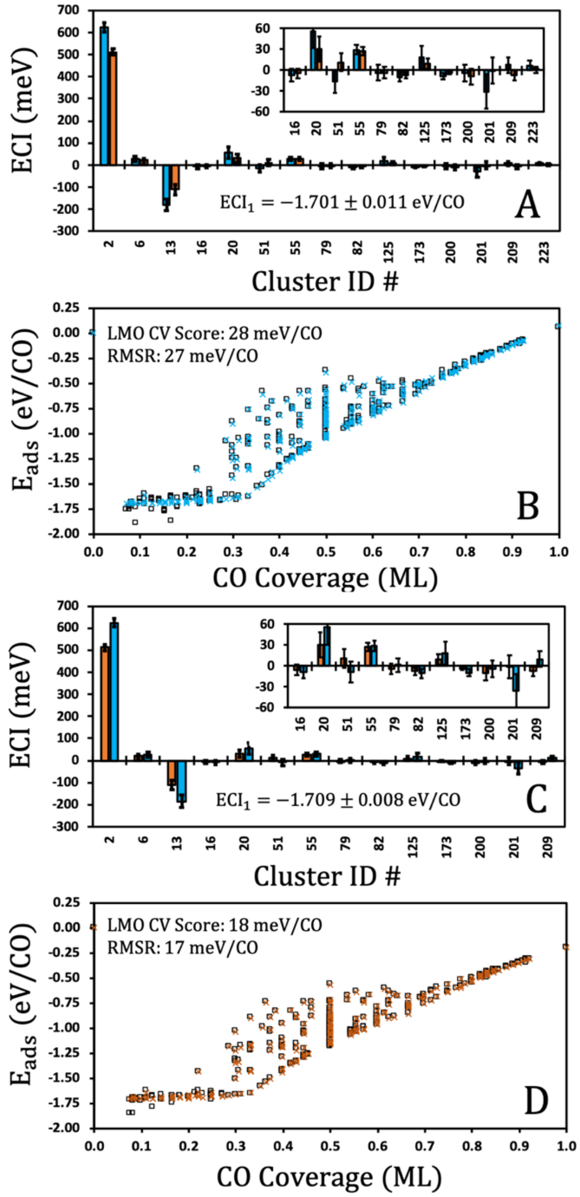


Figure 8.2. (A) ECIs for the optimum LG CE for the CO/Co(111) system (blue bars). The error bars indicate each ECIs associated "two sigma" ECI errors. The CO adsorption energy (cluster ID #1) is given as an inset due to magnitude discrepancy. ECIs for Clusters 16 – 223 are also shown inset at a smaller energy scale to make their values discernable. The orange bars are the ECIs found for the CO/Co(0001) shown for comparison only. (B) The predicted CO adsorption energies (blue crosses) found from optimum LG CE fit to the CO/Co(111) dataset set against the DFT calculated Co adsorption energies (black squares). The LMO-CV score and RMSR of the final fit are shown inset. (C) ECIs for the optimum LG CE for the CO/Co(0001) system (orange bars). The error bars indicate each ECIs associated "two sigma" ECI errors. The CO adsorption energy (cluster ID #1) is shown inset due to magnitude discrepancy. ECIs for Clusters 16 – 209 are also shown inset at a smaller energy scale to make their values discernable. The blue bars are the ECIs found for the CO/Co(111) are shown for comparison only. (B) The predicted CO adsorption energies (orange crosses) found from optimum LG CE fit to the CO/Co(0001) dataset set against the DFT calculated Co adsorption energies (black squares). The LMO-CV score and RMSR of the final fit are given as an inset.

Figure 8.2B shows the fcc system LG CE predicted (blue crosses) against the DFT calculated (black squares) CO adsorption energies (E_{ads}) for the fcc system data set. Visually, the quality of the fit is striking with most deviations occurring at coverages less than 0.33 ML. Given that the LMO-CV score is nearly as low as the RMSR, this asymmetry in the fit strongly indicates that there are inconsistencies in the data set. In principle, if the data set is consistent, the ECIs found for the lower coverage structures, which have far fewer cluster terms to fit to, would be easily “corrected” by the higher order cluster terms available to the higher-coverage structures. Since this does not appear to happen (the fits are poorer for the low coverage structures), it is likely that the data set exhibits inconsistencies. We posit these are most likely due to inconsistent **k**-point sampling and inconsistent artificial boundary conditions imposed by the widely varying supercell sizes that are produced by ATAT.

Figure 8.2C shows the optimized LG CE for the hcp system with the orange bars showing the resultant ECIs; the blue bars correspond to the ECIs for the hcp system using the same clusters shown for comparison. Here, the LMO-CV score and RMSR demonstrate parity similar to the fcc system. However, these values are reduced by ~10 meV/CO indicating much more ideal data in the data set. However, despite this, many of the ECIs are found to be insignificant when their ECI errors are accounted for. As was seen in the fcc system, Figure 8.1D shows that the fit to the lower coverage structures is worse in the hcp data set despite the lower LMO-CV score and RMSR. This again suggests that there are inconsistencies in its data set. The preponderance of insignificant ECIs in the optimum LG CEs shown in Figure 8.2 motivates us to include these ECI errors as a “filter” in the optimization process as described in section 8.2. The idea here is that many of these insignificant ECIs may simply be trying to capture the inconsistencies in the data set mentioned above. The result of this process is shown in Figure 8.3.

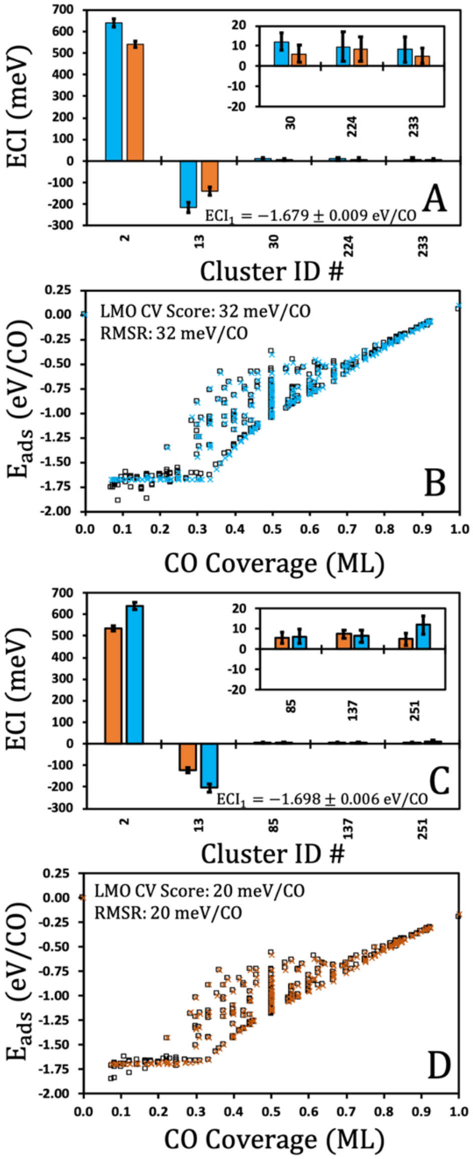


Figure 8.3. (A) ECIs for the optimum LG CE for the CO/Co(111) system (blue bars) using ECI errors as a filter. The error bars indicate each ECIs associated "two sigma" ECI errors. The CO adsorption energy (cluster ID #1) is shown inset due to magnitude discrepancy. ECIs for Clusters 30 – 233 are also given as an inset at a smaller energy scale to make their values discernable. The orange bars are the ECIs found for the CO/Co(0001) shown for comparison only. (B) The predicted CO adsorption energies (blue crosses) found from optimum LG CE fit to the CO/Co(111) dataset set against the DFT calculated Co adsorption energies (black squares). The LMO-CV score and RMSR of the final fit are shown inset. (C) ECIs for the optimum LG CE for the CO/Co(0001) system (orange bars) using ECI errors as a filter. The error bars indicate each ECIs associated "two sigma" ECI errors. The CO adsorption energy (cluster ID #1) is shown inset due to magnitude discrepancy. ECIs for Clusters 85 – 251 are also shown inset at a smaller energy scale to make their values discernable. The blue bars are the ECIs found for the CO/Co(111) are shown for comparison only. (D) The predicted CO adsorption energies (orange crosses) found from optimum LG CE fit to the CO/Co(0001) dataset set against the DFT calculated Co adsorption energies (black squares). The LMO-CV score and RMSR of the final fit are given as an inset.

The clusters found in both Figure 8.2 and Figure 8.3 are shown in Figure 8.4 for reference. When LG CEs with insignificant ECIs are filtered out during optimization, the number of retained clusters in the LG CE falls dramatically. In both the fcc and hcp system, only six clusters are found to be both predictive and significant, and of those, only three are greater than 20 meV in magnitude. In both systems, both the first nearest neighbor ("1NN", cluster #2, see Figure 8.4) and 1-1-1 triangular trio ("1TT", cluster #13, see Figure 8.4) dominate the energetics. The LMO-CV scores of the filtered fcc and hcp system LG CEs in Figure 8.3 are only 4 eV/CO and 2 eV/CO higher than their unrestricted counterparts in Figure 8.2. This is incredibly encouraging and demonstrates the utility of using ECI errors as a filter: (1) the nine to ten additional long-range and higher-body clusters of the unrestricted LG CEs in Figure 8.2 provide, at most, ~0.44 meV per additional cluster suggesting they may be superfluous artifacts of the fitting process, (2) it is expected that such simple systems (with only a single site and the xy-position of the carbon of the CO fixed) should intuitively not display overly complex LG CEs, and (3) it makes a lot of physical sense that these particular clusters (the 1NN and 1TT) dominate as they are the most compact 2- and 3-body clusters available representing, respectively, repulsions from close proximity and a correction for the tilting allowed to the CO molecule. The three higher-body interactions in both unrestricted LG CEs in Figure 8.3 are difficult to assign to any physical significance as they are very small, and the majority can only be guaranteed to be at most 2 meV when their ECI errors are accounted for.

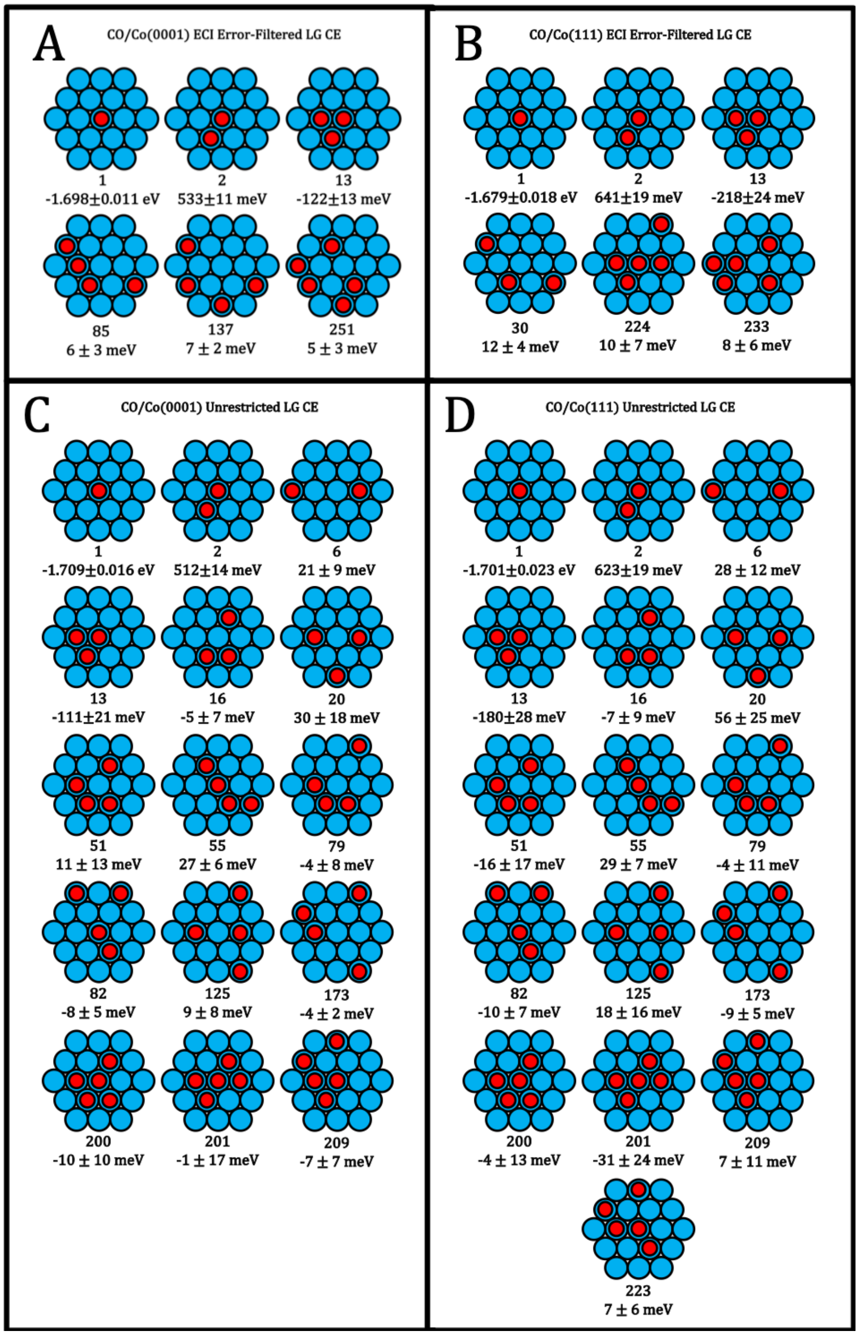


Figure 8.4. Clusters in the optimized LG CEs shown in Figures 8.2 and 8.3. Cluster IDs and ECIs are shown below each cluster graphic.

The differences in the ECIs of the same clusters that are present between the filtered and unrestricted LG CEs of the fcc and hcp systems is worth noting. In both fcc and hcp systems, cluster #13 is present in both the filtered and unrestricted LG CEs. However, the ECI for cluster 13, the 1TT, in the fcc system is -218 meV in the filtered LG CE but only -180 meV in the unrestricted LG CE. However, notice that clusters 200, 201, and 209 contain cluster 13 within them. Ideally, if there were no concern over the consistency of the DFT data, these higher-body clusters would be providing corrections to cluster 13 when more adsorbates are in proximity around it thus allowing the energetics of the isolated cluster 13 to be more accurately represented. However, in only one case (cluster 201 of the fcc system) is the ECI error smaller than the magnitude of the ECI itself. This means that these "corrections" are insignificant or, more exactly, too variable to suggest that the energetic contribution of isolated cluster 13 is being captured accurately. We posit that ECI filtering allows the most accurate version of ECIs for clusters like the 1TT in the fcc and hcp systems here to be found given the data set provided, and that this process mitigates some of the superfluous artificial fitting inherent to the linear regression procedure used on the total data set to get the final ECIs.

With the effect of ECI filtering stated, we turn now to the elucidation of the effect that the crystallographic morphology of the Co nanoparticle has on the CO adsorption energetics on Co(111) and Co(0001) within the LG CE framework presented here. For this purpose, we use the ECI filtered LG CEs presented in Figure 8.2. First of note, the adsorption energy (the ECI of cluster #1, "ECI_1" in Figure 8.2A and 8.2C) are shown to differ by ~20 meV with no overlapping ECI errors, suggesting this difference is significant. The hcp system adsorption energy is stronger than that of the fcc system which is the general trend in the data for the low coverage structures that are similar between the two systems. This difference is significant, but is unlikely to lead to great differences in, say, CO dissociation rates unless C and O binding energies are asymmetrically affected or otherwise not similarly affected. However, the 1NN ECIs are considerably different between the two system (641 meV vs 533 meV for the fcc and hcp systems, respectively). Both are large enough to be considered a 1NN exclusion energy (i.e. effectively infinite) at most temperatures. However, at large enough chemical potentials, such as those created by the high pressures in FT, the hcp 1NN may be accessed where the fcc 1NN is still prohibited. Proximal CO may have effects on CO dissociation, but if this discrepancy in 1NN is a trend for other CO – R interactions, the implications could be considerable as the presence of proximal CO has been shown to change the favorability of certain FT coupling reactions.[41] That this trend should continue is suggested because the source of this increased 1NN interaction energy appears to be entirely due to the decreased lattice spacing in the Co(111) surface as compared to the Co(0001) surface. Thus it should be expected that any two adspecies should have a more difficult time attaining proximity in fcc Co nanoparticles as compare to hcp Co nanoparticles.

The 1TT ECIs also significantly differ between the fcc and hcp systems here (-218 meV vs -122 meV for the fcc and hcp systems, respectively). However, their effect on the system must be contextualized by their constituent 1NNs. The ECIs of higher body clusters are always additive with the ECIs of their constituent lower-body clusters. In the case of the 1TT, three 1NN ECIs worth of energy must be expended before the attractive 1TT interaction is assessed. If two CO form a 1NN pair on the surface, the addition of a third CO in proximity to this 1NN pair can minimize the energy expenditure by only forming a single additional 1NN pair via creation of a 1-1-2 linear trio or 1-1-$\sqrt{3}$ bent trio instead of creating the 1-1-1 triangular trio since the energy of two repulsive 1NN ECIs is still less than three repulsive 1NN and one attractive 1TT. A 1TT can be avoided completely all the way up to 2/3 ML, above which 1TT must start being created. For this reason, we expect that the difference in 1TT ECIs between these two systems will remain inconsequential at all but the highest chemical potentials, likely inaccessible in any experimental setting.

8.4. Conclusions

Differences in the CO/Co(0001) and CO/Co(111) systems were assessed via the construction of first principles lattice gas cluster expansions (LG CEs). Two LG CEs were created for both systems to demonstrate the utility of using our recently developed method of determining ECI errors to effectively filter out LG CEs with “insignificant” clusters. The method is shown to drastically reduce the number of clusters included in the LG CEs of these two systems. Additionally, the most physically important clusters are still retained whose energetics, we posit, are ultimately better described. We thus recommend using this method to produce the most reliable LG CEs in a general setting.

Using the ECI filtered LG CEs for the two systems, we are able to show that the primary difference produced, given the simplified single-site model we have constructed, is in the first nearest neighbor (1NN) repulsion. While the adsorption energies are shown to be minimally different between the fcc crystal and hcp crystal, the 1NN energies have a large discrepancy. This means that, at high enough chemical potentials, 1NN pairs may be formed preferentially on the hcp crystal where they are still prohibitively energy costly on the fcc crystal. Since this discrepancy is likely due to the roughly 2% compressive strain induced in the surface of fcc crystals as compared to that of hcp crystals, we posit that this effect is likely replicated for other species pairings, ones where proximal spectators may stabilize or destabilize important FT reaction intermediates. This hints at an explanation for the reduced activity of fcc Co nanoparticles observed experimentally[11-13] beyond the relative ratios of stepped vs flat surfaces prevalent in the fcc vs. hcp in-vacuum Wulff construction.

Acknowledgements

Work at WSU was primarily funded by the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Chemical Sciences, Biosciences and Geosciences (DE-SC0014560), and by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1347973. G.C., Seattle Chapter ARCS Fellow and NSF Graduate Research Fellow, gratefully acknowledges financial support from the Achievement Rewards for College Scientists foundation as well as the National Science Foundation Graduate Research Fellowship Program. This research also used resources from the Center for Institutional Research Computing at Washington State University. PNNL is a multi-program national laboratory operated for the U.S. DOE by Battelle.

REFERENCES

[1] F. Fischer, H. Tropsch, BRENNSTOFF-CHEMIE 7 (1926) 97-104.

[2] T. Nishizawa, K. Ishida, Bull. Alloy Phase Diagr. 4 (1983) 387-390.

[3] O. S. Edwards, H. S. Lipson, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 180 (1942) 268-277.

[4] C. G. Granqvist, R. A. Buhrman, J. Appl. Phys. 47 (1976) 2200-2219.

[5] S. Gangopadhyay, G. C. Hadjipanayis, C. M. Sorensen, K. J. Klabunde, IEEE Trans. Magn. 28 (1992) 3174-3176.

[6] E. Anno, Phys. Rev. B 50 (1994) 17502-17506.

[7] J. L. Maurice, J. Briático, J. Carrey, F. Petroff, L. F. Schelp, A. Vaurès, Philos. Mag. A 79 (1999) 2921-2934.

[8] O. Kitakami, T. Sakurai, Y. Miyashita, Y. Takeno, Y. Shimada, H. Takano, H. Awano, K. Ando, Y. Sugita, Jpn. J. Appl. Phys. 35 (1996) 1724-1728.

[9] O. Kitakami, H. Sato, Y. Shimada, F. Sato, M. Tanaka, Phys. Rev. B 56 (1997) 13849-13854.

[10] H. Sato, O. Kitakami, T. Sakurai, Y. Shimada, Y. Otani, K. Fukamichi, J. Appl. Phys. 81 (1997) 1858-1862.

[11] D. I. Enache, B. Rebours, M. Roy-Auberger, R. Revel, J. Catal. 205 (2002) 346-353.

[12] M. Sadeqzadeh, H. Karaca, O. V. Safonova, P. Fongarland, S. Chambrey, P. Roussel, A. Griboval-Constant, M. Lacroix, D. Curulla-Ferré, F. Luck, A. Y. Khodakov, Catal. Today 164 (2011) 62-67.

[13] M. K. Gnanamani, G. Jacobs, W. D. Shafer, B. H. Davis, Catal. Today 215 (2013) 13-17.

[14] G. L. Bezemer, J. H. Bitter, H. P. C. E. Kuipers, H. Oosterbeek, J. E. Holewijn, X. Xu, F. Kapteijn, A. J. van Dillen, K. P. de Jong, J. Am. Chem. Soc. 128 (2006) 3956-3964.

[15] J.-X. Liu, H.-Y. Su, D.-P. Sun, B.-Y. Zhang, W.-X. Li, J. Am. Chem. Soc. 135 (2013) 16284-16287.

[16] J. P. den Breejen, P. B. Radstake, G. L. Bezemer, J. H. Bitter, V. Frøseth, A. Holmen, K. P. d. Jong, J. Am. Chem. Soc. 131 (2009) 7197-7203.

[17] J. Wilson, C. de Groot, J. Phys. Chem. 99 (1995) 7860-7866.

[18] A. Banerjee, A. P. van Bavel, H. P. C. E. Kuipers, M. Saeys, ACS Catal. 5 (2015) 4756-4760.

[19] P. Strasser, S. Koh, T. Anniyev, J. Greeley, K. More, C. Yu, Z. Liu, S. Kaya, D. Nordlund, H. Ogasawara, Nature chemistry 2 (2010) 454.

[20] K. Amakawa, L. Sun, C. Guo, M. Hävecker, P. Kube, I. E. Wachs, S. Lwin, A. I. Frenkel, A. Patlolla, K. Hermann, Angew. Chem. Int. Ed. 52 (2013) 13553-13557.

[21] L. A. Kibler, A. M. El-Aziz, R. Hoyer, D. M. Kolb, Angew. Chem. Int. Ed. 44 (2005) 2080-2084.

[22] S. Zhang, X. Zhang, G. Jiang, H. Zhu, S. Guo, D. Su, G. Lu, S. Sun, J. Am. Chem. Soc. 136 (2014) 7734-7739.

[23] I. Kasatkin, P. Kurr, B. Kniep, A. Trunschke, R. Schlögl, Angew. Chem. Int. Ed. 46 (2007) 7324-7327.

[24] X.-Q. Zhang, R. A. van Santen, E. J. M. Hensen, ACS Catal. 5 (2015) 596-601.

[25] G. Kresse, J. Hafner, Phys. Rev. B 47 (1993) 558.

[26] G. Kresse, J. Furthmüller, Comput. Mater. Sci. 6 (1996) 15-50.

[27] G. Kresse, J. Furthmüller, Phys. Rev. B 54 (1996) 11169.

[28] P. E. Blöchl, Phys. Rev. B 50 (1994) 17953.

[29] G. Kresse, D. Joubert, Phys. Rev. B 59 (1999) 1758.

[30] W. Kohn, L. J. Sham, Phys. Rev. 140 (1965) A1133-A1138.

[31] J. P. Perdew, K. Burke, M. Ernzerhof, Phys. Rev. Lett. 77 (1996) 3865-3868.

[32] G. Collinge, Y. Xiang, R. Barbosa, J.-S. McEwen, N. Kruse, Surf. Sci. 648 (2016) 74-83.

[33] G. Collinge, N. Kruse, J.-S. McEwen, J. Phys. Chem. C 121 (2017) 2181-2191.

[34] G. Collinge, N. Kruse, J.-S. McEwen, J. Catal. 368 (2018) 31-37.

[35] A. van de Walle, M. Asta, G. Ceder, Calphad 26 (2002) 539-553.

[36] A. van de Walle, G. Ceder, Journal of Phase Equilibria 23 (2002) 348.

[37] A. van de Walle, Calphad 33 (2009) 266-278.

[38] J.-S. McEwen, A. Eichler, J. Chem. Phys. 126 (2007) 094701.

[39] G. Collinge, K. Groden, C. Stampfl, J.-S. McEwen, Journal of Chemical Theory and Computation (2018) (Submitted).

[40] G. Collinge, A. J. R. Hensley, J.-S. McEwen, Journal of Chemical Theory and Computation (2019) (Submitted).

[41] C. J. Weststrate, J. W. Niemantsverdriet, ACS Catal. 8 (2018) 10826-10835.

CHAPTER NINE:
CONCLUSIONS

Two different but complimentary research efforts to determine the structure, reaction environment, and catalytically active phase of Co-based Fischer-Tropsch (FT) catalysts have been covered in great detail in this dissertation from a first principles computational perspective (at the level of density functional theory (DFT)). The first endeavor was covered in the chapters three through five and concerned cobalt-copper (CoCu) nanoparticles and the ways in which the strongest adsorbing FT reactant, carbon monoxide (CO), interacts with them. The second endeavor was covered in chapters six through eight and involved the development and implementation of a reformulation of lattice gas (LG) cluster expansion (CE) theory and code, new methods to assess their errors, and application of these error-minimized LG CEs to CO adsorption energetics on the face centered cubic (fcc) and hexagonal close packed (hcp) phases of Co nanoparticles. These latter crystal phases have been shown to have marked effect on the activity of the FT reaction.

In the first of these endeavors, we were able to show that in the absence of CO, CoCu nanoparticles have a strong thermodynamic tendency to self-assemble into a Co@Cu core-shell morphology. However, we were able to show that CO adsorption induces an antisegregation of Co to the surface of these nanoparticles. Interestingly, at FT temperatures and relevant pressures, the amount of Co enrichment of the surface was found to be limited to at most 50% of step edges and 25% of terraces due to the formation of highly favorable geminal subcarbonyl complexes ($Co(CO)_x$, x=2 or 3). Favorability was determined via the construction of surface phase diagrams, which require the application of statistical mechanical principles to our DFT data to determine Gibbs free energies. These subcarbonyl structures, highly metal bound at steps, were then computationally subjected to “rupturing”, wherein the $Co(CO)_x$ complex is removed from the step—creating one to two kink sites—and adsorbed on CoCu terraces. The thermodynamic favorability of this process was demonstrated for the cobalt tricarbonyl complex, $Co(CO)_3$, and we were further able to show that these Co tricarbonyl complexes can dimerize to begin the nucleation process for nanoisland formation. These results, in sum total, contribute to our understanding of CoCu reconstruction and what surface moieties can be expected on the surface during FT to oxygenates.

In the second of the two research endeavors covered in this dissertation, a reformulation of multi-component LG CE theory was developed alongside the algorithms needed to implement this new theory. These algorithms were used to create the framework of a new LG CE development code called the *ab initio* Mean-field Augmented Lattice Gas Modeling (AMALGM) code, which is designed to parameterize LG CEs using *ab initio* data, particularly from DFT calculations. A new method, minimally reliant on statistical assumption about the inputted data, was also developed to assess the errors of the energetic fitting coefficients within the LG CE. This method was shown to be useful in quantifying the degree to which LG CEs are able (or unable) to capture geometric relaxations away from ideal lattice behavior in the inputted DFT data. While confidence intervals rely heavily on the statistical assumption of linear regression and only capture uncertainty due to poor configuration space sampling, we demonstrated that these error estimates capture the uncertainty brought about due solely to the non-ideality of the energetic data itself. We were then able to use these error estimate as criteria within the LG CE optimization routines of AMALGM to effectively filter out unreliable LG CEs while still maximizing their predictiveness. This process was used on the fcc CO/Co(111) and hcp CO/Co(0001) systems to reveal the fundamental differences inherent to CO adsorption on the two crystal phases of cobalt. We showed that the first nearest neighbor repulsion in the CO/Co(0001) system is significantly smaller (by nearly 100 meV) than that of the CO/Co(111) system, and posited that at high chemical potentials, CO-CO first nearest neighbor pairs may form on the Co(0001) surface where they are still prohibited on the Co(111) surface. These types of pairs, involving possibly CO and other key FT intermediates, may contribute to the stabilization or destabilization of such species which may explain the increased activity of the hcp phase over the fcc phase in FT. These results contribute to our understanding of LG CE theory and provide a means to quantifying LG CE reliability, a capability currently missing from the tools available to researchers in the fields of surface science and heterogenous catalysis.

The work performed in this dissertation provides a clear “jumping off point” for future research into FT on Co and CoCu, both for the purposes of increasing selectivity to oxygenates and determining the structure-activity relationships in FT synthesis in general. We expect that the development of AMALGM and our LG CE error estimation technique will provide a heretofore missing tool to the computational catalysts community that will aid in research into myriad catalytic systems, mitigating the problems inherent in using relaxed surface calculations with the ideal lattice assumption of general CE theory.

APPENDIX A

I. Derivation of Gibbs Free Energy from Statistical Mechanical First Principles

In general − for ideal, isothermal, constant volume chemical reactions − we can write the Helmholtz free energy in terms of the canonical partition function, Q, for each molecular species thusly[1]:

$$F(N,V,T) = -k_B T ln(Q)$$

$$Q = \frac{q^N}{N!} = \frac{(q_{elec} q_{trans} q_{rot} q_{vib})^N}{N!}$$

$$\Rightarrow F(N,V,T) = \Bigg[-k_B T ln(q_{elec}(N,T)^N) - k_B T ln\left(\frac{q_{trans}(N,V,T)^N}{N!}\right) - k_B T ln(q_{rot}(N,T)^N) - k_B T ln(q_{vib}(N,T)^N)\Bigg]$$

$$\Rightarrow F(N,V,T) = [F_{elec}(N,T) + F_{trans}(N,V,T) + F_{rot}(N,T) + F_{vib}(N,T)]$$

where small q's are the molecular canonical partition functions, overall as well as for various contributions, electronic (elec), translational (trans), rotational (rot), and vibrational (vib). It will be helpful (as will become apparent shortly) to point out that the electronic, rotational, and vibrational molecular canonical partition functions do not depend on V, while the translational partition function is the only contribution that does.

Reuter and Scheffler[2], as an example, proceed by pointing out that the pV term in

$$G = F + pV$$

is generally quite small on a per unit area basis and can thus be neglected in a restricted pressure range. We opt to create an exact, unrestricted connection to G by evaluating the isothermal-isobaric partition function, Δ, directly:

$$G(N,p,T) = -k_B T ln(\Delta)$$

$$\Delta(\mathrm{N,p,T}) = \int_0^\infty Q(N,V,T)\ e^{-\frac{pV}{k_BT}}dV$$

where Q is the canonical partition function as defined previously. We have chosen to use the notation and ensemble definition provided in Shell's *Thermodynamic and Statistical Mechanics*[3]. From here, it is quite easy to show that for the molecular canonical partition functions that do not depend on V, their contribution to the isothermal-isobaric partition function becomes:

$$\Delta_{elec,rot,vib}(\mathrm{N,p,T}) = \int_0^\infty q(N,T)^N e^{-\frac{pV}{k_BT}}dV = q(N,T)^N\left(\int_0^\infty e^{-\frac{pV}{k_BT}}dV\right) = q(N,T)^N\left(\frac{k_BT}{p}\right)$$

and taking advantage of chemical potentials, we differentiate the logarithm with respect to N, and generate:

$$\mu_{elec,rot,vib} = -k_BT\frac{\partial(\ln\left(\Delta_{elec,rot,vib}(N,p,T)\right)}{\partial N} = -k_BT\frac{\partial(\ln\left(q(N,T)^N\left(\frac{k_BT}{p}\right)\right)}{\partial N}$$

$$= -k_BTln\big(q(N,T)\big)$$

which, due to summability rules (i.e. $G = \sum N_i\mu_i$) results in

$$\Longrightarrow G_{elec,rot,vib}(N,T) = F_{elec,rot,vib}(N,T)$$

for any molecular species and no approximation is consequently necessary.

The translational canonical partition function depends on V thusly:

$$\frac{q_{trans}(N,V,T)^N}{N!} = \left(\frac{T}{\Lambda}\right)^{\frac{3N}{2}}\frac{V^N}{N!}$$

where Λ is the translational "temperature" equal to $\frac{h^2}{2\pi mk_B}$. This changes the evaluation of its contribution to Δ as shown:

$$\Delta_{trans}(\mathrm{N,p,T}) = \int_0^\infty \left(\frac{T}{\Lambda}\right)^{\frac{3N}{2}} \frac{V^N}{N!}\, e^{-\frac{pV}{k_BT}} dV = \left(\frac{T}{\Lambda}\right)^{\frac{3N}{2}} \left[\int_0^\infty \frac{V^N}{N!}\, e^{-\frac{pV}{k_BT}} dV\right] = \left(\frac{T}{\Lambda}\right)^{\frac{3N}{2}} \left[\left(\frac{k_BT}{p}\right)^{N+1}\right]$$

and, once again, we use the definition of chemical potentials and summability rules to give:

$$\mu_{trans}(p,T) = -k_BTln\left[\left(\frac{T}{\Lambda}\right)^{\frac{3}{2}}\left(\frac{k_BT}{p}\right)\right]$$

$$G_{trans}(N,p,T) = -Nk_BTln\left[\left(\frac{T}{\Lambda}\right)^{\frac{3}{2}}\left(\frac{k_BT}{p}\right)\right]$$

To confirm these conclusions, the translational contribution to the Helmholtz free energy is commonly calculated (from the canonical partition function) as

$$F_{trans}(N,V,T) = -Nk_BTln\left[\left(\frac{T}{\Lambda}\right)^{\frac{3}{2}}\left(\frac{V}{N}\right)e\right]$$

which we can see is exactly the same as $G_{trans}(N,p,T)$ except for the last term(s) in the logarithm. Because we assumed ideal gas behavior from the very start, the $\left(\frac{V}{N}\right)$ term is equal to $\left(\frac{k_BT}{p}\right)$ and $F_{trans}(N,V,T)$ actually only differs from $G_{trans}(N,p,T)$ by the exponential "e" in the logarithm. Making ideal gas equation of state substitutions and separating this out gives

$$F_{trans}(N,V,T) = -Nk_BTln\left[\left(\frac{T}{\Lambda}\right)^{\frac{3}{2}}\left(\frac{V}{N}\right)\right] - Nk_BTln[e]$$

$$= G_{trans}(N,p,T) - Nk_BT$$

$$= G_{trans}(N,p,T) - pV$$

which can be compared to the thermodynamic definition $F = G - pV$ to show that the entire pV term is consumed by the transitional contributions to G and F. This leaves all other contributions (vib., rot., elec., etc) exactly equal to each other (e.g. $G_{vib}(N,p,T) = F_{vib}(N,V,T)$) and confirms what was found before using the isothermal-isobaric ensemble: for all contributions not dependent on V or p, the Gibbs free energy can be calculated directly as the Helmholtz free energy without restriction. In summary:

$$G_{trans} = F_{trans} + pV = F_{trans} + Nk_BT\ (assuming\ ideal\ gas)$$

$$G_{elec} = F_{elec}$$

$$G_{rot} = F_{rot}$$

$$G_{vib} = F_{vib}$$

We should note that these relationships hold even without the assumption of an ideal gas so long as F_{trans} and pV are evaluated directly (likely within the grand canonical ensemble) to account for the non-ideality. However, the assumption that the contributions are uncoupled (i.e. $q = q_{elec}q_{trans}q_{rot}q_{vib}$) is still required.

We are now in a position to delineate and manipulate the expressions used to calculate the various contributions to the Gibbs free energy in our calculations:

a) We assume, with negligible loss of accuracy, that only the ground state electronic energy − found from DFT − is important:

$$G_{elec} = N\mu_{elec} = -Nk_BTln\left[e^{-\frac{E^{DFT}}{k_BT}}\right] = NE^{DFT}$$

b) Rotational contributions are found from the "high T" limit of the rigid-rotor approximated rotational molecular canonical partition function:

$$G_{rot} = N\mu_{rot}(T) = N\left[k_BTln\left(\frac{8\pi^2\mu r^2k_BT}{\sigma h^2}\right)\right]$$

where μ is the reduced mass and μr^2 is(are) the moment(s) of inertia, and σ is the symmetry number of the molecule in question (equal to 1 for CO).

c) Vibrational contributions are found from the harmonic oscillator vibrational molecular canonical partition function:

$$G_{vib} = N\mu_{vib}(T) = N\left[k_BT\sum_{k}^{norm.modes}\left(ln\left[\frac{e^{-\frac{h\nu_k}{2k_BT}}}{1-e^{-\frac{h\nu_k}{k_BT}}}\right]\right)\right]$$

d) And translational contributions are found via the equation(s) derived previously. As is traditionally done, we break this down into a "standard" chemical potential (p^0= 1bar) plus a pressure correction term as such:

$$G_{trans} = N\mu_{trans}(T,p) = N\left[-k_BTln\left[\left(\frac{2\pi mk_BT}{h^2}\right)^{\frac{3}{2}}\left(\frac{k_BT}{p^0}\frac{p^0}{p}\right)\right]\right]$$

$$= N\left[-k_BTln\left[\left(\frac{2\pi mk_BT}{h^2}\right)^{\frac{3}{2}}\left(\frac{k_BT}{p^0}\right)\right] + k_BTln\left(\frac{p}{p^0}\right)\right]$$

$$= N\left[\mu_{trans}(T,p^0) + k_BTln\left(\frac{p}{p^0}\right)\right]$$

with

$$\mu_{rot}(T) + \mu_{vib}(T) + \mu_{trans}(T,p^0) = \mu^0(T,p^0)$$

II. Calculating ΔG and Δγ for Phase Diagrams

In our system, we have the following chemical reaction occurring:

$$(N_{CO})CO(g) + CoCu \overset{ads}{\longrightarrow} CO(\theta = \frac{N_{CO}}{N_s})/CoCu$$

where N_s is the total number of adsorption sites. We assume adsorption is in chemical equilibrium with a CO reservoir. From this we, at least intermediately, need ΔG:

$$\Delta G = G_{CO(\theta)/CoCu} - G_{CoCu} - G_{CO} = \mu_{CO(\theta)/CoCu} - \mu_{CoCu} - N_{CO}\mu_{CO}$$

$$= \left(E^{DFT}_{CO(\theta)/CoCu} + \mu^{vib}_{CO(\theta)/CoCu}(T)\right) - \left(E^{DFT}_{CoCu} + \mu^{vib}_{CoCu}(T)\right)$$

$$- \left(N_{CO}\left[E^{DFT}_{CO} + \mu^{0}_{CO}(T, p^0) + k_B T ln\left(\frac{p_i}{p^0}\right)\right]\right)$$

Which, as is often done, can be expressed as

$$\Delta G(N_{CO}, p_{CO}, T) = \Delta G^0(N_{CO}, p^0, T) - N_{CO} k_B T ln\left(\frac{p_{CO}}{p^0}\right)$$

and since this value is now on an ambiguous “per supercell” basis, we divide through by the surface area , A, per supercell to get Δγ. Variation in p_{CO} (on a log scale) at a chosen T results in the generation of our phase diagram.

III. Surface Vibrations

At this point, we address the calculation of the non-DFT chemical potential terms. We use our specific terms to illustrate some general examples.

For the gas phase CO, $\mu^0_{CO}(T, p^0)$, two choices are commonly taken: 1) As, for example, Reuter and Scheffler[2] and Getman et al.[4], taken from tabulated values in thermodynamic tables, or 2) as, for example, the group of Mark Saeys[5, 6], evaluated from the vibrational partition function and frequency calculations as implemented in VASP or other *ab initio* software. We have opted here to use the latter choice and calculate vibrational frequencies using VASP since we will do so for the surfaces, as well.

For the adsorbed system, $\mu^{vib}_{CoCu\sim CO}(T)$, and the clean surface, $\mu^{vib}_{CoCu}(T)$, there are a few methods to choose from: 1) As, for example, Getman et al.[4], assume that the adsorbate vibrations are all that matter in the calculations and that the surface metal vibrations are unperturbed by the adsorbate(s); scale the vibrational contribution from one adsorbate by the total number of adsorbates in the system or calculate them explicitly. 2) As, for example, Reuter and Scheffler[2], use the Einstein model to evaluate the phonon density of states (aka "spectral density function") and evaluate the surface vibrational contributions by selecting a range of characteristic frequencies for the surface atoms. 3.) Assume the two terms will approximately cancel each other out[7, 8]. For reasons expounded upon in the full article, we have chosen to eschew all of these approaches and calculate the surface vibrational modes for the clean and adsorbate-covered surfaces directly, allowing all atoms to relax except for those kept fixed in their bulk positions (the bottom two to three layers, generally). The vibrational partition function is then calculated via an explicit sum of the resultant normal modes. In order to fully capture these modes, the supercell is quadrupled in size (doubled in the x and y directions).

III.A. VASP Vibrational Mode Calculation Details

The vibrational calculations were run at a tighter SCF tolerance in order to recover all the real-valued vibrational modes. In only one case was this criterion not sufficient to eliminate imaginary modes. For that one case, the geometric optimization tolerance was set to 1.0×10^{-2} eV/Å prior to running the vibrational calculation (still at the tighter SCF tolerance). Visually, there was no difference in the newly optimized structure, and running the extra optimization corresponded to an error in the original calculation of less than 2 meV. As such, the original optimized structure was still used for energy comparison purposes. To ensure that running the geometric optimizations at the higher SCF tolerance was not necessary for the rest of the structures, we re-ran a few select geometric optimization calculations at the 1.0×10^{-8} eV SCF criteria and found that 1.0×10^{-4} eV was in error by less than 0.1 meV, and no additional geometric optimization steps had to be taken during these runs. This confirms that the optimized structures − ran with the tighter SCF tolerance − actually correspond to their energy minima. These results are shown in Table A1.

Table A1. Comparison of total DFT energies calculated using a looser and tighter self-consistent field (SCF) tolerance for a few select structures. As can be seen, little error in the looser tolerance is implied by the use of a tighter tolerance.

System Tested	Adsorption Energy (eV/CO)		Energy Differences (eV)			
	1×10^{-4} eV SCF Tolerance	1×10^{-8} eV SCF Tolerance	system diff	clean E diff	CO E diff	ads E diff
0.50 ML CO/ 0.25 ML Co Cu/Co(0001)	-1.15685	-1.15027	**0.00009**	0.00005	-0.00656	0.00658
0.75 ML CO/ 0.25 ML Co Cu/Co(0001)	-0.85377	-0.84722	**0.00003**			0.00655
0.50 ML CO/ 0.25 ML Co Cu/Co($10\bar{1}2$)	-1.43024	-1.42408	**0.00001**	0.00081		0.00616
0.75 ML CO/ 0.25 ML Co Cu/Co($10\bar{1}2$)	-1.22391	-1.21762	**0.00002**			0.00629

IV. Co patching in CoCu models

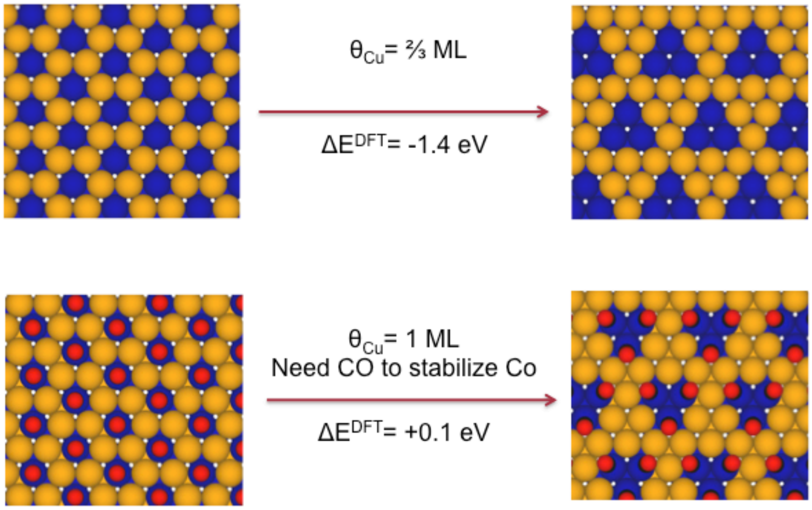


Figure A1. A clean surface with 1/3 ML Co terminating the surface (shown to be thermodynamically unfavorable) and the same surface with CO adsorbed (shown to be more stable by >1 eV). As can be seen, the Co atoms in the clean surface do in fact have a thermodynamic tendency to cluster and create patches - a model proposed for CoCu catalysts. However, in the presence of CO, such patching is far less likely to occur.

V. Comparing the Favorability of the Geminal Dicarbonyl across other GGA Functionals

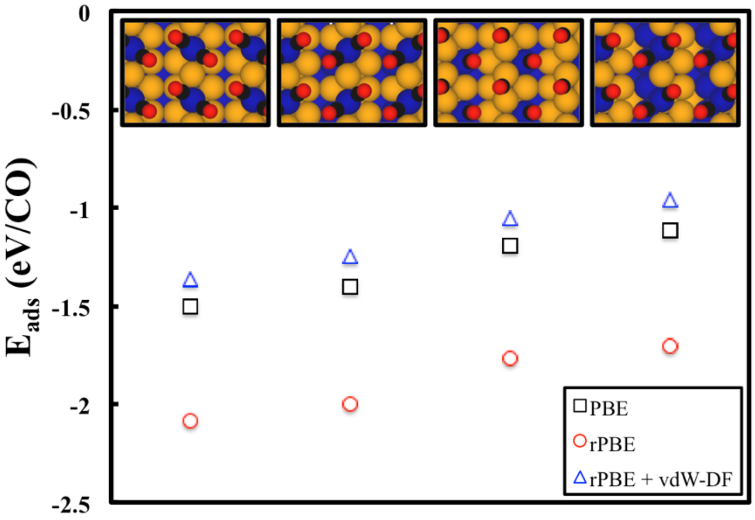


Figure A2. Calculated adsorption energies for the geminal dicarbonyl (far left inset picture and data points) with the three next most favorable coverage-equivalent configurations. In the 2nd and 4th pictures from the left, there is a terrace Co at the surface that is partially obscured by the CO adsorbed to the step Co above it. Here, Co is blue, Cu is orange, C is black, and O is red. The structures do not change noticeably when a different functional is used.

VI. Phase Diagram for CO/Cu/Co(0001)

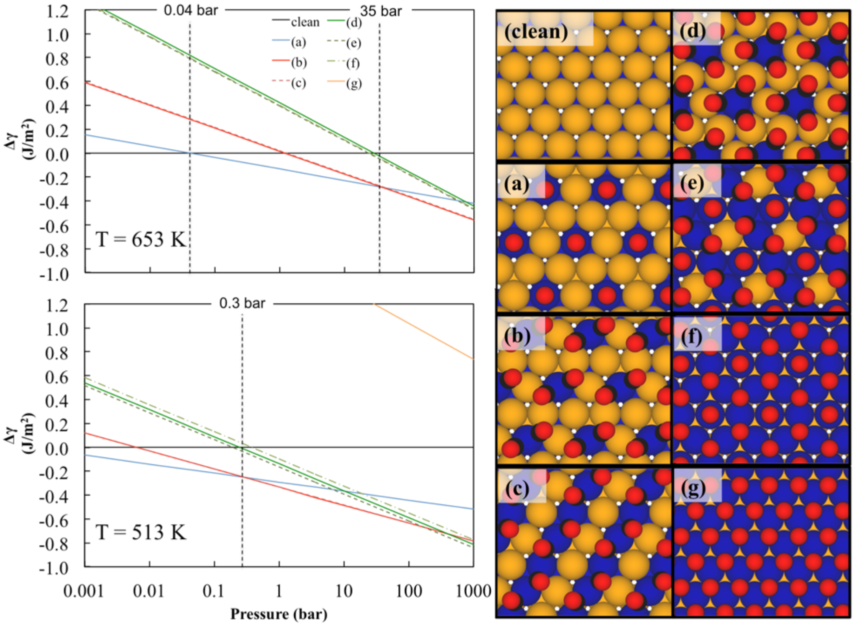


Figure A3. CO/Cu/Co(0001) Phase diagram of minimal DFT energy configurations for each CO coverage tested in our previous study[9] at 513 K and 653 K. Vertical dotted lines are placed at its marked pressure to delineate a phase transition. Here, numerous low-lying energy configurations exist for 0.5 ML CO and 0.75 ML CO and are thus assigned different shades of the same color in the legend (reds for 0.5 ML CO and greens for 0.75 ML CO).

We show a similarly constructed phase diagram for our previously studied CO on Cu/Co(0001) as a comparison to the stepped surface presented here and as a fleshing out of the CO on CoCu story. This is shown in Figure A2. At 513 K, the clean surface (no Co enrichment) is only favorable at pressures less than 1 mbar - similar to the stepped surface. The 0.25 ML CO coverage on Cu$_{0.25}$Co$_{0.75}$/Co(0001) (8(a)) is thermodynamically favored up to ~0.3 bar, at which point we find that the 0.50 ML CO coverage on Cu$_{0.25}$Co$_{0.75}$/Co(0001) (8(b)), corresponding to the formation of geminal dicarbonyl Co, is most thermodynamically favorable. At no reasonable pressure does another surface phase transition occur. The story is similar at 653 K. The transition between the clean Cu-terminated surface and monocarbonyl-coordinated Co (8(a)) occurs at 40 mbar, and the transition to the geminal carbonyl Co covered surface (8(b)) is predicted at a considerably higher pressure of ~35 bar (since gas phase CO starts to become significantly non-ideal at this point, the transformation might occur at a higher pressure). In both cases, the adsorbed CO induces only 0.25 ML Co enrichment of the surface. Therefore, Co "pumping" is expected to be driven to at most 0.25 ML surface enrichment at these temperatures. Our calculations indicate that the thermodynamically driven transition to 0.75 ML CO coverage, corresponding to 1.00 ML surface Co enrichment (the complete inversion of the CoCu surface layer sequence), does not occur at 10 bar until the temperature is lowered to ~331 K and does not occur at 1 bar until it is lowered to ~283 K, at which point the kinetics might be prohibitively controlling. This is in contrast to the stepped surface, of course, where Co pumping to 1.00 ML surface enrichment is never favorable regardless of temperature or pressure.

VII. Surface Chemical Potential Change for the Flat Surface

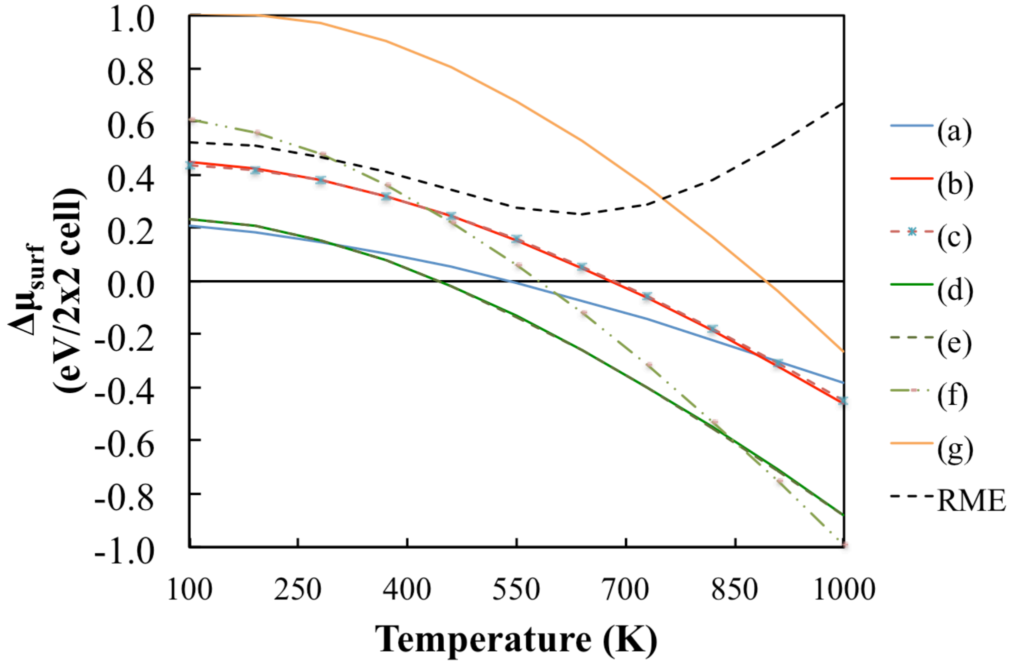


Figure A4. The Cu/Co(0001) surface chemical potential change (given in eV per p(2×2) supercell) for the minimum DFT energy Co/CO configurations studied here as a function of temperature. A common practice is to assume this value is zero or very close to zero. The black dashed line shows the root mean error (RME), or root mean deviation from a value of zero, for all configurations. This error is as great as ~0.6 eV/p(2×2) supercell and only as low as ~0.25 eV/p(2×2) supercell depending on the temperature and, like the stepped surface, does not appear to be bounded.

VIII. Comparison of Phase Diagrams for CO/Cu/Co(0001)

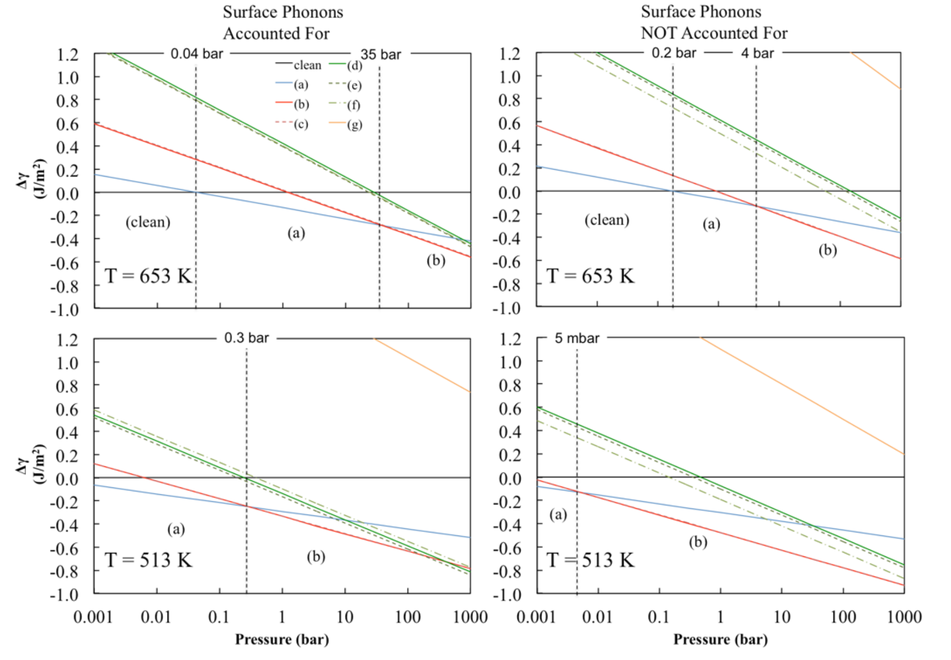


Figure A5. Comparison of phase diagrams constructed for the Cu/Co(0001) surface that have had surface phonon modes accounted for vs. unaccounted for. To highlight the largest differences, the low energy surface phase (bold lettering in parenthesis) is shown in each region delineated by the vertical dashed lines. The legend corresponds to the configurations shown in Figure A2.

IX. Vibrational Density of States

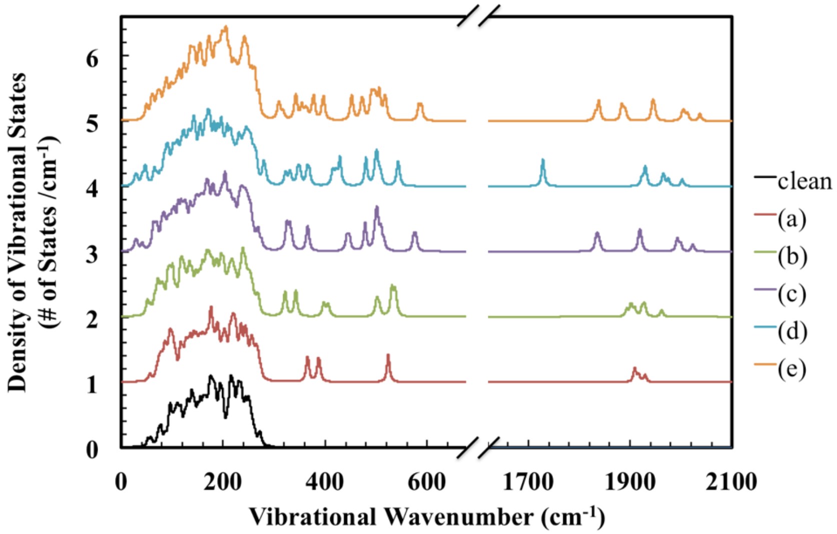


Figure A6. Vibrational Density of States for the CO adsorbed on Cu/Co($10\bar{1}2$) systems. Here (a)-(e) correspond to the structures referenced in Figure 4.7, representing (a) the single CO molecules adsorbed on single surface Co atoms at the terrace (0.25 ML CO, 016 ML surface Co); (b) the geminal dicarbonyl structure (0.50 ML CO, 0.16 ML surface Co); (c) the geminal tricarbonyl structure (0.75 ML CO, 0.16 ML Co); (d) the CO bridging a geminal dicarbonyl Co and terrace Co (0.75 ML CO, 0.33 ML Co); and (e) the geminal tricarbonyl structure with single CO adsorbed on single Co at the terrace (1.00 ML CO, 0.33 ML Co).

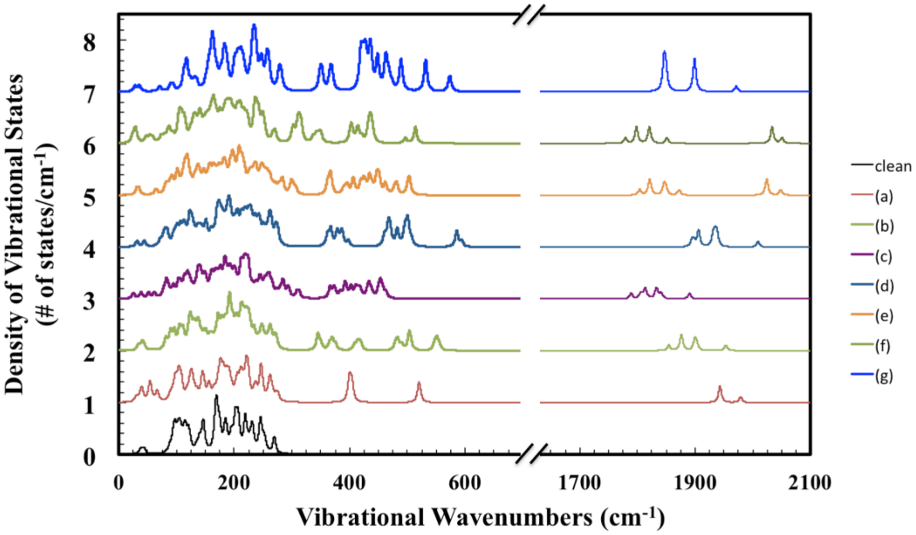


Figure A7. Vibrational Density of States for the CO adsorbed on Cu/Co(0001) systems. Here (a)-(g) correspond to the structures referenced in Figure A3. Briefly: (a) 0.25 ML CO, 0.25 ML Co; (b) 0.50 ML CO, 0.25 ML Co; (c) 0.50 ML CO, 0.50 ML Co; (d) 0.75 ML CO, 0.25 ML Co; (e) 0.75 ML CO, 0.75 ML Co; (f) 0.75 ML CO, 1.00 ML Co; and (g) 1.00 ML CO, 1.00 ML Co.

Vibrational density of states were obtained by applying a Lorentz distribution at each VASP-calculated vibrational mode with a lambda value of 3 cm^{-1} and intensity of 1.0 then summing over all Lorentz-distributed modes to obtain a single distribution across all wavenumbers. Each distribution was then normalized such that numerically integrating each distribution returned the 3N VASP-calculated vibrational modes originally inputted. As such, we have constructed a surface "spectral density function"[1].

X. Propagation of Error in DFT Calculated Vibrational Frequencies.

Concerning the DFT error in the calculated frequencies, we have run small vibrational mode calculations on the p(2x2) cell of clean Cu/Co(0001) and the geminal dicarbonyl on Cu/Co(0001) using revPBE and revPBE with vdW-DF to find an approximate error in the calculated frequencies obtained using the pure PBE functional. While these calculations could be characterized as preliminary, they reveal that the calculated frequencies are sensitive to the functional used (and therefore DFT error) by *at most* 22 cm^{-1} or ~0.003 eV, where the high CO stretch frequencies show the most error.

Error propagation analysis shows that the error in Gibbs energy resulting from an error in the vibrational frequency is

$$dG_i = \left(\frac{1}{2} + \frac{1}{e^{\frac{hv_i}{k_B T}} - 1}\right) d(hv_i)$$

or

$$dG_i = \left(\frac{1}{2} + n_i\right) dE_i,$$

where n_i is the vibrational occupation number. This shows that lower frequencies result in larger errors since lower frequencies have higher occupation numbers. The occupation numbers at the lowest frequencies of our systems tend to be between 10 and 20, and thus our Gibbs energy would have a maximal error of ~0.06 eV due to that frequency's error. Depending on the number of low modes and the degree of error cancelation between systems, this could result in errors in excess of the usual DFT error. However, if we ignore the calculated high CO stretch frequencies when determining our approximate frequency error (which have occupation numbers near zero, anyway), the error drops to about 8 cm^{-1} or ~0.001 eV and even with a significant number of low modes and only modest error cancelation would give errors in the calculated Gibbs energy that are within DFT error.

APPENDIX A REFERENCES

[1] D. A. McQuarrie, *Statistical thermodynamics*. 1st ed. Vol. 0. 1973: HarperCollins.

[2] K. Reuter, M. Scheffler, Phys. Rev. B 65 (2001).

[3] M. S. Shell, *Thermodynamics and Statistical Mechanics: An Integrated Approach*. 1st ed. 2015: Cambridge University Press.

[4] R. B. Getman, Y. Xu, W. F. Schneider, J. Phys. Chem. C 112 (2008) 9559-9572.

[5] A. Banerjee, A. P. van Bavel, H. P. C. E. Kuipers, M. Saeys, ACS Catal. 5 (2015) 4756-4760.

[6] G. T. K. K. Gunasooriya, A. P. van Bavel, H. P. C. E. Kuipers, M. Saeys, Surf. Sci. 642 (2015) L6-L10.

[7] F. Che, S. Ha, J.-S. McEwen, Applied Catalysis B: Environmental 195 (2016) 77-89.

[8] J.-S. McEwen, T. Anggara, W. F. Schneider, V. F. Kispersky, J. T. Miller, W. N. Delgass, F. H. Ribeiro, Catal. Today 184 (2012) 129-144.

[9] G. Collinge, Y. Xiang, R. Barbosa, J.-S. McEwen, N. Kruse, Surf. Sci. 648 (2016) 74-83.

APPENDIX B

I. DFT Parameters and Theory

All calculations were carried out in the Vienna *Ab-initio* Simulation Package (VASP)[1-3], which uses Projector Augmented Wave (PAW) pseudopotentials[4] and a plane wave basis set to solve the Kohn-Sham Equations[5, 6]. For all the calculations except those reported in Section III of Appendix B we utilized the Perdew-Burke-Enzerhof (PBE) Generalized Gradient Approximation (GGA) functional[7, 8] to describe the exchange-correlation. For the calculations reported in Section III, we used the vdW-DF functional[9, 10]. We determined that a plane wave basis set expanded to a 450 eV energy cutoff[11] and the first Brillouin zone of the *p*(1×3) Cu/Co(755) supercell sampled with a Monkhorst-Pack k-point mesh of 3 × 4 × 1 (see Figure B1) accurately described the underlying energetics of the adsorbate system. Spin polarization was incorporated to account for the presence of ferromagnetic Co and dipole corrections were included to eliminate the fictitious dipole created by the asymmetric finite metal slab. Self-consistent field (SCF) and geometric optimization criteria were set at 1.0×10^{-4} eV and 3.0×10^{-2} eV/Å, respectively. For minimum energy pathway calculations and finding transition states, the nudged elastic band (NEB)[12, 13] and climbing image nudged elastic band (CINEB)[14] method was used, and geometric optimization criteria was decreased (made more stringent) to 1.0×10^{-5} eV and 1.0×10^{-2} eV/Å, respectively. For vibrational mode calculations, SCF optimization criteria of 1.0×10^{-6} eV and central finite differences with atomic displacements of ±0.01 Å were used. To eliminate z-direction interaction between slabs, a ~15 Å vacuum layer was imposed. The computed lattice constant for the fcc phase of Co is given in Table B1 and compared to its values in the hcp phase. Surface free energies changes, $\Delta\gamma(T, p)$, and Gibbs free energies, $G(T, p_{CO})$, were calculated using the statistical mechanical procedures and equations outlined previously,[15] where the harmonic oscillator approximation is used to determine the surface species' contribution to each system's free energy and all relaxed metal atoms are included in the evaluation of the Hessian. If a low-lying real or imaginary mode is shown to correspond to a surface translation or rotation, the appropriate free translator/rotor partition function is used instead, otherwise the structure is reoptimized to remove all imaginary modes. Equilibrium constants were calculated as

$$K_{eq}(T, p_{CO}) = e^{\frac{-\Delta G(T, p_{CO})}{k_B T}} \tag{B1}$$

where $K_{eq}(T, p_{CO})$ is the equilibrium constant at absolute temperature, T, and CO partial pressure, p_{CO}; k_B is Boltzmann's constant. The DFT energy and adsorption energy are defined as

$$E^{DFT} = E_{adsorbates/slab} - E_{slab} - N_{adsorbates} E^{gas}_{adsorbate} \tag{B2}$$

$$E_{ads} = \frac{E^{DFT}}{N_{adsorbates}} \tag{B3}$$

where $E_{adsorbates/slab}$, E_{slab}, and $E^{gas}_{adsorbate}$ are the DFT-calculated energies of the total adsorbed system (on a per supercell basis), the clean slab (on a per supercell basis), and the gas-phase adsorbate (on a per molecule basis), respectively. $N_{adsorbates}$ is the number of adsorbates per supercell.

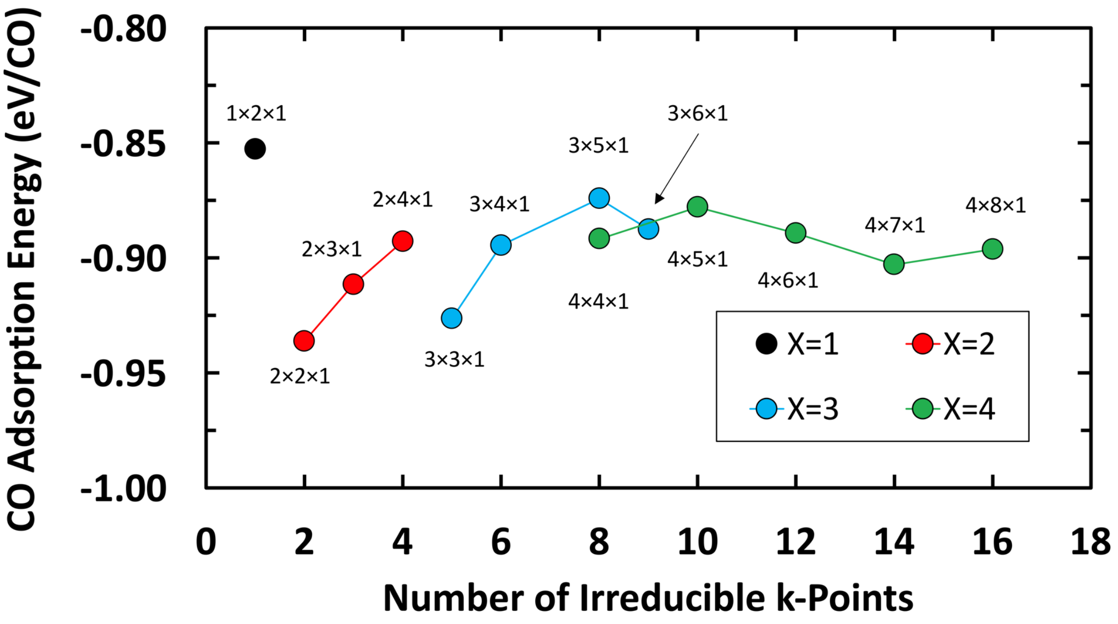


Figure B1. K-point mesh (X×Y×1) convergence test performed for CO adsorbed on the Cu covered terrace of the p(1×3) Cu/Co(755) surface.

The lengths of the surface (reciprocal space) vectors in the 1st Brillouin zone suggest that the k-point mesh should be larger in the "Y" direction than in the "X" direction by a factor of no more than 2. Thus, for each "X" value, "Y" is varied from X to 2X and the adsorption energy is plotted as a function of the number of irreducible k-points and the "X" value. When "X" is increased from a value of 3 to 4, the adsorption energy does not change significantly from the final value on the X=3 (blue) line (9 irreducible k-points), suggesting that the "X" value is sufficiently converged at a value of 3. All remaining values are less than ~20 meV/CO from the value at 6 irreducible k-points (k-point mesh of 3×4×1) on the blue line, which being roughly 1/10th the error of GGA-DFT, we consider sufficiently converged for this study. The value at 6 irreducible k-points is also within 2 meV/CO of the highest k-point sampling tested at 16 irreducible k-points, further reassuring that it is well converged.

Table B1. Lattice constants (in Å) calculated for cobalt in both its fcc and hcp phases using both the PBE and revPBE + vdW-DF functionals.

Co phase		PBE	revPBE + vdW-DF
hcp	a	2.494	2.526
	c/a	1.618	1.612
fcc	a	3.457	3.501

II. Model Justification

Here, we choose to work with the Cu/Co(755) surface because it is comprised of a 6 atom long (111) terrace with a single atom high (100) step (see Figure B2). This can be regarded as a good approximation of a step defect in a (111) surface. Ideally, a step surface with an even longer terrace would be used but performing DFT calculations on supercells much greater in size would become computationally prohibitive.

We have previously worked with the hcp phase of Co[11, 15], but we wish to move toward models that are more congruent with experimental work, where particle sizes or synthesis conditions can be small enough to induce the transition of Co to its fcc phase[16-19]. Due to the similarity in the surface electronic structure of these two phases, we do not anticipate any difficulty in transferring the information gained from work on hcp structures to fcc structures.

In our previous work[11, 15], we concerned ourselves with a catalyst surface that had been sufficiently equilibrated with its reaction environment. That investigation lead to the conclusion that a CoCu catalyst would present a predominantly Cu surface, with adsorbed CO coordinated to a low concentration of disperse surface Co. This is admittedly simplified—the larger backdrop of potential early-reaction species will affect this picture to some degree (which we note as relevant to future work). However, this picture of the surface is likely a good first-order approximation since it has been indicated that adsorbed CO, undissociated, is predominant on the surface[20]. We also showed that surface geminal Co di- and tricarbonyls occupy up to

50% of the step sites, with "clean" Cu atoms acting as spacers between adjacent carbonyl structures. In the early transient phase of the reaction, we argue that these step site Co carbonyls will form long before any of the CO adsorbed on the terrace sites has had time to equilibrate with the underlying Co (as we have predicted should eventually occur). We base this on two main arguments: (1) our calculations show that these step site carbonyls are lower in free energy than the simultaneously predicted CO-adsorbed terrace structures, meaning the thermodynamic driving force will generally favor step site CO over terrace CO and (2), the kinetic barriers of Co bulk diffusion should be significantly lower for diffusion from the bulk to steps sites than for diffusion from the bulk to terrace sites, which is simply due to the lower number of metal bonds that must be distorted to accomplish such a diffusion process. This suggests terrace CO/Co equilibration is slowed compared to that of step CO/Co. Therefore, the terrace is modeled such that any CO is adsorbed to the Cu without inducing surface Co antisegregation in our models of CO/Cu/Co(755).

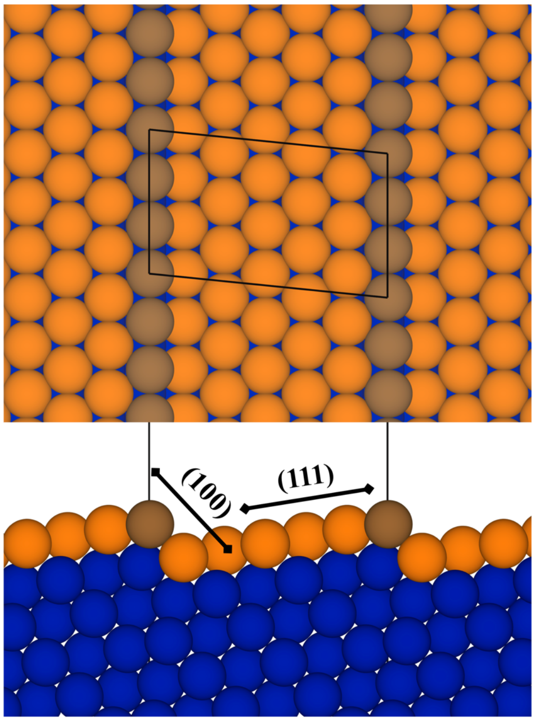


Figure B2. Model of the Cu/Co(755) system used here. The p(1×3) supercell is shown outlined in black. The blue spheres are Co, the orange spheres are Cu, and the brown spheres are Cu at the step sites.

With regard to the rupturing favorability shown in Figure B2 of the main text, it is important to note that our model has only one layer of Cu on the surface and so when dissolution occurs it necessarily exposes a Co atom. If more layers of Cu were to make up the Cu shell, then, if we neglect thermal entropy-driven mixing, no Co atoms would be exposed after rupturing. However, as the temperature is increased, such thermal entropy-driven effects will become more significant. Further, geminal Co carbonyl formation is still predicted even when no bare Co is exposed at the surface[15]. This implies that our proposed dissolution scheme is still applicable to even higher Cu loadings than what we explicitly model here and not necessarily dependent on the exposure of surface Co during the rupturing process.

III. PBE vs. vdW-DF

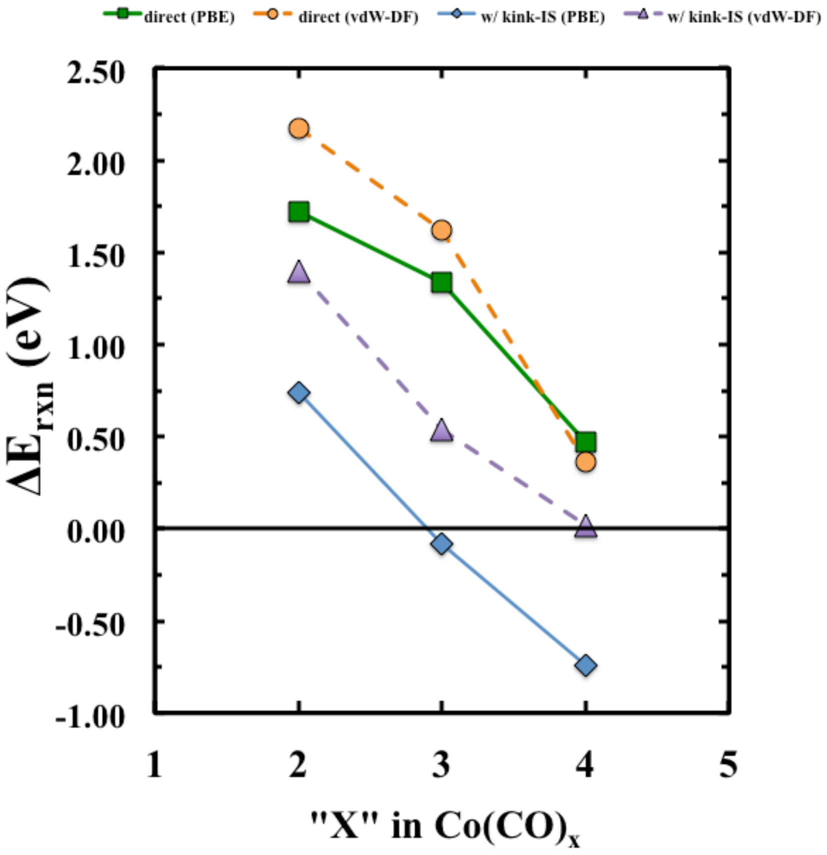


Figure B3. Energy differences, defined as E(ruptured) – E(unruptured), for systems A-F found in Figure 5.2 of chapter five. Here, we present the results of the PBE functional as well as those found when using the vdW-DF functional. For systems D-F, where the kink-IS is formed in the FS of each process, there appears to be an additional ~0.5 eV cost associated with rupturing when using the vdW-DF functional.

Table B2. DFT energy (ΔE^{DFT}), standard Gibbs free energy (ΔG^0) at 573K, and average DFT CO adsorption energy (ΔE_{ads}) of each structure, A-F, presented in Figure 5.1 of chapter five as calculated with both the PBE and vdW-DF functionals. Units are in eV, eV, and eV/CO, respectively. Here, the vibrational component of the thermal correction added to the DFT energy to arrive at the standard Gibbs free energy has been calculated using the PBE functional. This makes the ΔG^0 for the vdW-DF case approximate, though it is not expected that the vdW-DF functional would change the calculated vibrational modes enough to make up for the differences seen herein.

	PBE			vdW-DF		
Structure	ΔE^{DFT}	$\Delta G°(573K)$	ΔE_{ads}	ΔE^{DFT}	$\Delta G°(573K)$*	ΔE_{ads}
(A) IS	-2.864	-0.773	-1.432	-6.195	-4.105	-3.098
(A) FS	-1.157	1.008	-0.578	-4.030	-1.865	-2.015
(B) IS	-4.182	-0.934	-1.394	-9.116	-5.868	-3.039
(B) FS	-2.847	0.232	-0.949	-7.499	-4.420	-2.500
(C) IS	-4.234	0.155	-1.058	-10.548	-6.160	-2.637
(C) FS	-3.759	0.577	-0.940	-10.186	-5.850	-2.547
(D) IS	-4.634	-0.235	-1.158	-11.531	-7.132	-2.883
(D) FS	-3.898	0.628	-0.975	-10.129	-5.603	-2.532
(E) IS	-6.825	-0.555	-1.137	-17.195	-10.925	-2.866
(E) FS	-6.909	-0.639	-1.151	-16.659	-10.390	-2.777
(F) IS	-7.323	1.350	-0.915	-21.101	-12.428	-2.638
(F) FS	-8.087	0.658	-1.011	-21.076	-12.332	-2.635

*approximated from vibrational modes calculated with the PBE functional

Calculation with the vdW-DF functional shows a massive increase in the average adsorption strength of the CO in each system. Using the PBE functional, the CO adsorption strength is within the range expected for CO given the values obtained on either Co(111) or Cu(111) (~1.7 eV/CO[21] and ~0.8 eV/CO[22], respectively). The discrepancy was concerning enough that the PBE functional was deemed the more conservative choice. However, it can be noted that if one uses the vdW-DF functional, the tetracarbonyl rupturing process seen in Figure 5.2F is predicted to be most favorable across all pressure ranges of interest at 573 K. Pressure effects are then negligible: at no reasonable pressure does the next-most favorable process (tricarbonyl rupturing as in Figure 5.2E) become the most favorable process. Thus, in the end, the story does not change substantially despite the choice of functional.

IV. CO Adsorption Energy Calculations

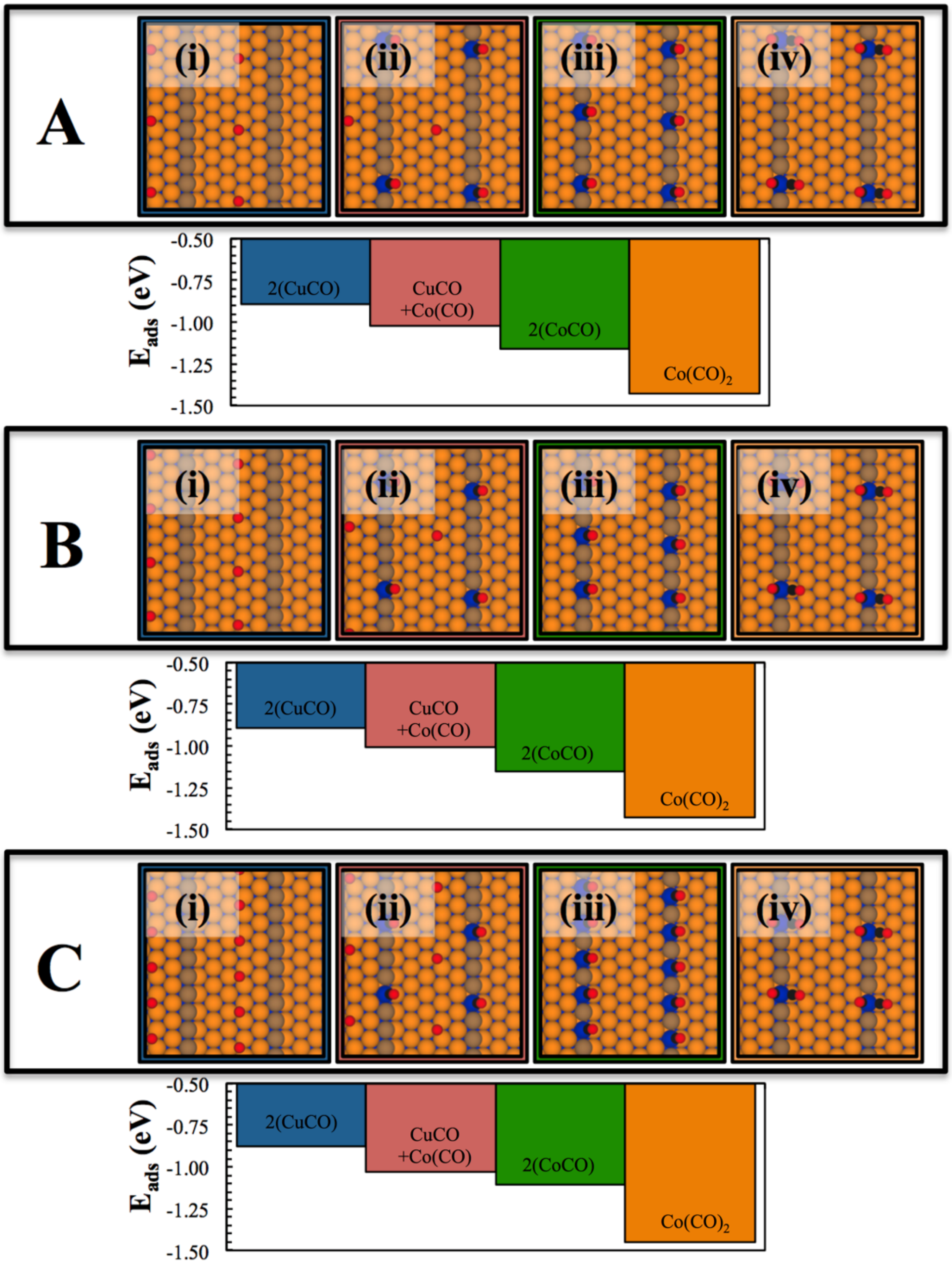


Figure B4. PBE DFT adsorption energy calculations for three step-site-equivalent coverages: (A) 0.050 ML, (B) 0.067 ML, and (C) 0.100 ML. Each coverage; A, B, and C; contains two adsorbed CO molecules per p(1×8), p(1x6), and p(1x4) Cu/Co(755) supercell, respectively. Within each coverage, we compare four systems where, from left two right, (i) both CO are on Cu terrace sites, (ii) one CO is on a Cu terrace site and one is on a Co step site, (iii) both CO are on Co step sites, and (iv) both CO are on the same Co step site in the form of a dicarbonyl. In all three system, the energetic trends are the same and the step site Co dicarbonyl is predicted to be significantly more favorable than the other adsorption configurations.

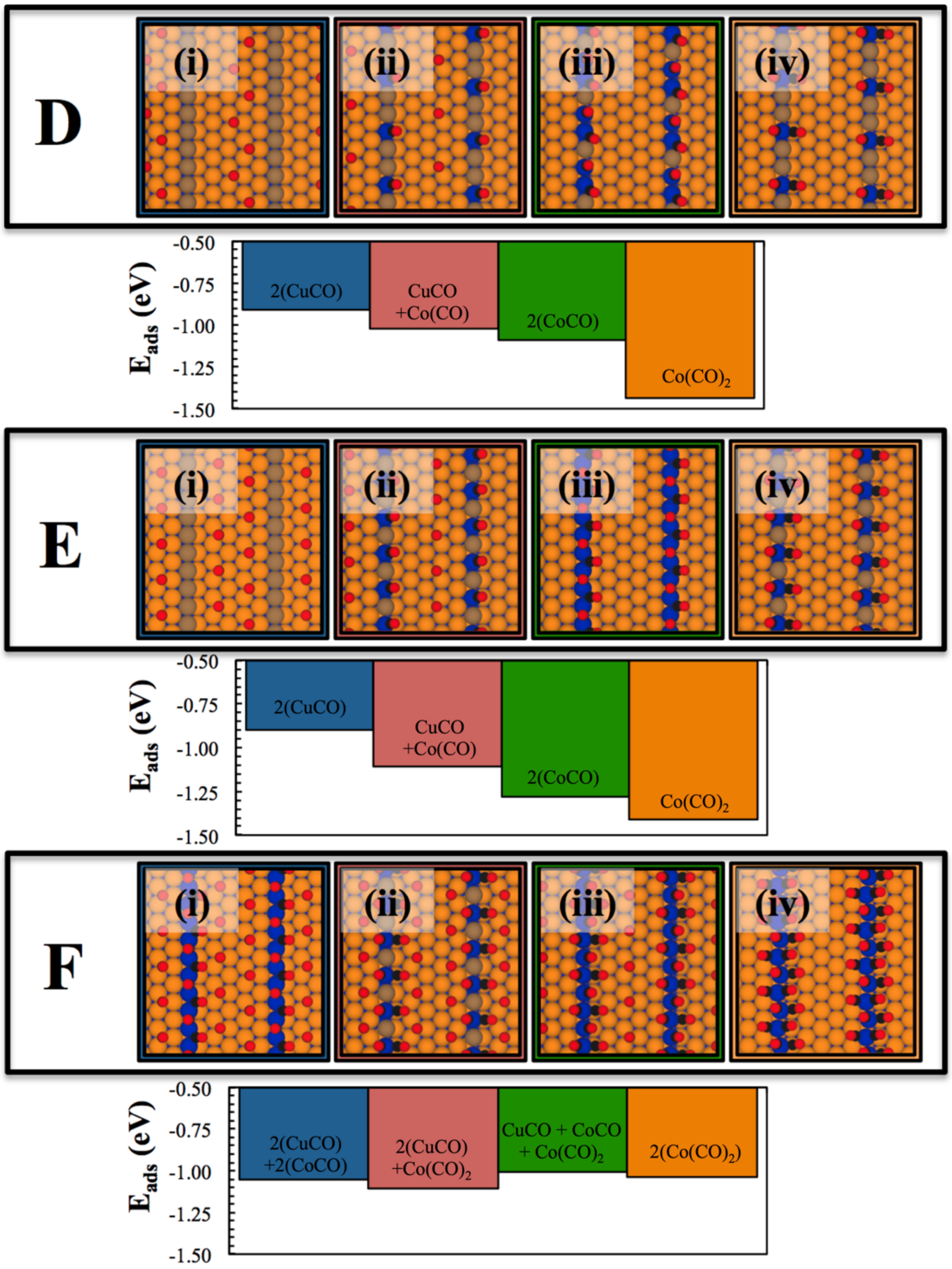


Figure B5. PBE DFT adsorption energy calculations for three coverages: (D) 0.133 ML, (E) 0.200 ML, and (F) 0.400 ML. D and E contain two adsorbed CO molecules per p(1×3), p(1x2) Cu/Co(755) supercell, respectively; while F contains four CO molecules per p(1x2) Cu/Co(755) supercell. Within D and E, we compare four systems where, from left two right, (i) both CO are on Cu terrace sites, (ii) one CO is on a Cu terrace site and one is on a Co step site, (iii) both CO are on Co step sites, and (iv) both CO are on the same Co step site in the form of a dicarbonyl. In F the step site coverage is kept at 1.00 ML and the remaining two CO add to the step in a similar fashion as in the previous two systems. Systems D and E have the same energetic trends scene in Figure B4, and only at 2.00 ML do we see a change in this trend.

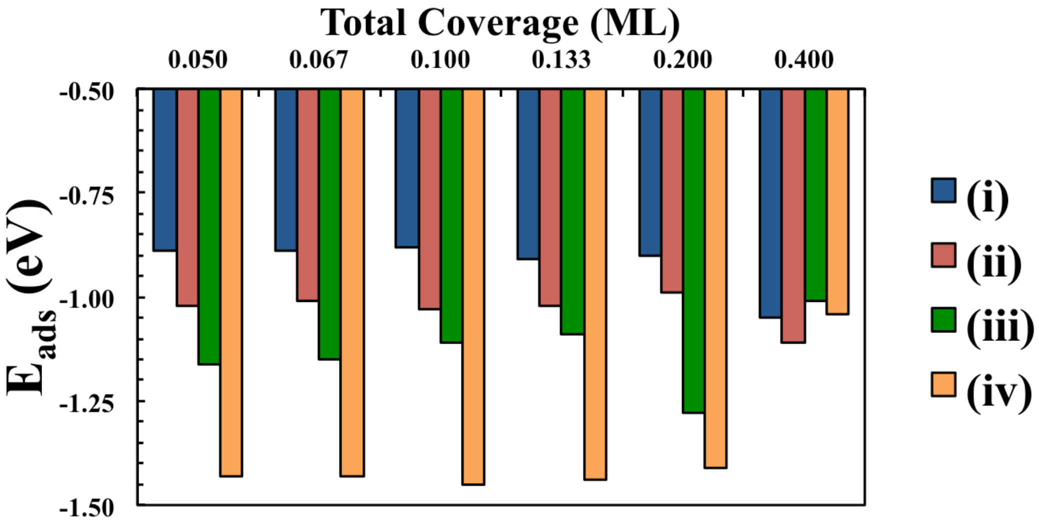


Figure B6. Summary of all data presented in Figures B4-B5.

V. Potentially IR-Active Vibrational Modes

Table B3. Normal modes of the relevant species in this paper extracted from the direct rupturing structures' vibrational mode calculations. Unless otherwise stated in the description, all modes are "stretch" modes. All wavenumber values have been rounded to the nearest 10 cm^{-1}. The notation in the normal mode description shows the number of species involved, in what ratio, as well as the type of vibration. e.g. "2/1-CO antisymmetric" means that 2 CO vibrate symmetrically, while 1 CO (the third CO) vibrates antisymmetric to the other two.

Species	Normal Mode Description	Calculated Wavenumbers (cm^{-1})
Step-Bound $Co(CO)_2$	2-CO symmetric	1970
	2-CO antisymmetric	1940
	1-(CO-Co)	550
Adsorbed/Free FS $Co(CO)_2$	2-CO symmetric	1990
	2-CO antisymmetric	1950
	1-(CO-Co)	520—530
Step-Bound $Co(CO)_3$	3-CO symmetric	1940
	2-CO antisymmetric	1890
	2/1-CO antisymmetric	1870
	1-CO	500
Adsorbed/Free $Co(CO)_3$	3-CO symmetric	2000
	2-CO antisymmetric	1970
	2/1-CO antisymmetric	1920
	1-CO (coupled to Co translation and 2-(Co-C-O) bending)	510
Step-Bound $Co(CO)_4$	top-CO	2050
	top+1-CO antisymmetric	1880
	2-CO antisymmetric	1850
	2/1-CO antisymmetric	1790
	Various coupled (Co-C-O) stretching and bending	370—530
Adsorbed/Free $Co(CO)_4$	top-CO	2050
	3/top-CO antisymmetric	1920
	2/1-CO antisymmetric	1860
	2-CO antisymmetric	1850
	Various coupled (Co-C-O) stretching and bending	400—560

APPENDIX B REFERENCES

[1] J. Tang, L. Deng, S. Xiao, H. Deng, X. Zhang, W. Hu, J. Phys. Chem. C (2015).

[2] G. Kresse, J. Hafner, Phys. Rev. B 47 (1993) 558.

[3] G. Kresse, J. Furthmüller, Comput. Mater. Sci. 6 (1996) 15-50.

[4] G. Zvejnieks, A. Ibenskas, E. E. Tornau, J. Alloys Compd. 649 (2015) 313-319.

[5] B. C. Han, A. Van der Ven, G. Ceder, B. Hwang, Phys. Rev. B 72 (2005) 205409.

[6] W. Kohn, L. J. Sham, Phys. Rev. 140 (1965) A1133-A1138.

[7] A. Lopes, G. Tréglia, C. Mottet, B. Legrand, Phys. Rev. B 91 (2015) 035407.

[8] J. P. Perdew, K. Burke, M. Ernzerhof, Phys. Rev. Lett. 77 (1996) 3865-3868.

[9] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, B. I. Lundqvist, Phys. Rev. Lett. 92 (2004) 246401.

[10] J. Klimeš, D. R. Bowler, A. Michaelides, Phys. Rev. B 83 (2011) 195131.

[11] G. Collinge, Y. Xiang, R. Barbosa, J.-S. McEwen, N. Kruse, Surf. Sci. 648 (2016) 74-83.

[12] H. Jonsson, G. Mills, K. W. Jacobsen, Nudged elastic band method for finding minimum energy paths of transitions: Classical and Quantum Dynamics in Condensed Phase Simulations, WORLD SCIENTIFIC, 1998, pp. 385-404.

[13] G. Henkelman, H. Jónsson, The Journal of chemical physics 111 (1999) 7010-7022.

[14] G. Henkelman, B. P. Uberuaga, H. Jónsson, The Journal of Chemical Physics 113 (2000) 9901-9904.

[15] G. Collinge, N. Kruse, J.-S. McEwen, J. Phys. Chem. C 121 (2017) 2181-2191.

[16] O. Kitakami, H. Sato, Y. Shimada, F. Sato, M. Tanaka, Phys. Rev. B 56 (1997) 13849-13854.

[17] D. I. Enache, B. Rebours, M. Roy-Auberger, R. Revel, J. Catal. 205 (2002) 346-353.

[18] M. Sadeqzadeh, H. Karaca, O. V. Safonova, P. Fongarland, S. Chambrey, P. Roussel, A. Griboval-Constant, M. Lacroix, D. Curulla-Ferré, F. Luck, A. Y. Khodakov, Catal. Today 164 (2011) 62-67.

[19] M. K. Gnanamani, G. Jacobs, W. D. Shafer, B. H. Davis, Catal. Today 215 (2013) 13-17.

[20] B. T. Loveless, C. Buda, M. Neurock, E. Iglesia, J. Am. Chem. Soc. 135 (2013) 6107-21.

[21] Š. Pick, Surf. Sci. 601 (2007) 5571-5575.

[22] M. Gajdoš, J. Hafner, Surf. Sci. 590 (2005) 117-126.

APPENDIX C

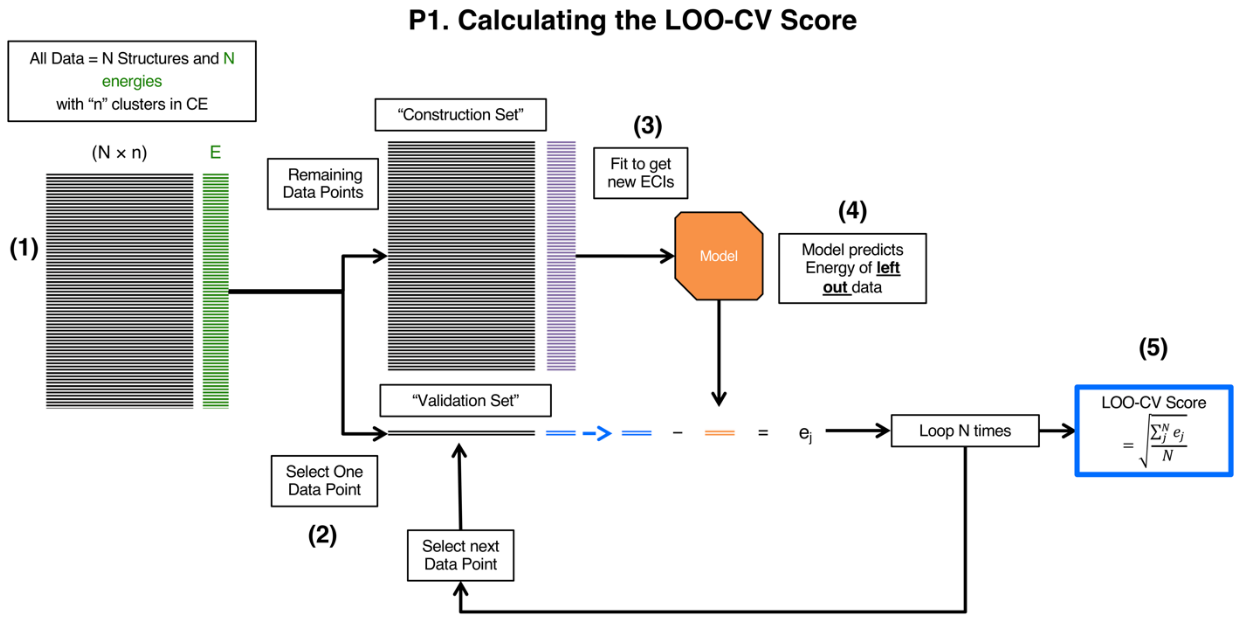


Figure C1. Algorithm for calculation of the LOO-CV score.

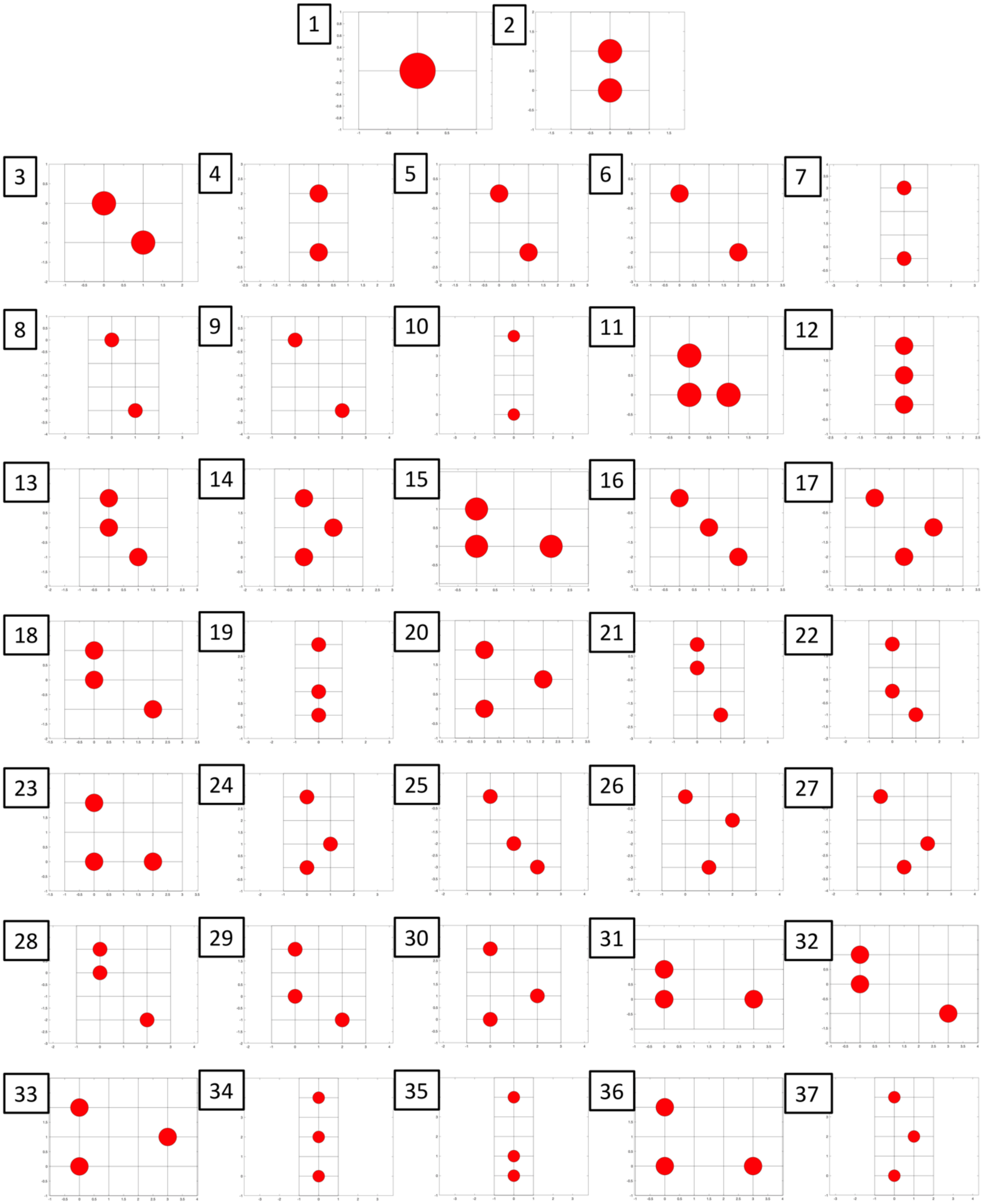


Figure C2. Potential clusters found for the O/Fe(100) system with a cutoff radius of 4 lattice constants for the 2- and 3- body clusters and 3 lattice constants for 4-body clusters. The numbers at the upper left-hand corner of each cluster is its cluster ID number. Additional potential clusters are shown in Figure C3.

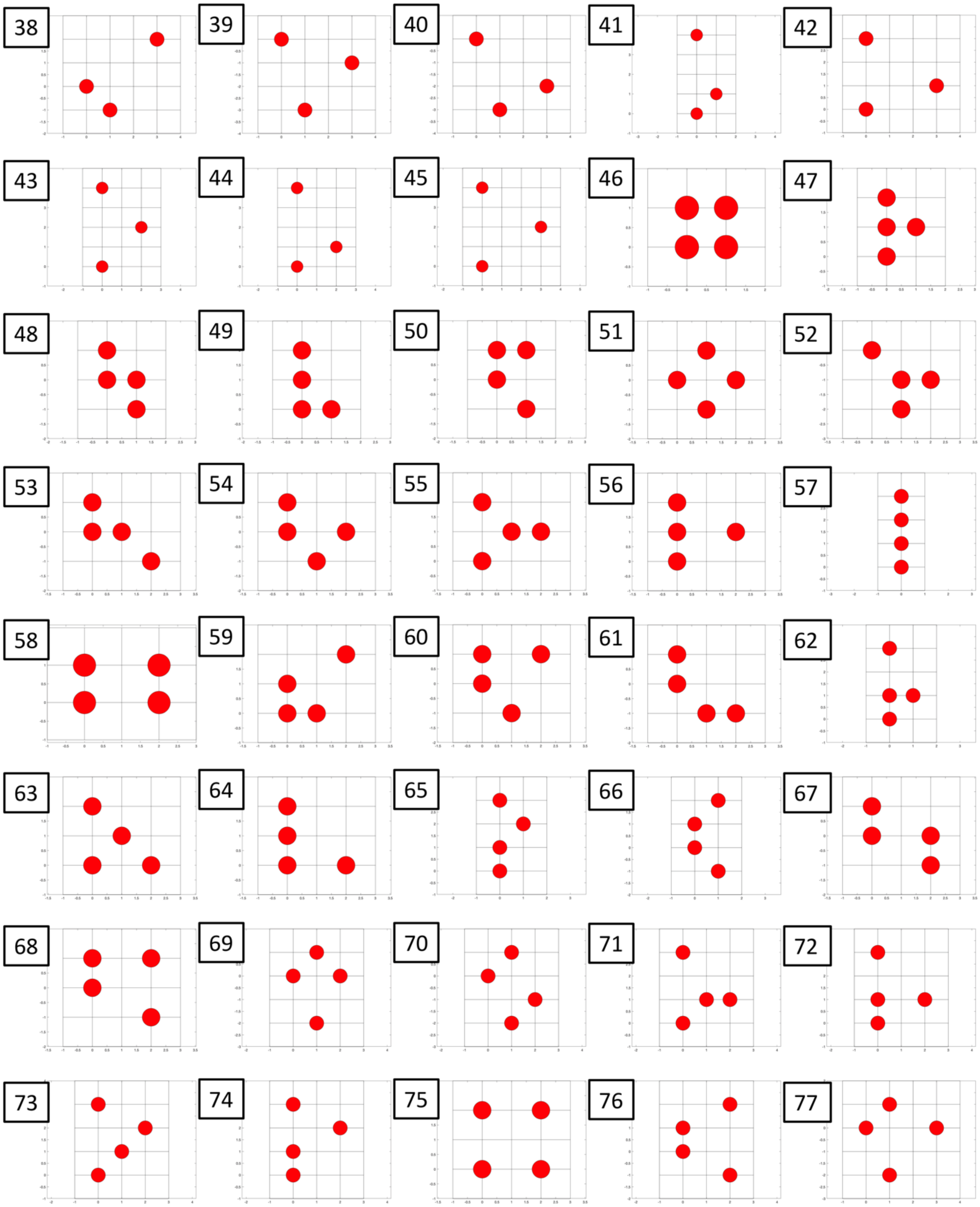
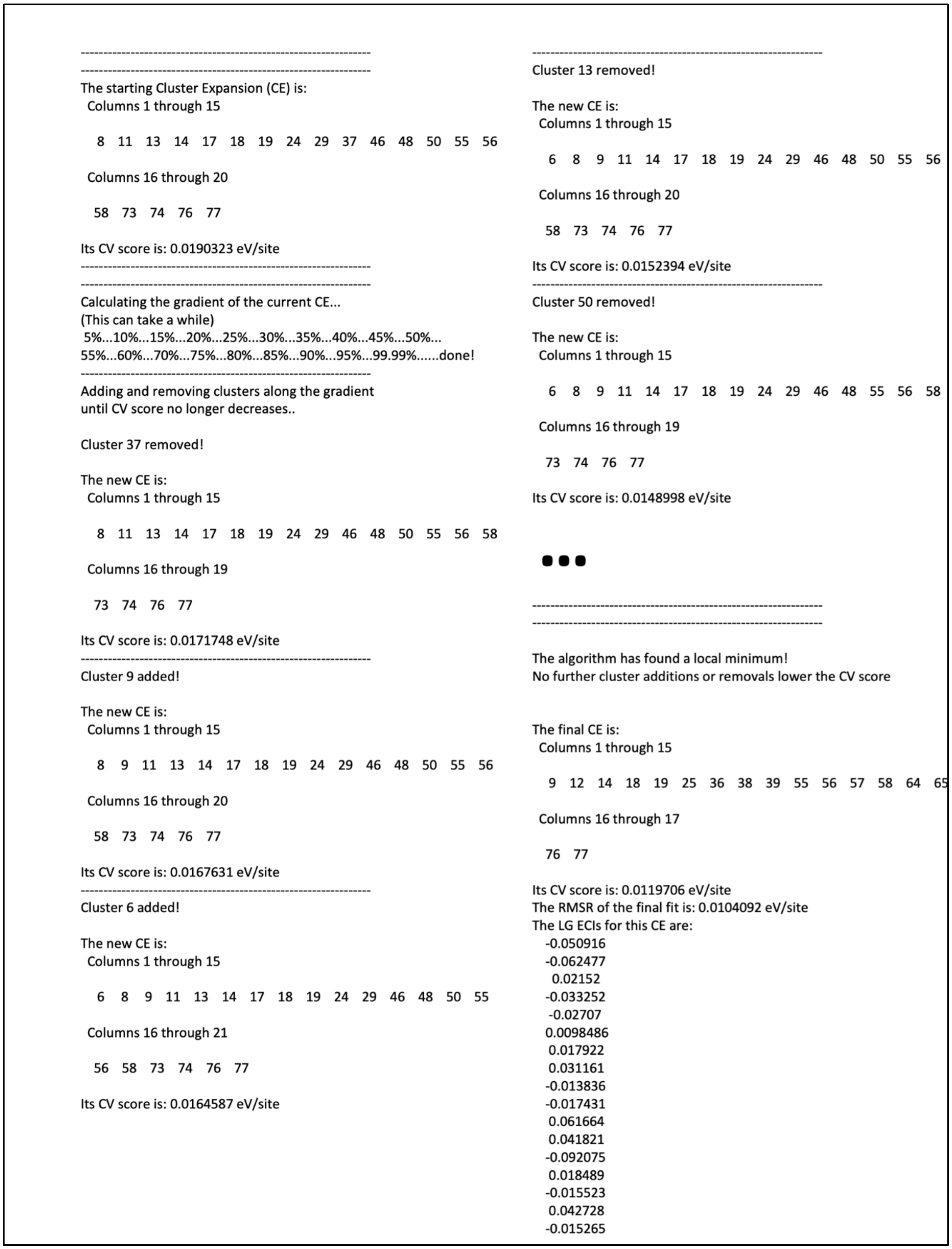


Figure C3. Potential clusters found for the O/Fe(100) system with a cutoff radius of 4 lattice constants for the 2- and 3- body clusters and 3 lattice constants for 4-body clusters. The numbers at the upper left-hand corner of each cluster is its cluster ID number. Additional potential clusters are shown in Figure C2.



```
---------------------------------------------------------------
---------------------------------------------------------------
The starting Cluster Expansion (CE) is:
  Columns 1 through 15

    8   11   13   14   17   18   19   24   29   37   46   48   50   55   56

  Columns 16 through 20

   58   73   74   76   77

Its CV score is: 0.0190323 eV/site
---------------------------------------------------------------
---------------------------------------------------------------
Calculating the gradient of the current CE...
(This can take a while)
 5%...10%...15%...20%...25%...30%...35%...40%...45%...50%...
55%...60%...70%...75%...80%...85%...90%...95%...99.99%......done!
---------------------------------------------------------------
Adding and removing clusters along the gradient
until CV score no longer decreases..

Cluster 37 removed!

The new CE is:
  Columns 1 through 15

    8   11   13   14   17   18   19   24   29   46   48   50   55   56   58

  Columns 16 through 19

   73   74   76   77

Its CV score is: 0.0171748 eV/site
---------------------------------------------------------------
Cluster 9 added!

The new CE is:
  Columns 1 through 15

    8    9   11   13   14   17   18   19   24   29   46   48   50   55   56

  Columns 16 through 20

   58   73   74   76   77

Its CV score is: 0.0167631 eV/site
---------------------------------------------------------------
Cluster 6 added!

The new CE is:
  Columns 1 through 15

    6    8    9   11   13   14   17   18   19   24   29   46   48   50   55

  Columns 16 through 21

   56   58   73   74   76   77

Its CV score is: 0.0164587 eV/site
```

```
---------------------------------------------------------------
Cluster 13 removed!

The new CE is:
  Columns 1 through 15

    6    8    9   11   14   17   18   19   24   29   46   48   50   55   56

  Columns 16 through 20

   58   73   74   76   77

Its CV score is: 0.0152394 eV/site
---------------------------------------------------------------
Cluster 50 removed!

The new CE is:
  Columns 1 through 15

    6    8    9   11   14   17   18   19   24   29   46   48   55   56   58

  Columns 16 through 19

   73   74   76   77

Its CV score is: 0.0148998 eV/site


 •••

---------------------------------------------------------------
---------------------------------------------------------------

The algorithm has found a local minimum!
No further cluster additions or removals lower the CV score


The final CE is:
  Columns 1 through 15

    9   12   14   18   19   25   36   38   39   55   56   57   58   64   65

  Columns 16 through 17

   76   77

Its CV score is: 0.0119706 eV/site
The RMSR of the final fit is: 0.0104092 eV/site
The LG ECIs for this CE are:
   -0.050916
   -0.062477
     0.02152
   -0.033252
    -0.02707
   0.0098486
    0.017922
    0.031161
   -0.013836
   -0.017431
    0.061664
    0.041821
   -0.092075
    0.018489
   -0.015523
    0.042728
   -0.015265
```

Figure C4. AMALGM output for "P2" in Figure 6.4 of chapter six.

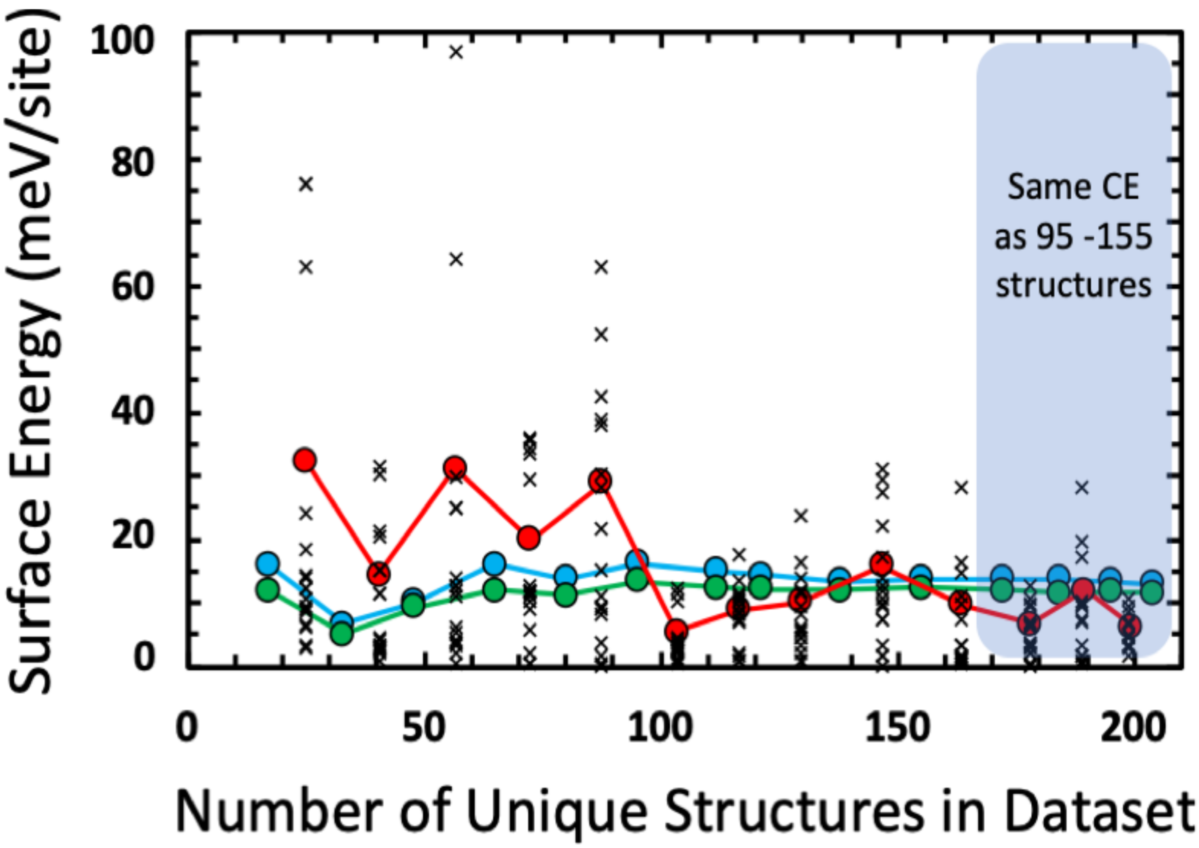


Figure C5. Progression of the LMO-CV score (blue), RMSR (green), external CV score (red), and the absolute prediction errors (black X's) as a function of the number of unique structures in the dataset. Here, instead of allowing the CE to change after 155 structures to the slightly better CE presented in the main text, the prior CE is used instead. This demonstrates how little the quantities tracked change.

APPENDIX D

I. Density Functional Theory (DFT) Calculation Parameters for the Fixed-O/Fe(100) Systems

DFT calculations were performed using the Vienna *Ab Initio* Simulation Package (VASP) with the core electrons modeled using the projector augmented wave (PAW) method and the valence electrons modeled with a plane wave basis set expanded to a cutoff energy of 400 eV. The exchange-correlation functional used was either the RPBE or optB88-vdW functionals, as denoted in the main text. Methfessl-Paxton smearing (N = 1) with a smearing width of 0.1 eV was used to perform electron smearing. Spin polarization was used to model the magnetization of Fe, and dipole corrections in the $\hat{z}$-direction were applied. The energies and structures used in the "fixed" O/Fe(100) system datasets, i.e. those generated using the Alloy Theoretic Automated Toolkit (ATAT), were from single point calculations where the self-consistent field cycle tolerance was set to 10^{-4} eV. **K**-point grids were generated automatically with the condition of 1200 **k**-points/reciprocal atom with the Gamma distribution. The Fe lattice constants used were 2.868 and 2.825 Å for the RPBE and optB88-vdW calculations, respectively.

The Fe(100) slab was modeled using four atomic layers with the bottom two layers fixed into their bulk positions. The vacuum space above the Fe(100) slab was at least 14 Å. The top two Fe(100) layers were fixed into their bulk $\hat{x}$- and $\hat{y}$-positions, and the oxygen adatoms were fixed into their ideal, 4-fold hollow $\hat{x}$- and $\hat{y}$-positions, as shown in Figure D1. The $\hat{z}$-position for the top two Fe(100) layers and oxygen adatoms were chosen based on ground state optimizations of a p(4x4) Fe(100) supercell, where the surface was clean and with a single oxygen adatom, respectively. The force tolerance for these optimizations was 0.03 eV/Å, and a Gamma point centered **k**-point mesh of (4×4×1) was used.

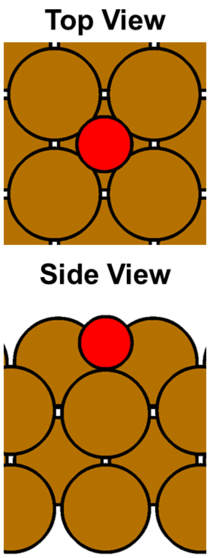


Figure D1. Top and side views of the fixed-O/Fe(100) system. The red and gold spheres represent oxygen and Fe, respectively. The oxygen is adsorbed in the ideal, 4-fold hollow site and the top two Fe(100) layers are fixed in their ideal, bulk $\hat{x}$- and $\hat{y}$-positions. The $\hat{z}$-position of the oxygen is set to that found by ground state optimization of a p(4x4) Fe(100) supercell with a single oxygen adatom. The $\hat{z}$-position of the top two Fe(100) layers are set to that found by ground state optimization of a clean p(4x4) Fe(100) supercell.

II. Unweighted ECI Error Deviations

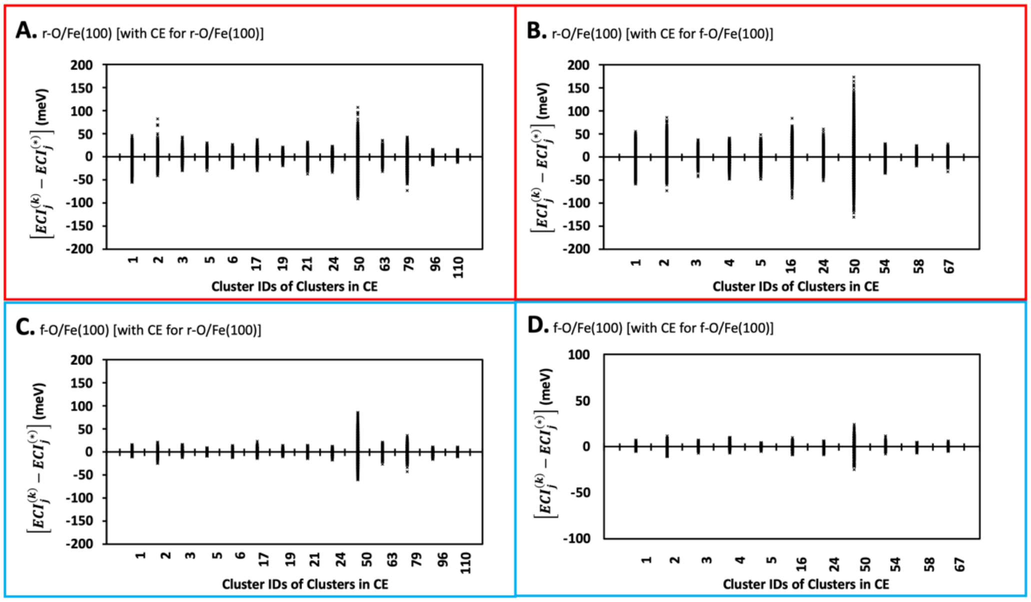


Figure D2. Unweighted ECI error deviations for the systems found in Figure 7.4 of chapter seven.

APPENDIX E

AMALGM MATLAB CODES

INTERACTIONS_GEN.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%% INTERACTIONS_GEN.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Build and find interaction terms in a cluster expansion for adsorbates
% on metal surfaces.

% v3: Hopefully output MC_POSITIONS for use in MC simulations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clearvars
format short g

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%% BEGIN USER INPUTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% User provided "natural" vectors of the metal surface. Be sure to use
% the vectors that create an obtuse angle. For FCC(111) this would be
% [1 ; 0 ; 0] and [-1/2 ; sqrt(3)/2 ; 0] instead of [1 ; 0; 0] and
% [1/2 ; sqrt(3)/2 ; 0]. Technically these are the a, b, and c vectors of
% p(1 x 1) unit cell of your surface. If you intend to use the z
% coordinate position in defining adsorption site locations (only important
% if you have numerous adsorption sites and they aren't all on the same
% plane.)...i.e. you can easily define them as having the same z position),
% you should provide the c vector as it appears in your POSCAR divided by
% the 1st NN distance. Otherwise, you can (and should) leave it as
% [0 ; 0 ; 1]. Mind, whatever length a unit vector within this coordinate
% system is will be the "natural unit" used from here on out.
% NOTE: This is the only place where cartesian coordinates should be
% encountered!

ux = [1 ; 0 ; 0];
uy = [-1/2 ; sqrt(3)/2 ; 0];
uz = [0 ; 0 ; 22.98170/2.47558];

% Alternatively, change infile to "1" and provide a file called
% "NATURAL_COORDINATES.txt" with each ux uy and uz provided as column
% vectors.

infile = 0;

% This file must be written in decimal (floating point) format.
% An example for FCC(11) or HCP(0001) follows:
%
%     1 0 0
%     -0.5 0.866025403784439 0
%     0 0 1
%
%     end of example
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% How many sites within this "natural" cell?

sites_per_cell = 1;

% If there are more than one site, please provide their location within the
% natural unit cell (in "natural coordinates") along with a number to
% signify the type of site. The site "type" can be any positive non-zero
% integer and does not need to be continuous.
% e.g. if you have an FCC(111) surface, there are potentially top sites (1),
% bridge sites (2), fcc hollow sites (3), and hcp hollow sites (4). If
% you want to specify more than one type of adsorbate, you can do that here
% by repeating the same adsorption site, but changing the "type" number
% (i.e. the 4th element)
    Site(:,1) = [0; 0 ; 0 ; 1]; % O site (hollow site)

% If any of the sites are linked, as in through a bond, then identify
% below. THis will simply remove the point EIC (V naught) for the linked
% site

linked = [];
```

```
% User specified overall maximum N-body clusters to include (even if the
% max is different for different site types, still specify the max of all
% types here)

maxNbody= 5;

%%%%%%%%%%%%%%%%%%% BOOK KEEPING, PLEASE DON'T TOUCH %%%%%%%%%%%%%%%%%%%%%%
        vecbody = [ones(1,maxNbody)*maxNbody maxNbody];
        Rmax = zeros(vecbody);
%%%%%%%%%%%%%%%%%%%%%%%%%% OKAY DONE, CONTINUE %%%%%%%%%%%%%%%%%%%%%%%%%%%%

% User defined maximum interaction distances "Rmax" in units of
% natural unit vectors. Each matrix (e.g. Rmax(:,:,2)) corresponds to a
% n-body interaction (e.g. 2 body interaction). Each row and column
% correspond to each site type (so if there are 3 site TYPES, these
% will be 3 x 3 symmetric matrices. Each element corresponds to
% interactions between the types designated by the row and column. For
% example, if there are 3 site types (1, 2, and 3), Rmax(:,:,3) contains
% the maximum site distances (or "cluster sizes") for 3-body interactions
% and Rmax(1,3,3) is the maximum 3-body site distance between sites
% 1 and 3 (corresponding to the sites entered above). If you want (say) 4
% body interactions between site 1 and itself (Rmax(1,1,4)) but not between
% site 1 and 2, just enter "0" for that entry (i.e. Rmax(1,2,4) = 0).

Rmax(1,1,:,:,:,2) = 5;
Rmax(1,1,1,:,:,3) = 3.7;
Rmax(1,1,1,1,:,4) = 3.7;
Rmax(1,1,1,1,1,5) = 3;

% User defined minimum interaction distance "Rmin" in units of
% natural unit vectors. This is the same for all types of interactions.

Rmin = 0.001;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%& END USER INPUTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic

% manipulate/fill in Rmax array
for ii=2:maxNbody
    Ncombs = nmultichoosek(unique(Site(4,:)),ii);
    if size(Ncombs,1) == 1
        Ncombs = ones(1,ii)*unique(Site(4,:));
    end
    sz_Ncombs = size(Ncombs,1);
    RtoUse = zeros(sz_Ncombs,1);
    for jj = 1:sz_Ncombs
        Nperms = unique(perms(Ncombs(jj,:)),'rows');
        sz_Nperms = size(Nperms,1);
        for kk = 1:sz_Nperms
            vecp = [Nperms(kk,:) ones(1,maxNbody-ii) ii];
            vecind = num2cell(vecp);
            Rvecind(jj,kk) = Rmax(sub2ind(size(Rmax),vecind{:})); %#ok<SAGROW>
        end
        RtoUse(jj) = max(Rvecind(jj,:));
    end
    if any(RtoUse == 0)
        whichjj = find(RtoUse == 0);
        uniqtype = unique(Ncombs(whichjj,:));
        for jj = 1:sz_Ncombs
            if all(unique(Ncombs(jj,:)) == uniqtype)
                matchjj = jj;
                break
            end
        end
        RtoUse(whichjj) = RtoUse(matchjj);
    end
    for jj = 1:sz_Ncombs
        Nperms = unique(perms(Ncombs(jj,:)),'rows');
        sz_Nperms = size(Nperms,1);
        for kk = 1:sz_Nperms
            vecp = [Nperms(kk,:) ones(1,maxNbody-ii) ii];
            vecind = num2cell(vecp);
            Rmax(sub2ind(size(Rmax),vecind{:})) = RtoUse(jj);
        end
    end
end
Rmax = Rmax.^2;
Rmin = Rmin.^2;

%if size(Rmax,2) ~= max(max(Nbody)) - 1
    %error('Error. Number of defined maximum distances per n-body interaction inconsistent with the number of n-body
interactions specified.');
```

```
%end
fprintf('----------------------------------------------------------------\n')
fprintf('----------------------------------------------------------------\n')
fprintf('\nWorking...\n\n')
fprintf('----------------------------------------------------------------\n\n')

% Get the natural coordinate system from file "NATURAL_COORDINATES.txt" if
% infile flag has been turned on
if infile == 1
    fID = fopen('NATURAL_COORDINATES.txt');
    tline = fgetl(fID);
    ux = cell2mat(textscan(tline, '%f'));
    tline = fgetl(fID);
    uy = cell2mat(textscan(tline, '%f'));
    tline = fgetl(fID);
    uz = cell2mat(textscan(tline, '%f'));
    fclose(fID);
end

% Determine the "norm conserving matrix" for later determination of
% distances
natcoor = [ux uy uz];
normR = natcoor'*natcoor;

% determine the max X and Y values needed to reach the maximum Rmax value
% specified
maxRmax = max(Rmax(:));
unitX = [1;0;0];
unitY = [0;1;0];
lengthX = unitX'*normR*unitX;
lengthY = unitY'*normR*unitY;
maxX = ceil(sqrt(maxRmax)/lengthX);
maxY = ceil(sqrt(maxRmax)/lengthY);
% Total number of sites and bodies
Site = Site(:,Site(4,:)~=0);        % get rid of Sites where the "type" is 0
sMax = size(unique(Site(4,:)),2);
nMax = maxNbody;

% Total number of body-to-body pair distances
numRs = nchoosek(nMax,2);
site_max = size(Site,2);
% The "CELL" and "SIG" matrix:
CELL = [1 0 ; 0 1 ];
SIG = [zeros(site_max,2) (1:site_max)'];

num_adsorbates = size(SIG,1);

% Initial Size of sig matrix (dynamic preallocation of memory)
BLOCK = 100;
col_BLOCK = 4;
sig = zeros(BLOCK,col_BLOCK);

% Find SigMaxX and SigMaxY: the dimensions needed to create a surface
% large enough to encompass the maxR distance for each adsorbate
SigMaxX = maxX;
SigMaxY = maxY;

cellvecX = CELL'\[2*SigMaxX ; -2*SigMaxY];
cellvecY = CELL'\[-2*SigMaxX ; 2*SigMaxY];

xMin = cellvecY(1);
xMax = cellvecX(1);

xMin = sign(xMin)*ceil(abs(xMin));
xMax = sign(xMax)*ceil(abs(xMax));

yMin = cellvecX(2);
yMax = cellvecY(2);

yMin = sign(yMin)*ceil(abs(yMin));
yMax = sign(yMax)*ceil(abs(yMax));

xTot = xMax - xMin;
yTot = yMax - yMin;

% Populate "sig" matrix
kk = 1;
SIG_dif = zeros(num_adsorbates,2);
for ii = 1:num_adsorbates
    for xx = 0:xTot
        sigx = xx + xMin;
        for yy = 0:yTot
```

```
            sigy = yy + yMin;
            repeat = SIG(ii,1:2)+(CELL'*[sigx;sigy])';
            SIG_dif = SIG(:,1:2) - repeat;                      % difference between this repeated position and all
adsorbates' positions
            R_check = (diag([SIG_dif zeros(num_adsorbates,1)]*normR*[SIG_dif zeros(num_adsorbates,1)]'));
            if any(R_check <= maxRmax+0.1)
                Shift = Site(1:3,SIG(ii,3))';
                sig_type = Site(4,SIG(ii,3));
                sig(kk,:) = [[repeat 0]+Shift sig_type];

                kk = kk + 1;
            end
            % Add new block of memory to sig matrix if needed
            check = sig(any(sig~=0,2),:);
            if size(check,1)/size(sig,1) > 0.95                 % Only 5% of the current allocation is left
                list_size = size(check,1) + BLOCK;
                sig(kk+1:list_size,:)=0;                        % ...so add a new block of memory
            end
        end
    end
end

sig = sig(any(sig~=0,2),:);
sig = sortrows(sig,1:3);

nTot = size(sig,1);
X = sig(:,1);
Y = sig(:,2);
Z = sig(:,3);
% Find where each site within the unit cell is within the sig matrix
sigPos = zeros(1,num_adsorbates);
sig_coord = zeros(num_adsorbates,4);
for ii = 1:num_adsorbates
    Shift = Site(1:3,SIG(ii,3))';
    sig_type = Site(4,SIG(ii,3));
    check = [[SIG(ii,1:2) 0]+Shift sig_type];
    [sig_coord(ii,:),sigPos(ii)] = intersect(sig,check,'rows');
end

% All combinations of pairs for every adsorbate on the SigMaxX by
% SigMaxY surface
CombTot = sortrows(nchoosek(1:nTot,2),2);
nCombs = size(CombTot,1);
frstn = CombTot(:,1)';
scndn = CombTot(:,2)';

% Find the delta X,Y,Z between each pair of bodies with each site
delX = zeros(1,nCombs);
delY = zeros(1,nCombs);
delZ = zeros(1,nCombs);
delXYZ = zeros(3,nCombs);
R = zeros(1,nCombs);
for ii = 1:nCombs
    delX(ii) = X(scndn(ii))-X(frstn(ii));
    delY(ii) = Y(scndn(ii))-Y(frstn(ii));
    delZ(ii) = Z(scndn(ii))-Z(frstn(ii));
    delXYZ(:,ii) = [delX(ii);delY(ii);delZ(ii)];
    R2 = delXYZ(:,ii)'*normR*delXYZ(:,ii);
    R(ii) = (R2);
end

% Associate R's with each of its associated pair's positions.
% Swaping these positions changes nothing so do that now.
Rassoc = zeros(nTot);
for h = 1:size(CombTot,1)
    Rassoc(CombTot(h,1),CombTot(h,2)) = R(h);
    Rassoc(CombTot(h,2),CombTot(h,1)) = Rassoc(CombTot(h,1),CombTot(h,2));
end

% Now find all pairs that contain the number of the position of each
% supercell adsorbate. Use this to extract all the R's that connect the
% various n-body interactions
clusters = zeros(BLOCK, nMax+numRs);
posclusters = zeros(BLOCK, 4*nMax+numRs);     % create a matrix with enough position for each n-body, each R, and each
one's 3 coordinates
min_clusters = zeros(1,BLOCK);
Int_sz = BLOCK;
int_ptr = 1;

% Create point (V naught) clusters first
principle_sites = unique(Site(4,:));
if size(principle_sites,1) > 1
    uu = unique(linked);
    nn = histcounts(linked);
    over_linked = uu(nn>1);
    principle_site = intersect(principle_sites,uu);

    for ii = 1:size(over_linked,2)
        over_linked_sites = linked(linked==over_linked(ii),:);
        principle_sites = [principle_sites min(over_linked_sites(:))]; %#ok<AGROW>
    end
    non_linked_sites = principle_sites;
    for ii = 1:size(linked,1)
        if isempty(intersect(linked(ii,:),principle_sites))
            non_linked_sites = [non_linked_sites min(linked(ii,:))]; %#ok<AGROW>
        end
    end
end

    non_linked_sites = principle_sites;
for ii = 1:size(non_linked_sites,2)
    clusters(int_ptr,1) = non_linked_sites(ii);
    posclusters(int_ptr,1) = non_linked_sites(ii);
    int_ptr = int_ptr + 1;
end

% Now go through and find the rest of the clusters
for ii = 2:nMax
    fprintf('\tFinding all %d-body clusters...\n',ii)
    for kk = 1:num_adsorbates
        fprintf('\t\tSite #%d\n',kk)
        curr_ads_type = sig(sigPos(kk),4);
        vec_dif = sig(:,1:3) - sig(sigPos(kk),1:3);
        R_vec_dif = (diag(vec_dif*normR*vec_dif'));
        sigwR = [sig R_vec_dif];
        irrel_sigA = sigwR(sig(:,1)==sig(sigPos(kk),1) & sig(:,2)<sig(sigPos(kk),2),:);       % Find the "irrelevant"
positions in sig (A) those where Y < X when X = the X of adsorbate kk
        irrel_sigB = sigwR(sig(:,1)<sig(sigPos(kk),1),:);                                      % ...(B) those where X is
lower than the adsorbate
        ind_sigC = zeros(10000,1);
        ind_ptr = 1;
        for bb = 1:sMax                                                              % ...(C) those where their distance
from the adsorbate is greater than this type's n-body Rmax
            nd = ndims(Rmax);                                                        % Since it can change, how many
dimension in the Rmax array?
            hold_ind = repmat({':'},1,nd-3);                                         % Create nd-3 ":" indices to insert
into the Rmax indexing
            subsetR = Rmax(curr_ads_type,bb,hold_ind{:},ii);                         % Get the 2-body subset of Rmax of
the current site type and bb
            maxSubsetR = max(subsetR(:));                                            % Find the maximum value for this
subset
            temp_ind = find(sigwR(:,4)==bb & sigwR(:,5) > maxSubsetR);              % Grab the sigwR positions that are
outside the maxRmax distance from the current position
            sz_temp = size(temp_ind,1);                                              % Number of these positions?
            ind_sigC(ind_ptr:(ind_ptr+sz_temp-1)) = temp_ind;                        % Add their indices to ind_sigC
            ind_ptr = ind_ptr + sz_temp;                                             % advance pointer
        end
        ind_sigC = ind_sigC(ind_sigC ~= 0,:);
        ind_sigC = sort(ind_sigC);
        irrel_sigC = sigwR(ind_sigC,:);
        irrel_sig = [irrel_sigA ; irrel_sigB ; irrel_sigC];
        irrel_sig = setdiff(irrel_sig,sigwR(sigPos(kk),:),'rows');
        [rel_sig,rel_bodies] = setdiff(sigwR,irrel_sig,'rows');                                % Find which bodies
(indices in sig) fit the above criteria
        rel_combs = nchoosek(rel_bodies',ii);                                                % Use these bodies to find
the relevant n-body combos
        rel_combs = rel_combs(any(rel_combs == sigPos(kk),2),:);                             % Restrict the relevant
bodies to those with the kk adsorbate in it
        bTot = size(rel_combs,1);
        dist = zeros(bTot,numRs);
        types = reshape(sig(rel_combs',4),ii,[])';                                           % rel_bodies are read row
THEN column by sig, so transpose rel_bodies first, then read their types (col 4 of sig). Reshape this back into a
matrix the same shape as rel_bodies
        rel_pos = [];
        for yyy = 1:ii
            rel_pos = [rel_pos sig(rel_combs(:,yyy)',1:3)]; %#ok<AGROW>
        end
        good_types = zeros(bTot,ii);
        good_pos = zeros(bTot,3*ii);
        zz = 1;
        for ll = 1:bTot
            pairs = sortrows(nchoosek(rel_combs(ll,:),2),2);
            num_pairs = size(pairs,1);
            dist_temp = diag(Rassoc(pairs(:,1),pairs(:,2)))';
            index_vec = num2cell([types(ll,:) ones(1,nMax-ii) ii]);
            if all(dist_temp <= Rmax(sub2ind(size(Rmax),index_vec{:})))
                dist(ll,1:num_pairs) = dist_temp;
                good_types(ll,:) = types(ll,:);
                good_pos(ll,:) = rel_pos(ll,:);
            end
        end
        fill_zeros = zeros(bTot,nMax-ii);
        fill_zeros2 = zeros(bTot, 3*(nMax-ii));
        dum_types = [good_types fill_zeros];
        dum_pos = [good_pos fill_zeros2];
        dum = [dum_types dist];
        dum2 = [dum dum_pos];
        dum(all(dum == 0,2),:) = [];
        dum2(all(dum2 == 0,2),:) = [];
        dum = dum(all(dum(:,nMax+1:nMax+num_pairs) >= Rmin-0.01,2),:);
        dum_sz = size(dum,1);
            % Fill memory as needed
            check = clusters(all(clusters == 0,2),:);
            check_sz = size(check,1);
            while check_sz < dum_sz
                Int_sz = Int_sz + BLOCK;
                clusters(int_ptr+1:Int_sz,:)=0;
                posclusters(int_ptr+1:Int_sz,:)=0;
                min_clusters(int_ptr+1:Int_sz) =0;
                check = clusters(all(clusters == 0,2),:);
                check_sz = size(check,1);
            end

        clusters(int_ptr:int_ptr+dum_sz-1,:) = dum;
        posclusters(int_ptr:int_ptr+dum_sz-1,:) = dum2;
        int_ptr = int_ptr + dum_sz + 1;
    end

end
% Clean up clusters matrix: remove unused space
clusters = clusters(any(clusters~=0,2),:);
posclusters = posclusters(any(posclusters~=0,2),:);
TotInts = size(clusters,1);

% Find where each n-body interaction starts and ends
section_ctr = zeros(1,nMax);
dummy_int = clusters;
for mm = 2:nMax
    if mm ~= nMax
        section = dummy_int(all(dummy_int(section_ctr(mm-1)+1:end,mm+1)==0,2),:);
    else
        section = dummy_int(section_ctr(mm-1)+1:end,:);
    end
    section_ctr(mm) = section_ctr(mm-1) + size(section,1);
end
section_ctr(nMax+1) = TotInts;
fprintf('\nAll n-body clusters found!\n\nNow reducing to non-equivalent interactions...\n\n')

% Initial Size of INTERACTIONS matrix (dynamic preallocation of memory)
BLOCK = 100;
list_size = BLOCK;
col_BLOCK = nMax+numRs;
INTERACTIONS = zeros(BLOCK,col_BLOCK);
POS_INTERACTIONS = zeros(BLOCK,4*nMax+numRs);
list_ptr = 1;

% Now permute the n-bodies and check against all previously found
% interactions, if it's a new one, add it to INTERACTIONS

for qq = 2:nMax
    fprintf('\n\nFinding equivalent site-permutations of %d-body clusters\nCounting unique clusters as they are
found:\n',qq)
    Nperm = sortrows(perms(1:qq));        % Find all permutations of qq bodies
    Nperm_sz = size(Nperm,1);
    tpair = sortrows(nchoosek(1:qq,2),2); % Total combinations of pairs used here according to number qq
    tpair_sz = size(tpair,1);
    for ii = section_ctr(qq-1)+1:section_ctr(qq)
        if size(INTERACTIONS(any(INTERACTIONS ~= 0,2),:),1) == 0
            fprintf('1...')
            INTERACTIONS(list_ptr+1,:) = clusters(1,:);
            POS_INTERACTIONS(list_ptr+1,:) = posclusters(1,:);
            list_ptr = list_ptr + 1;
            continue
        end
        newRassoc = zeros(nMax);
        for kk = 1:tpair_sz
            newRassoc(tpair(kk,1),tpair(kk,2)) = clusters(ii,nMax+kk);
            newRassoc(tpair(kk,2),tpair(kk,1)) = newRassoc(tpair(kk,1),tpair(kk,2));
        end

        swappedRs = zeros(Nperm_sz,numRs);
        swappedTypes = zeros(Nperm_sz,nMax);
        type_holder = clusters(ii,1:nMax);
        for ll = 1:Nperm_sz
            for kk = 1:tpair_sz
                BodyA = Nperm(ll,tpair(kk,1));
                BodyB = Nperm(ll,tpair(kk,2));
                swappedRs(ll,kk) = newRassoc(BodyA,BodyB);
                if qq ~= nMax
                    swappedTypes(ll,:) = [type_holder(Nperm(ll,:)) zeros(1,nMax-qq)];
                else
                    swappedTypes(ll,:) = type_holder(Nperm(ll,:));
                end
            end
        end
        allpossperms = [swappedTypes swappedRs];
        % Check each permutation against the interactions already found
        flag = 0;

        check = INTERACTIONS(any(INTERACTIONS~=0,2),:);
        INT_sz = size(check,1);
        for nn = 1:Nperm_sz
            for mm = 1:INT_sz
                checkdif = abs(allpossperms(nn,:) - check(mm,:));
                if all(checkdif < 0.05)
                    flag = 1;
                    break
                elseif nn == Nperm_sz && mm == INT_sz && flag ==0
                    fprintf('%d...',list_ptr)
                    if mod(list_ptr,7)==0
                        fprintf('\n')
                    end
                    INTERACTIONS(list_ptr+1,:) = clusters(ii,:);
                    POS_INTERACTIONS(list_ptr+1,:) = posclusters(ii,:);
                    list_ptr = list_ptr + 1;
                    % Add memory as it's needed
                    if list_ptr+BLOCK/20 > list_size
                        list_size = list_size + BLOCK;
                        INTERACTIONS(list_ptr+1:list_size,:)=0;
                        POS_INTERACTIONS(list_ptr+1:list_size,:)=0;
                    end
                end
            end
            if flag == 1
                break
            end
        end
    end
end
blah = INTERACTIONS(any(INTERACTIONS~=0,2),:);
pblah = POS_INTERACTIONS(any(POS_INTERACTIONS~=0,2),:);
blah = [blah sum(blah(:,nMax+1:end),2)];

% Create headings
Combs = sortrows(nchoosek(1:nMax,2),2);
numCombs = size(Combs,1);
for i = 1:nMax
    bawd{i} = sprintf('B%d',i); %#ok<SAGROW>
end
for i = 1:numCombs
    dists{i} = sprintf('R%d%d',Combs(i,1),Combs(i,2)); %#ok<SAGROW>
end
for i = 1:3:3*nMax
    poses{i} = sprintf('X%d',ceil(i/3)); %#ok<SAGROW>
    poses{i+1} = sprintf('Y%d',ceil(i/3)); %#ok<SAGROW>
    poses{i+2} = sprintf('Z%d',ceil(i/3)); %#ok<SAGROW>
end

Headings = [bawd dists];
Headings2 = [bawd dists poses];
% Get rid of unused rows
clusters(list_ptr:end,:)=[];
posclusters(list_ptr:end,:)=[];
new_inter = zeros(size(clusters,1),size(clusters,2));
pos_new_inter = zeros(size(posclusters));
tblah = blah;
ptblah = pblah;
front_pntr = 1;

% Sort the output so that interactions are clearly separated into the
% various n-body interactions and then increase in "size"
for i = 2:nMax
    zrow = nMax + nchoosek(i,2) + 1;
    if i == nMax
        [tempA, ti] = sortrows(tblah,[nMax:-1:1 size(tblah,2)]);
        tempB = ptblah(ti,:);
    else
        [tempA, ti] = sortrows(tblah(tblah(:,zrow)==0,:),[nMax:-1:1 size(tblah,2)]);
        tempB = ptblah(ptblah(:,zrow)==0,:);
        tempB = tempB(ti,:);
    end
    back_pntr = front_pntr+size(tempA,1)-1;
    tblah = setdiff(tblah,tempA,'rows');
    ptblah = setdiff(ptblah,tempB,'rows');
    new_inter(front_pntr:back_pntr,:) = tempA(:,1:end-1);
    pos_new_inter(front_pntr:back_pntr,:) = tempB;
    front_pntr = back_pntr + 1;
end
num_interactions = size(blah,1);


%%%% Generate MC_POSITIONS.txt %%%%
fprintf('\Working on MC_POSITIONS.txt now...\n\n')


clusters = zeros(BLOCK, nMax+numRs);
posclusters = zeros(BLOCK, 4*nMax+numRs+4);     % create a matrix with enough position for each n-body, each R, and
each one's 3 coordinates
min_clusters = zeros(1,BLOCK);
Int_sz = BLOCK;
int_ptr = 1;


for ii = 2:nMax
    fprintf('\tFinding all %d-body clusters...\n',ii)
    for kk = 1:num_adsorbates
        fprintf('\t\tSite #%d\n',kk)
        curr_ads_type = sig(sigPos(kk),4);
        vec_dif = sig(:,1:3) - sig(sigPos(kk),1:3);
        R_vec_dif = (diag(vec_dif*normR*vec_dif'));
        sigwR = [sig R_vec_dif];
        irrel_sigA = [];
        irrel_sigB = [];
        ind_sigC = zeros(10000,1);
        ind_ptr = 1;
        for bb = 1:sMax                                                     % ...(C) those where their distance
from the adsorbate is greater than this type's n-body Rmax
            nd = ndims(Rmax);                                               % Since it can change, how many
dimension in the Rmax array?
            hold_ind = repmat({':'},1,nd-3);                                % Create nd-3 ":" indices to insert
into the Rmax indexing
            subsetR = Rmax(curr_ads_type,bb,hold_ind{:},ii);                % Get the 2-body subset of Rmax of
the current site type and bb
            maxSubsetR = max(subsetR(:));                                   % Find the maximum value for this
subset
            temp_ind = find(sigwR(:,4)==bb & sigwR(:,5) > maxSubsetR);      % Grab the sigwR positions that are
outside the maxRmax distance from the current position
            sz_temp = size(temp_ind,1);                                     % Number of these positions?
            ind_sigC(ind_ptr:(ind_ptr+sz_temp-1)) = temp_ind;               % Add their indices to ind_sigC
            ind_ptr = ind_ptr + sz_temp;                                    % advance pointer
        end
        ind_sigC = ind_sigC(ind_sigC ~= 0,:);
        ind_sigC = sort(ind_sigC);
        irrel_sigC = sigwR(ind_sigC,:);
        irrel_sig = [irrel_sigA ; irrel_sigB ; irrel_sigC];
        irrel_sig = setdiff(irrel_sig,sigwR(sigPos(kk),:),'rows');
        [rel_sig,rel_bodies] = setdiff(sigwR,irrel_sig,'rows');                       % Find which bodies
(indices in sig) fit the above criteria
        rel_combs = nchoosek(rel_bodies',ii);                                       % Use these bodies to find
the relevant n-body combos
        rel_combs = rel_combs(any(rel_combs == sigPos(kk),2),:);                    % Restrict the relevant
bodies to those with the kk adsorbate in it
        bTot = size(rel_combs,1);
        dist = zeros(bTot,numRs);
        types = reshape(sig(rel_combs',4),ii,[])';                                  % rel_bodies are read row
THEN column by sig, so transpose rel_bodies first, then read their types (col 4 of sig). Reshape this back into a
matrix the same shape as rel_bodies
        rel_pos = [];
        for yyy = 1:ii
            rel_pos = [rel_pos sig(rel_combs(:,yyy)',1:3)]; %#ok<AGROW>
        end
        good_types = zeros(bTot,ii);
        good_pos = zeros(bTot,3*ii);
        zz = 1;
        for ll = 1:bTot
            pairs = sortrows(nchoosek(rel_combs(ll,:),2),2);
            num_pairs = size(pairs,1);
            dist_temp = diag(Rassoc(pairs(:,1),pairs(:,2)))';
            index_vec = num2cell([types(ll,:) ones(1,nMax-ii) ii]);
            if all(dist_temp <= Rmax(sub2ind(size(Rmax),index_vec{:})))
                dist(ll,1:num_pairs) = dist_temp;
                good_types(ll,:) = types(ll,:);
                good_pos(ll,:) = rel_pos(ll,:);
            end
        end
        fill_zeros = zeros(bTot,nMax-ii);
        fill_zeros2 = zeros(bTot, 3*(nMax-ii));
        dum_types = [good_types fill_zeros];
        dum_pos = [good_pos fill_zeros2];
        dum = [dum_types dist];
        all_sig_coord = ones(bTot,4).*sig_coord(kk,:);
        dum2 = [dum dum_pos all_sig_coord];
        dum(all(dum == 0,2),:) = [];
        dum2(all(dum2(:,1:end-1) == 0,2),:) = [];
        dum = dum(all(dum(:,nMax+1:nMax+num_pairs) >= Rmin-0.01,2),:);
        dum_sz = size(dum,1);
            % Fill memory as needed
            check = clusters(all(clusters == 0,2),:);
            check_sz = size(check,1);
            while check_sz < dum_sz
                Int_sz = Int_sz + BLOCK;
                clusters(int_ptr+1:Int_sz,:)=0;
                posclusters(int_ptr+1:Int_sz,:)=0;
                min_clusters(int_ptr+1:Int_sz) =0;
                check = clusters(all(clusters == 0,2),:);
                check_sz = size(check,1);
            end

        clusters(int_ptr:int_ptr+dum_sz-1,:) = dum;
        posclusters(int_ptr:int_ptr+dum_sz-1,:) = dum2;
        int_ptr = int_ptr + dum_sz + 1;
    end

end
% Clean up clusters matrix: remove unused space
clusters = clusters(any(clusters~=0,2),:);
posclusters = posclusters(any(posclusters~=0,2),:);
TotInts = size(clusters,1);

% Find where each n-body interaction starts and ends
section_ctr = zeros(1,nMax);
dummy_int = clusters;
for mm = 2:nMax
    if mm ~= nMax
        section = dummy_int(all(dummy_int(section_ctr(mm-1)+1:end,mm+1)==0,2),:);
    else
        section = dummy_int(section_ctr(mm-1)+1:end,:);
    end
    section_ctr(mm) = section_ctr(mm-1) + size(section,1);
end
section_ctr(nMax+1) = TotInts;
fprintf('\nAll n-body clusters found!\n\n')

% Initial Size of INTERACTIONS matrix (dynamic preallocation of memory)
BLOCK = 100;
list_size = BLOCK;
col_BLOCK = nMax+numRs;
INTERACTIONS2 = zeros(BLOCK,col_BLOCK);
POS_INTERACTIONS2 = zeros(BLOCK,4*nMax+numRs+4);
list_ptr = 1;

% Now permute the n-bodies and check against all previously found
% interactions, if it's a new one, add it to INTERACTIONS

pnt = ones(INT_sz,1);
MC_POSITIONS = zeros(100,size(posclusters,2),INT_sz); % Initialize MC_POSITIONS 3D array (1st_ind: position of NNs ;
2nd_ind: X Y Z of each NN; 1st_ind: cluster #)

for qq = 2:nMax
    fprintf('\n\nFinding equivalent site-permutations of %d-body clusters\nCounting unique clusters as they are
found:\n',qq)
    Nperm = sortrows(perms(1:qq));        % Find all permutations of qq bodies
    Nperm_sz = size(Nperm,1);
    tpair = sortrows(nchoosek(1:qq,2),2); % Total combinations of pairs used here according to number qq
    tpair_sz = size(tpair,1);
    for ii = section_ctr(qq-1)+1:section_ctr(qq)
        if size(INTERACTIONS2(any(INTERACTIONS2 ~= 0,2),:),1) == 0

            INTERACTIONS2(list_ptr+1,:) = clusters(1,:);
            POS_INTERACTIONS2(list_ptr+1,:) = posclusters(1,:);
            list_ptr = list_ptr + 1;
            continue
        end
        newRassoc = zeros(nMax);
        for kk = 1:tpair_sz
            newRassoc(tpair(kk,1),tpair(kk,2)) = clusters(ii,nMax+kk);
            newRassoc(tpair(kk,2),tpair(kk,1)) = newRassoc(tpair(kk,1),tpair(kk,2));
        end

        swappedRs = zeros(Nperm_sz,numRs);
        swappedTypes = zeros(Nperm_sz,nMax);
        type_holder = clusters(ii,1:nMax);
        for ll = 1:Nperm_sz
            for kk = 1:tpair_sz
                BodyA = Nperm(ll,tpair(kk,1));
                BodyB = Nperm(ll,tpair(kk,2));
                swappedRs(ll,kk) = newRassoc(BodyA,BodyB);
                if qq ~= nMax
                    swappedTypes(ll,:) = [type_holder(Nperm(ll,:)) zeros(1,nMax-qq)];
                else
                    swappedTypes(ll,:) = type_holder(Nperm(ll,:));
                end
            end
        end
        allpossperms = [swappedTypes swappedRs];
        % Check each permutation against the list of known interactions
        % (i.e. INTERACTIONS matrix). Increase the count of that
        % interaction
        flag = 0;
        INT_sz = size(new_inter,1);

        for nn = 1:Nperm_sz
            for mm = 1:INT_sz
                checkdif = abs(allpossperms(nn,:) - new_inter(mm,:));
                if all(checkdif < 0.05)             % Adjust this tolerance if using substantially large Rmax.
                    flag = 1;
                    MC_POSITIONS(pnt(mm),:,mm) = posclusters(ii,:);
                    pnt(mm) = pnt(mm) + 1;
                    break
                elseif mm == INT_sz && nn == Nperm_sz && flag == 0
                    fprintf('A structure could not be assigned to an interaction!\n')
                end
            end
            if flag == 1
                break
            end
        end
    end
end
mx_pnt = max(pnt);
MC_POSITIONS(mx_pnt:end,:,:)=[];
% Fix output for use in MC simulations where coordinates are of the type
% [X Y Z S] where S is the site type and X Y Z correspond to the INTEGER
% cell coordinate (unshifted)

sz1 = size(MC_POSITIONS,1);
sz2 = size(MC_POSITIONS,2);
sz3 = size(MC_POSITIONS,3);
bg = nMax+numRs;

for ii = 1:sz3
    for jj = 1:sz1
        dm = MC_POSITIONS(jj,:,ii);
        bodies = dm(1:nMax);
        bds = sum(bodies~=0);
        if bds == 0
            continue
        end
        part2 = dm(nMax+1:bg);
        MC_POSITIONS(jj,1,ii) = dm(end);

        for kk = 1:bds
            MC_POSITIONS(jj,2+4*(kk-1):4*kk,ii) = dm(bg+1+3*(kk-1):bg+3*kk);
            MC_POSITIONS(jj,5+4*(kk-1),ii) = bodies(kk);
            shift = Site(1:3,Site(4,:)==bodies(kk));
            MC_POSITIONS(jj,2+4*(kk-1):4*kk,ii) = MC_POSITIONS(jj,2+4*(kk-1):4*kk,ii) - shift';
        end
        % Fill in remaining places with zeros
        MC_POSITIONS(jj,2+4*bds:end,ii) = 0;
        % Find the [ 0 0 0 S ] coordinate and move to front for eventual deletion
        part = zeros(nMax,4);
        for kk = 1:bds
            part(kk,:) = MC_POSITIONS(jj,2+4*(kk-1):1+4*kk,ii);
            if isequal(part(kk,1:3),[0 0 0]) && part(kk,4) == MC_POSITIONS(jj,1,ii)
                foundit = kk;
            end
        end
        dummy = MC_POSITIONS(jj,2:5,ii);
        MC_POSITIONS(jj,2:5,ii) = MC_POSITIONS(jj,2+4*(foundit-1):1+4*foundit,ii);
        MC_POSITIONS(jj,2+4*(foundit-1):1+4*foundit,ii) = dummy;
    end
end

% remove all unused space after the coordinates
MC_POSITIONS(:,2+4*nMax:end,:) =[];
% remove the central coordinates
MC_POSITIONS(:,2:5,:) = [];

%%%%%%%


%%% Write MC_POSITIONS to files
```

```
if exist('MC_POSITIONS','dir')==7
    rmdir 'MC_POSITIONS' s
    mkdir('MC_POSITIONS')
else
    mkdir('MC_POSITIONS')
end
cd './MC_POSITIONS'
for ii=1:sz3
    dlmwrite(num2str(ii),MC_POSITIONS(:,:,ii),'delimiter','\t','precision','%4.6g')
end
cd '../'

% Create a table from the sorted interactions with the heading created above
final_out = array2table(new_inter,'VariableNames',Headings);

fprintf('\n----------------------------------------------------------------\n\n')
fprintf('DONE!\nHere are the  n-body clusters that were found:\n\n')
disp(final_out)
fprintf('A total of %d unique clusters were found\n',num_interactions)
fprintf('These results have been written to "OUTPUT_INTERACTIONS.txt"\nRename as "INTERACTIONS.txt" in order to use as
input to "COUNTS_GEN.m"\n')




%%% Write the output to file "OUTPUT_INTERACTIONS.txt"
fileID = fopen('OUTPUT_INTERACTIONS.txt','w');
fprintf(fileID,'   %s\t',Headings{1:end-1});
fprintf(fileID,'   %s\n',Headings{end});
fclose(fileID);
dlmwrite('OUTPUT_INTERACTIONS.txt',new_inter,'delimiter','\t','precision','%4.6g','-append')

fileID = fopen('CLUSTER_POSITIONS.txt','w');
fprintf(fileID,'   %s\t',Headings2{1:end-1});
fprintf(fileID,'   %s\n',Headings2{end});
fclose(fileID);
dlmwrite('CLUSTER_POSITIONS.txt',pos_new_inter,'delimiter','\t','precision','%4.6g','-append')

%Timing stuff
elapsed=toc;
inmin = elapsed/60;
fprintf('\nThis run took %9.2f seconds (or %3.2f min) to run.\n',elapsed,inmin)
fprintf('\n----------------------------------------------------------------\n')
fprintf('\n----------------------------------------------------------------\n\n')
```

COUNTS_GEN_v8.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%% COUNTS_GEN.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Build and find interaction terms in a cluster expansion for adsorbates
% on metal surfaces, along with occupancies of these interactions (given a
% certain configuration)

% v5:   finds "problem structures"
% v6:   find which of the new structures in "new_configs" is repeated, if
% any; does not include these in the calculation of the external CV score.
% v6.1: fixes "double counting" of interactions based on site number
% v7: finds ground states
% v8: determines the matrix of constraints needed to ensure that
% the ground states are recovered in the fitting of the ECIs to the dataset
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if exist('surf_normalized_counts','var') == 1
    MAT = surf_normalized_counts;
    EN = surf_energy;
end
clearvars -except MAT EN surf_normalized_counts surf_energy
format short g

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%% BEGIN USER INPUTS %%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% User provided "natural" vectors of the metal surface. Be sure to use
% the vectors that create an obtuse angle. For FCC(111) this would be
% [1 ; 0 ; 0] and [-1/2 ; sqrt(3)/2 ; 0] instead of [1 ; 0; 0] and
% [1/2 ; sqrt(3)/2 ; 0]. Technically these are the a, b, and c vectors of
% p(1 x 1) unit cell of your surface. If you intend to use the z
% coordinate position in defining adsorption site locations (only important
% if you have numerous adsorption sites and they aren't all on the same
% plane.)...i.e. you can easily define them as having the same z position),
% you should provide the c vector as it appears in your POSCAR. Otherwise,
% you can (and should) leave it as [0 ; 0 ; 1]. Mind, whatever length a
% unit vector within this coordinate system is will be the "natural unit"
% used from here on out.
% NOTE: This is the only place where cartesian coordinates should be
% encountered!

ux = [1 ; 0 ; 0];
uy = [-1/2 ; sqrt(3)/2 ; 0];
uz = [0 ; 0 ; 1];

% Alternatively, change infile to "1" and provide a file called
% "NATURAL_COORDINATES.txt" with each ux uy and uz provided as column
% vectors.

infile = 0;

% This file must be written in decimal (floating point) format. Be careful
% here, the script appears to suffer from round off errors and you might
% need to a "fudge factor" to your Rmax. Experiment. Otherwise, don't use
% this feature.
% An example for FCC(11) or HCP(0001) follows:
%
%     1 0 0
%     -0.5 0.866025403784439 0
%     0 0 1
%
%     end of example
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% If you know which clusters correspond to 1-body interactions in
% equilibrium with some chemical potential, you may want to subtract off
% the electronic (DFT) energy from that cluster. Add it here, if so, where
% the first entry is the cluster ID number and the second entry is the
% energy

mu_elec(1,:) = [1 -14.778323];

% User specified ECI value(s). First entry is the cluster ID #, second entry
% is the ECI value (in eV).

%ECI_val(1,:) = [1 -1.687979080];

% Specify whether to make regression constraints based on surface energy
% ("ads_flag = 0") or adsorption energy ("ads_flag = 1")
```

```
ads_flag = 1;

% How many sites within this "natural" cell?

sites_per_cell = 1;

% If there are more than one site, please provide their location within the
% natural unit cell (in "natural coordinates") along with a number to
% signify the type of site. The site "type" can be any positive non-zero
% integer and does not need to be continuous.
% e.g. if you have an FCC(111) surface, there are potentially top sites (1),
% bridge sites (2), fcc hollow sites (3), and hcp hollow sites (4). If
% you want to specify more than one type of adsorbate, you can do that here
% by repeating the same adsorption site, but changing the "type" number
% (i.e. the 4th element)
    Site(:,1) = [0; 0 ; 0 ; 1]; % Subsurface
   %Site(:,2) = [1/2; 1/2 ; 0 ; 2]; % Surface

% If any of the sites are linked, as in through a bond, then identify
% below. THis will simply remove the point EIC (V naught) for the linked
% site

linked = [];

% Which sites (not types) will be used to calculate the coverage of this
% system?

coverage_sites = [1];

% User specified overall maximum N-body clusters to include (even if the
% max is different for different site types, still specify the max of all
% types here)

maxNbody= 5;

% User specified "problematic length". When a supercell has a length that
% is equal to or smaller than this, the corresponding structure will be
% marked as problematic and REMOVED.

prob_length = 1;

%%%%%%%%%%%%%%%%%%% BOOK KEEPING, PLEASE DON'T TOUCH %%%%%%%%%%%%%%%%%%%%%%
        vecbody = [ones(1,maxNbody)*maxNbody maxNbody];
        Rmax = zeros(vecbody);
%%%%%%%%%%%%%%%%%%%%%%%%%% OKAY DONE, CONTINUE %%%%%%%%%%%%%%%%%%%%%%%%%%%%

% User defined maximum interaction distances "Rmax" in units of
% natural unit vectors. Each matrix (e.g. Rmax(:,:,2)) corresponds to a
% n-body interaction (e.g. 2 body interaction). Each row and column
% correspond to each site type (so if there are 3 site TYPES, these
% will be 3 x 3 symmetric matrices. Each element corresponds to
% interactions between the types designated by the row and column. For
% example, if there are 3 site types (1, 2, and 3), Rmax(:,:,3) contains
% the maximum site distances (or "cluster sizes") for 3-body interactions
% and Rmax(1,3,3) is the maximum 3-body site distance between sites
% 1 and 3 (corresponding to the sites entered above). If you want (say) 4
% body interactions between site 1 and itself (Rmax(1,1,4)) but not between
% site 1 and 2, just enter "0" for that entry (i.e. Rmax(1,2,4) = 0).

Rmax(1,1,:,:,:,2) = 5;
Rmax(1,1,1,:,:,3) = 3.7;
Rmax(1,1,1,1,:,4) = 3.7;
Rmax(1,1,1,1,1,5) = 3;

% User defined minimum interaction distance "Rmin" in units of
% natural unit vectors. This is the same for all types of interactions.

Rmin = 0.01;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%& END USER INPUTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
display_flag = 0;
% manipulate/fill in Rmax array
for ii=2:maxNbody
    Ncombs = nmultichoosek(unique(Site(4,:)),ii);
    sz_Ncombs = size(Ncombs,1);
    RtoUse = zeros(sz_Ncombs,1);
    for jj = 1:sz_Ncombs
```

```
        Nperms = unique(perms(Ncombs(jj,:)),'rows');
        sz_Nperms = size(Nperms,1);
        for kk = 1:sz_Nperms
            vecp = [Nperms(kk,:) ones(1,maxNbody-ii) ii];
            vecind = num2cell(vecp);
            Rvecind(jj,kk) = Rmax(sub2ind(size(Rmax),vecind{:})); %#ok<SAGROW>
        end
        RtoUse(jj) = max(Rvecind(jj,:));
    end
    if any(RtoUse == 0)
        whichjj = find(RtoUse == 0);
        uniqtype = unique(Ncombs(whichjj,:));
        for jj = 1:sz_Ncombs
            if all(unique(Ncombs(jj,:)) == uniqtype)
                matchjj = jj;
                break
            end
        end
        RtoUse(whichjj) = RtoUse(matchjj);
    end
    for jj = 1:sz_Ncombs
        Nperms = unique(perms(Ncombs(jj,:)),'rows');
        sz_Nperms = size(Nperms,1);
        for kk = 1:sz_Nperms
            vecp = [Nperms(kk,:) ones(1,maxNbody-ii) ii];
            vecind = num2cell(vecp);
            Rmax(sub2ind(size(Rmax),vecind{:})) = RtoUse(jj);
        end
    end
end
Rmax = Rmax.^2;
Rmin = Rmin.^2;

fprintf('----------------------------------------------------------------\n')
fprintf('----------------------------------------------------------------\n')
fprintf('Running COUNTS_GEN.m.\nAll configurations available should be placed in a folder name "configs".\n')
fprintf('----------------------------------------------------------------\n')
fprintf('\nWorking...\n\n')
fprintf('----------------------------------------------------------------\n\n')

% Get the natural coordinate system from file "NATURAL_COORDINATES.txt" if
% infile flag has been turned on
if infile == 1
    fID = fopen('NATURAL_COORDINATES.txt');
    tline = fgetl(fID);
    ux = cell2mat(textscan(tline, '%f'));
    tline = fgetl(fID);
    uy = cell2mat(textscan(tline, '%f'));
    tline = fgetl(fID);
    uz = cell2mat(textscan(tline, '%f'));
    fclose(fID);
end

% Determine the "norm conserving matrix" for later determination of
% distances
natcoor = [ux uy uz];
normR = natcoor'*natcoor;

% determine the max X and Y values needed to reach the maximum Rmax value
% specified
unitX = [1;0;0];
unitY = [0;1;0];
lengthX = unitX'*normR*unitX;
lengthY = unitY'*normR*unitY;
maxRmax = max(Rmax(:));
factor = lengthX/lengthY;
if factor > 1
    big_vec = factor*unitY + unitX;

    leng_bigvec = big_vec'*normR*big_vec;
    max_fac = sqrt(maxRmax)/leng_bigvec;

    maxX = ceil(max_fac*big_vec(1));
    maxY = ceil(factor*max_fac*big_vec(2));
elseif factor < 1
    big_vec = unitY + unitX./factor;

    leng_bigvec = big_vec'*normR*big_vec;
    max_fac = sqrt(maxRmax)/leng_bigvec;

    maxX = ceil(max_fac*big_vec(1)/factor);
    maxY = ceil(max_fac*big_vec(2));
else
    big_vec = unitY + unitX;
    leng_bigvec = big_vec'*normR*big_vec;
    max_fac = sqrt(maxRmax)/leng_bigvec;

    maxX = ceil(max_fac*big_vec(1));
    maxY = ceil(max_fac*big_vec(2));
end
maxX= ceil(maxX);
maxY=ceil(maxY);
% Total number of sites and bodies
Site = Site(:,Site(4,:)~=0);
sMax = size(unique(Site(4,:)),2);
nMax = maxNbody;
site_max = size(Site,2);

% Total number of body-to-body pair distances
numRs = nchoosek(nMax,2);

% Get all possible interactions from file "INTERACTIONS.txt" which
% needs to be in the parent directory

fID = fopen('INTERACTIONS.txt');
tline = fgetl(fID);
line=0;
while ischar(tline)
    line = line + 1 ;
    tline = fgetl(fID);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%% Kill program if number of columns in INTERACTIONS.txt is inconsistent
    %%% with the number in those specified by nMax
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if line == 2
        COLS = cell2mat(textscan(tline,'%f'))';
        INT_COLS = size(COLS,2);
        nCalc = -1/2 + 1/2*sqrt(1+8*INT_COLS);

        if abs(nCalc-nMax)>0.1
            error('Error. The maximum number of n-body interactions suggested by the number of columns in the
INTERACTIONS.txt file is %3.0g, but you have specified %3.0g in this script. Correct this and try again.',nCalc,nMax)
        end
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
line = line - 1;

fclose(fID);
INTERACTIONS = zeros(line,nMax+numRs);
INT_sz = line;
fID = fopen('INTERACTIONS.txt');
tline = fgetl(fID);
for ii = 1:line
    tline = fgetl(fID);
    INTERACTIONS(ii,:) = cell2mat(textscan(tline, '%f'));
end
fclose(fID);

% Check if folder new_configs exists. If so, check that user has generated
% file LG_EICs.txt and/or mf_coeffs.txt. Prompt for user input where
% necessary.
if exist('new_configs','dir') == 7
    fprintf('Directory "new_configs" has been created...\n...checking if a LG model has been provided...\n')
    if exist('LG_EICs.txt','file') == 0
        no_LG = 1;
        fprintf('"LG_EICs.txt" not found.\nWould you like this script to simply move any new structures\nfound in
folder "new_configs" to folder "configs"?\n')
        move_input = input('\nEnter "1" (yes) or "0" (no): ');
    else
        no_LG = 0;
        fprintf('"LG_EICs.txt" found.\nThis script will calculate the external-CV score for any new
configurations\nfound in folder "new_configs".\n')
        fprintf('Do you want to move any non-problematic structures found in new_configs to folder "configs"?\n')
        move_input2 = input('\nEnter "1" (yes) or "0" (no): ');
        fprintf('Do you want to move any problematic structures found in new_configs to folder "prob_configs"?\n')
        move_input3 = input('\nEnter "1" (yes) or "0" (no): ');
    end
    if exist('mf_coeffs.txt','file') == 0
        no_mf = 1;
    else
        no_mf = 0;
        fprintf('\nmf_coeffs.txt found.\nThis script assumes the LG model in LG_EICs.txt is\nfit to the residuals of
this mean field model.\nIf this is not the case,the external-CV score will be useless.\n')
        kill_flag = input('Proceed? ("1") or end this script? ("0"): ');
        if kill_flag ~= 1
            return
        end
    end
    fprintf('Proceeding...\n\n')

    % Loop over configurations in directory "new configs", which should be located
    % in your working directory where this script is located
    all_configs = dir('./new_configs');             % command 'dir' assigns each file a fileID starting from 3
    newest_configs = {all_configs.name};
    newest_configs = newest_configs(3:end);
    num_configs = numel(all_configs)-2;             % since fileID 1 and 2 are STDIN and STDERR we want to discount
these
    counts = zeros(num_configs,line);               % A whole bunch of preallocation follows...
    energy = zeros(num_configs,1);
    coverages = zeros(num_configs,1);
    surf_normalized_counts = counts;
    surf_energy = energy;
    confnum = 0;
    for jj=1:num_configs                    % start working through the new configurations...
        if exist('move_input','var')== 1
            if move_input == 1
                movefile(['./new_configs/' all_configs(jj+2).name],'./configs/')
                fprintf('NEW CONFIGURATION #%d, %s:\n',jj, all_configs(jj+2).name)
                if move_input2 ~= 1
                    fprintf('\tMoved!');
                end
                continue
            end
            if move_input ~= 1
                fprintf('Ignoring new_configs.\n')
                break
            end
        end
        fprintf('NEW CONFIGURATION #%d, %s:\n',jj, all_configs(jj+2).name)
        backhome = cd('./new_configs/');
        confnum = confnum + 1;
        analyzed_flag = 0;

        % Determine how many lines there are in this file
        fID = fopen(all_configs(jj+2).name);
        tline = fgetl(fID);
        line=0;

        while ischar(tline)
            if isempty(tline)
                tline = fgetl(fID);
                continue
            end

            if strcmp(tline,'analyzed')                         % check if this configuration has already been analyzed
                analyzed_flag = 1;                              % Turn the analyzed_flag on
                tline = fgetl(fID);
                if strcmp(tline,'Confidence') || strcmp(tline,'No Confidence')
                    tline = fgetl(fID); %#ok<NASGU>
                end
                tline = fgetl(fID);                             % advance to where the counts should be
                counts(jj,:) = cell2mat(textscan(tline, '%f')); % grab these counts for the matrix
                break
            else
            line = line + 1 ;
            tline = fgetl(fID);
            end
        end
        fclose(fID);

        line = line -1;
        CELL = zeros(2,2);
        SIG = zeros(line-2,3);

        % Now extract the unit cell of the configuration (CELL) along with the
        % positions of the adsorbates (SIG) and the configuration energy
        % (energy)
        fID = fopen(all_configs(jj+2).name);

        for linenum = 1:2
            tline = fgetl(fID);
            CELL(linenum,:) = cell2mat(textscan(tline, '%f'));
        end
        for linenum = 3:line
            tline = fgetl(fID);
            SIG(linenum-2,:) = cell2mat(textscan(tline, '%f'));
        end
        tline = fgetl(fID);
        energy(jj) = str2double(tline);

        fclose(fID);
        cd('../')
```

```
Ns = norm(det([CELL(1,:);CELL(2,:)]))*sites_per_cell;
rel_SIG = [];
for ii = 1:max(size(coverage_sites))
    rel_SIG = [rel_SIG; SIG(SIG(:,3)==coverage_sites(ii),:)];  %#ok<AGROW>
end
num_adsorbates = size(rel_SIG,1);

coverages(jj) = num_adsorbates/Ns;

num_adsorbates = size(SIG,1);

% find length of all effective vector lengths in the unit cell
cell_length = zeros(1,4);
cell_length(1) = CELL(1,:)*normR(1:2,1:2)*CELL(1,:)';
cell_length(2) = CELL(2,:)*normR(1:2,1:2)*CELL(2,:)';
cell_length(3) = (CELL(1,:)-CELL(2,:))*normR(1:2,1:2)*(CELL(1,:)-CELL(2,:))';
cell_length(4) = (CELL(1,:)+CELL(2,:))*normR(1:2,1:2)*(CELL(1,:)+CELL(2,:))';

% If any of those length are equal to or less than the problematic
% length, flag this structure
prob_flag = 0;
if sum(cell_length-prob_length^2 < 1) > 0.999 && sum(cell_length-prob_length^2 < 1) < 1.999
    if exist('prob_configs','dir') ~= 7
        mkdir('prob_configs');
    end
    prob_flag = 1;
end

if analyzed_flag == 1                                   % if this configuration has already been analyzed
    fprintf('Already analyzed!\n')
    surf_normalized_counts(jj,:) = counts(jj,:)./Ns;
    surf_energy(jj) = energy(jj)./Ns;
    if move_input2 == 1
        movefile(['./new_configs/' all_configs(jj+2).name],'./configs/')
    end
    continue                                            % ...then we can skip the rest of this iteration
end

% Initial Size of sig matrix (dynamic preallocation of memory)
BLOCK = 100;
list_size = BLOCK;
col_BLOCK = 4;
sig = zeros(BLOCK,col_BLOCK);

% Find SigMaxX and SigMaxY: the dimensions needed to create a surface
% large enough to encompass the maxR distance for each adsorbate

diagonal_vec = (CELL(1,:)+CELL(2,:));
SigMaxX = diagonal_vec(1)+maxX;
SigMaxY = diagonal_vec(2)+maxY;

cellvecX = CELL'\[2*SigMaxX ; -2*SigMaxY];
cellvecY = CELL'\[-2*SigMaxX ; 2*SigMaxY];

xMin = cellvecY(1);
xMax = cellvecX(1);

if xMax < xMin
    dumX = xMax;
    xMax = xMin;
    xMin = dumX;
end

xMin = floor(xMin);
xMax = ceil(xMax);

yMin = cellvecX(2);
yMax = cellvecY(2);

if yMax < yMin
    dumY = yMax;
    yMax = yMin;
    yMin = dumX;
end

yMin = floor(yMin);
yMax = ceil(yMax);

xTot = xMax - xMin;
```

```
        yTot = yMax - yMin;

       % Populate "sig" matrix
        kk = 1;
        SIG_dif = zeros(num_adsorbates,2);
        for ii = 1:num_adsorbates
            for xx = 0:xTot
                sigx = xx + xMin;
                for yy = 0:yTot
                    sigy = yy + yMin;
                    repeat = SIG(ii,1:2)+(CELL'*[sigx;sigy])';
                    SIG_dif = SIG(:,1:2) - repeat;                      % difference between this repeated position and
all adsorbates' positions
                    R_check = (diag([SIG_dif zeros(num_adsorbates,1)]*normR*[SIG_dif zeros(num_adsorbates,1)]'));
                    if any(R_check <= maxRmax+1)
                        Shift = Site(1:3,SIG(ii,3))';
                        sig_type = Site(4,SIG(ii,3));
                        sig(kk,:) = [[repeat 0]+Shift sig_type];
                        kk = kk + 1;
                    end
                    % Add new block of memory to sig matrix if needed
                    check = sig(any(sig~=0,2),:);
                    if size(check,1)/size(sig,1) > 0.95                 % Only 5% of the current allocation is left
                        list_size = size(check,1) + BLOCK;
                        sig(kk+1:list_size,:)=0;                        % ...so add a new block of memory
                    end
                end
            end
        end

        sig = sig(any(sig~=0,2),:);
        sig = sortrows(sig,1:3);

        nTot = size(sig,1);
        X = sig(:,1);
        Y = sig(:,2);
        Z = sig(:,3);
        % Find where each site within the unit cell is within the sig matrix
        sigPos = zeros(1,num_adsorbates);
        for ii = 1:num_adsorbates
            Shift = Site(1:3,SIG(ii,3))';
            sig_type = Site(4,SIG(ii,3));
            check = [[SIG(ii,1:2) 0]+Shift sig_type];
            [~,sigPos(ii)] = intersect(sig,check,'rows');
        end

        % All combinations of pairs for every adsorbate on the SigMaxX by
        % SigMaxY surface
        CombTot = sortrows(nchoosek(1:nTot,2),2);
        nCombs = size(CombTot,1);
        frstn = CombTot(:,1)';
        scndn = CombTot(:,2)';

        % Find the delta X,Y,Z between each pair of bodies with each site
        delX = zeros(1,nCombs);
        delY = zeros(1,nCombs);
        delZ = zeros(1,nCombs);
        delXYZ = zeros(3,nCombs);
        R = zeros(1,nCombs);
        for ii = 1:nCombs
            delX(ii) = X(scndn(ii))-X(frstn(ii));
            delY(ii) = Y(scndn(ii))-Y(frstn(ii));
            delZ(ii) = Z(scndn(ii))-Z(frstn(ii));
            delXYZ(:,ii) = [delX(ii);delY(ii);delZ(ii)];
            R2 = delXYZ(:,ii)'*normR*delXYZ(:,ii);
            R(ii) = (R2);
        end

        % Associate R's with each of its associated pair's positions.
        % Swaping these positions changes nothing so do that now.
        Rassoc = zeros(nTot);
        for h = 1:size(CombTot,1)
            Rassoc(CombTot(h,1),CombTot(h,2)) = R(h);
            Rassoc(CombTot(h,2),CombTot(h,1)) = Rassoc(CombTot(h,1),CombTot(h,2));
        end

        % Now find all pairs that contain the number of the position of each
        % supercell adsorbate. Use this to extract all the R's that connect the
        % various n-body interactions
        clusters = zeros(BLOCK, nMax+numRs);
        min_clusters = zeros(1,BLOCK);
        Int_sz = BLOCK;
        int_ptr = 1;

        % Create point (V naught) clusters first
        principle_sites = unique(Site(4,:));
        if size(principle_sites,2) > 1
            uu = unique(linked);
            nn = histcounts(linked);
            over_linked = uu(nn>1);
            principle_site = intersect(principle_sites,uu);

            for ii = 1:size(over_linked,2)
                over_linked_sites = linked(linked==over_linked(ii),:);
                principle_sites = [principle_sites min(over_linked_sites(:))]; %#ok<AGROW>
            end
            non_linked_sites = principle_sites;
            for ii = 1:size(linked,1)
                if isempty(intersect(linked(ii,:),principle_sites))
                    non_linked_sites = [non_linked_sites min(linked(ii,:))]; %#ok<AGROW>
                end
            end
        end

            non_linked_sites = principle_sites;
        for ii = 1:size(non_linked_sites,2)
       num_sites_here = size(SIG(SIG(:,3)==ii,:),1);
            for  ww= 1:num_sites_here
                clusters(int_ptr,1) = non_linked_sites(ii);
                posclusters(int_ptr,1) = non_linked_sites(ii);
                int_ptr = int_ptr + 1;
            end
        end

        % Now go through and find the rest of the clusters
        for ii = 2:nMax
            fprintf('\tFinding all %d-body clusters...\n',ii)
            for kk = 1:num_adsorbates
                fprintf('\t\tSite #%d\n',kk)
                curr_ads_type = sig(sigPos(kk),4);
                vec_dif = sig(:,1:3) - sig(sigPos(kk),1:3);
                R_vec_dif = (diag(vec_dif*normR*vec_dif'));
                sigwR = [sig R_vec_dif];
                irrel_sigA = sigwR(sig(:,1)==sig(sigPos(kk),1) & sig(:,2)<sig(sigPos(kk),2),:);       % Find the
"irrelevant" positions in sig (A) those where Y < X when X = the X of adsorbate kk
                irrel_sigB = sigwR(sig(:,1)<sig(sigPos(kk),1),:);                                     % ...(B) those
where X is lower than the adsorbate
                irrel_sigD = sigwR(sig(:,3)<sig(sigPos(kk),3),:);                                     % ...(D) whose
site number is less than the adsorbates
            ind_sigC = zeros(10000,1);
            ind_ptr = 1;
            for bb = 1:sMax                                                            % ...(C) those where their
distance from the adsorbate is greater than this type's n-body Rmax
                nd = ndims(Rmax);                                                       % Since it can change, how man
dimension in the Rmax array?
                hold_ind = repmat({':'},1,nd-3);                                        % Create nd-3 ":" indices to
insert into the Rmax indexing
                subsetR = Rmax(curr_ads_type,bb,hold_ind{:},ii);                        % Get the 2-body subset of Rmax
of the current site type and bb
                maxSubsetR = max(subsetR(:));
                temp_ind = find(sigwR(:,4)==bb & sigwR(:,5) > maxSubsetR);
                sz_temp = size(temp_ind,1);
                ind_sigC(ind_ptr:(ind_ptr+sz_temp-1)) = temp_ind;
                ind_ptr = ind_ptr + sz_temp;
            end
            ind_sigC = ind_sigC(ind_sigC ~= 0,:);
            ind_sigC = sort(ind_sigC);
            irrel_sigC = sigwR(ind_sigC,:);
            irrel_sig = [irrel_sigA ; irrel_sigB ; irrel_sigC; irrel_sigD];
                irrel_sig = setdiff(irrel_sig,sigwR(sigPos(kk),:),'rows');
                [rel_sig,rel_bodies] = setdiff(sigwR,irrel_sig,'rows');                               % Find which
bodies (indices in sig) fit the above criteria
                if size(rel_bodies,1) <= 1
                    continue
                end
                rel_combs = nchoosek(rel_bodies',ii);                                               % Use these bodies
to find the relevant n-body combos
                rel_combs = rel_combs(any(rel_combs == sigPos(kk),2),:);                            % Restrict the
relevant bodies to those with the kk adsorbate in it
                bTot = size(rel_combs,1);
                dist = zeros(bTot,numRs);
                types = reshape(sig(rel_combs',4),ii,[])';                                          % rel_bodies are
read row THEN column by sig, so transpose rel_bodies first, then read their types (col 4 of sig). Reshape this back
into a matrix the same shape as rel_bodies
                good_types = zeros(bTot,ii);
                zz = 1;
                for ll = 1:bTot
                    pairs = sortrows(nchoosek(rel_combs(ll,:),2),2);
                    num_pairs = size(pairs,1);
                    dist_temp = diag(Rassoc(pairs(:,1),pairs(:,2)))';
                    index_vec = num2cell([types(ll,:) ones(1,nMax-ii) ii]);
                    if all(dist_temp <= Rmax(sub2ind(size(Rmax),index_vec{:})))   % will need to generalize this part
another time
                        dist(ll,1:num_pairs) = dist_temp;
                        good_types(ll,:) = types(ll,:);
                    end
                end
                fill_zeros = zeros(bTot,nMax-ii);
                dum_types = [good_types fill_zeros];
                dum = [dum_types dist];
                dum(all(dum == 0,2),:) = [];
                dum = dum(all(dum(:,nMax+1:nMax+num_pairs) >= Rmin-0.01,2),:);
                dum_sz = size(dum,1);
                    % Fill memory as needed
                    check = clusters(all(clusters == 0,2),:);
                    check_sz = size(check,1);
                    while check_sz < dum_sz
                        Int_sz = Int_sz + BLOCK;
                        clusters(int_ptr+1:Int_sz,:)=0;
                        min_clusters(int_ptr+1:Int_sz) =0;
                        check = clusters(all(clusters == 0,2),:);
                        check_sz = size(check,1);
                    end

                clusters(int_ptr:int_ptr+dum_sz-1,:) = dum;
                int_ptr = int_ptr + dum_sz + 1;
            end

        end
        % Clean up clusters matrix: remove unused space
        clusters = clusters(any(clusters~=0,2),:);
        TotInts = size(clusters,1);

        % Find where each n-body interaction starts and ends
        section_ctr = zeros(1,nMax);
        dummy_int = clusters;
        for mm = 2:nMax
            if mm ~= nMax
                section = dummy_int(all(dummy_int(section_ctr(mm-1)+1:end,mm+1)==0,2),:);
            else
                section = dummy_int(section_ctr(mm-1)+1:end,:);
            end
            section_ctr(mm) = section_ctr(mm-1) + size(section,1);
        end
        section_ctr(nMax+1) = TotInts;
        fprintf('\nAll n-body clusters found!\n\nNow assigning them to the provided interactions...\n')

        % Now permute the n-bodies and check against the INTERACTIONS matrix

        for qq = 2:nMax
            Nperm = sortrows(perms(1:qq));        % Find all permutations of qq bodies
            Nperm_sz = size(Nperm,1);
            tpair = sortrows(nchoosek(1:qq,2),2);              % Total combinations of pairs used here according to
number qq
            tpair_sz = size(tpair,1);
            for ii = section_ctr(qq-1)+1:section_ctr(qq)
                newRassoc = zeros(nMax);
                for kk = 1:tpair_sz
                    newRassoc(tpair(kk,1),tpair(kk,2)) = clusters(ii,nMax+kk);
                    newRassoc(tpair(kk,2),tpair(kk,1)) = newRassoc(tpair(kk,1),tpair(kk,2));
                end

                swappedRs = zeros(Nperm_sz,numRs);
                swappedTypes = zeros(Nperm_sz,nMax);
                type_holder = clusters(ii,1:nMax);
                for ll = 1:Nperm_sz
                    for kk = 1:tpair_sz
                        BodyA = Nperm(ll,tpair(kk,1));
                        BodyB = Nperm(ll,tpair(kk,2));
                        swappedRs(ll,kk) = newRassoc(BodyA,BodyB);
                        if qq ~= nMax
                            swappedTypes(ll,:) = [type_holder(Nperm(ll,:)) zeros(1,nMax-qq)];
                        else
                            swappedTypes(ll,:) = type_holder(Nperm(ll,:));
                        end
                    end
                end
                allpossperms = [swappedTypes swappedRs];
                % Check each permutation against the list of known interactions
                % (i.e. INTERACTIONS matrix). Increase the count of that
                % interaction
                flag = 0;
                for nn = 1:Nperm_sz
                    for mm = 1:INT_sz
                        checkdif = abs(allpossperms(nn,:) - INTERACTIONS(mm,:));
                        if all(checkdif < 0.05)             % Adjust this tolerance if using substantially large Rmax.
```

```
                            flag = 1;
                            counts(jj,mm) = counts(jj,mm) + 1;
                            break
                        elseif mm == INT_sz && nn == Nperm_sz && flag == 0
                            fprintf('A structure could not be assigned to an interaction!\n')
                        end
                    end
                    if flag == 1
                        break
                    end
                end
            end
        end
        fprintf('...Done!\n')
        Int_sz = size(INTERACTIONS,1);
        output_stuff = [(1:Int_sz)' INTERACTIONS];
        surf_normalized_counts(jj,:) = counts(jj,:)./Ns;
        surf_energy(jj) = energy(jj)./Ns;
        ads_energy(jj) = energy(jj)./num_adsorbates;
        % Put "analyzed" at end of the file for this configuration
        backhome = cd('./new_configs');
        fID = fopen(all_configs(jj+2).name,'a');
        fprintf(fID,'\nanalyzed\n');
        if prob_flag == 0
            fprintf(fID,'Confidence\n');
        else
            fprintf(fID,'No Confidence\n');
        end
        dlmwrite(all_configs(jj+2).name,1:size(counts,2),'delimiter','\t','-append');
        dlmwrite(all_configs(jj+2).name,counts(jj,:),'delimiter','\t','-append');
        fprintf(fID,'Coverage:     %6.5g ML\nNumber of sites: %g\nSurface Energy: %7.4g eV/site\nAdsorption Energy: 
%8.4',coverages(jj),Ns,surf_energy(jj),ads_energy(jj));
        fprintf(fID,'\n\nHere are the interactions types for easy reference:\n');
        dlmwrite(all_configs(jj+2).name,output_stuff,'delimiter','\t','precision','%4.3g','-append');
        fclose(fID);
        cd(backhome)
        if prob_flag == 1 && move_input3 == 1
            movefile(['./new_configs/' all_configs(jj+2).name],'./prob_configs/')
            fprintf('File Moved to "prob_configs"\n');
        elseif prob_flag == 0 && move_input2 == 1
            movefile(['./new_configs/' all_configs(jj+2).name],'./configs/')
            fprintf('File Moved to "configs"\n');
        end
    end
    if num_configs > 0
        % Extract zero coverage energy from 'zero_energy.txt'
        fID = fopen('zero_energy.txt');
        tline = fgetl(fID);
        zero_en = cell2mat(textscan(tline, '%f'));
        fclose(fID);

        if no_LG == 0
            % Extract LG EICs from LG_EICs.txt
            fID = fopen('LG_EICs.txt');
            tline = fgetl(fID);
            line=0;
            while ischar(tline)
                line = line + 1 ;
                tline = fgetl(fID);
            end
            line = line - 1;

            fclose(fID);
            LG_EICs = zeros(line-1,1);
            fID = fopen('LG_EICs.txt');
            tline = fgetl(fID);
            CE = cell2mat(textscan(tline, '%f'));
            for ii = 1:line-1
                tline = fgetl(fID);
                LG_EICs(ii) = cell2mat(textscan(tline, '%f'));
            end
            fclose(fID);



           %%% Determine if any of the new structures are not unique
           %%% amongst themselves. Remove all not unique structures from
           %%% calculation of external CV score, keeping the lowest energy
           %%% version
            repeat_flag = 0;
            uCovs = unique(coverages);
            sMat = surf_normalized_counts;
            testing = unique(surf_normalized_counts,'rows','stable');
            indMat = 1:size(surf_normalized_counts,1);
            sEn = surf_energy;
            totoss = [];
            for ii = uCovs'
                sub_mat= sMat(abs(coverages-ii)<0.001,:);
                sub_en = sEn(abs(coverages-ii)<0.001);
                sub_ind = indMat(abs(coverages-ii)<0.001);
                [uSubMat,jja,jjb] = unique(sub_mat,'rows','stable');
                sz_dup = numel(jjb)-numel(jja);
                if sz_dup == 0
                    continue
                end
                [count,~,idxcount] = histcounts(jjb,numel(jja));
                kk = 0;
                count(count>1) = 2;
                for jj = 1:numel(count)
                    if count(jj)>1
                        count(jj) = count(jj) +kk;
                        kk = kk + 1;
                    end
                end
                what = count(idxcount);
                mincount = min(what(what>1));
                maxcount = max(what);
                for jj = mincount:maxcount
                    test = find(what==jj);
                    [~, iik] = min(sub_en(test));
                    totoss = [totoss sub_ind(test(test~=test(iik)))]; %#ok<AGROW>
                end
                if totoss > 0
                    repeat_flag = 1;
                end
            end
            tokeep = setdiff(indMat,totoss);
            surf_normalized_counts = surf_normalized_counts(tokeep,:);
            coverages = coverages(tokeep);                                         % grab the associated coverages
            surf_energy = surf_energy(tokeep);
            newest_configs = newest_configs(tokeep);

            %%% Determine if any of these new structures are equivalent to
            %%% previous structures: remove these from evaluation of
            %%% external CV score

            if ~isempty(MAT)
               [~,repi] = intersect(surf_normalized_counts,MAT,'rows','stable');
               if ~isempty(repi)
                   repeat_flag = 1;
                   surf_normalized_counts(repi,:) = [];
                   relevant_counts = surf_normalized_counts(:,CE);
                   surf_energy(repi,:) = [];
                   coverages(repi) = [];
                   surf_energy = surf_energy - zero_en;
                   labels_tokeep = setdiff(1:numel(newest_configs),repi);
                   newest_configs = newest_configs(labels_tokeep);
               else
                   relevant_counts = surf_normalized_counts(:,CE);
                   surf_energy = surf_energy - zero_en;
               end
            end
            %%%%
             % Subtract off the mu_elec energy provided by the user (if it exists)
            if exist('mu_elec','var')==1
                if ~isempty(mu_elec)
                    sz_mu_elec = size(mu_elec,1);

                    for ii = 1:sz_mu_elec
                        cluster_num = mu_elec(ii,1);
                        cluster_ECI = mu_elec(ii,2);

                        part_en = surf_normalized_counts(:,cluster_num)*cluster_ECI;
                        surf_energy = surf_energy - part_en;
                    end
                end
            end

            if no_mf == 1
                % Calculate predicted surface energy based on this LG model
                predicted_energy = relevant_counts*LG_EICs;
            else
                % Extract mean field coefficients from mf_coeffs.txt
                ID = fopen('mf_coeffs.txt');
                tline = fgetl(fID);
                line=0;
                while ischar(tline)
                    line = line + 1 ;
                    tline = fgetl(fID);
                end

                fclose(fID);
```

```
                mf_coeffs = zeros(line,1);
                fID = fopen('mf_coeffs.txt');
                for ii = 1:line
                    tline = fgetl(fID);
                    mf_coeffs(ii) = cell2mat(textscan(tline, '%f'));
                end
                fclose(ID);

                % Calculate the predicted mean field energy:
                mf_energy = 0;
                for ii = 1:line
                    mf_energy = mf_energy + mf_coeffs(ii)*coverages.^ii;
                end
                % ...and then predicted surface energy:
                predicted_energy = mf_energy + relevant_counts*LG_EICs;
            end
            residuals = surf_energy - predicted_energy;
            predicted_ads_en = predicted_energy./coverages;
            ads_en = surf_energy./coverages;
            sqrd_residuals = residuals.^2;
            ext_CV = sqrt(mean(sqrd_residuals));
            resid_stdev = sqrt(mean(sqrd_residuals)-(mean(residuals))^2);
            Rr = sqrd_residuals;
            ext_stdev = (mean(Rr.^2)-(mean(Rr))^2)^(1/4);
            if num_configs > 0
                display_flag = 1;
            end
            % Plot up the results
            mf_model = 0;
            th = 0:0.01:1;
            zeroline = zeros(size(th,2),2);

            subplot(2,1,1)
            scatter(coverages,ads_en,'sk')
            hold on
            scatter(coverages,predicted_ads_en,'r+')
            ylabel('Ads. En. (eV/adsorbate)')
            title('Current LG Model')
            xlim([0 1])
            hold off

            subplot(2,1,2)
            scatter(coverages,residuals,'r+')
            hold on
            plot(th,zeroline,'k')
            xlabel('OH Coverage (ML)')
            ylabel('Residuals (eV/site)')
            xlim([0 1])
            hold off
        end
    end
end


% Loop over configurations in directory "configs", which should be located
% in your working directory where this script is located

all_configs = dir('./configs');         % command 'dir' assigns each file a fileID starting from 3
num_configs = numel(all_configs)-2;     % since fileID 1 and 2 are STDIN and STDERR we want to discount these
counts = zeros(num_configs,INT_sz);       % A whole bunch of preallocation follows
energy = zeros(num_configs,1);
coverages = zeros(num_configs,1);
surf_normalized_counts = counts;
surf_energy = energy;
confnum = 0;
for jj=1:num_configs                    % start working through the configurations...
    fprintf('CONFIGURATION #%d, %s:\n',jj, all_configs(jj+2).name)
    backhome = cd('./configs/');
    confnum = confnum + 1;
    analyzed_flag = 0;
    % Determine how many lines there are in this file
    fID = fopen(all_configs(jj+2).name);
    tline = fgetl(fID);
    line=0;
    emptylines = 0;
    while ischar(tline)
        if isempty(tline)
            tline = fgetl(fID);
            continue
        end

        if strcmp(tline,'analyzed')                         % check if this configuration has already been analyzed
            analyzed_flag = 1;                              % Turn the analyzed_flag on
            tline = fgetl(fID);
            if strcmp(tline,'Confidence') || strcmp(tline,'No Confidence')
                tline = fgetl(fID); %#ok<NASGU>
            end
            tline = fgetl(fID);                             % advance to where the counts should be
            counts(jj,:) = cell2mat(textscan(tline, '%f')); % grab these counts for the matrix
            break
        else
        line = line + 1 ;
        tline = fgetl(fID);
        end
    end
    fclose(fID);

    line = line -1;
    CELL = zeros(2,2);
    SIG = zeros(line-2,3);

    % Now extract the unit cell of the configuration (CELL) along with the
    % positions of the adsorbates (SIG) and the configuration energy
    % (energy)
    fID = fopen(all_configs(jj+2).name);

    for linenum = 1:2
        tline = fgetl(fID);
        CELL(linenum,:) = cell2mat(textscan(tline, '%f'));
    end
    for linenum = 3:line
        tline = fgetl(fID);
        SIG(linenum-2,:) = cell2mat(textscan(tline, '%f'));
    end
    tline = fgetl(fID);
    energy(jj) = str2double(tline);

    fclose(fID);
    cd('../')



    Ns = norm(det([CELL(1,:);CELL(2,:)]))*sites_per_cell;
    rel_SIG = [];
    for ii = 1:max(size(coverage_sites))
        rel_SIG = [rel_SIG; SIG(SIG(:,3)==coverage_sites(ii),:)];  %#ok<AGROW>
    end
    num_adsorbates = size(rel_SIG,1);

    coverages(jj) = num_adsorbates/Ns;

    num_adsorbates = size(SIG,1);

    % find length of all effective vector lengths in the unit cell
    cell_length = zeros(1,4);
    cell_length(1) = CELL(1,:)*normR(1:2,1:2)*CELL(1,:)';
    cell_length(2) = CELL(2,:)*normR(1:2,1:2)*CELL(2,:)';
    cell_length(3) = (CELL(1,:)-CELL(2,:))*normR(1:2,1:2)*(CELL(1,:)-CELL(2,:))';
    cell_length(4) = (CELL(1,:)+CELL(2,:))*normR(1:2,1:2)*(CELL(1,:)+CELL(2,:))';

    % If any of those length are equal to or less than the problematic
    % length, flag this structure
    prob_flag = 0;
    if sum(cell_length-prob_length^2 < 1) > 0.999 && sum(cell_length-prob_length^2 < 1) < 1.999
        if exist('prob_configs','dir') ~= 7
            mkdir('prob_configs');
        end
        prob_flag = 1;
    end

    if analyzed_flag == 1                                   % if this configuration has already been analyzed
        fprintf('Already analyzed!\n')
        surf_normalized_counts(jj,:) = counts(jj,:)./Ns;
        surf_energy(jj) = energy(jj)./Ns;
        if prob_flag == 1
            movefile(['./configs/' all_configs(jj+2).name],'./prob_configs/')
            fprintf('File Moved to "prob_configs"\n');
        end
        continue                                            % ...then we can skip the rest of this iteration
    end

    % Initial Size of sig matrix (dynamic preallocation of memory)
    BLOCK = 100;
    list_size = BLOCK;
    col_BLOCK = 4;
    sig = zeros(BLOCK,col_BLOCK);

    % Find SigMaxX and SigMaxY: the dimensions needed to create a surface
    % large enough to encompass the maxR distance for each adsorbate
```

```
    diagonal_vec = (CELL(1,:)+CELL(2,:));
    SigMaxX = diagonal_vec(1)+maxX;
    SigMaxY = diagonal_vec(2)+maxY;

    cellvecX = CELL'\[2*SigMaxX ; -2*SigMaxY];
    cellvecY = CELL'\[-2*SigMaxX ; 2*SigMaxY];

    xMin = cellvecY(1);
    xMax = cellvecX(1);

    if xMax < xMin
        dumX = xMax;
        xMax = xMin;
        xMin = dumX;
    end

    xMin = floor(xMin);
    xMax = ceil(xMax);

    yMin = cellvecX(2);
    yMax = cellvecY(2);

    if yMax < yMin
        dumY = yMax;
        yMax = yMin;
        yMin = dumY;
    end

    yMin = floor(yMin);
    yMax = ceil(yMax);

    xTot = xMax - xMin;
    yTot = yMax - yMin;

    % Populate "sig" matrix
    kk = 1;
    SIG_dif = zeros(num_adsorbates,2);
    for ii = 1:num_adsorbates
        for xx = 0:xTot
            sigx = xx + xMin;
            for yy = 0:yTot
                sigy = yy + yMin;
                repeat = SIG(ii,1:2)+(CELL'*[sigx;sigy])';
                SIG_dif = SIG(:,1:2) - repeat;                      % difference between this repeated position and all
adsorbates' positions
                R_check = (diag([SIG_dif zeros(num_adsorbates,1)]*normR*[SIG_dif zeros(num_adsorbates,1)]'));
                if any(R_check <= maxRmax+1)
                    sig_type = Site(4,SIG(ii,3));
                    Shift = Site(1:3,SIG(ii,3))';
                    sig(kk,:) = [[repeat 0]+Shift sig_type];
                    kk = kk + 1;
                end
                % Add new block of memory to sig matrix if needed
                check = sig(any(sig~=0,2),:);
                if size(check,1)/size(sig,1) > 0.95                 % Only 5% of the current allocation is left
                    list_size = size(check,1) + BLOCK;
                    sig(kk+1:list_size,:)=0;                        % ...so add a new block of memory
                end
            end
        end
    end

    sig = sig(any(sig~=0,2),:);
    sig = sortrows(sig,1:3);

    nTot = size(sig,1);
    X = sig(:,1);
    Y = sig(:,2);
    Z = sig(:,3);
    % Find where each site within the unit cell is within the sig matrix
    sigPos = zeros(1,num_adsorbates);
    for ii = 1:num_adsorbates
        Shift = Site(1:3,SIG(ii,3))';
        sig_type = Site(4,SIG(ii,3));
        check = [[SIG(ii,1:2) 0]+Shift sig_type];
        [~,sigPos(ii)] = intersect(sig,check,'rows');
    end

    % All combinations of pairs for every adsorbate on the SigMaxX by
    % SigMaxY surface
    CombTot = sortrows(nchoosek(1:nTot,2),2);
    nCombs = size(CombTot,1);
    frstn = CombTot(:,1)';
    scndn = CombTot(:,2)';

    % Find the delta X,Y,Z between each pair of bodies with each site
    delX = zeros(1,nCombs);
    delY = zeros(1,nCombs);
    delZ = zeros(1,nCombs);
    delXYZ = zeros(3,nCombs);
    R = zeros(1,nCombs);
    for ii = 1:nCombs
        delX(ii) = X(scndn(ii))-X(frstn(ii));
        delY(ii) = Y(scndn(ii))-Y(frstn(ii));
        delZ(ii) = Z(scndn(ii))-Z(frstn(ii));
        delXYZ(:,ii) = [delX(ii);delY(ii);delZ(ii)];
        R2 = delXYZ(:,ii)'*normR*delXYZ(:,ii);
        R(ii) = (R2);
    end

    % Associate R's with each of its associated pair's positions.
    % Swaping these positions changes nothing so do that now.
    Rassoc = zeros(nTot);
    for h = 1:size(CombTot,1)
        Rassoc(CombTot(h,1),CombTot(h,2)) = R(h);
        Rassoc(CombTot(h,2),CombTot(h,1)) = Rassoc(CombTot(h,1),CombTot(h,2));
    end

    % Now find all pairs that contain the number of the position of each
    % supercell adsorbate. Use this to extract all the R's that connect the
    % various n-body interactions
    clusters = zeros(BLOCK, nMax+numRs);
    Int_sz = BLOCK;
    int_ptr = 1;

    % Create point (V naught) clusters first
    principle_sites = unique(Site(4,:));
    if size(principle_sites,1) > 1
        uu = unique(linked);
        nn = histcounts(linked);
        over_linked = uu(nn>1);
        principle_site = intersect(principle_sites,uu);

        for ii = 1:size(over_linked,2)
            over_linked_sites = linked(linked==over_linked(ii),:);
            principle_sites = [principle_sites min(over_linked_sites(:))]; %#ok<AGROW>
        end
        non_linked_sites = principle_sites;
        for ii = 1:size(linked,1)
            if isempty(intersect(linked(ii,:),principle_sites))
                non_linked_sites = [non_linked_sites min(linked(ii,:))]; %#ok<AGROW>
            end
        end
    end

        non_linked_sites = principle_sites;
    for ii = 1:size(non_linked_sites,2)
        num_sites_here = size(SIG(SIG(:,3)==ii,:),1);
        for  ww= 1:num_sites_here
            clusters(int_ptr,1) = non_linked_sites(ii);
            posclusters(int_ptr,1) = non_linked_sites(ii);
            int_ptr = int_ptr + 1;
        end
    end

    % Now go through and find the rest of the clusters
    for ii = 2:nMax
        fprintf('\tFinding all %d-body clusters...\n',ii)
        for kk = 1:num_adsorbates
            fprintf('\t\tSite #%d\n',kk)
            curr_ads_type = sig(sigPos(kk),4);
            vec_dif = sig(:,1:3) - sig(sigPos(kk),1:3);
            R_vec_dif = (diag(vec_dif*normR*vec_dif'));
            sigwR = [sig R_vec_dif];
            irrel_sigA = sigwR(sig(:,1)==sig(sigPos(kk),1) & sig(:,2)<sig(sigPos(kk),2),:);       % Find the
"irrelevant" positions in sig (A) those where Y < X when X = the X of adsorbate kk
            irrel_sigB = sigwR(sig(:,1)<sig(sigPos(kk),1),:);                                    % ...(B) those where
X is lower than the adsorbate
            irrel_sigD = sigwR(sig(:,3)<sig(sigPos(kk),3),:);                                    % ...(D) whose site
number is less than the adsorbates
            ind_sigC = zeros(10000,1);
            ind_ptr = 1;
            for bb = 1:sMax                                                           % ...(C) those where their
distance from the adsorbate is greater than this type's n-body Rmax
                nd = ndims(Rmax);                                                     % Since it can change, how man
dimension in the Rmax array?
```

```
                hold_ind = repmat({':'},1,nd-3);                                        % Create nd-3 ":" indices to
insert into the Rmax indexing
                subsetR = Rmax(curr_ads_type,bb,hold_ind{:},ii);                        % Get the 2-body subset of Rmax
of the current site type and bb
                maxSubsetR = max(subsetR(:));
                temp_ind = find(sigwR(:,4)==bb & sigwR(:,5) > maxSubsetR);
                sz_temp = size(temp_ind,1);
                ind_sigC(ind_ptr:(ind_ptr+sz_temp-1)) = temp_ind;
                ind_ptr = ind_ptr + sz_temp;
            end
            ind_sigC = ind_sigC(ind_sigC ~= 0,:);
            ind_sigC = sort(ind_sigC);
            irrel_sigC = sigwR(ind_sigC,:);
            irrel_sig = [irrel_sigA ; irrel_sigB ; irrel_sigC; irrel_sigD];
            irrel_sig = setdiff(irrel_sig,sigwR(sigPos(kk),:),'rows');
            [rel_sig,rel_bodies] = setdiff(sigwR,irrel_sig,'rows');                               % Find which bodies
(indices in sig) fit the above criteria
            if size(rel_bodies,1) <= 1
                    continue
            end
            rel_combs = nchoosek(rel_bodies',ii);                                               % Use these bodies to
find the relevant n-body combos
            rel_combs = rel_combs(any(rel_combs == sigPos(kk),2),:);                            % Restrict the relevant
bodies to those with the kk adsorbate in it
            bTot = size(rel_combs,1);
            dist = zeros(bTot,numRs);
            types = reshape(sig(rel_combs',4),ii,[])';                                          % rel_bodies are read
row THEN column by sig, so transpose rel_bodies first, then read their types (col 4 of sig). Reshape this back into a
matrix the same shape as rel_bodies
            good_types = zeros(bTot,ii);
            zz = 1;
            for ll = 1:bTot
                pairs = sortrows(nchoosek(rel_combs(ll,:),2),2);
                num_pairs = size(pairs,1);
                dist_temp = diag(Rassoc(pairs(:,1),pairs(:,2)))';
                index_vec = num2cell([types(ll,:) ones(1,nMax-ii) ii]);
                if all(dist_temp <= Rmax(sub2ind(size(Rmax),index_vec{:})))
                    dist(ll,1:num_pairs) = dist_temp;
                    good_types(ll,:) = types(ll,:);
                end
            end
            fill_zeros = zeros(bTot,nMax-ii);
            dum_types = [good_types fill_zeros];
            dum = [dum_types dist];
            dum(all(dum == 0,2),:) = [];
            dum = dum(all(dum(:,nMax+1:nMax+num_pairs) >= Rmin-0.01,2),:);
            dum_sz = size(dum,1);
                % Fill memory as needed
                check = clusters(all(clusters == 0,2),:);
                check_sz = size(check,1);
                while check_sz < dum_sz
                    Int_sz = Int_sz + BLOCK;
                    clusters(int_ptr+1:Int_sz,:)=0;

                    check = clusters(all(clusters == 0,2),:);
                    check_sz = size(check,1);
                end

            clusters(int_ptr:int_ptr+dum_sz-1,:) = dum;
            int_ptr = int_ptr + dum_sz + 1;
        end

    end
    % Clean up clusters matrix: remove unused space
    clusters = clusters(any(clusters~=0,2),:);
    TotInts = size(clusters,1);

    % Find where each n-body interaction starts and ends
    section_ctr = zeros(1,nMax);
    dummy_int = clusters;
    for mm = 2:nMax
        if mm ~= nMax
            section = dummy_int(all(dummy_int(section_ctr(mm-1)+1:end,mm+1)==0,2),:);
        else
            section = dummy_int(section_ctr(mm-1)+1:end,:);
        end
        section_ctr(mm) = section_ctr(mm-1) + size(section,1);
    end
    section_ctr(nMax+1) = TotInts;
    fprintf('\nAll n-body clusters found!\n\nNow assigning them to the provided interactions...\n')

    % Now permute the n-bodies and check against the INTERACTIONS matrix

    for qq = 2:nMax
        Nperm = sortrows(perms(1:qq));        % Find all permutations of qq bodies
        Nperm_sz = size(Nperm,1);
        tpair = sortrows(nchoosek(1:qq,2),2);              % Total combinations of pairs used here according to number
qq
        tpair_sz = size(tpair,1);
        for ii = section_ctr(qq-1)+1:section_ctr(qq)
            newRassoc = zeros(nMax);
            for kk = 1:tpair_sz
                newRassoc(tpair(kk,1),tpair(kk,2)) = clusters(ii,nMax+kk);
                newRassoc(tpair(kk,2),tpair(kk,1)) = newRassoc(tpair(kk,1),tpair(kk,2));
            end

            swappedRs = zeros(Nperm_sz,numRs);
            swappedTypes = zeros(Nperm_sz,nMax);
            type_holder = clusters(ii,1:nMax);
            for ll = 1:Nperm_sz
                for kk = 1:tpair_sz
                    BodyA = Nperm(ll,tpair(kk,1));
                    BodyB = Nperm(ll,tpair(kk,2));
                    swappedRs(ll,kk) = newRassoc(BodyA,BodyB);
                    if qq ~= nMax
                        swappedTypes(ll,:) = [type_holder(Nperm(ll,:)) zeros(1,nMax-qq)];
                    else
                        swappedTypes(ll,:) = type_holder(Nperm(ll,:));
                    end
                end
            end
            allpossperms = [swappedTypes swappedRs];
            % Check each permutation against the list of known interactions
            % (i.e. INTERACTIONS matrix). Increase the count of that
            % interaction
            flag = 0;
            for nn = 1:Nperm_sz
                for mm = 1:INT_sz
                    checkdif = abs(allpossperms(nn,:) - INTERACTIONS(mm,:));
                    if all(checkdif < 0.05)             % Adjust this tolerance if using substantially large Rmax.
                        flag = 1;
                        counts(jj,mm) = counts(jj,mm) + 1;
                        break
                    elseif mm == INT_sz && nn == Nperm_sz && flag == 0
                        fprintf('A structure could not be assigned to an interaction!\n')
                    end
                end
                if flag == 1
                    break
                end
            end
        end
    end
    fprintf('...Done!\n')
    Int_sz = size(INTERACTIONS,1);
    output_stuff = [(1:Int_sz)' INTERACTIONS];
    surf_normalized_counts(jj,:) = counts(jj,:)./Ns;
    surf_energy(jj) = energy(jj)./Ns;
    ads_energy(jj) = energy(jj)./num_adsorbates;
        % Put "analyzed" at end of the file for this configuration
        backhome = cd('./configs');
        fID = fopen(all_configs(jj+2).name,'a');
        fprintf(fID,'\nanalyzed\n');
        if prob_flag == 0
            fprintf(fID,'Confidence\n');
        else
            fprintf(fID,'No Confidence\n');
        end
        dlmwrite(all_configs(jj+2).name,1:size(counts,2),'delimiter','\t','-append');
        dlmwrite(all_configs(jj+2).name,counts(jj,:),'delimiter','\t','-append');
        fprintf(fID,'Coverage:     %6.5g ML\nNumber of sites: %g\nSurface Energy: %7.4g eV/site\nAdsorption Energy:
%8.4',coverages(jj),Ns,surf_energy(jj),ads_energy(jj));
    fprintf(fID,'\n\nHere are the interactions types for easy reference:\n');
    dlmwrite(all_configs(jj+2).name,output_stuff,'delimiter','\t','precision','%4.3g','-append');
    fclose(fID);
    cd(backhome)
end
% Subtract off the zero coverage (i.e. clean surface) energy
% If this energy is not available warn the user
no_zero_flag = 0;
if any(ismember(coverages,0,'rows')) == 1
    [~,iz] = intersect(coverages,0);
    zero_en = surf_energy(iz);
    surf_energy = surf_energy - zero_en;

    fID = fopen('zero_energy.txt','w');
    fprintf(fID,'%12.8f',zero_en);
    fclose(fID);
else
    no_zero_flag = 1;
end
```

```
%%%% Find all unique structures and amongst duplicates, select the
%%%% structure corresponding to the lowest energy inputted.
uCovs = unique(coverages);
sMat = surf_normalized_counts;
testing = unique(surf_normalized_counts,'rows','stable');
indMat = 1:size(surf_normalized_counts,1);
sEn = surf_energy;
totoss = [];
for ii = uCovs'
    sub_mat= sMat(abs(coverages-ii)<0.001,:);
    sub_en = sEn(abs(coverages-ii)<0.001);
    sub_ind = indMat(abs(coverages-ii)<0.001);
    [uSubMat,jja,jjb] = unique(sub_mat,'rows','stable');
    sz_dup = numel(jjb)-numel(jja);
    if sz_dup == 0
        continue
    end
    [count,~,idxcount] = histcounts(jjb,numel(jja));
    kk = 0;
    count(count>1) = 2;
    for jj = 1:numel(count)
        if count(jj)>1
            count(jj) = count(jj) +kk;
            kk = kk + 1;
        end
    end
    what = count(idxcount);
    mincount = min(what(what>1));
    maxcount = max(what);
    for jj = mincount:maxcount
        test = find(what==jj);
        [~, iik] = min(sub_en(test));
        totoss = [totoss sub_ind(test(test~=test(iik)))]; %#ok<AGROW>
    end



end
tokeep = setdiff(indMat,totoss);
surf_normalized_counts = surf_normalized_counts(tokeep,:);
coverages = coverages(tokeep);                                          % grab the associated coverages
surf_energy = surf_energy(tokeep);



atat_names = all_configs(3:end);
atat_names = {atat_names.name};
atat_names = atat_names(tokeep);
% Subtract off the mu_elec energy provided by the user (if it exists)
part_en = 0;
if exist('mu_elec','var')==1
    if ~isempty(mu_elec)
        sz_mu_elec = size(mu_elec,1);

        for ii = 1:sz_mu_elec
            cluster_num = mu_elec(ii,1);
            cluster_ECI = mu_elec(ii,2);

            part_en = part_en + surf_normalized_counts(:,cluster_num)*cluster_ECI;
            surf_energy = surf_energy - part_en;
        end
    end
end
ads_normalized_counts = surf_normalized_counts./coverages;
ads_normalized_counts(~isfinite(ads_normalized_counts(:))) = 0;
ads_energy = surf_energy./coverages;
ads_energy(~isfinite(ads_energy)) = 0;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%% v7 addition %%%%%%%%%%%%%%%%%%%

% Find isosteric ground states
[uniq_covs] = unique(round(coverages*1E9)/1E9);             % i.e. unique out to 9 decimal points
num_uniq_covs = numel(uniq_covs);
iso_gs = (1:num_uniq_covs)*0;



for ii = 1:num_uniq_covs
    subset = find(abs(coverages - uniq_covs(ii)) <= 1E-8);     % grab indices of coverages equal to the ii'th unique
coverage
    subset_energies = surf_energy(subset);                     % Find the energies for their corresponding structures
    [~,isu] = min(subset_energies);                                 % which index within subset energies corresponds to
the lowest energy structure (the isosteric ground state)
    iso_gs(ii) = subset(isu);                                  % and this corresponds to which index within the entire
coverages vector
```

```
end

iso_gs_ens = surf_energy(iso_gs);                                   % grab the energies of the iso_gs
iso_gs_covs = coverages(iso_gs);                                    % grab the coverages of the iso_gs



% Find true ground states
is_true_gs = (1:num_uniq_covs)*0;                                   % preallocating (is_true_gs = "is this a true
ground state")
is_true_gs(1) = 1;                                                  % the first (empty coverage) iso_gs is always a
true_gs

ii = 1;
while ii < num_uniq_covs
    Slope = (1:num_uniq_covs)*inf;                                  % preallocating
    for jj = ii+1:num_uniq_covs                                     % look at all slopes connecting this iso_gs to all
future iso_gs
        Slope(jj) = (iso_gs_ens(jj) - iso_gs_ens(ii))/(iso_gs_covs(jj) - iso_gs_covs(ii));    % slope (a forward finite
difference)
    end
    minS = min(Slope);                                              % the minimum slope
    minjj = find(Slope == minS,1);                                        % find which iso_gs is the next true_gs
    is_true_gs(minjj) = 1;                                          % mark this is as a true_gs

    ii = minjj;                                                     % move 'ii' counter to the location of the new
true_gs
end

true_gs = iso_gs(is_true_gs == 1);                                  % find the indices of the true ground states

%%% Now that we know the final surf energies we can grab those
%%% corresponding to the ground states %%%

true_gs_covs = coverages(true_gs);                                  % find the corresponding true_gs coverages
true_gs_ens = surf_energy(true_gs);                                 % find the corresponding true_gs surf energies
true_gs_ads_en = ads_energy(true_gs);                               % find the corresponding true_gs adsorption
energies
true_gs_names = atat_names(true_gs);                                % grab the true ground states' structure names

true_gs_slopes = true_gs.*0;                                        % preallocating for the gs_slopes
for ii = 2:numel(true_gs)
    true_gs_slopes(ii) = (true_gs_ens(ii) - true_gs_ens(ii-1))/(true_gs_covs(ii) - true_gs_covs(ii-1)); %forward finite
difference slope
end

form_E = surf_energy - surf_energy(coverages == max(coverages))*coverages/max(coverages);   % calculate the formaiton
energy

convex_hull = true_gs_ens - surf_energy(coverages == max(coverages))*true_gs_covs/max(coverages);   % do the same for
the ground states to get convex hull

convex_hull_ads_E = true_gs_ens./true_gs_covs;                      % adsorption energy convex hull
%convex_hull_ads_E(~isfinite(convex_hull_ads_E)) = [];               % convert NaN due to division by zero to "0"

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% v8 Find the normal matrix of constraints
if ads_flag == 1
    temp_counts = ads_normalized_counts;
    temp_energy = ads_energy;
else
    temp_counts = surf_normalized_counts;
    temp_energy = surf_energy;
end

% First all structures except the ground states
constraints = temp_counts;                       % preallocate
delta_else = temp_energy;                            % preallocate
kk = 2;
for ii = 1:num_uniq_covs
    phi = (uniq_covs(ii) - true_gs_covs(kk-1))/(true_gs_covs(kk) - true_gs_covs(kk-1));         % fractional distance
between the previous gs and the next
    if phi - 1 > 0                                                  % if phi is greater than 1, we are between two new
ground states
        kk = kk + 1;                                                % ...and must increment to that pair
        phi = (uniq_covs(ii) - true_gs_covs(kk-1))/(true_gs_covs(kk) - true_gs_covs(kk-1));         % fractional
distance between the previous gs and the next
    end

    subset = find(abs(coverages - uniq_covs(ii)) <= 1E-8);      % grab indices of coverages equal to the ii'th unique
coverage
    subset_energies = temp_energy(subset);                      % Find the energies for their corresponding structures
    subset_counts = temp_counts(subset,:);             % the counts for this isosteric subset
    if abs(phi - 1) < 1E-8
        gs_line = temp_counts(kk,:);               % if phi = 1 then we're at ground state kk and can just use the
ground state directly
        delta_else(subset) = 0;
    else
        gs_line = temp_counts(kk-1,:) + phi*(temp_counts(kk,:) - temp_counts(kk-1,:));  % counts that produce the
predicted point on the ground state line
        delta_else(subset) = temp_energy(iso_gs(ii)) - (temp_energy(true_gs(kk-1)) + phi*(temp_energy(true_gs(kk)) -
temp_energy(true_gs(kk-1)))); % use the actual distance between iso_gs and the gs line as the minimum constrained
distance
    end
    constraints(subset,:) = subset_counts - gs_line;
end
% And then the ground states
constraints(true_gs,:) = temp_counts(true_gs,:);
delta_else(true_gs) = 0;
energy_gs = temp_energy(true_gs);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Now print out a summary and plot
fprintf('----------------------------------------------------------\n')
fprintf('The provided dataset exhibits the following ground states:\n')
T = table(true_gs_names',true_gs_covs,true_gs_ads_en, true_gs_ens,true_gs_slopes');
T.Properties.VariableNames = {'Structure' 'Coverage' 'Ads_Energy' 'Surf_Energy','GS_Chem_Pot'};
disp(T)

figure
plot(coverages(coverages~=0), ads_energy(coverages~=0), 'rx')
hold on
plot(true_gs_covs, convex_hull_ads_E, 'b-o')
hold off
title('Adsorption Energy')

figure
plot(coverages, form_E, 'rx')
hold on
plot(true_gs_covs, convex_hull, 'b-o')
hold off
title('Formation Energy')

figure
plot(coverages,surf_energy,'rx')
hold on
plot(true_gs_covs,true_gs_ens,'b-o')
hold off
title('Surface Energy')



% Subtract off energy due to specified ECI values (if they exist)

part_en = 0;
part_ads_en = 0;
part_gs_slope_en = 0;
part_slope_en = 0;
if exist('ECI_val','var')==1
    if ~isempty(ECI_val)
        sz_ECI_vals = size(ECI_val,1);

        for ii = 1:sz_ECI_vals
            cluster_num = ECI_val(ii,1);
            cluster_ECI = ECI_val(ii,2);

            part_en = part_en + surf_normalized_counts(:,cluster_num)*cluster_ECI;
            part_ads_en = part_ads_en + ads_normalized_counts(:,cluster_num)*cluster_ECI;

                orig_surf_normalized_counts = surf_normalized_counts;   % Don't think I need this anymore...
            surf_normalized_counts(:,cluster_num) = 0;
            ads_normalized_counts(:,cluster_num) = 0;

        end
    end
end
surf_energy = surf_energy - part_en;
ads_energy = ads_energy - part_ads_en;
ads_energy(~isfinite(ads_energy)) = 0;

dlmwrite('COUNTS_MATRIX.txt',counts,'delimiter',' ');
dlmwrite('NORMALIZED_MATRIX.txt',surf_normalized_counts,'delimiter',' ');
dlmwrite('COVERAGES.txt',coverages,'delimiter',' ');
dlmwrite('SURF_ENERGY',surf_energy,'delimiter',' ');
```

```
fprintf('----------------------------------------------------------------\n')
fprintf('----------------------------------------------------------------\n')
fprintf('FINSIHED!\nEach configuration file has had its results added to it.\nThe strict total counts have been written
to "COUNTS_MATRIX.txt"\nThe normalized counts have been written to "NORMALIZED_MATRIX.txt"\nThe coverage of each system
has been written to "COVERAGES.txt"\nThe surface normalized energies have been written to "SURF_ENERGY.txt"\n')
if no_zero_flag == 1
    fprintf('\nWARNING. You have not provided a zero-coverage structure in the configs directory.\nThe current results
cannot be fit to a lattice gas model!\n\n')
end
fprintf('----------------------------------------------------------------\n')
fprintf('For ease of manual manipulation of the data,\nall the above data can be found in the following
variables:\n"counts"\n"surf_normalized_counts"\n"coverages"\n"surf_energy"\n')
fprintf('----------------------------------------------------------------\n')
if exist('repeat_flag','var') == 1
    if repeat_flag == 1
        if numel(repi)+numel(totoss) == 1
            fprintf('\t\tOOPS!\n\t\tBased on the available data,\n\t\t%d of these new structures is equivalent to one
of\n\t\tthe structures that have already been added!\n\t\tThis structure will be ignored in evaluation of the external
CV-score\n',numel(repi)+numel(totoss))
        elseif numel(repi)+numel(totoss) > 1
            fprintf('\t\tOOPS!\n\t\tBased on the available data,\n\t\t%d of these new structures are equivalent to each
other\n\t\tor to structures that have already been added!\n\t\tThese structures will be ignored in evaluation of the
external CV-score\n',numel(repi)+numel(totoss))
        end
    else
        fprintf('\t\tCongrats!\n\t\tBased on the available data,\n\t\tALL of these new structures are unique!\n')
    end
end
fprintf('----------------------------------------------------------------')
if display_flag == 1
    fprintf('\nThe external-CV score is %8.6g eV/site\n',ext_CV);
    fprintf('The standard deviation of the residuals is %8.6g eV/site\n',resid_stdev);
    fprintf('The external-CV deviation is %8.6g eV/site\n',ext_stdev);
    fprintf('The residuals are:\n');
    for ii = 1:size(residuals,1)
        fprintf('%s   %8.6g \n', newest_configs{ii}, residuals(ii));
    end

end
%Timing stuff
elapsed=toc;
inmin = elapsed/60;
fprintf('\nThis run took %9.2f seconds (or %5.4f minutes) to run.\n',elapsed,inmin)
fprintf('----------------------------------------------------------------')
fprintf('\n----------------------------------------------------------------\n')
clearvars -except constraints energy_gs delta_else slope_counts slope_energy part_slope_en true_gs_slopes_counts
true_gs_slopes part_ads_en ads_normalized_counts true_gs_ens true_gs_ads_ens true_gs_covs true_gs iso_gs
diff_ads_counts diff_surf_counts diff_surf_energies diff_ads_energies true_gs_covs true_gs_ads_e  true_gs_ens
true_gs_slopes ads_energy part_en orig_surf_normalized_counts atat_names newest_configs residuals mf_resids counts
surf_normalized_counts coverages surf_energy sMax INTERACTIONS
```

CONSTRUCT_CE_v4.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONSTRUCT_CE.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% A script to hopefully find the optimal set of clusters that best
% describes and predicts the training data provided in matrix
% "surf_normalized_counts" and either "surf_energy" or "mf_resids".

% Must have CV_calc_v2.m in the active directory.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

format short g

% If you'd like to use the mean field residuals as your fitting energies,
% please set "mf_flag" to "1". Else, set to "0". Default is "0"

mf_flag = 0;

% Would you like to add clusters 1 at a time of both 1 and 2 at a time?
% Adding 2 at a time increases the total number of gradent calculation
% loops by M choose 2...so it scales as M^2 ('M squared') where M is the
% total number of clusters available in "INTERACTIONS.txt".
% Select 1 at a time (1) or 1 AND 2 at a time (2)

at_a_time = 2;

% User defined percentage (fraction) of total number of known energies that become
% validation set during CV calculation (for a constant leave-x-out, set "a"
% to x/N)

a = 0.60;

% If a cluster is underrepresented in the known structures, it can be
% removed from consideration to speed up the algorithm. Set "rep" as the
% minimum number of structures that have to have a cluster for that cluster
% to be considered

rep = 20;

% If you'd like to specify that certain clusters be added at the start (no
% guarantee the algorithm won't remove them, mind you) add them here. Note:
% if you don't want ANY starting clusters, just delete the numbers and
% leave an empty set...DO NOT comment this out.

start_clusters = [ 1    2   3   4   5   6   13  25  53  55  56  58  81  104 139 159 193 195 199];

protect_clusters = [];

% User defined fraction of clusters to delete from start_clusters (must be
% less than 1)

start_fraction_to_remove = 0; %0.8*rand;

% User defined initial size of *randomly* generated clusters

max_stsz = 30 - round(start_fraction_to_remove*size(start_clusters,2));
min_stsz = 3;

start_sz = 0; %round(min_stsz +(max_stsz-min_stsz)*rand);

% Specify a CV lowering tolerance, making it so additions/removals only
% occur if the CV lowers by AT LEAST this amount (0.005 eV is a good start)

tol = 0.00002;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
unrep = find(~any(surf_normalized_counts,1));
zero_clust=find(sum(surf_normalized_counts,1)==0);
[MAT,iaa,~] = unique(surf_normalized_counts,'rows','stable');
N=size(MAT,1);                                                              % total number of known (unique) structures
candidate_clusters = find(sum(counts > 0,1)>=rep);
M=size(MAT,2);                                                             % total number of clusters
if isempty(start_clusters)
    nonzero_M = union(candidate_clusters, start_clusters)';
else
    nonzero_M = union(candidate_clusters, start_clusters);
end
```

```
% Grab appropriate energies
if mf_flag == 1
    energy = mf_resids;
else
    energy = surf_energy(iaa);
end

% create random vector of interactions that will be used as a starting CE
% in addition to specified starting CE
starting_flag = 0;
while starting_flag ==0

    out_CE = setdiff(nonzero_M,start_clusters);                 % which clusters aren't part of the starting clusters?
    rsCE = randperm(size(out_CE,2),start_sz);                   % A randon permutation to use as indices
    test_CE = out_CE(rsCE);
    sz_CE = size(test_CE,2);

    sz_start_clusters = size(start_clusters,2);                             % How many clusters in the start_clusters
vector?
    start_remove = round(start_fraction_to_remove*sz_start_clusters);       % How many to remove?
    rrCE = randperm(sz_start_clusters,start_remove);                        % a random permutation to use as indices
    start_clusters(rrCE) = [];                                              % Remove those indices from the start
clusters
    start_clusters = union(start_clusters,protect_clusters);                % add back the protected clusters in any
have been removed.
    CE = sort([test_CE start_clusters]);



    test_mat = MAT(:,CE);
    if size(CE,2) >= rank(MAT)
        error('There are not enough known structures for this many starting clusters. Reduce the amount or add more
structures and try again.')
    end
    [CV_score,std_dev] = CV_calc_v2(test_mat,energy,a);        %calulate the intial CV score
    if CV_score < 100
        starting_flag = 1;
    end
end
fprintf('----------------------------------------------------------------\n')
fprintf('----------------------------------------------------------------\n')
fprintf('\nThe starting Cluster Expansion (CE) is:\n')
disp(CE)
fprintf('Its CV score is: %8.6g eV/site\n',CV_score)
fprintf('Its standard deviation is: %8.6g eV/site\n',std_dev)
fprintf('----------------------------------------------------------------\n')
fprintf('----------------------------------------------------------------\n')
% Run through every cluster and either adding it or subtracting it from the
% current CE. Keep a record of

flag = 1;
AA = zeros(M.^2,4);        % Column 1 and 2 are clusters ii and jj. Col 3 is used to track what kind of
addition/removal was performed. Col 4 is the CV score
newl = 0;
if at_a_time == 2
    totloops = 1/2*size(nonzero_M,2)*(size(nonzero_M,2)-1);
else
    totloops = size(nonzero_M,2);
end
while flag < 2
    fprintf('Calculating the gradient of the current CE...\n(This can take a while)\n')
    loop = 0;
    for kk = nonzero_M
        loop = loop + 1;
        AA(loop,1) = kk;
        if any(zero_clust==kk)
            AA(loop,4) = 1000;
        elseif any(CE == kk)
            AA(loop,3) = -1;                       % This is used to track which clusters were added or removed along
the 1 thru M cluster additions/removals
            test_CE = setdiff(CE,kk);
            out_CE = setdiff(nonzero_M,test_CE);
            test_mat = MAT(:,test_CE);
            [AA(loop,4),~] = CV_calc_v2(test_mat,energy,a);

        else
            AA(loop,3) = 1;
            test_CE = sort([CE kk]);
            out_CE = setdiff(nonzero_M,test_CE);
            test_mat = MAT(:,test_CE);
            [AA(loop,4),~] = CV_calc_v2(test_mat,energy,a);

        end
        if mod(loop,round(totloops/20)) == 0                    % Display a percent complete
            pcent = round(round(loop/totloops/.05)*0.05*100);
            if pcent >= 100
```

```
                pcent = 99.99;
            end
            fprintf('%2g%%...',pcent);
            newl = newl + 1;
        end
        if mod(newl,10) < 0.001 && mod(loop,round(totloops/20)) == 0
            fprintf('\n');
        end
    end
    if at_a_time == 2
        tt = 0;
        for kk = nonzero_M
            tt = tt + 1;
            for jj = nonzero_M(tt+1:end)
                loop = loop + 1;
                AA(loop,[1 2]) = [kk jj];
                if any(zero_clust==kk) || any(zero_clust==jj)
                    AA(loop,4) = 1000;
                elseif any(CE == kk) && any(CE == jj)
                    AA(loop,3) = -3;                                               % This is used to track which
clusters were added or removed along the 1 thru Mchoose2 cluster additions/removals
                    test_CE = setdiff(CE,[kk jj]);
                    out_CE = setdiff(nonzero_M,test_CE);
                    test_mat = MAT(:,test_CE);
                    [AA(loop,4),~] = CV_calc_v2(test_mat,energy,a);
                elseif any(CE == kk) && ~any(CE == jj)
                    AA(loop,3) = -2;
                    test_CE = setdiff(CE,kk);
                    test_CE = [test_CE jj];                                         %#ok<AGROW>
                    out_CE = setdiff(nonzero_M,test_CE);
                    test_mat = MAT(:,test_CE);
                    [AA(loop,4),~] = CV_calc_v2(test_mat,energy,a);
                elseif ~any(CE == kk) && any(CE == jj)
                    AA(loop,3) = 2;
                    test_CE = setdiff(CE,jj);
                    test_CE = [test_CE kk];                                         %#ok<AGROW>
                    out_CE = setdiff(nonzero_M,test_CE);
                    test_mat = MAT(:,test_CE);
                    [AA(loop,4),~] = CV_calc_v2(test_mat,energy,a);
                else
                    AA(loop,3) = 3;
                    test_CE = [CE kk jj];
                    out_CE = setdiff(nonzero_M,test_CE);
                    test_mat = MAT(:,test_CE);
                    [AA(loop,4),~] = CV_calc_v2(test_mat,energy,a);
                end
                if mod(loop,round(totloops/20)) == 0                    % Display percent complete
                    pcent = round(round(loop/totloops/.05)*0.05*100);
                    if pcent >= 100
                        pcent = 99.99;
                    end
                    fprintf('%2g%%...',pcent);
                    newl = newl + 1;
                end
                if mod(newl,10) < 0.001 && mod(loop,round(totloops/20)) == 0
                    fprintf('\n');
                end
            end
        end
    end
    fprintf('...done!\n\n')
    fprintf('-----------------------------------------------------------------\n')
    fprintf('Adding and removing clusters along the gradient\nuntil CV score no longer decreases..\n\n')
    AA = AA(any(AA ~= 0,2),:);  % Remove all unused rows
    sorted_AA = sortrows(AA,4);
    sz_AA = size(AA,1);

    new_CE = CE;
    attempt = 0;
    for kk = 1:sz_AA
        ii = sorted_AA(kk,1);
        jj = sorted_AA(kk,2);
        if at_a_time == 2
            if (kk ~= 1 && any(any(sorted_AA(1:kk-1,1:2) == ii)))==1 || (kk ~= 1 && any(any(sorted_AA(1:kk-1,1:2) ==
jj)))==1
                 sorted_AA(kk,[1,2]) = [-1 -1];
                continue
            end
        else
            if (kk ~= 1 && any(any(sorted_AA(1:kk-1,1:2) == ii)))==1
                 sorted_AA(kk,[1,2]) = [-1 -1];
                continue
            end
        end
        if sorted_AA(kk,3) == -3
            test_CE = setdiff(CE,[ii jj]);
            out_CE = setdiff(nonzero_M,test_CE);
            test_mat = MAT(:,test_CE);
            [new_CV_score,new_std_dev] = CV_calc_v2(test_mat,energy,a);

        elseif sorted_AA(kk,3) == -2
            test_CE = setdiff(CE,ii);
            test_CE = [test_CE jj];                        %#ok<AGROW>
            out_CE = setdiff(nonzero_M,test_CE);
            test_mat = MAT(:,test_CE);
            [new_CV_score,new_std_dev] = CV_calc_v2(test_mat,energy,a);

        elseif sorted_AA(kk,3) == -1
            test_CE = setdiff(CE,ii);
            out_CE = setdiff(nonzero_M,test_CE);
            test_mat = MAT(:,test_CE);
            [new_CV_score,new_std_dev] = CV_calc_v2(test_mat,energy,a);

        elseif sorted_AA(kk,3) == 1
            test_CE = sort([CE ii]);
            out_CE = setdiff(nonzero_M,test_CE);
            test_mat = MAT(:,test_CE);
            [new_CV_score,new_std_dev] = CV_calc_v2(test_mat,energy,a);

        elseif sorted_AA(kk,3) == 2
            test_CE = setdiff(CE,jj);
            test_CE = [test_CE ii];                        %#ok<AGROW>
            out_CE = setdiff(nonzero_M,test_CE);
            test_mat = MAT(:,test_CE);
            [new_CV_score,new_std_dev] = CV_calc_v2(test_mat,energy,a);

        elseif sorted_AA(kk,3) == 3
            test_CE = [CE ii jj];
            out_CE = setdiff(nonzero_M,test_CE);
            test_mat = MAT(:,test_CE);
            [new_CV_score,new_std_dev] = CV_calc_v2(test_mat,energy,a);

        end

        if CV_score - new_CV_score >= tol
            flag = 0;
            CE = sort(test_CE);
            CV_score = new_CV_score;
            std_dev = new_std_dev;

            if sorted_AA(kk,3) == -3
                fprintf('\nCluster %d and %d removed!\n',ii,jj)
            elseif sorted_AA(kk,3) == -2
                fprintf('\nCluster %d removed and %d added!\n',ii,jj)
            elseif sorted_AA(kk,3) == -1
                fprintf('\nCluster %d removed!\n',ii)
            elseif sorted_AA(kk,3) == 1
                fprintf('\nCluster %d added!\n',ii)
            elseif sorted_AA(kk,3) == 2
                fprintf('\nCluster %d added and %d removed!\n',ii,jj)
            elseif sorted_AA(kk,3) == 3
                fprintf('\nCluster %d and %d added!\n',ii,jj)
            end
            fprintf('\nThe new CE is:\n')
            disp(CE)
            fprintf('Its CV score is: %8.6g eV/site\n',CV_score)
            fprintf('Its standard deviation is: %8.6g eV/site\n',std_dev)
            fprintf('----------------------------------------------------------------\n')
        elseif CV_score - new_CV_score < tol

            if flag < 2
                if attempt == 4
                    flag = flag + 1;
                    if flag < 2
                        fprintf('\nReached a minimum along this gradient!\n')
                        fprintf('----------------------------------------------------------------\n')
                        fprintf('----------------------------------------------------------------\n')
                    end
                    break
                end
                attempt = attempt + 1;
                continue
            elseif flag >= 2
                continue
            end
        end
    end
    if kk == sz_AA
        fprintf('\nAll possible additions/removals along this gradient exhausted!\n')
        fprintf('----------------------------------------------------------------\n')
    end
```

```
end
interactions = INTERACTIONS(CE,:);

final_mat = MAT(:,CE);
final_coeffs = (final_mat'*final_mat)\(final_mat'*energy);
final_en = final_mat*final_coeffs;
final_resids = energy - final_en;
RMSE = sqrt(mean(final_resids.^2));

%%% Write the output to file "LG_EICs.txt"
dlmwrite('LG_EICs.txt',CE,'delimiter','\t')
dlmwrite('LG_EICs.txt',final_coeffs,'delimiter','\t','-append')
fID = fopen('LG_EICs.txt','a');
fprintf(fID,'%8.6f eV/site',CV_score);
fclose(fID);
fprintf('----------------------------------------------------------------\n')
fprintf('----------------------------------------------------------------\n')
fprintf('\nThe algorithm has found a local minimum!\nNo further cluster additions or removals lower the CV score\n\n')
fprintf('\nThe final CE is:\n')
disp(CE)
fprintf('Its CV score is: %8.6g eV/site\n',CV_score)
fprintf('Its standard deviation is: %8.6g eV/site\n',std_dev)
fprintf('The RMSE of the final fit is: %8.6g eV/site\n',RMSE)
fprintf('The LG EIC for this CE are:\n')
disp(final_coeffs)
%fprintf('The corresponding residuals are:\n')
%display(final_resids)
fprintf('This CE corresponds to the following interactions:\n')
disp(interactions)

clearvars -except max_pCs min_pCs pC CE interactions CV_score final_resids final_coeffs counts surf_normalized_counts
coverages surf_energy sMax final_en INTERACTIONS mean_coeffs mf_resids
```

CV_CALC.m

```
function [ CV_score, std_dev] = CV_calc_v2( test_mat,energy,a,optional_reference_CV_score)
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%% CV_calc.m: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Given an N x M submatrix corresponding to the M interaction coefficients
% of some (size M) cluster expansion (CE) for some N known DFT energies,
% calculates the Leave Multiple Out - Cross Validation (LMO-CV) score.
%
% The training set of N DFT energies is split into a construction set (size
% N-d) and validation set (size d). The relative size of these is
% determined by the user specified "a": the fractional portion of the N
% total data that becomes the d validation set data.
% Per K. Baumann, Trends Anal. Chem. 22 (2003) 395-406, "a" should not be
% set below 0.4
%
% v2: Enforces an unbiased sampling of the data set.
%     ...After N random cuts of the data, the algorithm chooses the
%     validation set from structures that have been biased against in the
%     original cuts until there is no more bias remaining and until at
%     least 9N more cuts have been made and the CV score stops changing by
%     at least 10^-6 eV.
%     This turns out to be a far more stable (repeatable) value than the
%     previous randomaly biased version.
%
%     2018-02-15: Optimized. Now allows for a 0.001% bias where bias is
%     defined as total imbalance divided by total number of cuts of the
%     data. Also skips a step if a matrix is 'close to singular'

if ~exist('optional_reference_CV_score','var')
    optional_reference_CV_score = 10;
end
ref_CV_score = optional_reference_CV_score;
lastwarn('')
warning('off','all')

% remove 0 coverage structure (the one with all zeros) from consideration



N = size(test_mat,1);
sz_constn_set = floor((1-a)*N);
sz_validn_set = ceil(a*N);



% calculate CV score for this CE

count=zeros(1,N);
diffCV = 1000;
new_CV_score = 100;
k = 1;
cnt_rng = 100;

tot_skips = 0;

while abs(diffCV) > 10^-7 || k < 10*N || cnt_rng/k > 0.0001
    old_CV_score = new_CV_score;

    if tot_skips > N
        CV_score = 10000;
        std_dev = 10000;
        return
    end

    if k <= N
        cc = randperm(N,sz_constn_set);
        constn_configs = cc;
        constn_set = test_mat(constn_configs,:);
        validn_configs = 1:N;
        validn_configs(constn_configs)=[];
        count(validn_configs)=count(validn_configs)+1;
        cnt_rng = max(count) - min(count);
    else
        [~, ii] = sort(count);
        validn_configs = ii(1:sz_validn_set);
        constn_configs = 1:N;
        constn_configs(validn_configs)=[];
        constn_set = test_mat(constn_configs,:);
        count(validn_configs)=count(validn_configs)+1;
        cnt_rng = max(count) - min(count);
    end
    constn_en = energy(constn_configs);
    validn_set = test_mat;
    validn_set(constn_configs,:) = [];
    validn_en = energy;
    validn_en(constn_configs) = [];
    [~,warcheck] = lastwarn;
    if strcmp(warcheck,'MATLAB:nearlySingularMatrix')
        lastwarn('');
        tot_skips = tot_skips + 1;
        continue
    end

    if any(all(constn_set==0,1))==1
        tot_skips = tot_skips + 1;
        continue
    end
    coeffs = (constn_set'*constn_set)\(constn_set'*constn_en);
    pred_en = validn_set*coeffs;
    predn_err = pred_en - validn_en;
    sq_predn_err(k) = dot(predn_err,predn_err); %#ok<AGROW>
    sum_predn_err(k) = sum(predn_err); %#ok<AGROW>
    new_CV_score = sqrt(mean(sq_predn_err)/sz_validn_set);
    diffCV = new_CV_score - old_CV_score;

    k = k + 1;
    if tot_skips > 1000
        new_CV_score = 10000;
        break
    end
    if new_CV_score - ref_CV_score > 0.0005  && abs(diffCV) < 0.001 && k > 2*N
        break
    end
    if new_CV_score - ref_CV_score > 0  && abs(diffCV) < 0.01 && k > 10*N
        break
    end
    if k > 500*N
        break
    end
    if new_CV_score > 1E6
        break
    end
end
%fprintf("...\n")
CV_score = new_CV_score;
avg_error = sum(sum_predn_err)/(k*sz_validn_set);
std_dev = sqrt(sum(sq_predn_err)/(k*sz_validn_set)-avg_error^2);
```

make_mf.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% make_mf.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% v2: exclude duplicate structures from fit
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           Choose the order of the fit to the mean field model:           %
                                  n = 3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%             Force the fit to pass through the 1 ML coverage?             %
%                            (yes = 1, no = 0)                             %
                                 flag = 0;
%                                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[AAA,ia,~] = unique(surf_normalized_counts,'rows','stable');
uniqcov = coverages(ia);
uniqen = surf_energy(ia);

[mean_coeffs,mf_resids,MSE,MAE,ME]=mean_field(uniqcov,uniqen,n,flag);

fprintf('The mean field coefficients are:\n')
disp(mean_coeffs)
fprintf('Root Mean Squared Error: %8.6f eV/site\nMean Absolute Error: %8.6f ev/site\nMean Signed Error: %8.6f
eV/site\n\n',MSE,MAE,ME)

% Make a pretty plot
mf_model = 0;
th = 0:0.01:1;
line = zeros(size(th,2),2);
for i = 1:n
    mf_model = mf_model + mean_coeffs(i)*th.^(i-1);
end
ads_en = uniqen./uniqcov;
ads_en(isnan(ads_en)) = [];
mod_coverages = uniqcov;
mod_coverages(mod_coverages==0) = [];
subplot(2,1,1)
scatter(mod_coverages,ads_en);
hold on
plot(th,mf_model)
ylabel('Ads. En. (eV/adsorbate)')
title('Mean Field Model')
hold off

subplot(2,1,2)
scatter(uniqcov,mf_resids)
hold on
plot(th,line,'k')
xlabel('Adsorbate Coverage (ML)')
ylabel('Residuals (eV/site)')
hold off

%%% Write the output to file "mf_coeffs.txt"
dlmwrite('mf_coeffs.txt',mean_coeffs)
```

OVERNIGHT_CE.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%% overnight_CE_v2.m %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Runs "CONSTRUCT_CE_v4.m" on a loop the numbers of times specified.
%%%% Output from "CONSTRUCT_CE_v4.m" is placed in a folder called "LG_EICs",
%%%% which must exist for this script to work.

%%%%%%% USER INPUT %%%%%%%%%
numRuns = 500;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%


tic


backhome = cd('./LG_EICs');
for i = 1:numRuns
    cd(backhome)
    run('CONSTRUCT_CE_v4.m')
    thisCV = sprintf('%6.6f',CV_score);
    backhome = cd('./LG_EICs');
    movefile('../LG_EICs.txt',['LG_EICs.' thisCV '.txt']);
end
cd(backhome)
toc
```