Spring 2023

Scanning Probe Microscopy Studies of Petroleum Chemistry: Substrate-Dependent Catalytic Properties of MoS2 and Automating Scanning Probe Microscopy with Machine Learning

Steven Arias
University of New Hampshire, Durham

SCANNING PROBE MICROSCOPY STUDIES OF PETROLEUM CHEMISTRY: Substrate-Dependent Catalytic Properties of MoS_2 and Automating Scanning Probe Microscopy with Machine Learning

BY

STEVEN ARIAS
Physics BS, University of New Hampshire, 2016

DISSERTATION

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy
in
Physics

May, 2023

This thesis/dissertation has been examined and approved in partial fulfillment of the requirements for the degree of Doctorate of Philosophy in Physics by:

Dissertation Director, Shawna Hollen, Physics Ph.D.

Jiadong Zang, Physics Ph.D.

Kai Germachewski, Physics Ph.D.

Percy Zahl, Physics Ph.D., Brookhaven National Lab

Yunlong Zhang, Chemistry Ph.D., ExxonMobil

On April 17, 2023

Approval signatures are on file with the University of New Hampshire Graduate School.

ACKNOWLEDGEMENTS

Hay una lista gigantesca de persona a las que tengo que agradecer durante todos estos años en UNH. A mis padres por apoyarme desde la distancia y también les quiero pedir perdón por el abandono temporal mientras terminaba la carrera. A mis sobrinas por ser la motivación para dar todo lo que puedo para que ellas tengan un mejor futuro. Y gracias a Vane por ser una voz calmante y un soporte enorme durante todo este proceso. Tantas veces quise rendirme y ellos no me dejaron y por eso infinitas gracias por todo.

Thanks to Katie, Michelle, and Michelle for all their hard work, helping me countless times, and for providing me with all the candy I ever needed. Thanks to Tammy, Erin, Selina, Helen, and Kate from the McNair office as well for all their support and friendships.

I want to give a special shout-out to all the people in the Hollen Lab. Tan, Cam, Page, Riley, and Chris always made the clean room a much happier place to work in. Thanks to Quinn for all the fun conversations about how not to hurt ourselves with magnets and 3D printing. To Diana for all the machine learning talks. Thanks to Alana, Gavin, Lihy, and Logan for teaching me about their research in the lab as well. Thanks to Olaiya and Muhammad for taking over the lab and keeping it up and running. And special thanks to lab manager Ben for putting up with all my fire safety questions the last couple of months and showing me all the optics parts he's scavenged from old labs.

I do not think I would have made it sane through grad school without the help of my friends. Thanks to Ben for the fun journey all the way to Colorado. Thanks to Abby for teaching me that drawing cards is all you ever need to do in a game of Magic. Thanks to Adam for teaching me how easily you can break yourself skiing and for putting up with all

my nonsense. Thanks to Jason for always knowing what to say and for planning out our little Zoom reunions. Thanks to Jake for his mixology lessons and little home guitar concerts. Thanks to Alex for all the fun VR sessions. Thanks to Caitlyn and Tierney for being there since undergrad and always finding a way to make me feel better with pictures of Clover and Drea. Thanks to Blondie for all her support and for exploring Boston with me.

I would like to thank my whole committee past and current members for all their advice and support. Thanks to Percy and Yunlong for all their help with the automation projects discussed in Chapter 4. Special thanks to Percy for being a great mentor this past year at Brookhaven. Thanks to Kai for making my first physics class at UNH exciting enough to make me want to stay in the major. Thanks to Jiadong for knowing when I needed help with all the quantum and solid state lectures. Thanks to Christine for all the chemistry lessons.

Special thanks to Shawna for all her patience, motivational talks, and support during my research projects. Thank you Shawna for helping me have more confidence in my writing and in how I present myself. And thank you for all the group outings and for making the lab a fun place to work.

Lastly, I want to thank Karsten for all his support through my years in undergrad and graduate school. Thank you for helping me find joy in physics research, for all the soccer and coffee talks, and for helping me get to where I am now.

Y'all are great.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

SCANNING PROBE MICROSCOPY STUDIES OF PETROLEUM CHEMISTRY; STUDYING THE SUBSTRATE DEPENDENCE OF MoS_2 AS A CATALYST FOR HYDRODESULFURIZATION REACTIONS AND USING MACHINE LEARNING TO AUTOMATE SCANNING PROBE MICROSCOPY

by

Steven Arias
University of New Hampshire, May, 2023

With the growth of the population, society's energy demands are mostly reliant on petroleum products that come from the refining of crude oil. Most of these refining reactions have been developed through averaging spectroscopic techniques, but scientists do not know exactly what is happening in these processes at the nano and atomic levels. This information is crucial when designing an efficient refining process that produces petroleum products that emit fewer harmful gases when combusting. Scanning probe microscopy techniques have become a powerful tool to look into the chemical structures found in petroleum products, to understand catalytic reactions in refining processes, and to find new non-combustible uses for these products. In this dissertation, I show how scanning probe microscopy (SPM) techniques, especially non-contact atomic force microscopy (NC-AFM) can provide an atomic-level understanding of the chemical structures and active catalytic sites that play a role in these refining processes. First, I studied hydrodesulfurization reactions that use molybdenum disulfide as a main catalyst to explore the effect of layer thickness, strain, and underlying substrates on its electronic and catalytic properties. Here, I present the first NC-AFM experiments investigating the active catalytic sites of molybdenum disulfide on industrially relevant substrates. Through these experiments, I found how NC-AFM techniques on insulators need to be improved to achieve high-resolution images that are comparable to those collected on metal substrates. Second, I created Auto-HR-AFM, a machine-learning script that collects optimal high-resolution NC-AFM images. Auto-HR-AFM is a modular and open-source script that provides an initial framework for a fully automated SPM. Expanding on this framework will widen the use of scanning probe microscopy techniques to non-experts and the automation will increase the time the system is kept running to collect large optimal datasets. Ultimately, these studies will broaden the use of high-resolution SPM techniques and help create more efficient catalysts and refining processes to produce cleaner and more efficient petroleum products.

CHAPTER 1

Introduction/Background

With the growth of the population, the demand for clean and accessible energy sources increases as well. For more than 160 years households and industries have primarily relied on the combustion of fossil fuels to supply their energy demand, but the burning of these products also produces harmful greenhouse emission gases like carbon dioxide. The major challenge the world faces today is how to continue to meet these energy demands while reducing the number of harmful emissions.

To face this challenge the main issues that have to be taken care of are reducing the number of emissions produced by fossil fuels and creating more efficient renewable energy sources. These issues are intertwined since the development of renewable energy sources means that the world can start to peak its fossil fuel usage and move away from combustion procedures, but we are not there yet. The world currently relies mostly on petroleum and natural gas to provide most of the energy demand and most likely coal will be the first fossil fuel to start a decline. For now, as the population continues to increase these fossil fuels will still play a huge role in the energy supply as renewable energy technologies catch up. The goal for large groups of the world is to reach net-zero emissions in 30 to 50 years [1].

For this, the US oil and gas industry will be a critical and essential part to meet these goals and the capabilities they develop and deploy will be applied around the world to meet global targets for emission reduction. To accelerate development and deployment, industry R&D teams have to collaborate with DOE national laboratories and universities while still getting the support of governmental policies at all levels. [1].

The difficulty in reducing emissions comes from understanding the refining procedures used to treat crude oil and gas. While we have come a long way since the first initial refineries of the 1800's that refined fossil fuels into Kerosene [2] and now can create through refining procedures thousands of products that are used in daily life like aspirin, plastics, fuel, etc [3], scientists do not fully understand what composition of hydrocarbons make up these fossil fuels. Knowing the exact composition of these crude products helps design refinery processes to efficiently remove any contaminants and help produce cleaner energy sources.

Most of the work done in the 19th century to understand fossil fuels has been in finding ways to refine the materials into useable products [4, 5] and to study their physical properties. Chromatography, mass spectrometry, and petroleomics have been used by chemists to understand the complexity of these materials and they helped develop the fuels and other byproducts we use today. But what most of these techniques lacks is a fundamental understanding of the atomic and molecular level procedures that are taking place during these refining steps. As we look into optimizing our petroleum products, we have to start implementing tools to look at the molecular structures and the chemical reactivities at an atomic level.

The invention of the scanning tunneling microscope (STM) in 1982, by Binnig and Rohrer, opened the door to studying the atomic world directly [6]. Four years after that the atomic force microscope (AFM) allowed us to see the topography of any material regardless of its conductivity [7]. These scanning probe microscopy (SPM) techniques provide topographic images, and local measurements of surface properties, and can manipulate surfaces. For 40 years, SPM techniques have been an essential part of surface science and material science research. Petroleum chemistry research has also benefited from the use of SPM technologies by using STM to study catalysis reactions related to the desulfurization [8–10] of crude oil. In recent years the development of the NC-AFM has opened the door to studying the molecular structure and reactivities of petroleum products [11, 12]. Although there are still limitations to using these techniques as a high throughput characterization tool of petroleum products, these SPM studies give us insight into designing new refining procedures to produce cleaner and more efficient fuel sources. These studies can also help us discover non-combustible techniques for these products like finding a pathway to directly refine composite materials for high strength construction [11].

In this dissertation, I show how SPM techniques can be used in petroleum research to study refining reactions at nano and atomic scales. This information is crucial when designing an efficient refining process that produces petroleum products that emit fewer harmful gases when combusting. In Chapter 2, I explain the operation principles of STM and AFM, which are the two main SPM techniques I used for my research projects. I also describe how to perform the sample preparation techniques I used for the projects. Two of the major projects I performed are explained in Chapters 3 and 4. I present in Chapter 3 the first NC-AFM experiments investigating the active catalytic sites of molybdenum disulfide on industrially relevant substrates. Through these experiments, I found how NC-AFM techniques on insulators need to be improved to achieve high-resolution images that are comparable to those collected on metal substrates. In Chapter 4 I present my work in creating an initial framework for a fully automated SPM. The major focus of Chapter 4 is the work done to create Auto-HR-AFM, a machine learning script that collects optimal high-resolution non-contact AFM images. My conclusions for these projects are presented in Chapter 5 where I describe how to overcome issues and limitations in both main projects and ideas for continuing the projects in the future. These studies will broaden the use of high-resolution SPM techniques and help create more efficient catalysts and refining processes to produce cleaner and more efficient petroleum products.

CHAPTER 2

Methods

In this chapter, I describe the tools and techniques that I used for my thesis research projects, starting with an explanation of scanning probe microscopy (SPM) tools and the vacuum chamber that houses them. I then go into more detail explaining the main SPM techniques used in my thesis; scanning tunneling microscopes (STM) and atomic force microscopes (AFM).

For STM I describe the basic principles of operation and then I go over the common modes of STM operation. For AFM, I also go over the basic principles of operation, then I describe the common mode of operations and describe in detail the relationship between the measured frequency shifts in AFM to the forces felt between the tip and sample. I end the AFM section with an explanation of how the probes have changed through the years to increase their resolution.

In the last sections of this chapter, I describe the sample preparation techniques I used for my experiments. I start by explaining the process of exfoliation techniques used to separate layered two-dimensional materials from bulk crystals, I then explain the steps to use a polymer transfer method and flake transfer station to move exfoliated flakes onto different supports and how this is used for creating heterostructures. I end this section with an explanation of cleaning metal substrates using sputter and anneal cycles and a description of the physical vapor deposition techniques used to deposit molecules on clean surfaces.

2.1 Ultrahigh Vacuum Chamber and Main Tools

The main instruments I used in my experiments are STM and AFM which are operated under ultra-high vacuum (UHV) conditions with pressures below 5x10^{-9} torr and at variable temperatures ranging from room temperature to low temperatures (5K to 10K). These two techniques have opened the doors to the atomic world and were the starting point in nanotechnology research [13]. Due to their broad applications and high resolution, these techniques have been used in the fields of physics, chemistry, biology, engineering, and materials science.

Typical SPM experiments require a proper clean and controlled environment so they are carried out in an ultra-high vacuum (UHV) chamber. The base pressure in the chamber is kept to 5x10^{-10} torr or below to minimize the number of residual gas contaminants in the chamber. With fewer contaminants, our surfaces stay clean for a long period of time. Lower pressure in the chamber also increases the mean free path of the particles inside the chamber limiting the interactions between them and the clean surfaces used for experiments.

At higher temperatures objects have higher kinetic energy, so to study mobile molecular samples and to ease the goal of achieving atomic resolution it is best to keep the SPM running at low and stable temperatures. Having high thermal stability reduces the effect of thermal drift while imaging.

Figure 2.1 shows a schematic of LEWIS, the dual STM/Non-Contact AFM (NC-AFM) used in the Hollen Lab, built by RHK Technologies. Although most UHV SPM systems are customized by each research group to fill specific experimental needs, they mostly all have the same key components. Typical chambers are made out of stainless steel. A series of valves and pumps bring the pressure from atmospheric pressure of around 760 torr to UHV pressures of around 1x10^{-10}. Ion gauges are used to monitor the pressure around the chamber and thermocouples are used at various stages to monitor temperatures inside the chamber. Mass spectrometers and other surface characterization tools are also commonly

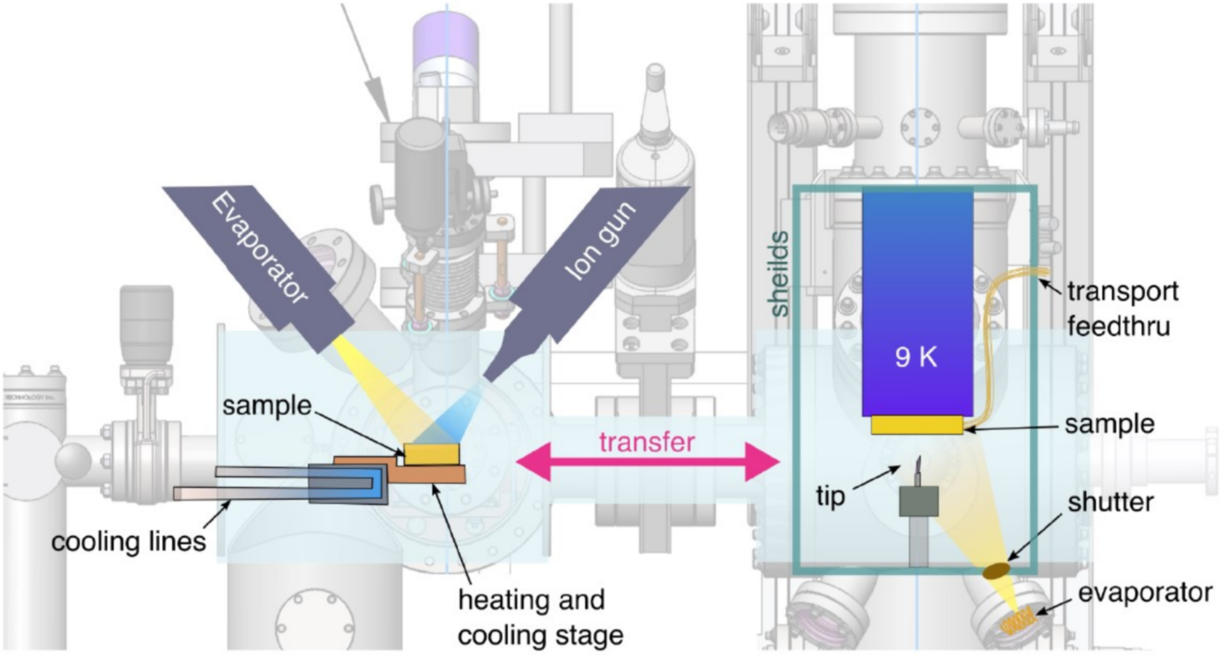


Figure 2.1: Schematic of the Hollen Lab's dual STM/NC-AFM system.

found in UHV chambers, and I used them occasionally. To obtain atomic resolution it is also important to isolate the SPM stage to minimize noise produced by thermal changes and vibrations.

Other components are added to help facilitate experiments. Most experiments require sample preparation techniques that are best done *in situ*, requiring heating stages, ion guns, evaporators, leak valves, and dosing ports to modify the surface. Multiple viewing ports let you see what is going on inside and most SPMs use a viewing camera as well to observe the SPM stage.

2.2 Scanning Probe Microscopy

Scanning probe microscopes have become an essential tool in the field of nanoscience. The basic principle for all scanning probes is the interaction between the probe and the sample. The resolution of SPM techniques is limited by the geometrical shape of the probe. Ideal probes have a cone-shaped tip and try to have the smallest radius at the tip apex.

As the probe is scanned across the surface of the sample, local interactions between them are measured. Piezoelectronics are used to precisely control the movement of the probes in 3 dimensions. An electronic control system is used to control the probe and is monitored by a computer. The computer takes the local measurements as a function of the probe position and turns that information into an image. The two main SPM techniques I used for my experiments are the Scanning Tunneling Microscope (STM) and the Atomic Force Microscope (AFM).

2.2.1 Scanning Tunneling Microscope

The STM invented by Binnig and Rohrer [6] gathers electronic and topographic data of conductive samples with atomic resolution. The setup for most systems can be seen in Figure 2.2 A [14]. The schematic shows a sharp metallic probe normally made out of tungsten or platinum/iridium wire that is approached to a conductive surface. During the approach, a bias voltage is applied between the tip and the sample. When the tip is close enough to the surface, around a gap of a couple of angstroms away, then a current can be measured as the electrons from the material that is biased, tunnel through the gap.

This tunneling is a quantum mechanical effect. Electrons live at specific energy levels. Electrons fill up energy valleys in the materials (Figure 2.3 A) [15]. The top energy where electrons can sit is the Fermi energy level. At every energy level, there is a number of electrons that live in the range $\Delta\epsilon$ away from ϵ. The density of states (DOS) for a specific energy is the number of electrons in a given $\Delta\epsilon$ divided by $\Delta\epsilon$.

When the tip and sample are close together there is a vacuum barrier between them (Figure 2.3 B). Classically, an energy greater than the work function ϕ is needed to move an electron outside of a material. Quantum mechanically, the wave nature of the electron allows it to tunnel through the barrier instead. A cartoon representation of this behavior can be seen in Figure 2.4 [16].

Tunneling electrons still need a place to go, so a bias voltage is applied to either the tip

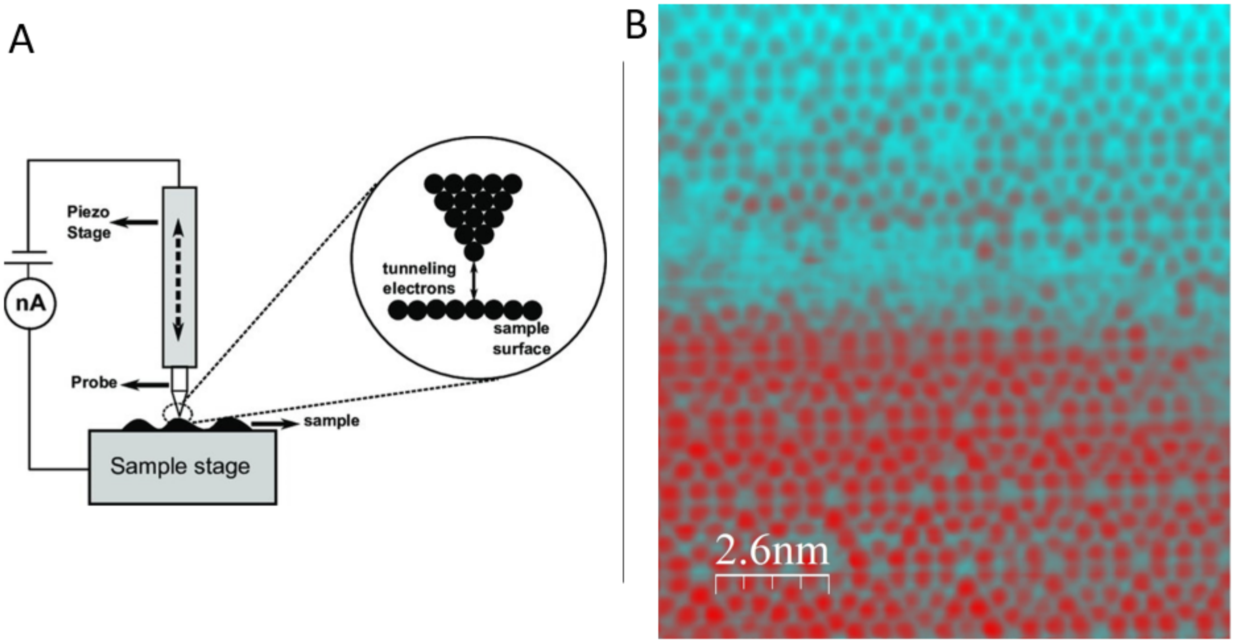


Figure 2.2: A) Illustration of an STM setup adapted from Marturi [14]. B) STM image of Si(111) (7x7) reconstruction taken by the Hollen Lab.

or sample. The bias voltage raises the Fermi energy of the material biased with respect to the other. This creates states for electrons to tunnel into (Figure 2.3 C).

In the vacuum gap, the wave function ψ of the electrons decays exponentially. Where m is the mass of the particle and $\hbar$ is Dirac's constant of 1.05x10^{-34} J·s.

$$\psi(z) = \psi(0) \exp - \frac{\sqrt{2m(\phi - E)}z}{\hbar} \tag{2.1}$$

The tunnel current (I_t) is based on the voltage difference between the tip and the sample. In Figure 2.3 C the electrons travel from the filled states of the sample into the empty states in the tip.

The elastic tunneling current from the sample to the tip for states of energy ϵ is:

$$I_t = -2e \cdot \frac{2\pi}{\hbar} \cdot |M|^2 (\rho_S(\epsilon) \cdot f(\epsilon))(\rho_t(\epsilon + eV) \cdot [1 - f(\epsilon + eV)]) \tag{2.2}$$

The factor of 2 in the front is from the spin of the electron, $-e$ is the electron charge,

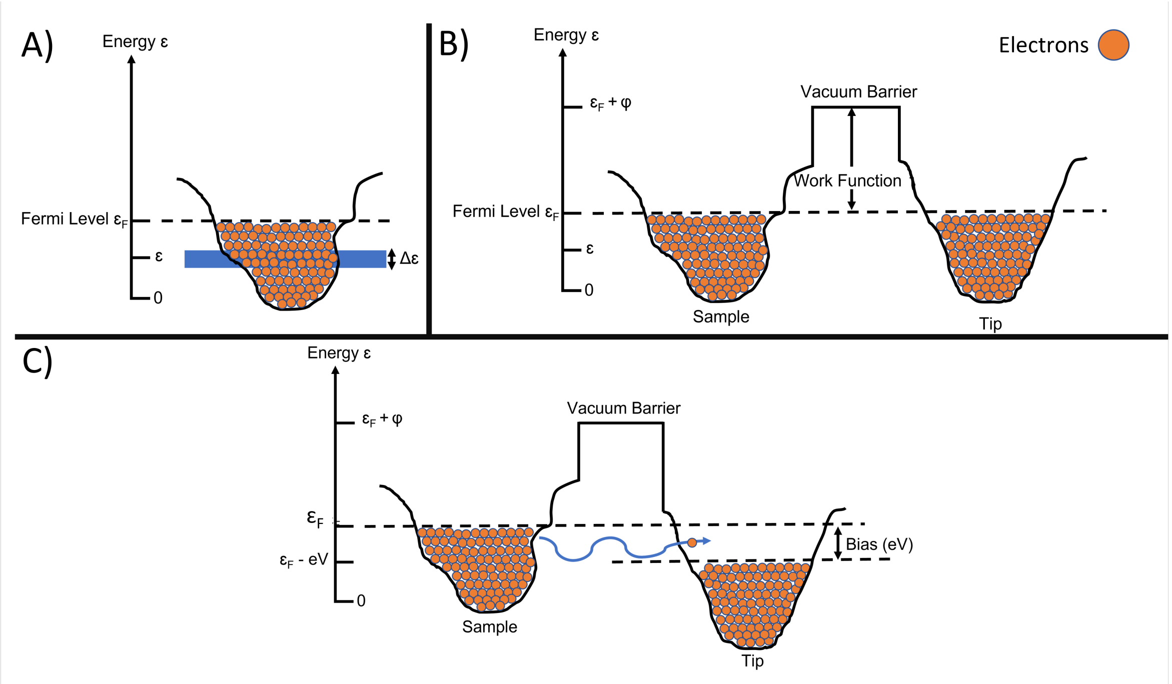


Figure 2.3: A) Representation of electrons filling up an energy valley up to the Fermi Level. B) Electrons in the energy valleys of the tip and sample with an energy barrier in between them caused by the vacuum gap. C) Applying a bias to the sample to raise the Fermi energy with respect to the tip's Fermi energy level. This creates empty states to tunnel into. Figures redesigned from the Hoffman Group's STM explanation [15].

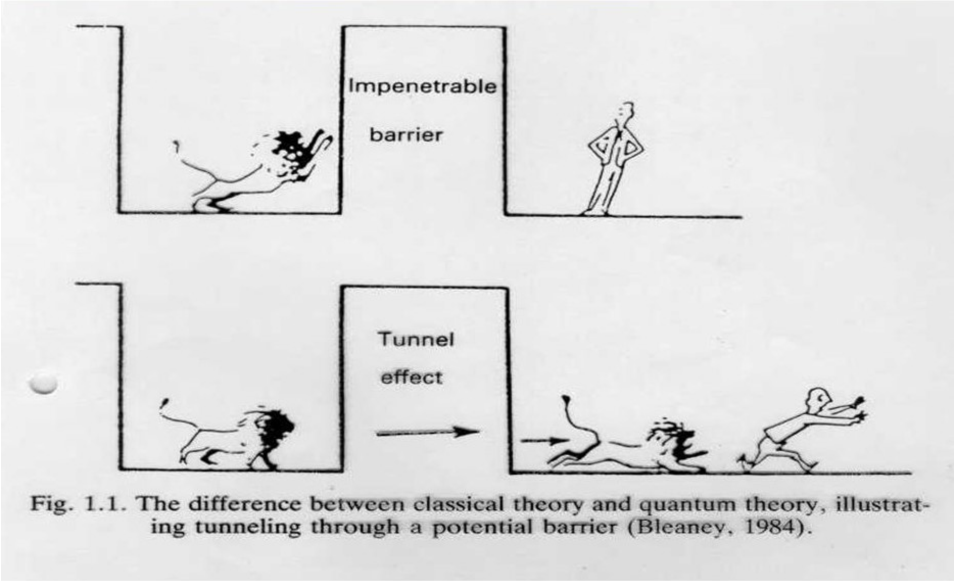


Fig. 1.1. The difference between classical theory and quantum theory, illustrating tunneling through a potential barrier (Bleaney, 1984).

Figure 2.4: Illustrating the difference of tunneling through a potential barrier. From Bleaney 1984

$2\epsilon/\hbar$ comes from the time-dependent perturbation theory, Equation 2.3 is the Fermi function $f(\epsilon)$, and Equation 2.4 is the tunneling matrix $|M|^2$.

$$f(\epsilon) = \frac{1}{1 + e^{\epsilon/k_B T}} \tag{2.3}$$

$$M = \frac{\hbar}{2m} \int_{sample} (\psi_S^* \frac{\partial \psi_t}{\partial z} - \psi_S \frac{\partial \psi_t^*}{\partial z}) dS \tag{2.4}$$

By applying a negative sample voltage of $-V$ there is a dominant tunneling current from sample to tip. There is also a smaller tunneling current flowing from the tip to the sample. To find the total tunneling current between tip and sample we have to add both current and integrate over all energies (ϵ):

$$I_t = -\frac{4\pi e}{\hbar} \int_{-\epsilon_F}^{\infty} |M|^2 \rho_S(\epsilon) \rho_t(\epsilon + eV) \cdot [f(\epsilon) \cdot [1 - f(\epsilon + eV)] - [1 - f(\epsilon)] \cdot f(\epsilon + eV)] d\epsilon \tag{2.5}$$

For measurements that are made at low temperatures and with a bias of $-eV$, this current is approximated to three energy regions: Above the biased Fermi level of the sample ($0 < \epsilon$), between the biased Fermi level of the sample and the Fermi level of the tip ($-eV < \epsilon < 0$), and below the Fermi level of the tip ($\epsilon < -eV$).

The relevant energy region to integrate over to find the tunneling current is between the biased Fermi level of the sample and the Fermi level of the tip, where $-eV < \epsilon < 0$. If the bias applied were a positive voltage then the range of integration would be from $0 < \epsilon < eV$.

$$I_t \approx -\frac{4\pi e}{\hbar} \int_{-eV}^{0} |M|^2 \rho_S(\epsilon) \rho_t(\epsilon + eV) d\epsilon \tag{2.6}$$

Typically STM tips are made out of a material that has a flat DOS within the energy range of the sample's Fermi surface. This means we can treat the DOS of the tip as a constant in our integral:

$$I_t \approx \frac{4\pi e}{\hbar}\rho_t(0)\int_{-eV}^{0}|M|^2\rho_S(\epsilon)d\epsilon \tag{2.7}$$

Since the tip and the sample have their own independent DOS, both their wavelengths fall exponentially to zero in the tunnel barrier. So if the tip-sample distance is large enough then the matrix element $|M|$ can be treated as a constant.

$$I_t \approx -\frac{4\pi e}{\hbar}|M|^2\rho_t(0)\int_{-eV}^{0}\rho_S(\epsilon)d\epsilon \tag{2.8}$$

Making the assumption that the vacuum barrier is a square barrier and using the WKB (Wentzel, Kramers, Brillouin) approximation we simplify the tunneling matrix to:

$$|M|^2 = e^{-2\gamma} \tag{2.9}$$

Where γ is Equation 2.10, m is the mass of the electron, z is the distance between the tip and the sample, and ϕ is a mix of the work functions of the tip and sample.

$$\gamma = \int_0^z \sqrt{\frac{2m\phi}{\hbar^2}}dx = \frac{z}{\hbar}\sqrt{2m\phi} \tag{2.10}$$

Putting this all together, the tunneling current is approximated by:

$$I_t \approx \frac{4\pi e}{\hbar}\rho_t(0)\exp\frac{-2z}{\hbar}\sqrt{2m\phi}\int_{-eV}^{0}\rho_S(\epsilon)d\epsilon \tag{2.11}$$

The tunneling current falls off exponentially as z increases. With work functions ranging from 3eV to 5eV for metals a change of 1 angstrom can cause a change of one order of magnitude in the current. This sensitivity is the reason STM can produce high-resolution images.

We also see from Equation 2.11, that the tunneling current is proportional to the local DOS (LDOS) of the sample at the Fermi level, meaning that STM probes the LDOS of the sample.

STM Modes

The 2 typical modes of operation for the STM are constant current mode and constant height.

STMs are most commonly run in constant current mode where the measured current is set to a specific value. While the tip is rastered across the surface it encounters changes in “topography” and to keep the current constant a software controller uses a feedback loop to change the height of the tip to measure the setpoint current. The changes in height are then used by the software controller to create a topographical map of the surface which in STM measurements topography means a combination of the changes in height and electronic structures on the surface. If the tip is well prepared then this technique can achieve atomic resolution similar and produce images like the one in Figure 2.2 B.

In constant height mode, the feedback loop is turned off and the height of the probe does not change while scanning. Instead, the changes in the current are measured and mapped out. The main advantage of this mode is that higher scan rates can be achieved to collect the data and that is better for observing dynamic processes. Since the feedback loop is turned off there is a higher risk of crashing the probe.

There also exist multiple spectroscopic techniques that are performed by STM systems like IV curves and dI/dV curves that provide information on the electronic structure of the materials. By measuring the current at different bias voltages or different heights the electronic states that can be tunneled into can be observed. These techniques are useful to find the bandgap and the local density of states of materials.

2.2.2 Atomic Force Microscopy AFM

STM produces stunning atomic resolution images, but the major limitation is that the technique only works on conductive materials. To solve this issue Binnig, Quate and, Gerber invented the AFM to be able to study the surface of bulk insulators as well [7]. Their technique measured the short and long range forces felt by a sharp probe while approaching

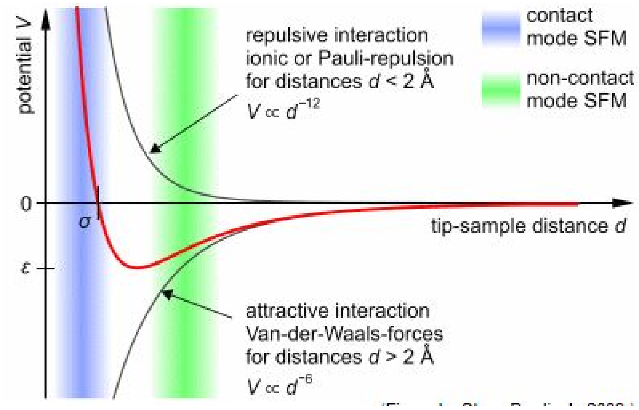


Figure 2.5: The Lennard Jones potential as a function of distances between the tip and the sample. The Pauli repulsion term and van-der-Waals forces are also shown. The regime for contact mode AFM is shaded in blue and the regime for non-contact AFM is shaded in green. Adapted from Pawlizak *et al* [17].

a sample surface regardless of its conductivity.

The forces felt by the probe are a combination of short range chemical forces from the overlap of electron wave functions and from the repulsion of ion cores and long range van-der-Waals forces caused by induced dipoles. These forces are described by the Lennard-Jones potential:

$$V_{Lennard-Jones} = 4\epsilon[(\frac{\sigma}{d})^{12} - (\frac{\sigma}{d})^{6}] \tag{2.12}$$

Here the d is the distance between the tip and the sample, ϵ is the depth of the potential minimum, and σ is a constant distance between the particles when $V_{Lennard-Jones} = 0$. The positive term describes the repulsive interactions and the negative describes the attractive interactions seen in Figure 2.5 [17]. Magnetic or electrostatic forces can also occur depending on the materials of the tip and the sample.

The setup for AFMs are similar to STMs, but instead of a tunneling tip AFM uses a force-sensing cantilever shown in Figure 2.6. The cantilever has a known spring constant k_N and a sharp tip mounted at the end. The forces bend the cantilever following Hooke's law seen in 2.6. The deflection of the probe Δz is measured as the tip of the cantilever interacts

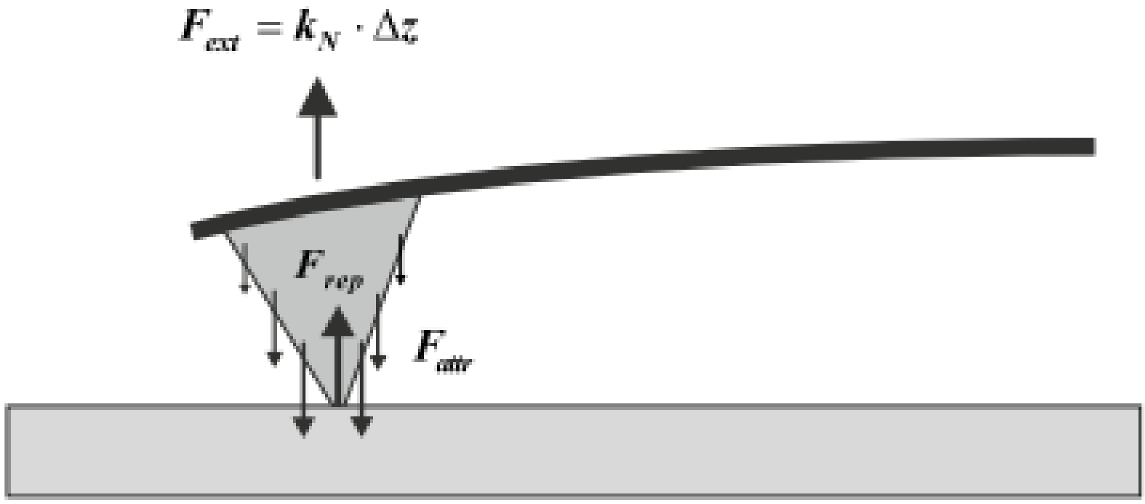


Figure 2.6: Schematic of the AFM probe showing Hooke's law and the equilibrium of the forces relevant to contact mode AFM. Adapted from Meyer *et al* [18].

with the surface. The force is used as an input signal to regulate the height with a feedback loop.

There are multiple different types of AFM systems, but the most common one is a table-top AFM that uses a laser beam detection system to measure the deflection of the cantilever as in Figure 2.7 [14]. The laser beam is reflected off the back of the cantilever and shines on a position-sensitive photodetector. As the tip is scanned across the surface, the forces between the tip and the sample deflect the cantilever. Those deflections change the position of the laser on the photodetector and that information is an input signal used to create a feedback loop where the controller adjusts the tip height to maintain a constant cantilever deflection. The changes in height are recorded and the controller uses those to generate topographic maps of the surface features.

AFM Modes

The most common operation modes for AFM are contact and non-contact mode. The ranges for these modes can be seen in Figure 2.5. In contact mode, the force acting between the tip and sample is used as the imaging signal. As the probe approaches the surface of the sample,

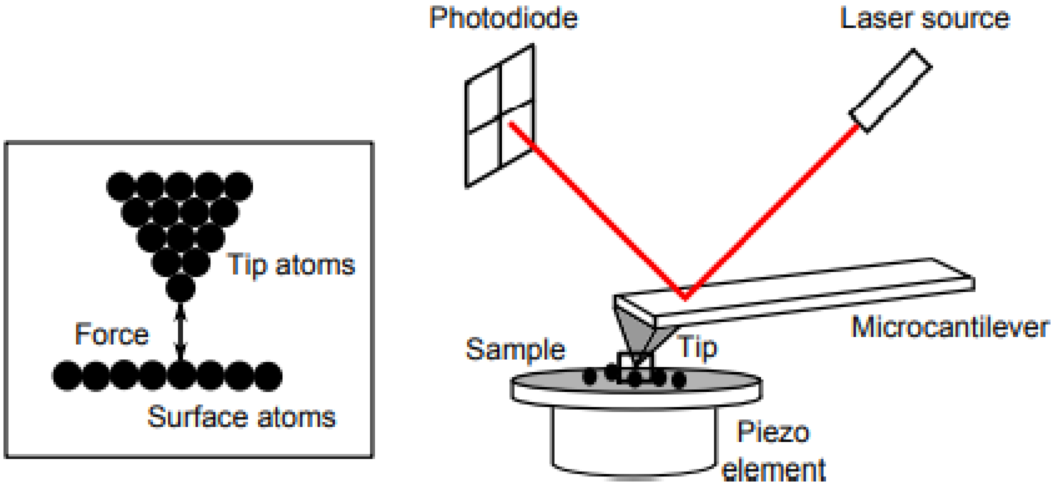


Figure 2.7: Illustration of an AFM setup. Adapted from Marturi [14].

the outermost atoms of both materials start to attract each other. Eventually, the distance between the tip and the sample is small enough that the Pauli exclusion principle triggers and starts to repulse the atoms deflecting the whole cantilever. Figure 2.6 shows how the forces are kept balanced by the controller to keep the deflection constant while scanning [18].

In non-contact AFM (NC-AFM) mode, a cantilever is mounted on an actuator and set to oscillate at its natural frequency. This mode has two operation methods seen in Figure 2.8: constant frequency shift and constant amplitude.

The feedback loop for the constant frequency shift mode is shown in Figure 2.9 [18]. An actuator is driven at a specific amplitude (A_{exc}) and fixed frequency ω_0 that is different from the natural frequency of the cantilever. The amplitude of the cantilever changes as the tip interacts with the sample. These changes in amplitude are used as a feedback signal to collect a topographical image while keeping the frequency constant. The changes in phase can also be used to reconstruct a phase image.

In constant amplitude mode, the cantilever is set to oscillate at a set amplitude by an actuator. A phase-locked loop measures the electrical signal proportional to the oscillation of the cantilever and applies that signal to the actuator after shifting the phase and amplifying

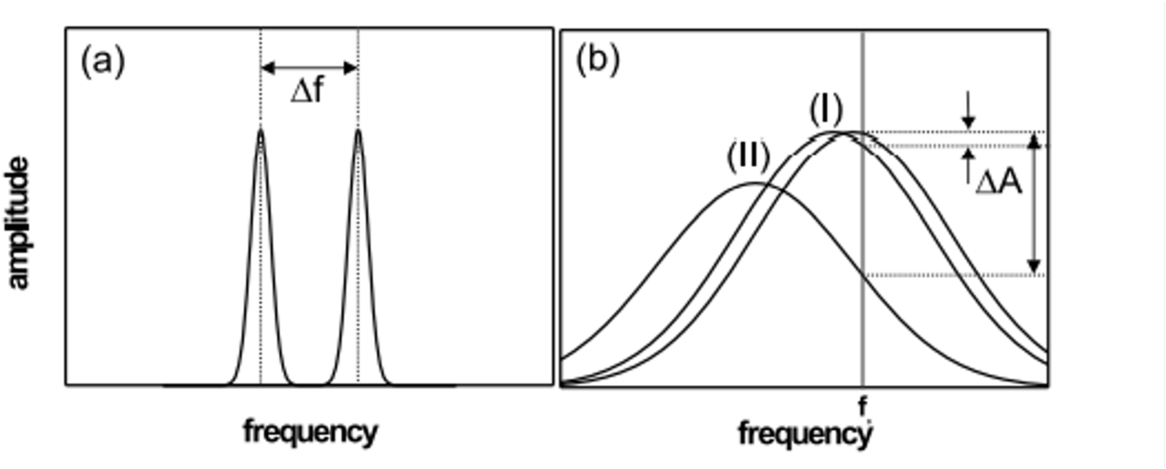


Figure 2.8: Amplitude versus Frequency curves representing the two operation modes in NC-AFM. a) Constant amplitude: The frequency change from the tip-sample interactions is detected. b) Constant frequency shift: The change of amplitude is detected at a constant frequency shift. Adapted from Meyer *et al* [18].

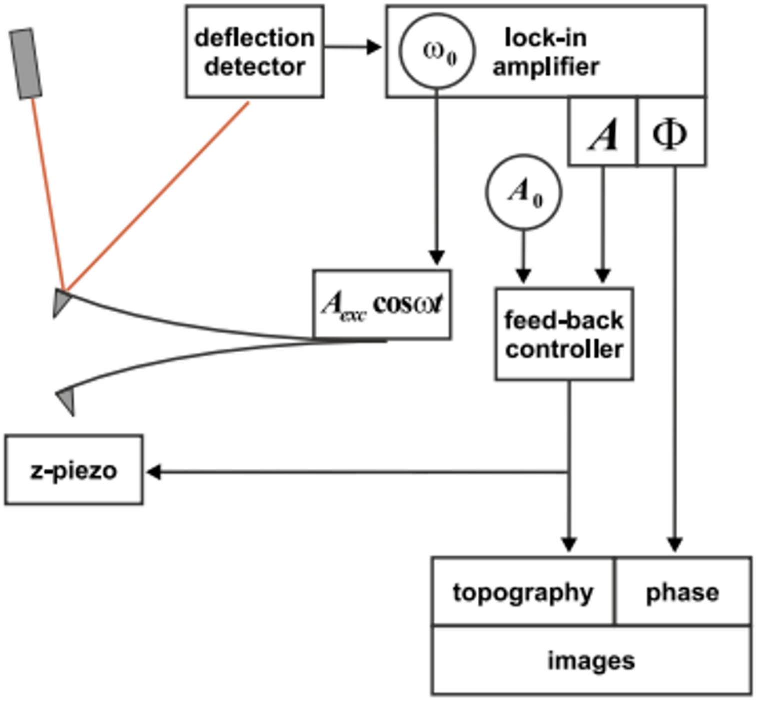


Figure 2.9: Schematic of the constant frequency shift feedback loop. Adapted from Meyer *et al* [18].

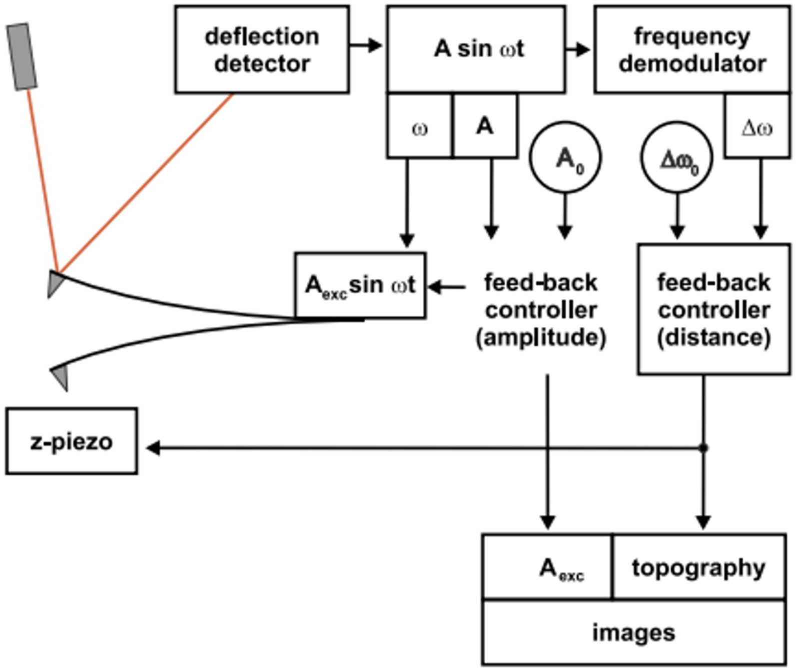


Figure 2.10: Schematic of the constant amplitude feedback loop. Reproduced from Meyer *et al* [18].

the signal. The tip-sample interactions change the frequency of the cantilever. To measure the change of frequency a fast frequency demodulator is employed, as introduced by Albrecht *et al* [19]. The change of frequency is used as the input signal to control the tip-sample distance. A schematic of this feedback loop can be seen in Figure 2.10 [18].

Relationship between tip-sample forces and frequency shifts

In NC-AFM the frequency shift is the measured observable. To understand how NC-AFM works, it is important to relate the frequency shift to the forces acting between the tip and the sample. Giessibl has published detailed reviews that explain the basic operations of the frequency shift modulation modes [20–22].

The motion of the cantilever, seen in Figure 2.11 with an effective mass m^* and a spring constant k can be described by a weakly disturbed harmonic oscillator.

The deflection of the cantilever is $q'(t)$ and the tip-sample distance is $q(t)$. The closest

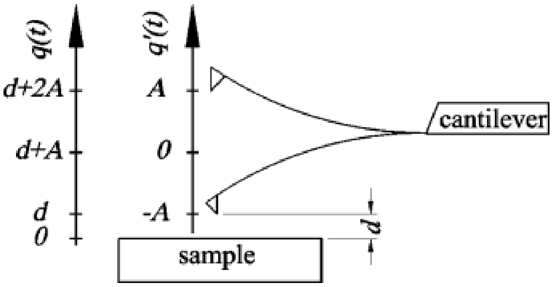


Figure 2.11: Schematic of an oscillating cantilever whose minimum tip-sample distance is d and the amplitude is A. Reproduced from Giessibl [20].

point to the sample us $q = d$ and $q(t) = q'(t) + d + A$. The Hamiltonian of the cantilever is:

$$H = \frac{p^2}{2m^*} + \frac{kq'^2}{2} + V_{ts}(q), \tag{2.13}$$

where $p = m^* dq'/dt$. The unperturbed motion is given by:

$$q'(t) = A\cos\left(2\pi f_0 t\right) \tag{2.14}$$

The frequency is given by:

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{k}{m^*}} \tag{2.15}$$

If the force gradient $k_{ts} - \partial F_{ts}/\partial z$ is constant during an oscillation cycle the frequency shift is:

$$\Delta f = f_0 \frac{k_{ts}}{2k} \tag{2.16}$$

In constant amplitude mode k_{ts} is not constant during the oscillation cycle, so a perturbation approach can be used to solve for Δf, Giessibl employed canonical perturbation theory and found Δf to be:

$$\Delta f = \frac{f_0}{kA^2}\langle F_{ts} q' \rangle \tag{2.17}$$

The terms between the brackets are averaged over one oscillation cycle.

Another approach used to calculating Δf is to solve the equations of motion of the cantilever with an effective mass μ^* and a spring constant k:

$$\mu^* \frac{d^2 q'}{dt^2} = -kq' + F_{ts}(q') \tag{2.18}$$

The motion of the cantilever $q'(t)$ is periodic and is expressed as a Fourier series with fundamental frequency f:

$$q'(t) = \sum_{m=0}^{\infty} a_m \cos{(m2\pi f t)} \tag{2.19}$$

Plugging $q'(t)$ into the equation of motion in Equation 2.18 gives:

$$\sum_{m=0}^{\infty} a_m[-(m2\pi f)^2 \mu^* + k] \cos{(m2\pi f t)} = F_{ts}(q') \tag{2.20}$$

Multiplying both sides by $\cos{(l2\pi f t)}$ and integrating over one oscillation period $t = 1/f$ gives:

$$a_m[-(m2\pi f)^2 \mu^* + k]\pi(1 + \delta_{m0}) = 2\pi f \int_0^{1/f} F_{ts}(q') \cos{(m2\pi f t)} dt \tag{2.21}$$

The orthogonality of the angular functions seen in Equation 2.22 is used to integrate the left-hand side of Equation 2.20

$$\int_0^{2\pi} \cos{(mx)} \cos{(lx)} dx = \pi \delta_{ml}(1 + \delta_{m0}) \tag{2.22}$$

With a weak perturbation $q'(t) \approx A\cos{(2\pi f t)}$ with $f = f_0 + \Delta f$, $f_0 = (1/2\pi)\sqrt{k/\mu^*}$, and $|\Delta f \ll f_0$. The first-order perturbation ($m = 1$) of the frequency shift is given by:

$$\Delta f = -\frac{f_0}{kA^2} \int_0^{1/f} F_{ts}(q') \cos{(2\pi f_0 t)} dt = -\frac{f_0}{kA^2} \langle F_{ts} q' \rangle \tag{2.23}$$

For small amplitudes, the frequency shift is independent of the amplitude and is proportional to the tip-sample force gradient seen in Equation 2.16. For amplitudes larger than the tip-sample force range, the frequency is a function of amplitude $\Delta f \propto A^{-1.5}$. When the amplitudes are larger than the range of the relevant forces used then it is useful to introduce a normalized frequency shift γ:

$$\gamma(z, A) = \frac{kA^{2/3}}{f_0}\Delta f(z, A) \tag{2.24}$$

A more detailed calculation of this term is done by Giessibl and Bielefeldt [21]. This normalized frequency shift is useful when comparing results recorded with different experimental parameters. The units of γ are the geometrical mean to the units of force and potential. This has been shown by Ke *et al* [23] and derived by Giessibl *et al* [21] as well.

Durig demonstrated that the force versus distance curve can be reconstructed from the frequency shift versus distance curves without the need to know the force law [24]. The force curve can also be recovered from the frequency curve using a matrix inversion. Sader and Jarvis introduced an inversion formula that is valid for large and small amplitudes compared to the interaction lengths. The Sader-Jarvis method converts frequency shifts to interactions forces $F(z)$ or energies $U(z)$ [25]:

$$F(z) = 2k\int_z^\infty (1 + \frac{A^{\frac{1}{2}}}{8\sqrt{\pi(t-z)}})\Omega(t) - \frac{A^{\frac{3}{2}}}{\sqrt{2(t-z)}}\frac{d\Omega(t)}{dt}dt \tag{2.25}$$

$$U(z) = 2k\int_z^\infty \Omega(t)((t-z) + \frac{A^{\frac{1}{2}}}{4}\sqrt{\frac{t-z}{\pi}} + \frac{A^{\frac{3}{2}}}{\sqrt{2(t-z)}})dt \tag{2.26}$$

where $\Omega(z) = \Delta\omega(z)/\omega_{res}$. These formulas can be used for any amplitude A.

AFM Probes

The cantilever design for AFM has changed through the years. Initial designs used tunneling [6, 26], capacitance [27, 28], and laser beam detection [29–31] methods to figure out the deflection of the probe. These methods relied on an external detection system which limited the imaging size. [32]

Piezoresistive cantilevers, seen in Figure 2.12 offer a detection scheme that does not

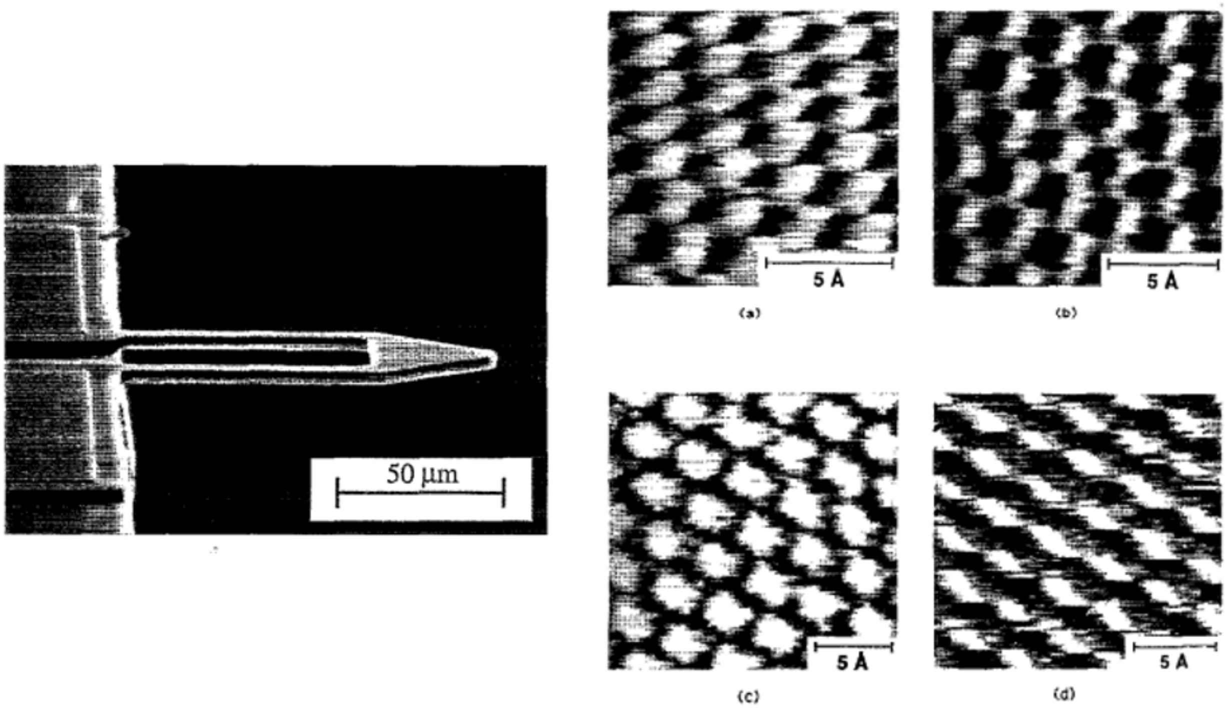


Figure 2.12: Piezoresistive AFM probe and atomic resolution images collected by this probe. a)graphite, b)boron nitride, c)molybdenum disulfide, d)tantulum diselenide.Adapted from Tortonese *et al* [32].

require aligning an external detector [32]. These cantilevers made from silicon exhibit a strong piezoresistive effect that makes it easy to measure the changes in bulk resistivity as the cantilever deflects [33]. These are especially useful for UHV and low temperatures where other detection schemes are difficult to implement.

While imaging in non-contact mode the stability of the frequency is the most important part when it comes to achieving atomic resolution. The piezoresistive probes have been used to image atomic resolution on multiple inert surfaces (Figure 2.12) [32]. Giessibl *et al* were the first to use these probes to obtain atomic resolution on the silicon(111) (7x7) reconstruction (Figure 2.13 A) [34], but noticed that the method still lacked stability [35].

This led Giessibl to design the qPlus probe, an alternative to the piezoresistive cantilever [36]. The qPlus probe is made from a quartz tuning fork that is normally found in wristwatches. These tuning forks are much stiffer than the silicon cantilevers used for AFM probes before which means that they oscillate at much lower amplitudes. Giessibl realized

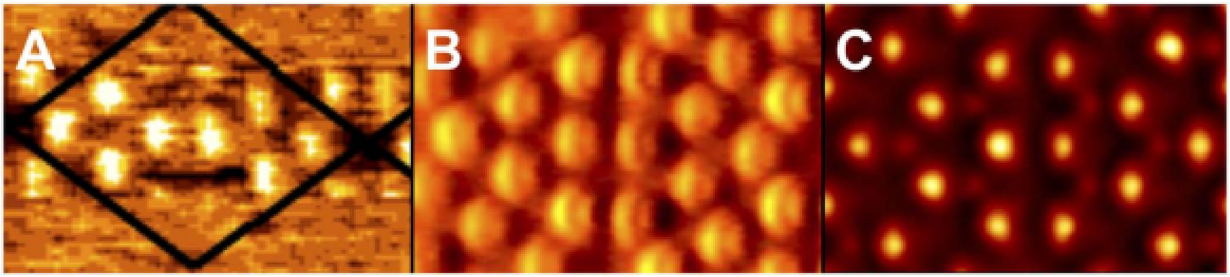


Figure 2.13: A) First Non-contact image of the Si(111) (7x7) reconstruction obtained with a piezoresistive Si cantilever, B) Si(111) (7x7) reconstruction using a qPlus probe, C) Si(111) (7x7) reconstruction using a qPlus probe with a functionalized CO tip. Adapted from Giessibl *et al* [33].

that utilizing smaller amplitudes improved the signal-to-noise ratio and attenuated the long-range forces felt by the cantilever increasing the spatial resolution. The result of using small amplitudes can be seen in Figure 2.13 B [34]. The qPlus probes are the most commonly used AFM sensors for low-temperature UHV systems.

The basic components of the qPlus probe can be seen in Figure 2.14 [37]. The quartz tuning fork is mounted on an actuator. One tine of the fork is fixed in place by epoxying it to the mount. The other tine is free to oscillate and resembles a conventional AFM cantilever. A sharp tip is attached to the end of the free tine. The deflection of the qPlus cantilever is measured by recording the current to keep the electrode on each tine at a constant potential. [34]

NC-AFM became an important tool to characterize nanostructures at an atomic scale [38–41]. Then in 2008 Gross *et al* were able to study the chemical structure of a single pentacene molecule using high-resolution NC-AFM (HR-AFM) by functionalizng a qPlus probe. Figure 2.15 shows their results comparing the resolution of STM and HR-AFM images of pentacene.

To achieve atomic resolution using the HR-AFM, it is necessary to operate in the short-range repulsive regime 2.5 where the Pauli exclusion principle dominates, because the long-range van-der-Waals and electrostatic forces do not contribute to the atomic contrast. To operate the HR-AFM in this regime it is necessary to use a cantilever with high stiffness,

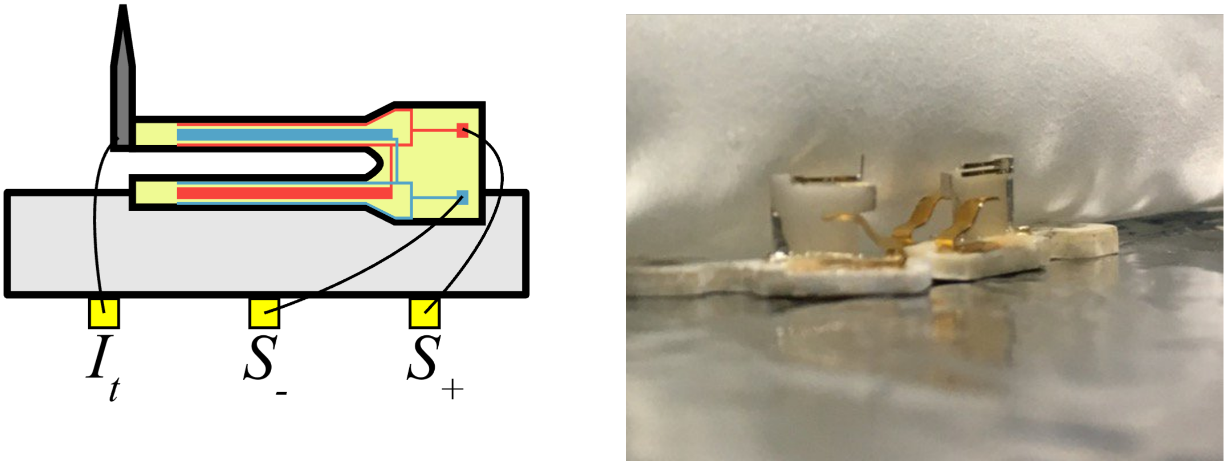


Figure 2.14: Schematic of a qPlus probe adapted from Stirling [37]. Picture of the Hollen Lab's qPlus Probes.

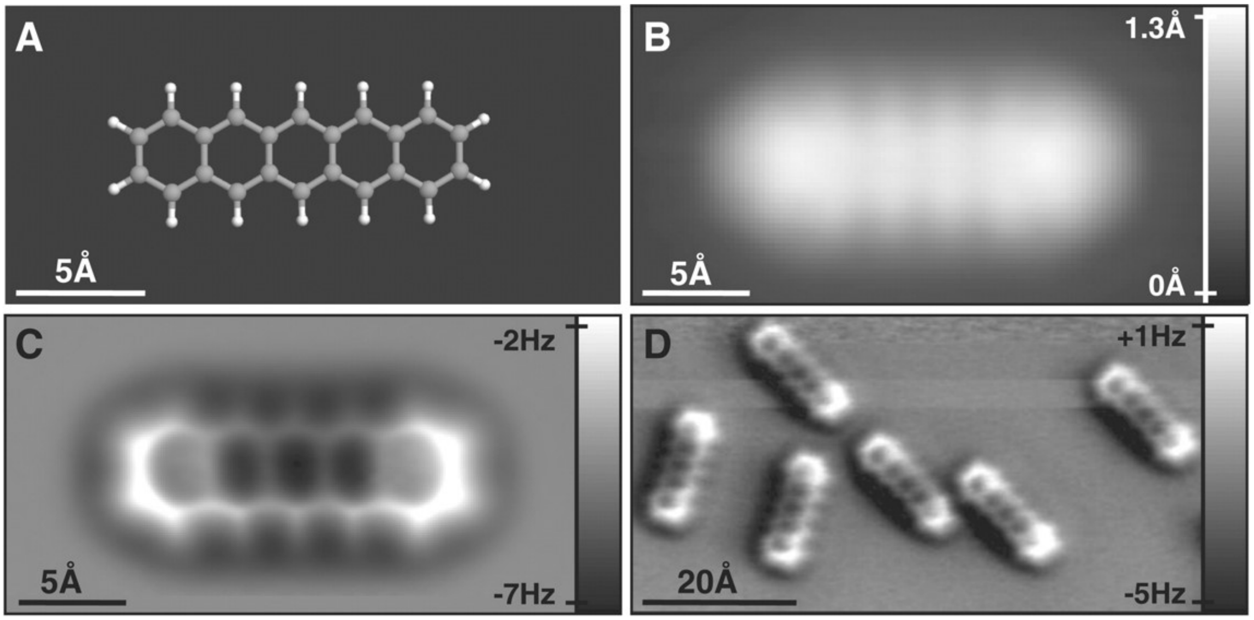


Figure 2.15: STM and NC-AFM imaging of pentacene on Cu(111). A) Ball-and-stick model of pentacene. B) STM image of pentacene using a CO functionalized probe. C and D) NC-AFM of pentacene obtained using a CO functionalized probe. Adapted from Gross *et al* [43].

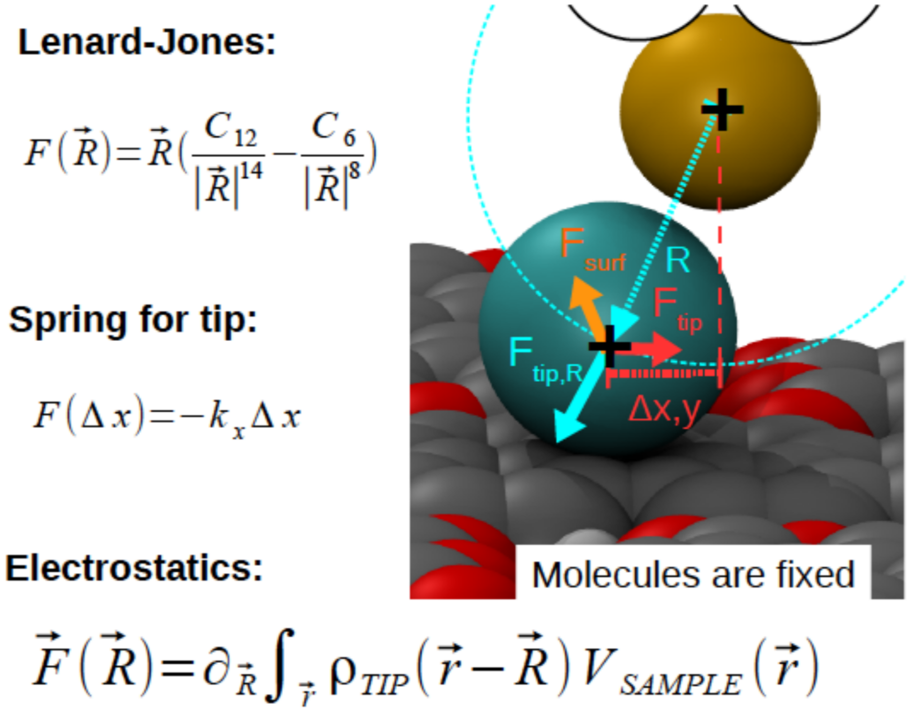

Lenard-Jones:
$F(\vec{R})=\vec{R}(\frac{C_{12}}{|\vec{R}|^{14}}-\frac{C_6}{|\vec{R}|^8})$
Spring for tip:
$F(\Delta x)=-k_x\Delta x$
Electrostatics:
$\vec{F}(\vec{R})=\partial_{\vec{R}}\int_{\vec{r}}\rho_{TIP}(\vec{r}-\vec{R})V_{SAMPLE}(\vec{r})$
F surf
R
F tip
F tip,R
Δx,y
Molecules are fixed

Figure 2.16: Forces acting on the relaxing probe particle represented by the blue ball, which is the last atom of the flexible tip apex. The forces are a) The spring force to keep the probe particle below the last atom of the metal tip, which is represented by the yellow ball. b) The Pauli repulsion and van-der-Waals forces acting between the probe particle and the fixed atoms on the substrate that are modeled by the Lennard Jones Potential. c) And the electrostatic forces between the sample and the charged probe particle if the apex is charged. Adapted from the Hapala *et al* [44].

operate with small oscillation amplitudes [42], and functionalize the probe. Gross *et al* [43] explored different tip terminations but found that CO molecules provided the best lateral resolution of the pentacene molecules.

Modeling the Forces in HR-AFM

Hapala *et al* found that the features seen in the HR-AFM images is due to the strong lateral relaxations of the particle attached to the apex of the tip [**?**, 44]. The particle relaxes away from the area where the Pauli repulsion is strong these are the sharp features seen in the HR-AFM images. Understanding the interactions between the functionalized probe and the molecule allows us to explain the experimentally recorded HR-AFM images.

The simulation calculates the Lennard Jones potential seen in Equation 2.12 finding the Pauli repulsive and van-der-Waals attractive forces, calculates the position where the probe particle relaxes over the surface to minimize its total energy, and generates a set of simulated HR-AFM images.

These simulated images can change based on parameters like the stiffness of the probe particle, the charge of the probe particle, parameters like stiffness and frequency of the cantilever, and oscillation amplitudes.

Ruben Perez's group has studied the interactions between functionalized probes and the molecules imaged by HR-AFM and created an AFM simulation package that has been incorporated into a database called QUAM-AFM [45, 46]. Their simulations split the tip-sample interactions into four contributions: short-range, electrostatic, van-der-Waals, and the tilting angle of the CO functionalizing the probe.

Carracedo-Cosme *et al* explain the contribution of each force when modeling the tip-sample interaction forces [45]. The van-der-Waals contribution comes from the atomic geometry given by the attractive term of the Lennard-Jones potential. The electrostatic contribution comes from the charge density of the molecule being imaged and the C and O atoms. The short-range contributions are from the Pauli repulsion between the overlap of the charge densities of the CO molecule and the molecule on the surfaces. The contribution from the CO tilting is modeled by having the CO molecule act like a spring that deflects when approaching the molecule on the surface.

2.3 Sample Preparation

For SPM experiments proper sample preparation and a clean environment are crucial for achieving atomic resolution. This section contains information on the exfoliation of 2D materials that were the main samples I used for the experiment I present in Chapter 3. This section also contains information on metal preparation procedures that I used for the experiments in Chapter 4.

2.3.1 Preparation of 2D Materials

2D materials are those that are a single atom or molecule thick. Popular materials in this 2D family are graphene, transition metal dichalcogenides, borophene, and hexagonal boron nitride. These 2D materials exhibit unique quantum states because the electronic states are confined to a single layer thickness. The growing family of 2D materials started with the isolation of graphene by Nosolev and Geim [47] and now there are thousands of materials discovered or theoretically thought to exist. Data mining has predicted over 1000 potential 2D material candidates [48], but most of these have not been isolated.

These materials can be isolated from bulk samples or grown from base materials. These 2D materials can be stacked and combined to create heterostructures that utilize their unique electronic properties. These heterostructures are the building blocks of semiconducting devices. [49] The electronic properties of 2D materials and their atomic flatness also make them an interesting playground for studying chemical reactions [8–10, 50, 51].

Mechanical Exfoliation

The simplest way to isolate 2D materials is by mechanical exfoliation. The bulk counterparts to these 2D materials have weak van-der-Waals forces acting between each layer. The forces are weak enough that by using simple scotch tape the layers can be separated from the bulk. In the Hollen lab, this is the most common method we use to exfoliate 2D crystals.[1]

The main steps for mechanical exfoliation using the scotch tape method are shown in Figure 2.17.

- Grab a long strip of scotch tape. Around 6 inches should be plenty.
- Place one end of the scotch tape strip on a bulk crystal. Apply a small amount of pressure to remove air bubbles between the tape and the top of the crystal. Using carbon tip tweezers has worked the best for this in the past.

[1] Shout out to Tan, Page, Chris, and Caitlyn for all their hard work on improving our exfoliation techniques.

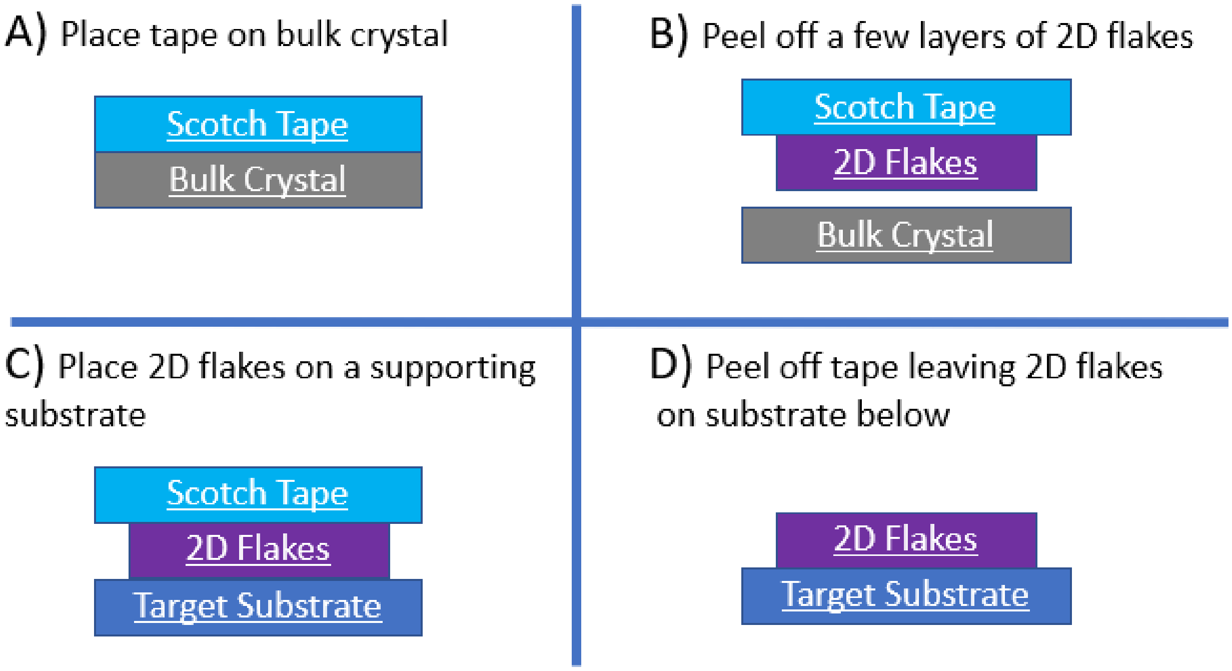

A) Place tape on bulk crystal
Scotch Tape
Bulk Crystal
B) Peel off a few layers of 2D flakes
Scotch Tape
2D Flakes
Bulk Crystal
C) Place 2D flakes on a supporting substrate
Scotch Tape
2D Flakes
Target Substrate
D) Peel off tape leaving 2D flakes on substrate below
2D Flakes
Target Substrate

Figure 2.17: Schematic showing the steps to mechanically exfoliate a 2D crystal.

- Peel off the tape from the crystal. The outermost layers will peel off as well. This leaves a fair amount of bulk material stuck on the tape. Keep exfoliating the material on the tape using spots on the tape strip that are empty until a patch of the tape has material that is barely visible.

- Place that barely visible patch on top of a target support substrate. Typically placed on a silicon dioxide/silicon chip.

- Remove the taper leaving behind the flakes on the silicon dioxide surface.

After the exfoliation process, the 2D flakes are observed with an optical microscope. The solicon dioxide layer used as a support has an oxide layer thickness of 300 nm. Under the optical microscope the oxide layer appears light purple. This exfoliation method often leaves a large number of flakes on the surface. Figure 2.18 shows how the color contrast changes on MoS_2 as you increase the layer number [52]. This change in color contrast with different layer thicknesses is similar for all 2D materials. Optical images are taken of the flakes that

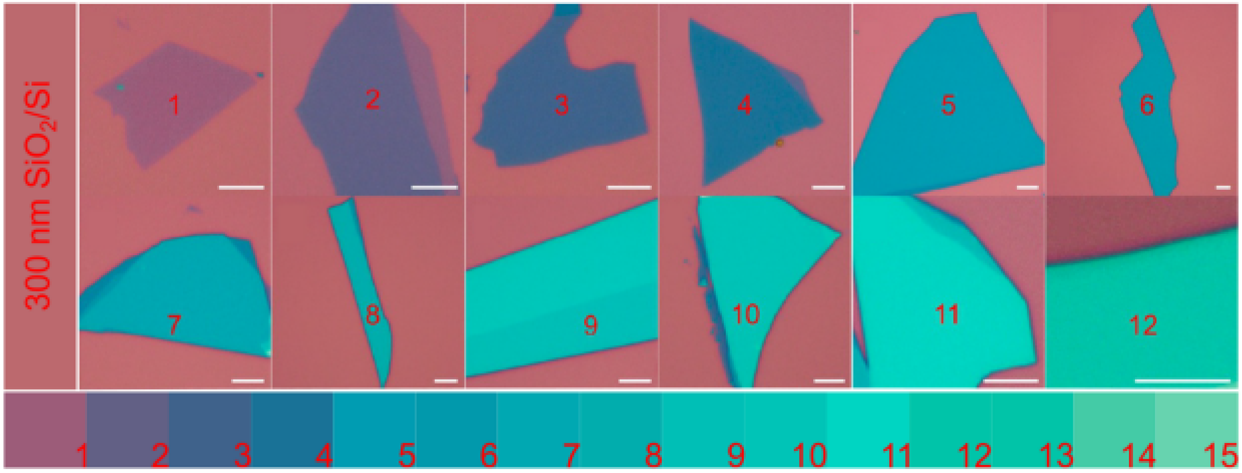


Figure 2.18: Color optical images of monolayer to 15 layered MoS2 on 300 nm SiO_2/Si substrate. The scale bars are 5 μm for images from monolayer to 11 layers and 10 μm for the image of 12 layers. Adapted from Li *et al* [52].

could be of potential use for any project.

In the Hollen lab, the images of the flakes are stored in the local computer in the clean room and uploaded to a shared drive on OneDrive. These images make up our flake library through the years. Most of the flakes exfoliated in the Hollen lab have been used for a specific project almost immediately.

While searching for the best mechanical exfoliation process, we have tried implementing different variations to these steps. Although there is not a lab consensus on what variations work best, we have used an alternative to scotch tape, Nitto Blue tape which does reduce the amount of tape residue on the underlying substrate, but does not necessarily yield a higher amount of desirable flakes compared to scotch tape.

Peeling the tape with a twist or adding a heating step before peeling off the tape are other variations that have been tried before in the lab, but no obvious improvement to the quality of the flakes yielded was noticed. These variations and others, like how much pressure is applied, humidity in the room, and source crystal, do have an effect on the flakes exfoliated, but this process is mostly a multiple trial effort. To get the best flake just repeat the process until a desired flake is found. A good project in the future would be to make this exfoliation

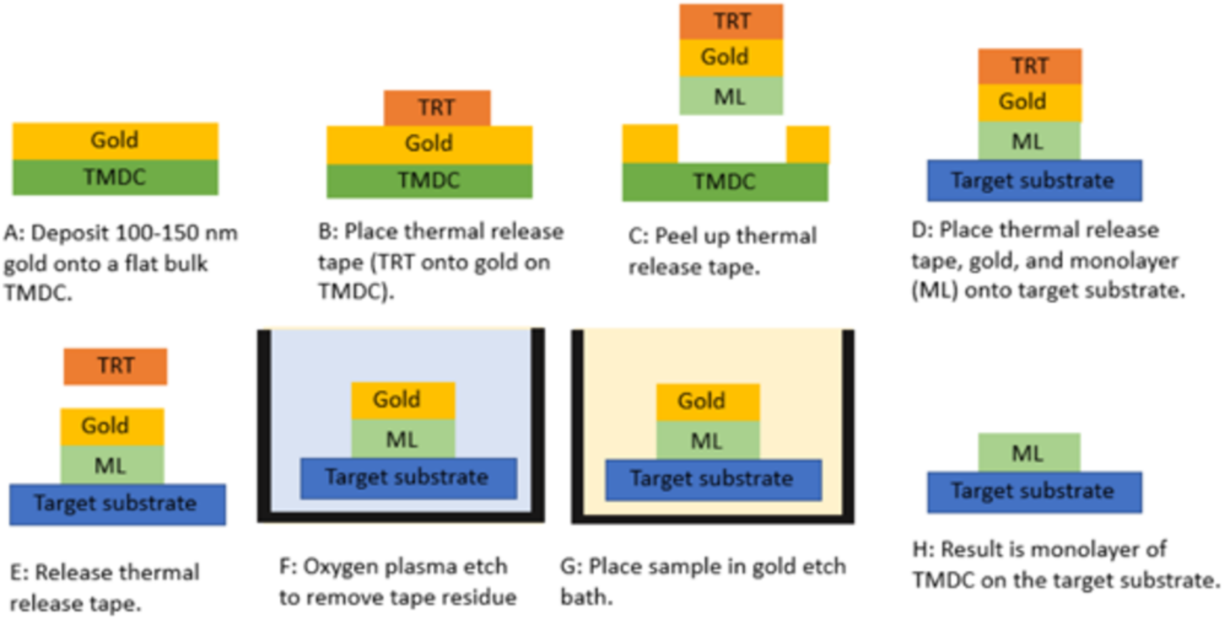

TRT
Gold
TMDC
ML
Target substrate
A: Deposit 100-150 nm gold onto a flat bulk TMDC.
B: Place thermal release tape (TRT onto gold on TMDC).
C: Peel up thermal release tape.
D: Place thermal release tape, gold, and monolayer (ML) onto target substrate.
E: Release thermal release tape.
F: Oxygen plasma etch to remove tape residue
G: Place sample in gold etch bath.
H: Result is monolayer of TMDC on the target substrate.

Figure 2.19: Schematic of the Gold Assisted Transfer Process. Adapted from Meditz [53].

procedure more reproducible in our lab, similar to the work being done at the Quantum Material Press (QPress) facility at Brookhaven National Lab to be able to create any flake in a controlled environment.

QPress is a modular cluster tool designed to study heterostructure materials and 2D materials. Tools in the QPress cluster include an exfoliator, cataloger, library, stacker, and characterization tools like AFM and Raman spectroscopy. Every section of the QPress is held under a vacuum or an inert atmosphere to be able to handle air sensitive materials. The tools are also being automated to maximize the speed and reproducibility of flake creation.

Gold Assisted Exfoliation

Another exfoliation technique implemented in the Hollen lab was gold assisted exfoliation mostly developed by Caitlyn Meditz [53] and adapted from Desai *et al* [54]. This method produces large area flakes of transition metal dichalcogenides (TMDC) thanks to the affinity that chacolgen atoms have with gold [55, 56]

The steps for the exfoliation process (Figure 2.19) [53].

- Place heat resistant tape (Capton tape) on top of the bulk crystal to be exfoliated. Apply pressure to remove any air bubbles.
- Figure 2.19 A: Evaporate a layer of gold around 100 nm to 150 nm of gold on top of the crystal/tape stack.
- Figure 2.19 B: Place a small piece of thermal release tape on top of the places where the gold is covering the bulk crystal.
- Figure 2.19 C: Peel the stack off the heat resistant tape leaving a stack of thermal release tape/gold/TMDC.
- Figure 2.19 D: This new stack is placed on a target supporting substrate.
- Figure 2.19 E: The stack is heated to release the thermal release tape.
- Figure 2.19 F: Oxygen plasma etch to remove excess tape residue. We did not see much improvement in cleanliness after this step.
- Figure 2.19 G: Etch away the gold in a potassium iodide and iodine wet solution. This process takes about 4 minutes
- Figure 2.19 H: Rinse the final TMDC/target substrate in an acetone bath for 10 minutes to remove any residues from the etching process.

Figure 2.20 shows some examples of 2D materials exfoliated with this technique. The technique provided a good yield of large area flakes, but they had a lot of residue and the number of layers was hard to control.

2.3.2 Flake Transfer Station

These exfoliated 2D materials exhibit various intrinsic physical and electronic properties. Those properties are modified when the 2D flakes are stacked to create heterostuctures or when the underlying substrate underneath the flake is changed. A flake transfer station

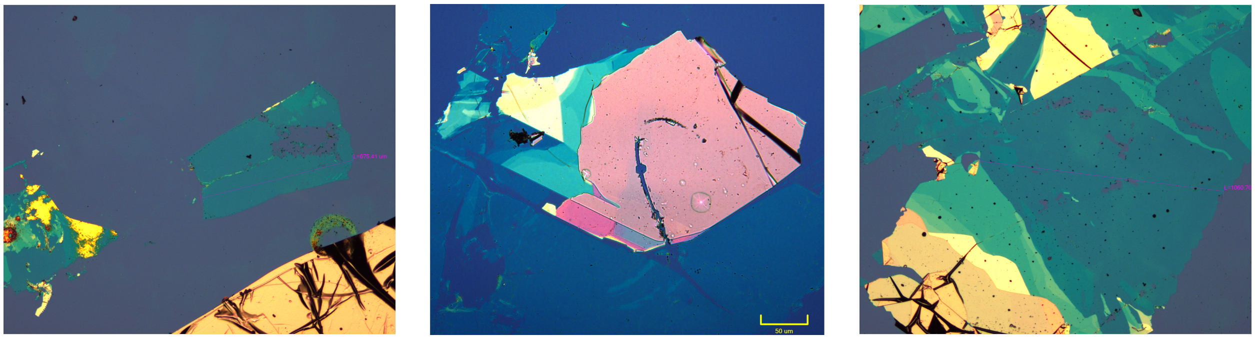


Figure 2.20: Optical Images of MoS_2 flakes exfoliated using the Gold Assisted Transfer Process

can pick up and move exfoliated flakes to new locations to create these layered structures. The Hollen lab has a manual transfer system from HQ Graphene. The parts of the transfer system are labeled in Figure 2.21 [53]. The sample and stamp stages tilt, rotate, and move in XYZ directions using micrometers. The micrometers control the exact location where the picked-up flake is meant to be placed. An attached camera above the stages helps the user visualize the whole flake transfer process. Substrates are held in place on the sample stage using a built-in vacuum and the stage has a built-in heater that can raise the temperature up to 200 ° C. [2]

The main transfer method used in the Hollen lab is a polymer assisted transfer. The steps for the procedure used in our lab were adapted from the literature by Page Waldo and Caitlyn Meditz (Figure 2.22) [53].

The polymers used for this transfer are a thin polycarbonate (PC) film and a small (around 0.5 cm by 0.5 cm) piece of polydimethysiloxane (PDMS). The PC film is made from a PC solution that is 6% of polycarbonate pellets dissolved in chloroform. The PC solution should be made in small batches that last roughly 6 months, after that time the PC solution starts to dry up. The PDMS piece is cut from a larger film that is made by combining a 10:1 ratio of Sylgard silicone elastomer base and Sylgard silicone elastomer curing agent.

The steps to make the PC film are:

[2] Since the camera is not supported very well, at high temperatures, the camera shakes making it difficult to visualize what is going on.

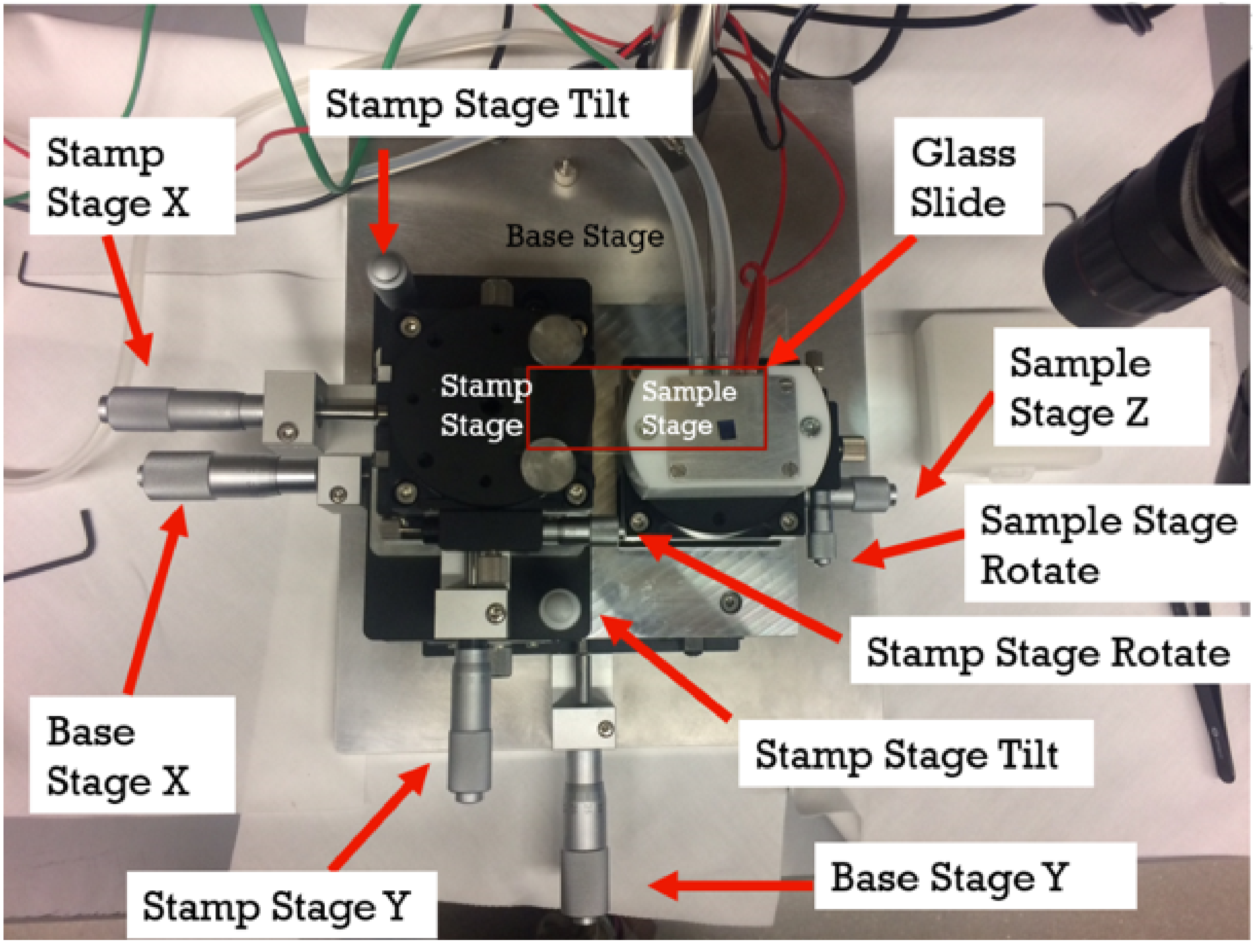


Figure 2.21: Flake transfer station from HQ Graphene setup with labels. Adapted from Meditz [53].

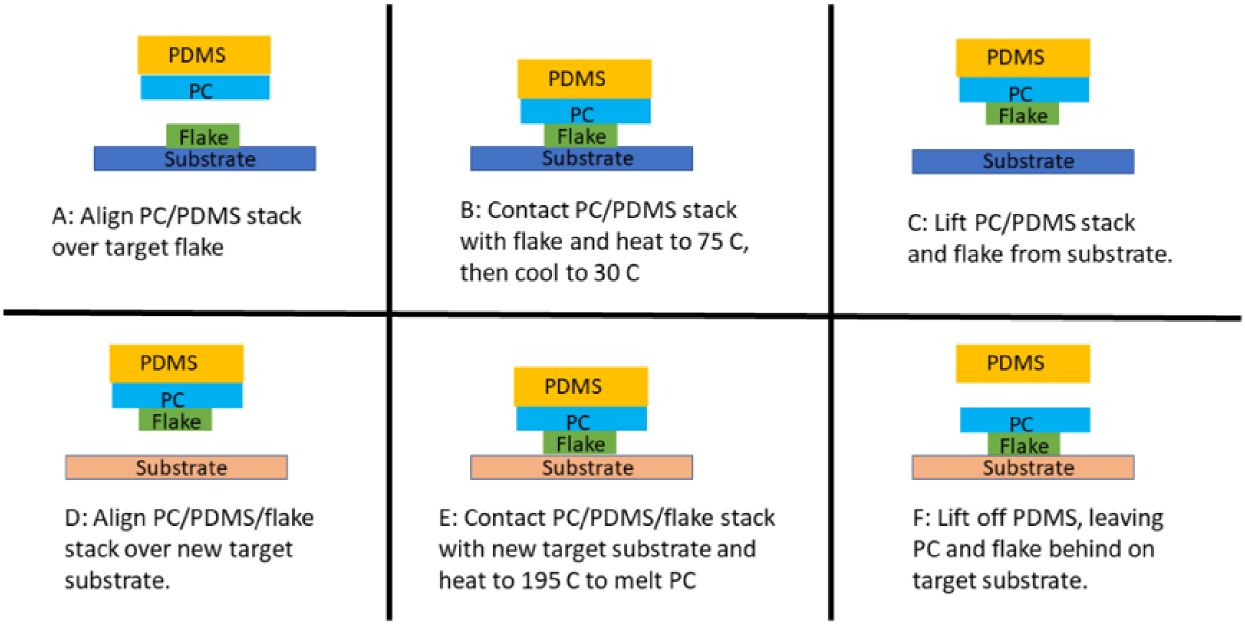

PDMS
PC
Flake
Substrate
A: Align PC/PDMS stack over target flake
PDMS
PC
Flake
Substrate
B: Contact PC/PDMS stack with flake and heat to 75 C, then cool to 30 C
PDMS
PC
Flake
Substrate
C: Lift PC/PDMS stack and flake from substrate.
PDMS
PC
Flake
Substrate
D: Align PC/PDMS/flake stack over new target substrate.
PDMS
PC
Flake
Substrate
E: Contact PC/PDMS/flake stack with new target substrate and heat to 195 C to melt PC
PDMS
PC
Flake
Substrate
F: Lift off PDMS, leaving PC and flake behind on target substrate.

Figure 2.22: Schematic of the Polymer Flake Transfer Procedure. Adapted from Meditz [53].

- Clean two glass slides.
- Place a drop of the PC solution on one end of a glass slide.
- Press the drop using the second glass slide.
- Slide the glass slides apart quickly to leave a thin PC film coating both glass slides.
- Let the PC films dry for at least 15 minutes before use.

The flake transfer process starts by preparing the polymer stack.

- Place the PDMS square on one end of a clean glass slide.
- Cover half of a second glass with scotch tape then make a small window on the tape that is slightly larger than the PDMS square.
- Peel the tape off the glass slide and place over a glass slide that has a PC film.
- Peel the tape off to pick up the PC film. The PC film should cover the small window cut in the second step.

- Place the PC film over the the glass slide with the PDMS aligning the small window with PC over the top of the PDMS square.
- Use extra tape if need to make sure the stack does not move. Tape down the sides of the scotch tape holding the PC film on to the glass slide. Make sure not to cover the back of the glass slide where the polymer stack sits so light can pass through.

The polymer stack is then ready to be used for the flake transfer and can be placed on the stamp stage. Following the steps in Figure 2.22, the polymer assisted transfer has been used in the Hollen lab to create stacks of 2D materials like graphene on hexagonal boron nitride and to move flakes to other supporting substrates. A common setback in this method is that monolayer flakes of 2D materials were difficult to pick up. Placing another flake partially on top of the monolayer helped pick up the stack, but this did not work all the time. I also ran into an issue of only partially picking up larger flakes as well. After a flake is picked up by the polymer stack it is fairly easy to place on top of a new surface.

The last step not described in Figure 2.22 is to rinse off the PC using a chloroform soak. This has to be a delicate step because the flake can either get washed off or folded during the rinse if not making proper contact with the new layer underneath.

2.3.3 Metal

Metal substrates like gold and copper crystals are commonly used surfaces in SPM experiments. Some examples of a clean gold surface are seen in Figure 2.23. Many of these metal surfaces present interesting physical phenomena that make them the perfect environment to act as supporting substrates to observe molecular self assembly [57], nanostructure formation [58] and surface catalysis procedures [9].

With time, contaminants can accumulate on the surface of the metals and this can be detrimental when it comes to imaging them with SPM. To remove the contaminants the surface of the crystals is cleaned with a series of sputter and anneal cycles. Before starting the sputter cycles, make sure to close off any valves to an ion pump. Ion pumps should not

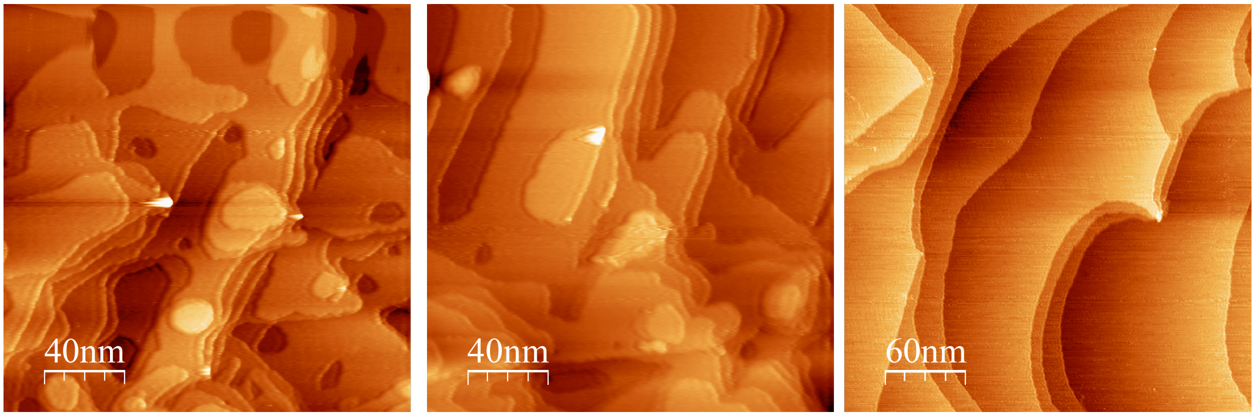


Figure 2.23: STM images of prepared gold surfaces after sputter and anneal cycles.

be operated at high pressures or exposed to high atomic number gases to avoid any damage to the pump. An inert gas, commonly argon, is leaked into the chamber through a leak valve to pressures of $5x10^{-5}$mbar. The argon gas is ionized by a voltage of around 0.6 keV to 1keV typically produced by an ion gun that is positioned in front of the metal crystal. The ions accelerated toward the crystal clear off any contaminants on the surface and can clear off the top metal atoms.

The crystal is then annealed to temperatures around 550 ° C to 600 ° C to smooth out the surface after the ion bombardment. The annealing process can also bring out impurities from the bulk to rise to the surface. Multiple sputter/anneal cycles should be performed to ensure a clean crystal surface. Controlled annealing creates steps and terraces on the surface of the crystal. These areas are the perfect playground to deposit and grow molecules to observe their molecular self-assembly [57] and to find adsorption and nucleation sites. Different crystal cuts allow for different steps and surfaces to be produced, but also controlling the temperature and time of the anneal can help control the number of steps and size of the terraces. It is always important to make sure not to go near the melting point of the crystal.

2.3.4 Freeze-Pump-Thaw Cycles and Thiophene Dosing

The Freeze-Pump-Thaw Cycle is an effective method to degas solvents, solutions, and liquid reagents. This technique is used in Chapter 3 to degas a thiophene solvent which we dosed into the vacuum chamber.

The Freeze-Pump-Thaw Cycle procedure:

- In a chemical fume hood, place a couple of ml of the solvent into a container that can be sealed and attached to the gas manifold.
- Once attached, freeze the solvent using liquid nitrogen. Should take less than a minute.
- Open the valve to a roughing pump to pump out any impurities found in the line.
- Let the solvent thaw this releases more solvent vapors.
- After multiple iterations of these steps the line will be filled with a pure solvent vapor.
- Close the valve to the main liquid solvent.

After the Freeze-Pump-Thaw cycles we filled the gas manifold in the chamber with thiophene vapor. A leak valve was used to dose into the chamber directed by another line pointed towards the SPM stage. The chamber was pressurized to $5E^{-5}$ torr.

2.3.5 Molecular Deposition Using Physical Vapor Deposition

Physical vapor deposition is an easy way to deposit molecules on a target surface. The molecules or molecule mixture to be studied starts as a chemical powder. The powder is placed in a holder that is heated inside the UHV chamber and degassed to remove water and trapped gasses initially. After degassing, the sample is placed in the line of sight of a target sample surface and heated again. As the holder is heated the powder starts to sublimate and molecules are deposited on the target surface.

The implementation varies between setups, but the setup that I have seen work the best is the one done by Dr. Zahl at Brookhaven National Lab. Dr Zahl places a few crumbs of the chemical powder on a silicon wafer. The powder is sublimated or flashed off the surface by heating the silicon wafer. After the deposition, the leftover crumbs can be seen visually through a viewport giving you initial information on the amount of material deposited. Dr.

Pohl at UNH had a similar setup but instead used an envelope made from a molybdenum sheet to hold the powder and the sample was heated through a tungsten wire wrapped around the envelope. Each source is calibrated to ensure a controlled deposition. The molecular coverage needed for calibration can be determined by an STM or other surface characterization techniques like auger electron spectroscopy.

CHAPTER 3

Scanned probe microscopy studies of MoS_2 catalysis on insulating substrates

3.1 Introduction

Refineries transform crude oil into petroleum products that we use daily through a series of chemical and physical processes. One of the most important processes is sulfur removal also known as hydrodesulfurization (HDS) reactions. These HDS reactions reduce the sulfur concentrations of the initial crude oil and the final refined petroleum products. The initial elimination of sulfur from crude oil protects noble metal catalysts that are easily poisoned by sulfur. Refineries also use a final HDS process to make sure that their petroleum products meet specific sulfur level requirements set by strict government regulations.

The combustion of these petroleum products containing sulfur produces harmful sulfur oxide byproducts. High concentrations of sulfur oxides can harm plants and trees by deteriorating their leaves and decreasing their growth, can make it harder for humans to breathe, and if mixed with other particles in the atmosphere can create a haze of particular matter that if deposited can stain and damage materials. For these reasons, the EPA has been reducing the allowed sulfur levels in fuels in the past decades and now they are around 10 to 15 ppm for gasoline and diesel fuels respectively. These HDS reactions also prevent palladium and platinum catalyst from getting poisoned. This expands the lifetime and efficiency of these catalysts.

Molybdenum disulfide (MoS_2) is one of the most commonly used catalysts for HDS and has been used for oil feedstocks since WWII [59]. The widely accepted model is that the sulfur vacancies in MoS_2 edge sites are involved in the HDS reactions, but this model was developed using spectroscopic techniques that do not provide information on the structure, morphology, or active sites of the MoS_2 catalysts. This information is crucial when developing new and improved catalyst designs. Surface characterization techniques have provided more information on the growth, structure and morphology of these edges [60], but still lack the resolution to provide information about the active catalytic sites. In recent years scanning probe microscopy (SPM) techniques have demonstrated that they can localize sulfur vacancies and pinpoint where sulfur containing molecules interact with the MoS_2 catalysts. [60]

However, all STM experiments of MoS_2 catalysis have employed metal substrates, [61–63] which are not industrially relevant catalyst supports. Most of the substrates in STM experiments used have been Au(111) surfaces due to their inertness, its a typical substrate used for nanoparticle growth and its a conductive sample needed for STM experiments [64]. Although the underlying gold substrates do not strongly interact with the MoS_2, DFT studies do show that the bonding strength of S atoms is slightly increased with the presence of gold [60], leading us to question the role of the gold substrates.

Other studies on MoS_2 have shown that strain, grain boundaries, and sulfur vacancies play a role in catalysis reactions [65–71]. The strain caused by the underlying substrate and at grain boundaries changes the electronic bandgap of the MoS_2. Controlling the amount of strain applied can help design a better catalyst and selecting a proper underlying substrate can tune the amount of strain applied.

In this chapter I describe the work I did to investigate the roles of substrate, strain, and lattice defects on the HDS reaction on two novel experimental systems using scanning tunneling microscopy (STM) and non-contact atomic force microscopy (NC-AFM), both atomic-scale probes. Improving HDS catalyst design will rely on our knowledge of the catalyst on industrially relevant, insulating substrates that do not have an effect on the catalytic sites.

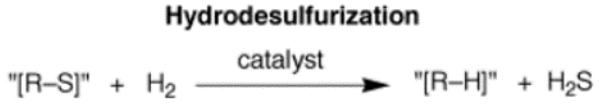


Figure 3.1: Hydrodesulfurization Reaction. R represents an alkyl, an alkane missing one hydrogen.

3.2 Scientific Background

Hydrodesulfurization is the catalytic chemical process that is used to remove sulfur from refined petroleum products. Removing the sulfur reduces the emission of sulfur dioxide when these products are combusted. [72] The early removal of sulfur from oil feeds can also prevent the poisoning of platinum and palladium catalysts down the process line. HDS is known as a hydrogenolysis reaction meaning it is also a hydrogenation reaction. Hydrogenation reactions occur when hydrogen reduces the double and triple bonds in hydrocarbons in the presence of a catalyst. The hydrogenolysis reaction cleaves the C-S bond resulting in the formation of C-H and H-S chemical bonds (Figure 3.1).

The most commonly used catalyst for this reaction is MoS_2 enhanced with either cobalt or nickel supported on alumina with high surface areas. [59]. Earlier models of these catalysts were developed by spectroscopic techniques that did not provide structural information about the catalyst or any information about the preferred adsorption sites of sulfur-containing molecules to react with. The catalytic activity of this process remains an active area of investigation and can provide insight on similar reactions that occur for hydrodenitrogenation and hydrodeoxygenation processes [60].

To understand HDS reactions and related catalysts there has been an emphasis on *in situ* characterization experiments. There are conflicting results between industrial and laboratory setting studies on how this reaction works [59]. The *in situ* characterization of HDS is limited to remote probes like spectroscopy. Chianelli's group at The University of Texas at El Paso studied industrial catalyst by synchrotron x-ray and high-resolution transmission electron microscopy (TEM). These studies determined the catalyst structure in the active phase [73]. The same group also found evidence that linkages between MoS_2 and the alumina supports destroy edge states [74]. X-ray photoelectron spectroscopy (XPS) compared a model catalyst to an industrial catalyst and found that they were similar [75]. Another approach used STM to correlate the concentration of the model catalyst to the activity of an electrocatalytic cell made using the imaged sample [63]. Based on these studies, there needs to be more emphasis on having agreeing experiments in both industrial and laboratory settings.

The structural models of this HDS are consistent: the active phase of this HDS reaction is a single layer of MoS_2 particles that interact with the alumina substrate underneath [60]. Investigating the interactions between the two-dimensional (2D) material catalysts and their surroundings in a controlled environment is needed to understand the differences between industry and laboratory findings.

SPM techniques have been to study the structure of these catalysts under controlled UHV environments. The first of these studies was done by Besenbacher, who performed an STM study of the structures of *in situ* grown nanoislands of MoS_2 and CoMoS on top of a Au(111) surface 3.2 [76]. Other surface characterization techniques have provided information on the growth and morphology of these catalysts on different substrates like carbon [50, 51] and alumina [60]

The advantage of using SPM techniques is that single defects can be imaged and SPM techniques can localize where molecules are interacting with the catalysts. Besenbacher's group studied the interaction of thiophene and these MoS_2 nanoislands. Thiophene was used because it is the simplest sulfur containing compound found in crude oil. The group performed STM experiments on the system before and after the thiophene dose (Figure 3.3 A) [8]. These STM experiments revealed one-dimensional edge states, or brim states, that demonstrated stronger binding and partial decomposition of the thiophene molecule (Figure 3.3 C) [8]. Their initial explanation was that sulfur vacancies play little role in the HDS reaction and that these brim states provide the mechanism for catalysis similar to noble

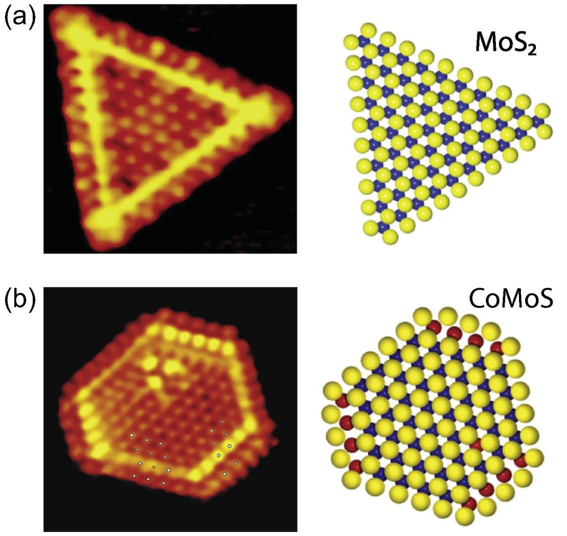


Figure 3.2: (a) Atom-resolved STM image of a triangular MoS_2 nanocrystal on Au(111) together with a ball model of the proposed edge structure reflecting Mo edge with a 100% S coverage (S: yellow, Mo: blue). b) Atom-resolved STM image of a truncated triangular Co-promoted (so-called CoMoS) nanoparticle. The interior part is MoS_2, but favorable substitution of Co is concluded to be at the S edge. (Co: red). Adapted from Grønborg *et al* [76]

metal catalyst. The discovery of these brim states led to the development of improved HDS catalyst named BRIM created by Besenbacher's group and Haldor Topsoe A/S.

STM experiments in different gas environments show the structural changes of the MoS_2 catalysts on Au(111) under HDS reactions [76,77]. Figure 3.4 shows how the structure of the MoS_2 is sulfided and sulfo reduced when exposed to hydrogen disulfide or hydrogen gas 3.4. Under ambient conditions this still holds true and Mom *et al* show that under HDS reaction conditions the structure of the edges changes to accomodate adsorbtion of the molecules that take part of the HDS reaction. Figure 3.5 shows the structure of the MoS_2 edges under various gas environments. These structural changes depend on the temperature, pressure, and original structure of the catalyst [77].

Besenbacher's proposed catalytic mechanism relies on the formation of one-dimensional edge states [8]. Supports and catalysts have been shown to have strong interactions between them. According to DFT calculations by Bollinger these edge states are intrinsic to the MoS_2 and the substrate underneath has little to no effect on the electronic structure of the

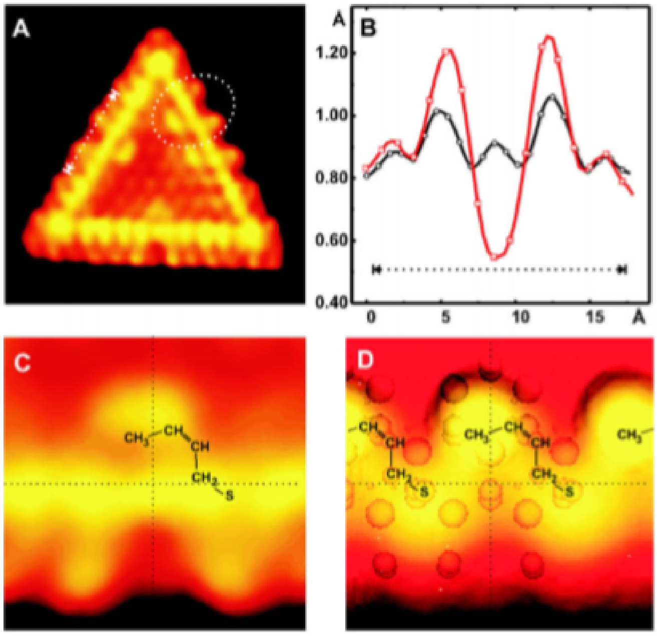


Figure 3.3: Thiophene is adsorbed on to a triangular MoS_2 nanocluster pre-exposed to atomic hydrogen. (A) Atom-resolved STM image (I_t =0.50 nA, V_t = 331 mV) showing species adsorbed at sites on the metallic edge states. The image dimensions are 50Å × 54Å. A bean-like structure is seen in a position adjacent to the bright brim. (B) STM line scans along the nearby edge protrusions of the clean edge (black) and an edge with a molecule adsorbed on the edge state (red). A decrease of 0.4Å of the protrusion located in front of the bean-like structure is observed, together with an increase of 0.2Å at the two neighboring protrusions. The associated with changes in the LDOS due to molecule adsorption. (C) Cut-out from the STM image in (A) illustrating the features associated with each molecule. (D) Simulated STM image from the DFT calculations of the structure, with individual molecules adsorbed in a repeated geometry along the edge. The hydrogenated thiophene species, C_4H_7S (cis-but-2-ene-thiolate), coordinated to the edge state primarily through the terminal S atom is seen to reproduce the details of the experimental STM image. Adapted from Lauritsen *et al.*[8]

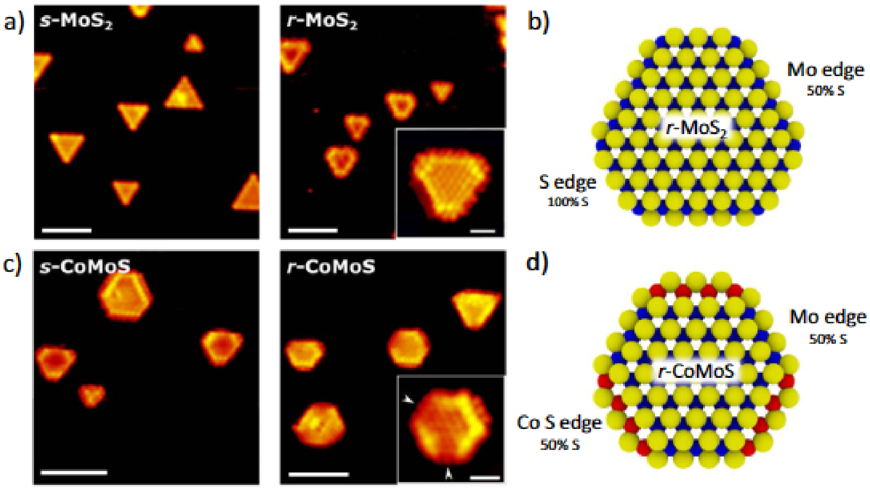


Figure 3.4: Reshaping effect of MoS_2 and CoMoS nanoparticles on Au(111) imaged with STM. a) MoS_2 nanoparticles in sulfiding conditions (s-MoS_2) and in sulfo-reductive conditions (r-MoS_2). Insert: atom-resolved STM image of a r-MoS_2 nanoparticle. b) Top view ball model showing an example of an r-MoS_2 nanoparticle. Ball model is shown without edge vacancies on Mo edges. c) CoMoS nanoparticles in sulfiding conditions (s-CoMoS) and in sulfo-reductive conditions (r-CoMoS). Insert: atom-resolved STM image of a r-CoMoS nanoparticle. d) Ball model of a r-CoMoS nanoparticle. Edges are shown without S vacancies and H adsorbates. Scale bars are 4 nm in large scale images, and 1 nm in the inserts, respectively. Color code: S: Yellow, Mo: blue, Co: red. Adapted from Grønborg *et al* [76]

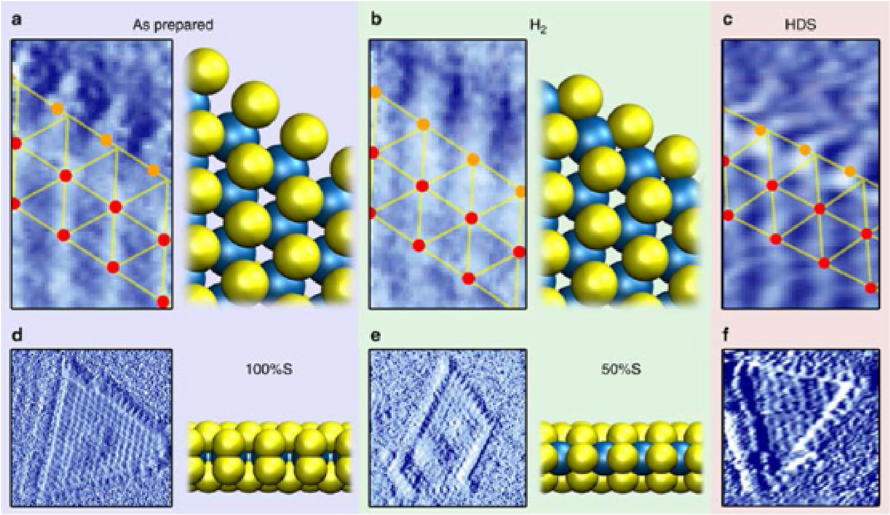


Figure 3.5: MoS_2 edge structure in various gas environments. a), b), and c) depict the averaged edge unit cell obtained from the original images seen in d), e), and f). Blue: Mo, yellow: S. Adapted from Mom *et al* [77]

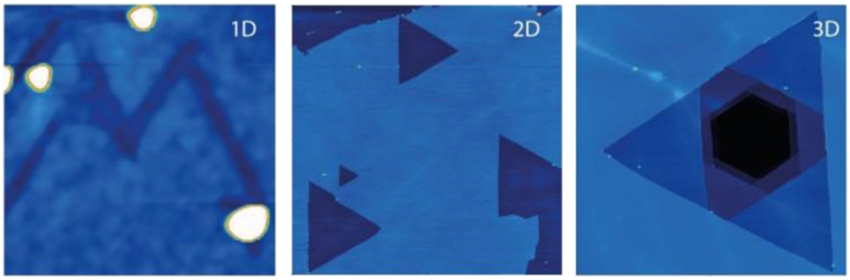


Figure 3.6: Steam vapor etching of Mo_2 flakes in a)1D, b)2D, and c)3D. Adapted from Wang *et al*[82]

MoS_2. [78] Other DFT calculations by Gronborg have shown that the substrate can affect the bonding of the MoS_2 flakes on the substrate. [76]

Looking more in detail at the electronic structure of MoS_2 DFT calculations by Davelou found that these metallic edge states exist on MoS_2 nanoribbons and these states change as the width of the ribbons increase. Hai studied these MoS_2 nanoribbons with STM and found that the location and binding to the underlying gold surface changed the electronic structure [65, 79]. These studies show that any electronic interaction of the catalyst with the substrate will play an important role in the mechanism.

Previous studies on the impact of substrates for MoS_2 catalysts found that MoS_2 interacts strongly with alumina substrates and forms Mo-O-Al linkages at MoS_2 edges, but these links are not formed on carbon or silica substrates making them easier to sulfide [74]. Controlling the edge structures of the MoS_2 catalysts is important when tuning the material for optimal chemical activity [80, 81]. New edges can be created by etching away the MoS_2 to create new catalytic sites that improve the electrocatalytic ability of MoS_2 catalysts (Figure 3.6) [82].

There has also been an interest in investigating how the catalytic activity of MoS_2 is improved by adding strain to the system [65–71]. It has been shown that the catalytic activity of MoS_2 for hydrogen evolution increases by a factor of three when the basal plane is activated by applying strain and combining that with the generation of S vacancies on nanopatterned Au substrates. [70] Even small factors of strain of about 0.02% are enough to improve the hydrogen evolution activity. [68] Other experimental results found that there is

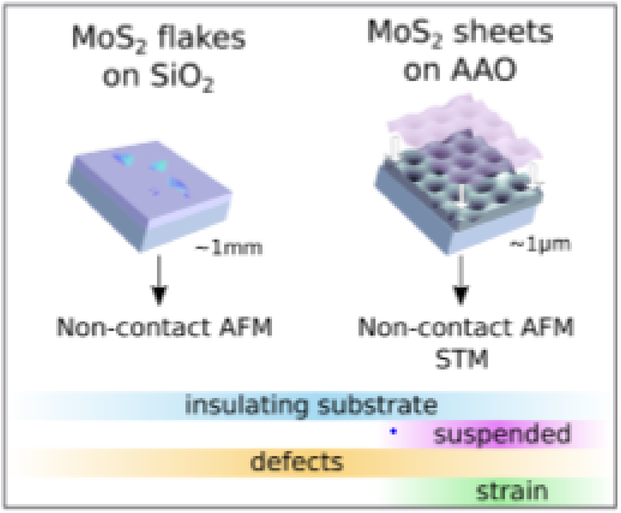


Figure 3.7: Schematic of the originally proposed experiment

limited role played by S vacancies, but there needs to be more work on how vacancies and strain affect the catalytic activity.

Recent progress in NC-AFM techniques have produced amazing atomic-scale images of single molecules. [12, 83–85] NC-AFM has the potential to study these catalytic systems on insulators that are industrially relevant in the future but also has not yet been developed. [86] Here I show the progress I made in studying HDS reactions using STM and NC-AFM on an industrially relevant substrate.

3.3 Experimental Design

For the original experiment, we developed two novel experimental systems shown in Figure 3.7: Flakes of MoS_2 on top of a silicon dioxide (SiO_2)/silicon (Si) substrate and MoS_2 on top of anodized alumina oxide (AAO). The flat SiO_2 substrate is a common support for exfoliating 2D materials like MoS_2. The AAO support was chosen so the MoS_2 can drape over the surface that is filled with nanoscale hills, valleys, and holes. On the AAO we see regions of strained and unsupported MoS_2.

The experimental plan to study the HDS reaction on these more industrially relevant supports was:

- Exfoliate flakes of MoS_2 on top of the support (SiO_2 or AAO).

- Image flakes using NC-AFM.

- Dose thiophene molecules into the chamber.

- Image flakes once more looking for thiophene binding sites.

For the HDS reaction, we chose to study thiophene, a simple and common sulfur-containing hydrocarbon found in crude oil. This study would also compare to the work done by Lauritsen *et al* [9]. Thiophene was dosed through a homemade Schlenk line integrated into our UHV system (LEWIS) after multiple Freeze-Pump-Thaw cycles described in Chapter 2. The Freeze-Pump-Thaw clear make sure we are dosing just pure thiophene into the chamber. The thiophene used for this experiment was obtained from Sigma Aldritch ($\geq$99%).

Understanding the role that insulating supports have on catalysts at an atomic scale has been difficult since electrical contact is needed to do STM. The small semiconductor flakes can be studied by STM, but making contact with them requires lithographic patterning techniques that introduce significant contamination. Recent advances in NC-AFM make it a great tool to replace STM to be able to perform these experiments.

Our goal was to use NC-AFM to image the flakes of MoS_2 before and after the thiophene dose. I wanted to explore the active catalytic sites of the MoS_2 where the thiophene would bind to. The results obtained by the work done by Besenbacher's group at Aarhus University showed that the bonded thiophene molecules appeared caused a topographical change in the original image by 20pm to 40pm that are seen in Figure 3.3 B.

Commercial NC-AFM systems like the SPM 150 Aarhus with KolibriSensor system have atomically resolved the Si(111)-(7x7) and the herrringbone reconstruction on Au(111). Figure 3.8 and Figure 3.9 show the capabilities of these commercial systems to resolve features which have corrugations on the order of 15 to 100 pm [87, 88]. The main assumption of

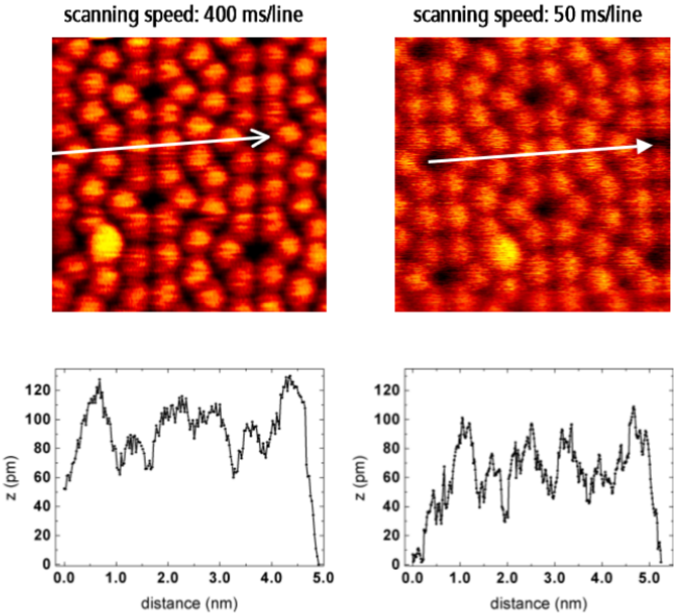


Figure 3.8: A corrugation up to 120 pm is typically observed. Adapted from SPeCS uploaded notes on their SPM 150 Aarhus with KolibriSensor system.[87]

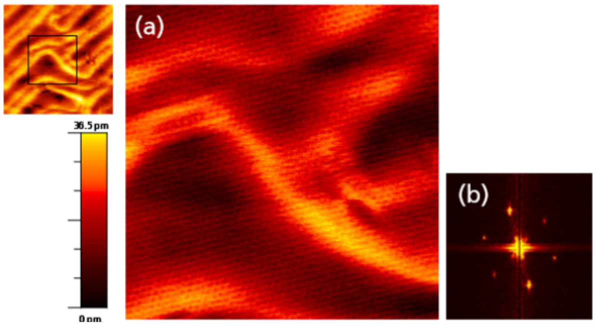


Figure 3.9: (a) High resolution NC-AFM image showing the atomic details of the herringbone reconstruction. (b) Fast fourier transformation of image (a) revealing the hexagonal ordering of the Au(111) surface. The atomic corrugation is about 15 pm. Adapted from SPECS uploaded notes on their SPM 150 Aarhus with KolibriSensor system.[88]

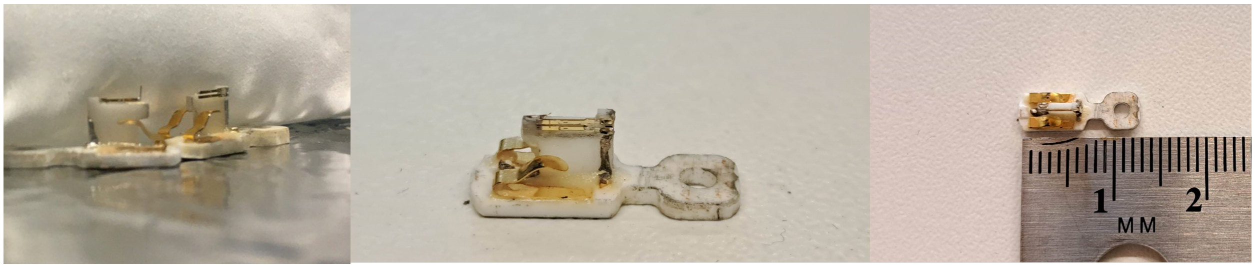


Figure 3.10: qPlus probes from RHK.

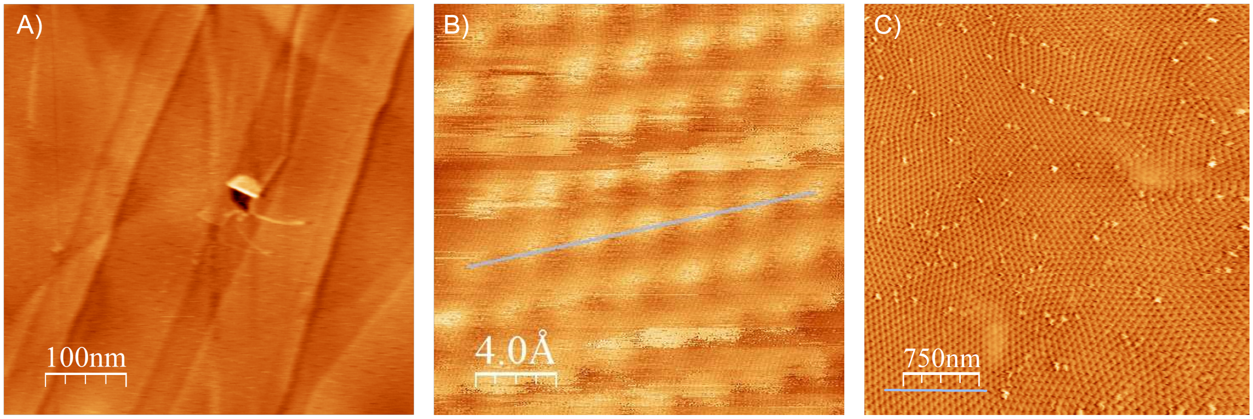


Figure 3.11: A) NC-AFM image of a HOPG step, B) STM image of gold on mica, C) NC-AFM of AAO.

our experiment was that the RHK system in the Hollen Lab would be able to resolve these features on this scale as well.

The qPlus probes from RHK (Figure 3.10) use cut Pt/Ir tips attached to a tuning fork. To check the resolution of the RHK system's qPlus probes in both STM and NC-AFM we tested multiple surfaces that can be seen in Figure 3.11including NC-AFM of HOPG steps, STM of gold atoms on mica, and NC-AFM of the AAO surface.

3.4 Results and Discussion

3.4.1 Creating Experimental Systems

MoS_2 bulk crystals were obtained from HQ Graphene and Graphenea. The flake samples were prepared by mechanically exfoliating bulk MoS_2 onto the insulators. The flakes were identified using an optical microscope. Figure 3.12 shows the bulk MoS_2 crystal and some

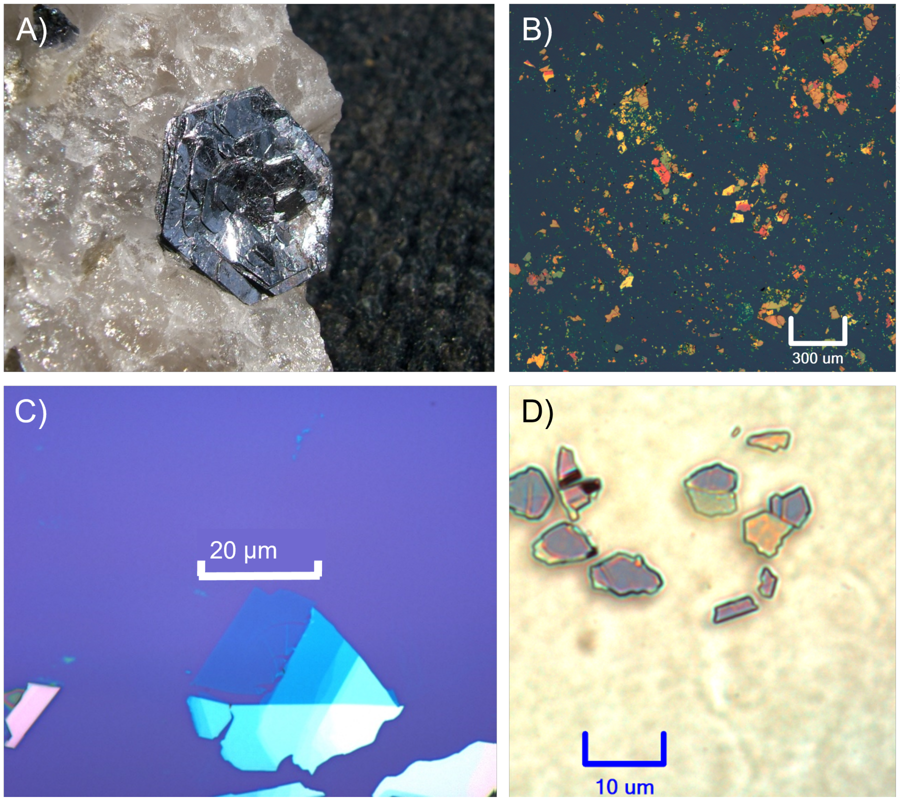


Figure 3.12: A) Bulk MoS_2 crystal 15mm across, B,C) Exfoliated flakes of MoS_2, D) Transferred Flakes on AAO

optical images of flakes on Si before being placed into the SPM.

Exfoliating on SiO_2 produced a large number of MoS_2 flakes. Although it was difficult to produce a high yield of large area monolayer MoS_2 using the mechanical exfoliation techniques. We explored multiple exfoliation techniques like gold assisted exfoliation[1] to produce larger MoS_2 flakes. Exfoliating on AAO did not have a large yield of flakes. Most times the exfoliation process just left depressions on the AAO surface with the outline of the flakes. Because it was difficult to exfoliate directly on the AAO, we transferred the flakes exfoliated on the SiO_2 onto the AAO, as seen in Figure 3.12 D, using a polymer transfer method[2].

[1] See Methods Section 2.2.1

[2] See Methods Section 2.2.2

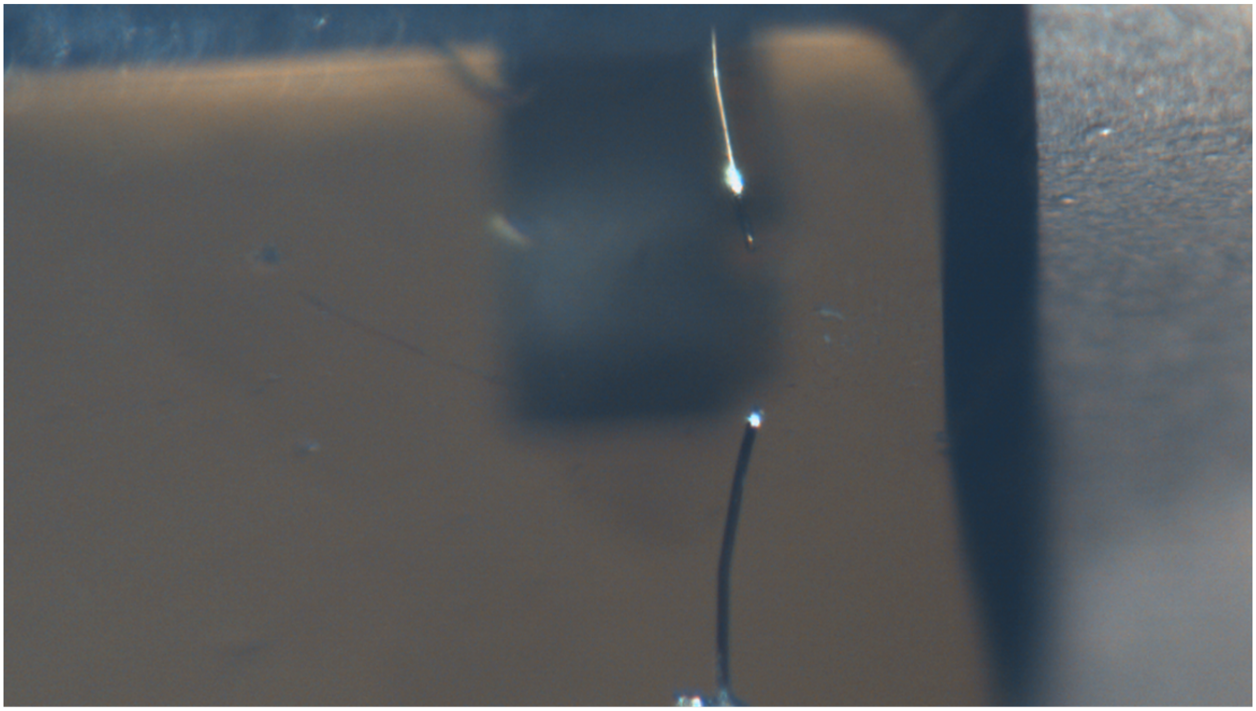


Figure 3.13: View from the optical camera on the MoS_2 on SiO_2 system

3.4.2 NC-AFM imaging MoS_2 on SiO_2

We started by imaging flakes of MoS_2 on SiO_2 in the SPM. The SiO_2 with flakes was placed on a sample holder and then into the SPM base. Figure 3.13 shows our view of the SiO_2 surface. The tip from one of our qPlus sensors is seen approaching in this image. The flakes we identified before with the optical microscope similar to the ones seen in Figure 3.12 could not be seen optically using our camera looking into the SPM stage.

Even though we could not locate the same flakes that we saw in the optical images we still attempted to image using NC-AFM. The feedback loop did not keep the frequency shift setpoint o while approaching on the SiO_2 surface, so we had to try to approach on the surface multiple times before we could image. While imaging at 9K we are limited to a small area of 1.5 x 1.5 micron^2, so it was difficult to know if we were imaging on the flakes or on the SiO_2 substrate. Eventually we managed to find the flakes and collect NC-AFM images seen in Figure 3.14. Since the flakes were in the range of 10's of micrometers lateral size and

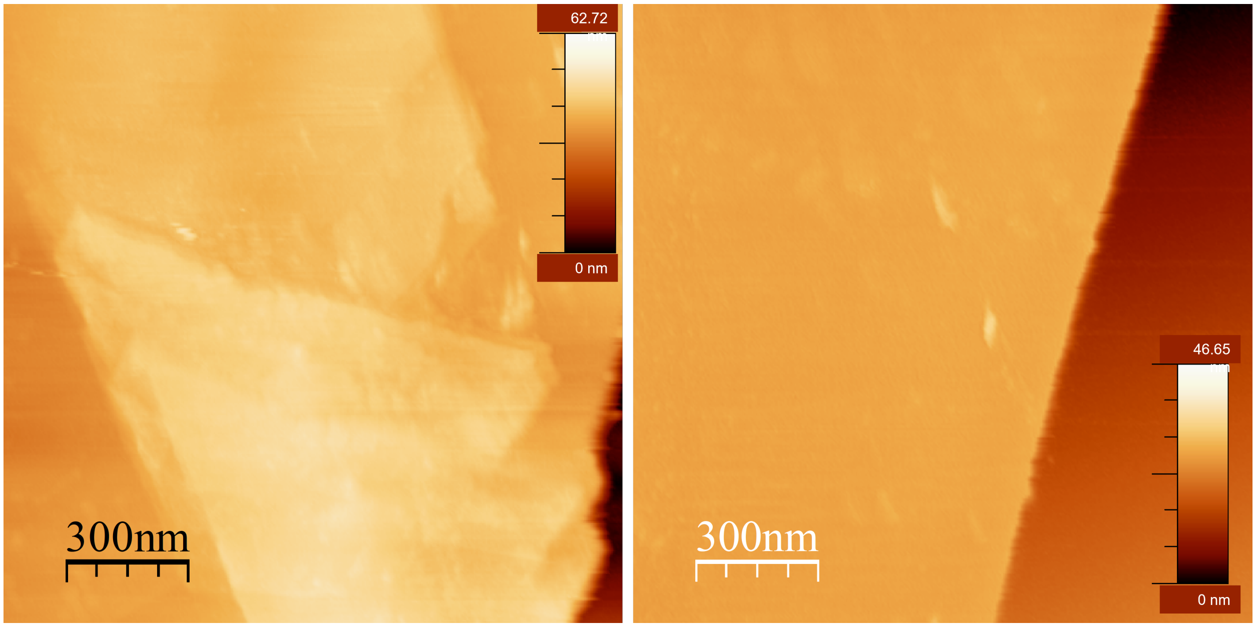


Figure 3.14: NC-AFM images of the MoS_2 flakes on SiO_2 at 9K

the distance between the flakes was much larger than that, it was time-consuming finding a single flake with the scan area available.

The flakes imaged were thicker than the range we had available in the Z direction at 9k. At some points during imaging the probe drifted away from the surface and we lost the feedback loop, so we had to reapproach. Upon reapproaching onto the surface we did not find the same areas that we imaged before.

To help find the location and make the imaging of the flakes more efficient and reproducible in the UHV chamber a gold grid was evaporated on the sample using a thermal evaporator. A TEM shadow mask covered the flake, then we evaporated a sticking layer of chromium and then a final layer of gold on top of the surface to create the grid.

Using optical images seen in Figure 3.15 A and 3.15 B we can pinpoint where the flake is and then using the view from the camera looking into the SPM stage seen in Figure 3.15 C we approached closer to the location of the target flake.

In a complimentary approach with the help of the Ishigami Group at the University of Central Florida, I designed an electron beam (e-beam) lithography pattern that makes the

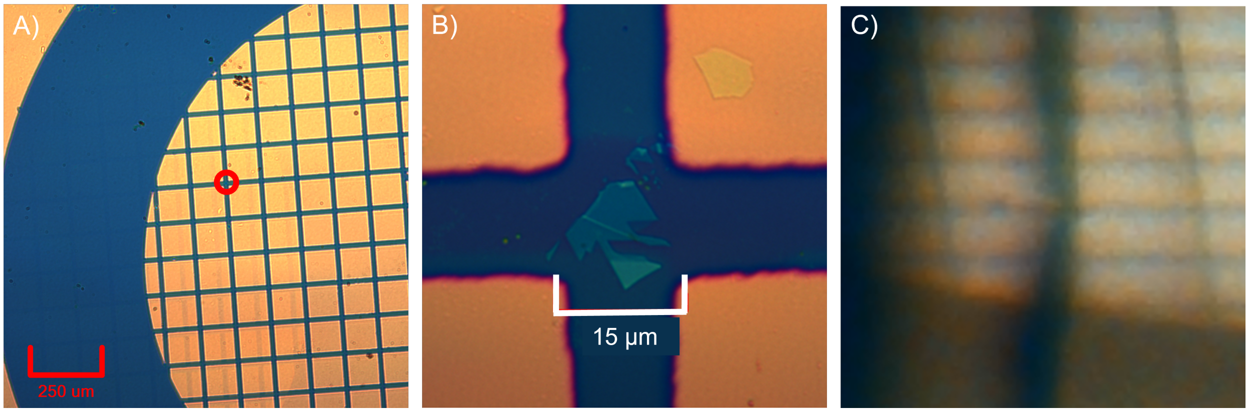


Figure 3.15: A) Optical image of the gold grid used to find the flakes. B) Zoomed-in image of the red dot in A, where the flake of interest is located. C) View from the optical camera looking into the SPM chamber.

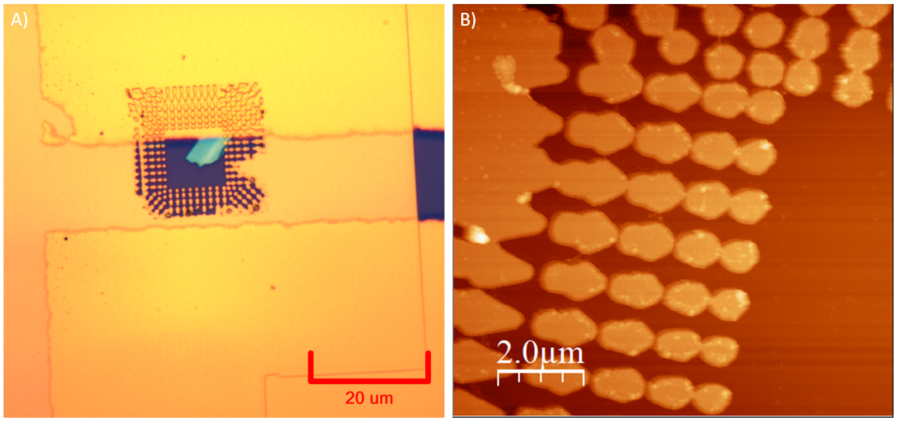


Figure 3.16: A) Optical image of the e-Beam arrow pattern around the flakes of MoS_2. B) Tabletop AFM image of the e-Beam arrow pattern around the flakes of MoS_2.

search for the flakes much easier. This e-beam pattern seen in Figure 3.16, has an array of arrows that get smaller as you approach the flake. The arrows in the grid were meant to provide recognizable features in the SPM images that could be used to find the location of the flake.

While the grids did help find the flakes faster, the imaging conditions were still very unstable, the setpoint was not kept by the feedback loop and we had to reapproach multiple times. It was hard to localize the flake after some scans. At one point we took the image seen in Figure 3.17 B. This NC-AFM image is the same location where the flake seen in

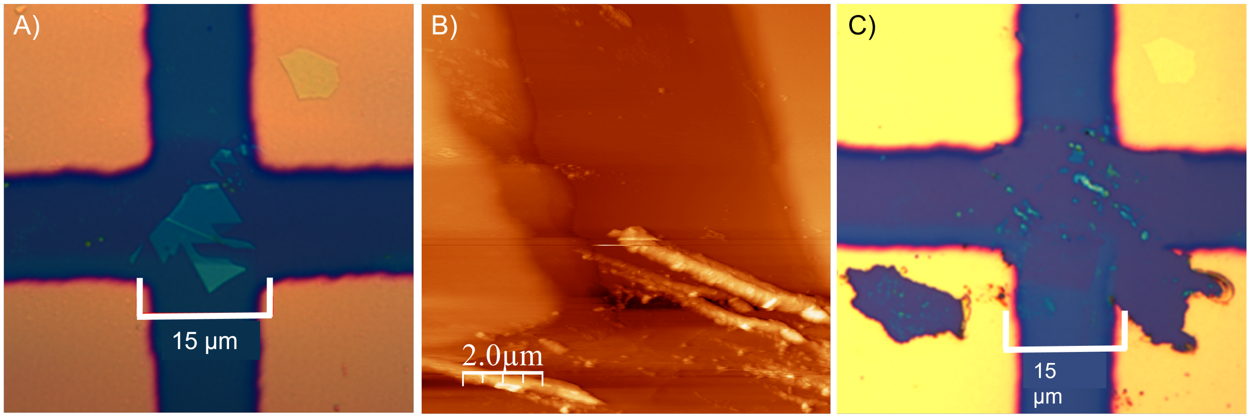


Figure 3.17: A) Optical images of the MoS_2 flake surrounded by gold grid. B) NC-AFM image of the area where the flake in A should be. C) Optical image of the same area in A and B after the NC-AFM imaging showing the flake is no longer there.

Figure 3.17 A should be, but we saw a different feature than what we expected. We decided to take out the sample and look at the area once more using an optical microscope. The optical image Figure 3.17 C showed that the flake was no longer there and that the areas of gold around were also being destroyed. It is clear the tip was making contact with the surface during the approach or while scanning. This also explains why it was difficult to image the flakes with and without the grid.

Despite these issues, multiple flakes were imaged during this set of experiments thanks to the help of the gold grids and by imaging at higher temperatures. Higher temperatures allowed us to have a larger scan window and made it easier to find the flakes. Figure 3.18 is an example of the NC-AFM images collected at 115K.

From the scale bars we see that there are high features on the surface of the sample. These high features are likely residue from the gold evaporation process. Figure 3.19 shows a line scan across one of the bright features showing it reached 100 nm in height. From the collected images none of them had a resolution compared to the 15 to 100 pm range we needed to collect the information of where the thiophene molecules would bind.

To see where the active catalytic sites are on the MoS_2 we need to be able to resolve where the thiophene will bind on the MoS_2. We tried dosing on the same sample seen in

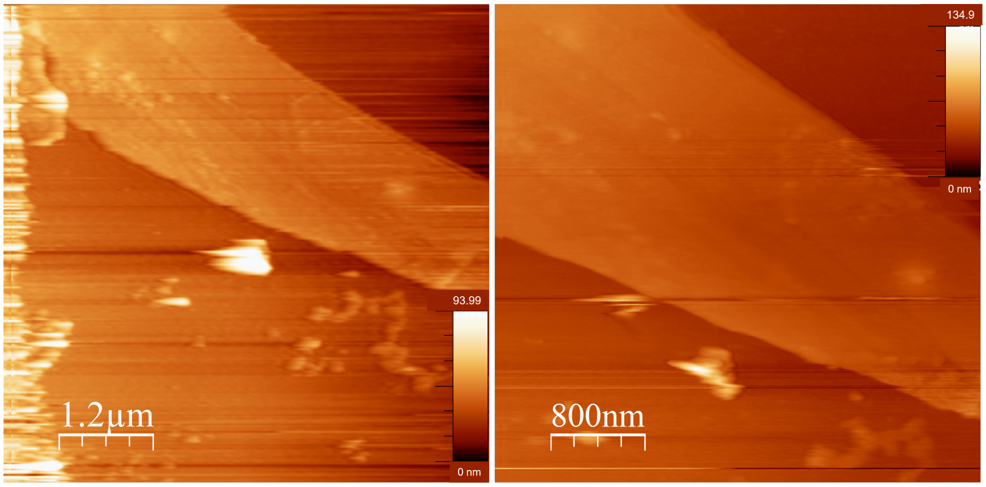


Figure 3.18: NC-AFM image of a MoS_2 flake surrounded by a gold grid taken at 112K.

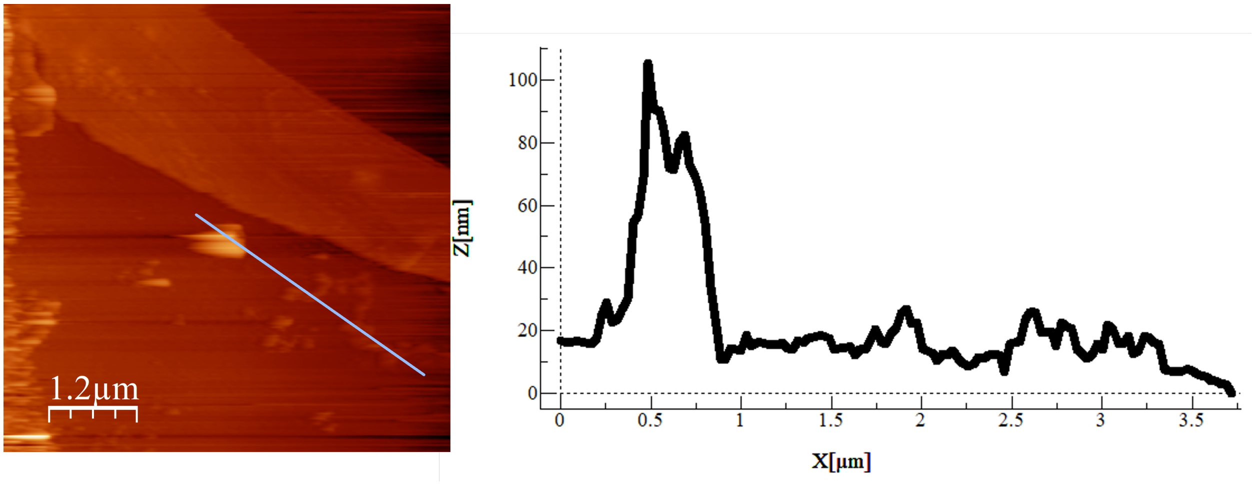


Figure 3.19: Line scan of the contaminants on the surface seen on the right.

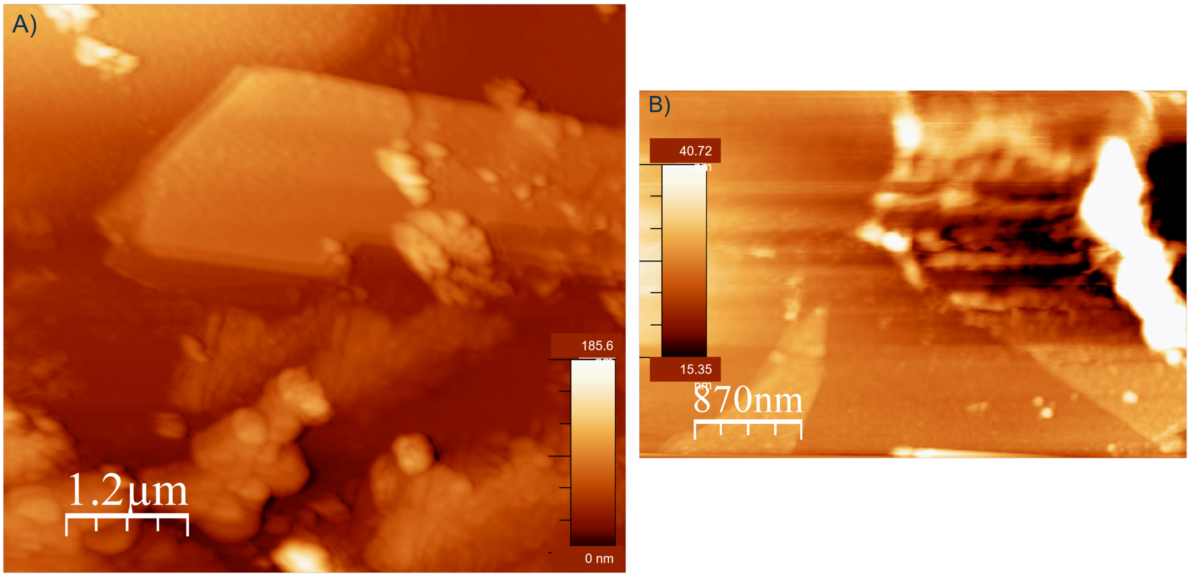


Figure 3.20: NC-AFM images of MoS_2 flakes after a thiophene dose. A) Taken at 85K. B) Taken at 293K.

Figure 3.18, but similar to before we could not find the same flake there, so we moved to a new area. Figure 3.20 A shows a NC-AFM image after a thiophene dose. It is impossible to determine if the contaminants on the surface were before or after the dose, but they seems similar to the contaminants seen in other images from the gold evaporation.

Since we could not tell if we had thiophene on the surface of the MoS_2 we increased the temperature to attempt to desorp the thiophene off the surface of MoS_2. After imaging once more we were again not able to see the same flake as before, likely we were still damaging the flakes or moving some of the contaminants while imaging. Figure 3.20 B show the NC-AFM imaging of the flakes after warming to room temperature. From these images it was inconclusive to say where the thiophene bind to the surface, since we could not get a reproducible before and after image to compare the surface of the MoS_2.

3.4.3 MoS_2 on AAO

The transferred flakes of MoS_2 draped over the AAO as we expected they would to create regions where the flake was strained. From optical images we could not tell that this was happening (Figure 3.21, but by characterizing the sample with a regular tabletop AFM we

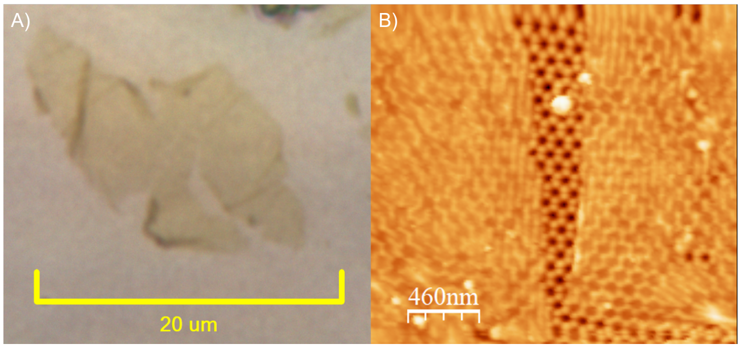


Figure 3.21: A) Optical image of an MoS_2 flake on top of AAO. B) Tabletop AFM image of an MoS_2 flake on top of AAO.

were able to see that the pattern of the AAO was still visible in the areas covered by the MoS_2, which can be seen in Figure 3.21 B. An undergrad in the Hollen Lab, Tan Dao did a lot of the work to help create and characterize these samples.

Dao used an AFM to characterize the MoS_2 flake on AAO and measured the corrugation of the flake seen in Figure 3.22. Using this information Dao was able to calculate the tensile strain (ϵ) felt by the MoS_2 by modeling the hills of the AAO as a sphere with a radius of 50 nm and the MoS_2 stretches on top of that sphere. The strain obtained is the average over the top hemisphere described by the equation:

$$\epsilon = \frac{\sqrt{2R_S}}{\sqrt{R_S + z}} - 1 \tag{3.1}$$

Where ϵ is the tensile strain, R_S is the 50 nm radius, and z is the change is the height of the corrugations seen in the AFM images. Using this Dao found a 0.3% tensile strain in the areas where the MoS_2 makes contact with the AAO. This is higher than the 0.02% strain calculated by Li *et al* that has an impact on the catalytic effect of MoS_2 during the hydrogen evolution reaction.

The goal was to image these samples with the NC-AFM, but with the difficulties experienced while using the NC-AFM to image the MoS_2 on SiO_2 samples, we decided that we would get similar inconclusive results on this sample and decided to not continue the

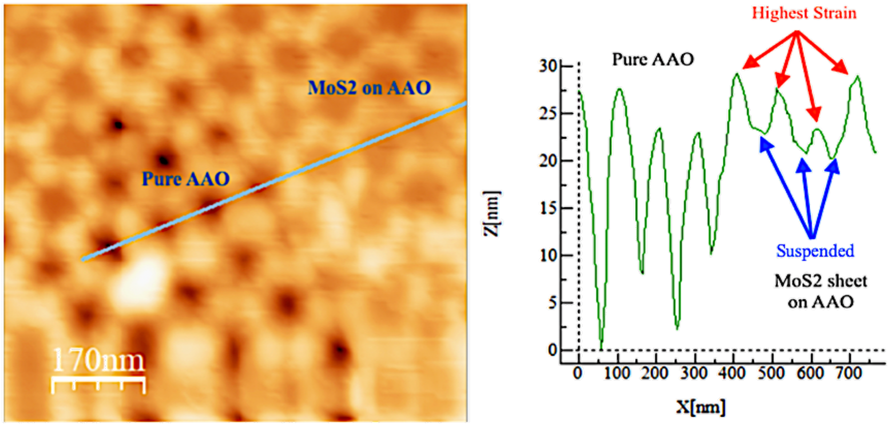


Figure 3.22: AFM image of MoS_2 on AAO. The line profile shows the corrugation of the bare AAO and the area covered by the MoS_2. Taken by Undergrad Tan Dao.

research project.

3.5 Conclusion

The 2D materials used as catalysts in hydrotreatment reactions of crude oil have been mostly developed by large averaging spectroscopic techniques. To fully understand the mechanisms of the catalysis reactions it is important to study them in a controlled environment and view the reactants, intermediates, and products at a molecular and atomic level. STM studies have shown promising results to explain the mechanism of why these 2D materials like MoS_2 make such good catalysts, but they are all studied on metal surfaces that are not industrially relevant substrate.

Here I presented my work done to study the catalytic sites of MoS_2 on industrially relevant substrates: SiO_2 and AAO. For these studies, we used NC-AFM, but found that with our current setup at the Hollen Lab, we were limited by our resolution. To find the active catalytic sites we wanted to image the MoS_2 on these insulators before and after a thiophene dose to see where the thiophene bound to the surface which would give us insight into where these hydrodesulfurization reactions take place.

To improve the resolution of the NC-AFM we need to be able to functionalize the tip of our qPlus probes with a CO molecule. This CO functionalization is done on metal surfaces or on a few layers of NaCL, but there does not seem to be a method developed to pick these up on an insulating surface. To be able to study these molecules on an insulator we need to develop a way to functionalize the probe on these insulating surfaces.

We also experienced multiple crashes while approaching the probe onto the insulating surfaces. This destroyed our samples and also ruined the quality of our tips. A way to tackle this problem could be to attempt to do a controlled approach by measuring the forces at various steps during the approach to create a force curve and then using that as a reference for future approaches instead of relying on the auto approach function from the RHK software.

The last limitation seemed to be the cleanliness of the samples. The exfoliation process produced nice flat flakes, but they were hard to find without a pattern on the surface to map out the location of the flakes. Using a pre-patterned substrate could be a cleaner approach instead of the gold grid we evaporated onto the surface. A clean and flat sample is really important while doing NC-AFM, this is why most groups only work on metals, since a CO functionalized tip can be lost easily when there are a lot of contaminants on the surface.

In conclusion, I took the first NC-AFM images of MoS_2 flakes on an industrially relevant insulating material. Limited by the resolution of our system, I could not image the active catalytic sites of MoS_2. Before NC-AFM can be used reliably a method has to be developed to functionalize the qPlus probes with a CO molecule or other molecules on insulating substrates, develop a more gentle approach method to reduce crashes, and improve the overall cleanliness of the samples.

CHAPTER 4

Automating Scanning Probe Microscopy With Machine Learning Algorithms

4.1 Introduction

Since the invention of the first scanning tunneling microscope at IBM Zurich, scanning probe microscopy (SPM) has revolutionized nanoscience. SPM techniques include a wide range of probes that can be used to measure the physical properties of a material like electronic, chemical, and magnetic signatures. The major drawback of these SPM techniques is that they are very time intensive. In order to collect any valuable data for scientific analysis, imaging parameters have to be optimized. Even for expert SPM users optimizing these parameters requires multiple iterations of tip prep and time spent searching for an optimal imaging site.

With recent advances in machine learning an artificial intelligence (AI) can be trained to perform tedious image optimization steps in SPM like finding features of interest and performing tip tuning procedures [89, 90]. The AI can be taught to recognize patterns commonly seen in SPM images to perform experiments autonomously [91] and to help with data analysis [92, 93]. These elements will help to build a fully automated SPM, but a framework is needed to integrate all these individual tasks into a single machine.

In this chapter, I explain my work to create a modular framework that is to be the base for a fully automated SPM AI controller. Utilizing open-source machine learning platforms and computer vision tools I created machine learning models to perform SPM tasks. My work led to the creation of Auto-HR-AFM, an AI script that autonomously controls the SPM to collect HR-AFM images of hydrocarbons. Future additions can be added to Auto-HR-AFM to make it an all-around autonomous SPM.

4.2 Scientific Background

While scanning across a surface an SPM probe measures a physical property of the surface. An image is constructed using the intensity of the measured signal as a function of the position. This technique has provided stunning images through the years and has given a visual representation of our atomic world. SPM data showcased in presentations and papers is a small fraction of the data actually collected. To produce good data a user has to consistently monitor the state of the probe and imaging conditions.

A user has to select areas to image and has to assess the quality of the data based on previous experience. If the user deems the quality of the images to be poor they have to perform either tip-tuning actions to improve the resolution or they have to find a new area to scan. These conditioning actions are performed multiple times until the user is satisfied with the results. This can be a very time-consuming process and takes away from the limited time some experiments have.

To improve imaging efficiency, we turn to automation. Most of the SPM actions are controlled using computer software and people have started creating automation scripts to perform these. [94] Still these actions depend on the user deciding when to do a specific task. With the advancement of AI, we can train a computer to assess current imaging conditions and make a decision on what the next best action to take is.

Machine learning is growing at a fast pace and we see it every day with the advancement of self-driving cars, better ways our emails detect spam mail, and YouTube algorithms deciding what new videos to recommend to us. An AI controls these actions, they are trained to learn from visuals around them and to learn from the data we give them. Similar tools can be used to automate SPM as well.

A good example of AI applied to SPM, is the work done by Krull *et al* in the development of the DeepSPM, an AI framework that is meant for SPM automation [89]. DeepSPM

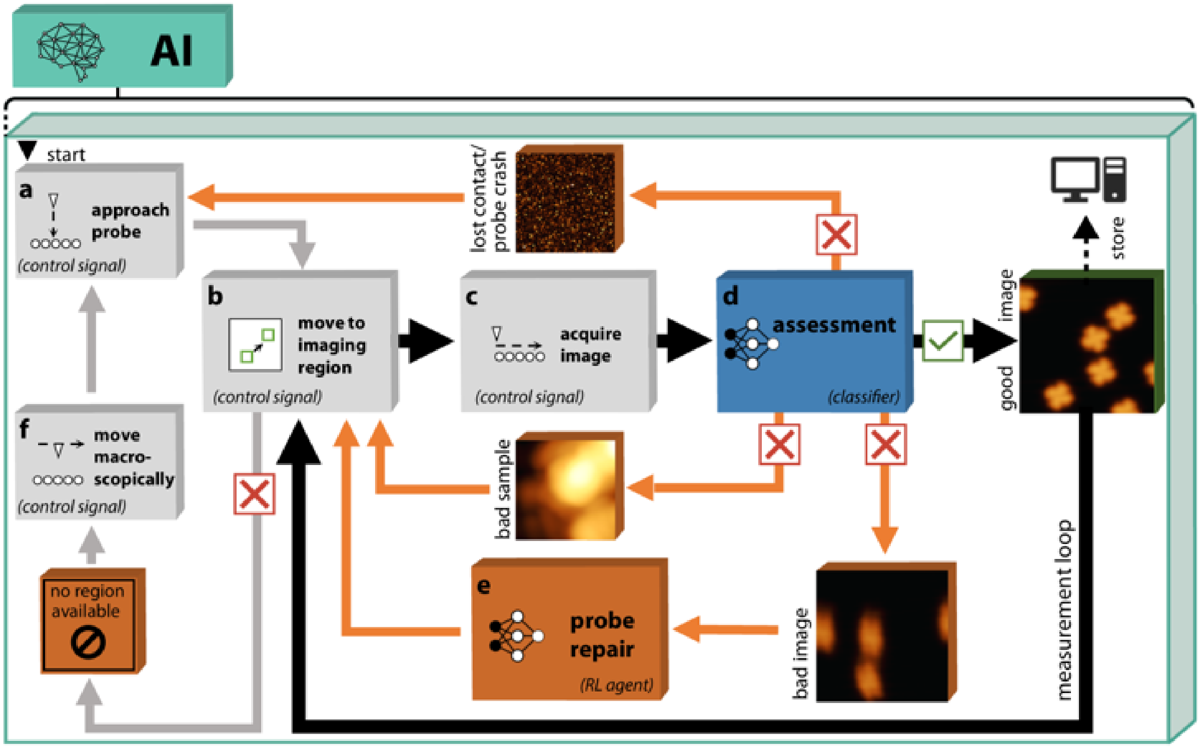


Figure 4.1: Machine learning architecture for DeepSPM and Map of automatic navigation. Adapted from Krull *et al* [89].

automates the navigation of the SPM probe and assesses the quality of the probe and imaging conditions. If DeepSPM decides the surface is not ideal, then it moves on to the next area to image. After multiple bad image regions, the AI then decides to tune the probe in a previously deemed good area.

The framework of DeepSPM can be seen in Figure 4.1. The two main AI components in DeepSPM are the image classification models that are used to assess the imaging quality (Blue box in Figure 4.1) and another reinforced learning model to assess the quality of the probe (Orange box in Figure 4.1).

The model to assess the quality of the probe is similar to Auto-CO-AFM created by Aldritt *et al* to recognize if an AFM tip is functionalized. Auto-CO-AFM uses an image classification script to determine if the probe is functionalized or not with a CO molecule [90]. Figure 4.2 shows an STM map of where the CO molecules are detected on a Cu(111) surface, Auto-CO-AFM then determines if the images are from a bad or good CO functionalized

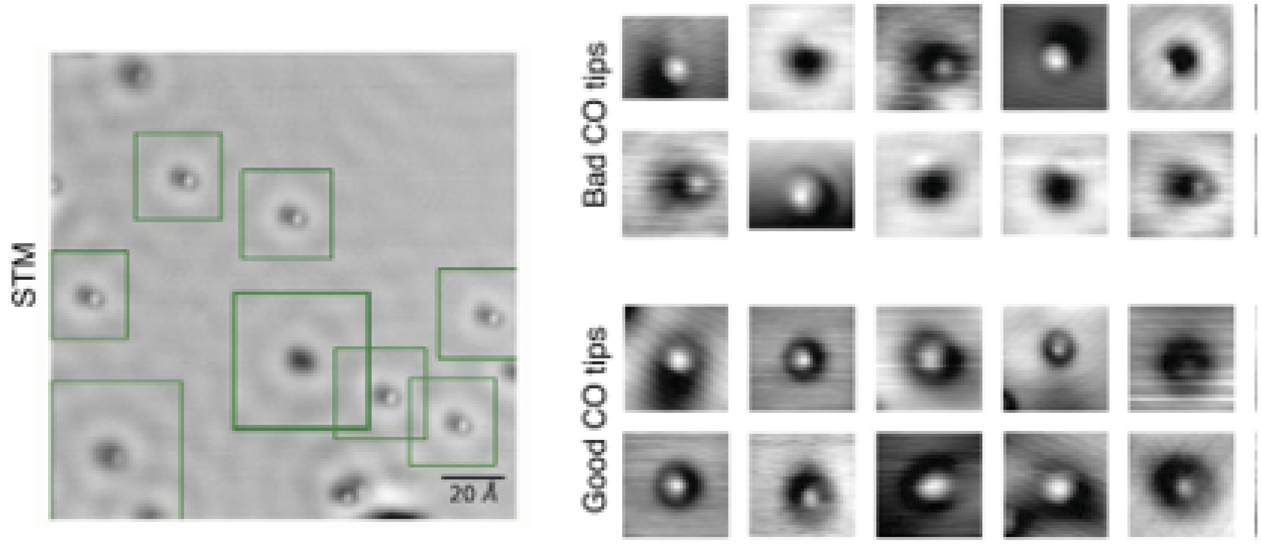


Figure 4.2: STM map of CO molecules on Cu(111) and a collection of NC-AFM images of CO that have good or bad CO functionalization. Adapted from Aldritt *et al* [93].

tip. Some examples of good and bad CO functionalized tips are seen in Figure 4.2. CO functionalization increases the sensitivity of AFM probes. The increased sensitivity allows AFM probes to collect high-resolution images. Auto-CO-AFM automates the assessment of the probe quality and even automates the task of CO functionalization.

Wang *et al* developed another AI tool to check the quality of their STM probe by assessing spectroscopy data on a Au(111) sample [95]. The AI was trained to recognize dI/dV spectrums of clean Au(111) surfaces. When their tip needs to be tuned they utilize their AI tool to find a clean area on the Au(111) surface. The AI then performs a tip-tuning procedure and then moves to a new clean area to check the quality of the probe by collecting and assessing a dI/dV spectrum. The process is repeated until the AI determines that the probe is optimized.

These machine learning automated SPM tasks focus on assessing the quality of the probe and surface. The AI replaces a user in looking over the data and determines what action to take immediately. The setback of these SPM automation techniques is that they are all trained on specific datasets and in order to generalize them to any SPM tasks the models have to be trained once more to include features specific to a new surface or whatever specific techniques, to recognize and to teach them what actions they can take. A more general framework is needed that can combine all of these. The fast-paced growth of machine learning models also means that the models require constant development to keep them up to date.

4.3 Computing Setup and Methods

There are multiple open-sourced machine learning framework packages that are toolboxes designed to create machine learning algorithms. The most common ones are Tensorflow (Combined with Keras) and PyTorch. There is a debate on which one of the two is better, but mostly it is a user preference. The two frameworks are not needed to create a machine learning model, but they both have the tools necessary to speed up the model design and tools to test and check model performance. Using Tensorflow or PyTorch allows developers to start a whole machine learning project from scratch with tools at hand to build on the base package.

Even though it is possible to run and develop machine learning codes on a CPU, it is recommended to switch over to a modern NVIDIA GPU, especially when working on projects that require imaging processing. The task of training models requires a significant amount of computational power and the multiple cores in a GPU are designed to perform parallel processing computations that expedite computational jobs by a factor of 5 to 10 compared to just using a CPU.

A good starting point for those who want to get into machine learning is to use online host networks that provide GPU run times like Google Collaboratory (Colab). Colab is great for small-sized dataset projects that do not require a lot computational power. Diving deeper into the field, and as the size of a machine learning project increases it gets harder to manage everything on host networks like Colab. For larger projects, the recommendation is to use either a large cloud service or to use a local GPU.

Cloud services are offered by most of the major tech companies and have become a profitable avenue for host startup companies. These services provide modern GPUs that can be accessed online and can provide cloud storage for data freeing up local drives. The cost of these cloud services scales with the number of resources used, but on average the most commonly used GPUs can be rented for $1 to $3 dollars per hour. This adds up over time but is generally lower than the cost of purchasing a local GPU.

Many research institutes also have high performance computer clusters available for researchers. These clusters are best suited for large computational problems that can be divided into multiple tasks. They are used for tasks that require large amounts of memory, storage, or runtime. If a machine learning project has a large workload, these computer clusters would be a perfect place for students and researchers to carry out their projects.

For the work done in this chapter, the size of the dataset and the workload for the machine learning projects was minimal compared to the size of most large collaborations that need a cloud server or computer cluster. All the work was done on an NVIDIA GEForce RTK 3080 GPU, running all the scripts on a Jupyter notebook.

Jupyter notebooks are widely used in the data science and machine learning communities. Most tutorials and machine learning classes offer Jupyter notebook files to follow along with. The notebooks are files that are edited in a browser window. Python code can be executed and annotated in notebooks. Longer code files can be broken down into components and the annotations can help explain what each part of the code does. Each component of the code can be run individually, which is useful when it comes to finding bugs in your code. This is a tool I recommend using while coding and learning, but the python scripts created for machine learning can be run on a terminal or other environments like PyCharm.

A good place to store coding projects and share them with others is Github. Software projects can be tracked on Github, which makes the tool especially useful when collaborating with multiple people. A record is kept by Github on all the changes done by collaborators. Most machine learning projects are kept in a repository uploaded to Github as well, so you can find projects related to your own and work from there. Many tutorials are also kept on

Github with all the required documentation needed to understand the codes fully.

The initial work done on this project was to select a proper setup and to choose what machine learning model frameworks would be the most useful to automate SPM tasks. Coming from an SPM background, having minimal coding experience, and knowing little to nothing about machine learning meant that I had to explore multiple options to teach myself machine learning. The most helpful tools for me were books and online video tutorials. A couple books worth mentioning are Chollet's Deep Learning with Python [96] and Geron's Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow [97]. Sergio Canu creator of Pysource and Nicholas Renotte offer very useful video tutorials on YouTube and blogs on machine learning projects as well. These tools helped me understand that the first thing to develop a machine learning model is to define the task that we want the model to perform. With a task in mind, we can start to collect and process data to train a model. With the processed dataset we create a machine learning model, starting from a base model provided by either Tensorflow or PyTorch. After multiple iterations of testing and optimizing we have a finished model.

4.4 Project Definition and Results

The original goal for this project was to build upon existing models to create a more generic and customizable framework that can lead to a fully autonomous SPM in the future. Previous SPM automation work seemed to cover the basic tasks of navigation and tip tuning, but they were either outdated codes or needed to be retrained to include a larger dataset.

The data I used to train my machine learning model was a collection of high-resolution NC-AFM (HR-AFM) images, mostly of hydrocarbons, provided by the low-temperature dual STM/NC-AFM system operated by Dr. Percy Zahl at Brookhaven National Lab. This system is used to collect STM and HR-AFM images of organic molecules on top of metal surfaces. We based the tasks for our machine learning model project on automating the data collection for these experimental systems. Implementing these machine learning automation tools at national user facilities makes these more efficient for visiting scientist.

Here I present the ideas and models we developed during the course of the project leading to the creation of Auto-HR-AFM [?], our AI framework that is the initial tool to build a fully automated SPM.

4.4.1 Computer vision feature detection

A major task in SPM experiments is finding regions of interest to probe. Most features in SPM images are simple geometric shapes. Vacancies and adatoms look like tiny circles in the imaging, steps have very defined sharp lines and molecules are mostly just blobs on the surface in STM images. Computer vision tools are being adopted for data processing and data analysis to find these features on the surface.

Edge detection scripts can be used to find features on the surface. These edge detectors apply filters on the images to pinpoint exactly where edges of the regions of interest are. The edges can be contoured and then overlapped over the original image to help the user locate the features. some examples of this are from Gudinas [98], who developed a tool to count defects in black phosphorus and Hellerstedt [99] created a code to find molecules on surfaces. When it comes to more complicated environments with multiple features with similar shapes these tools can get confused, and mislabeling can occur. Figure 4.3 shows an edge detector tool I developed to find petroleum molecules on a copper surface with different user-set thresholds. High thresholds found little to no molecules (Figure 4.3 A). Low threshold mislabeled areas, especially around step edges (Figure 4.3 B).

4.4.2 Image Classification with Quam AFM

To teach the AI what a good HR-AFM image looks like we required a labeled dataset of images. One of our original ideas was to have the AI recognize different heights in the HR-AFM image and then have the AI adjust the probe to collect a more optimal image. In our first attempt, we considered this an image classification problem.

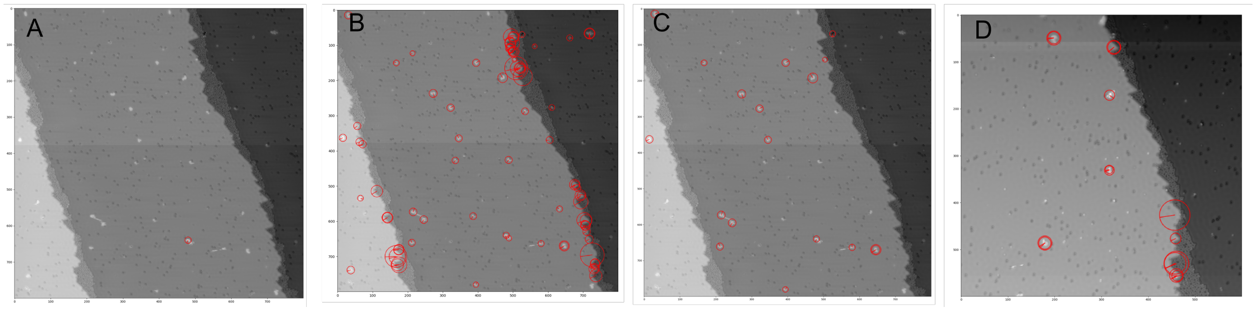


Figure 4.3: Edge detector script used to find molecules on a Cu(111) surface. The threshold conditions are changed for each image.

The idea is similar to having a neural network distinguish between a cat or a dog: we have the AI assess the HR-AFM image and label the different regions of the molecule to specific height regions. For this to work we needed to collect and label HR-AFM images of molecules with different heights. We were going to use old experimental data, but found a dataset of simulated AFM images called Quam AFM from the Autonomous University of Madrid [45]. Not only did this dataset resemble actual AFM images, the images were also already separated into ten different height folders that we used as labels. Figure 4.4 shows some examples of the data provided by Quam AFM. The first column is a ball and stick model of the molecules and the other columns show the simulated molecules at five different probe heights ranging from 2.9 angstroms to 3.7 angstroms.

Although the data was already labeled by height, we ran into issues when training the classification model. The model took in input AFM images and their respective heights as features to recognize. Ideally I wanted the model to recognize the aromatic rings in the molecules and base the decision of how close based on those, but the model was looking over the color contrast of each input image instead and basing the height off of that. Special difficulties arose when a molecule was not completely planar. This meant that the parts of the image had different contrast, the AI misunderstood this and labeled the molecule with an incorrect height.

I believe that using this dataset is still beneficial to train an in depth machine learning model, but the way that the data is labeled has to be modified in order to use it as a good

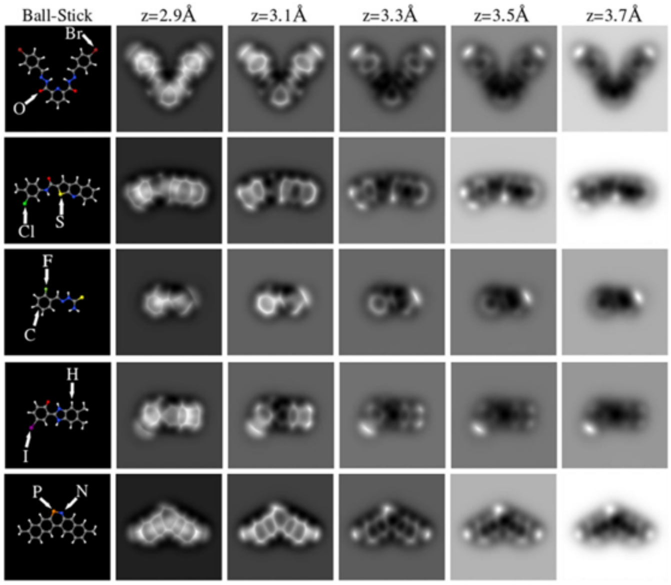


Figure 4.4: Collection of simulated NC-AFM images of hydrocarbon containing molecules used as a first training data set. The different heights were used as labels for 10 different heights. Adapted from Carracedo-Cosme *et al* [45].

assessment for the height of the probe above the molecule.

4.4.3 Instance Segmentation Using Detectron 2

Since the classification system using neural networks and Quam AFM labeled data did not produce the results we expected, we moved on to a different approach. While looking for new way to process and find features in images I learned about Detectron 2. Detectron 2 is an instance segmentation model created from the group at Facebook AI [100].

Instance segmentation models take an input image and find features of interest in the image. The models create an overlapping image with either bounding boxes or contour masks around the feature of interest and they label the item accordingly. These models are being used to automatically detect cancer cells [101] saving patients time to get results and used commonly in the development of self-driving cars to recognize using the car's cameras where objects near the car are while driving.

I started testing Detectron2 to train it to recognize a single class object: The aromatic rings that are the building blocks of these molecules. I started manually labeling the aromatic rings in both simulated and experimental data and worked with a Detectron2 base model. These base models are pre-trained on datasets from Common Objects in Context (COCO) [102] which contain more than two hundred thousand labeled images of everyday objects.

Using my own labeled data I retrained the base models from Detectron2. This process is called transfer learning, where a pre-trained model is used as the starting point to create a new one. The initial results of using this model to find these rings are seen in Figure 4.5. In simulated data, my instance segmentation model detected all of the rings in the original image. My initial model did make mistakes when looking at experimental data, but it still detected most of the rings add only a small amount of mislabeled spots.

I was pleasantly surprised that this model performed as well as it did to find rings, so I continued to expand this method to try and classify these rings into different regions. I expanded the single class model that recognized aromatic rings to a three class model that recognizes close, ideal imaging, and far regions. This formed the base for the AI decision-making for Auto-HR-AFM which shall be discussed in the next section.

4.4.4 Auto-HR-AFM

This section contains a combination of the draft and supplemental material of the submitted paper titled ”Autonomous Molecular Structure Imaging with High Resolution Atomic Force Microscopy for Molecular Mixture Discovery”. As of the date of my defense, the paper was submitted to the Journal of Physical Chemistry A and we were working on revisions.

Introduction

Recent advances in molecular imaging by high-resolution atomic force microscopy (HR-AFM) provide a first look at the diverse structures that make up complex molecular mixtures, such as petroleum [12, 83, 103–109], soot from combustion or pyrolysis [84, 110, 111], and organic molecules found in meteorites, oceans, and around Titan’s hazy atmosphere. [85,112,

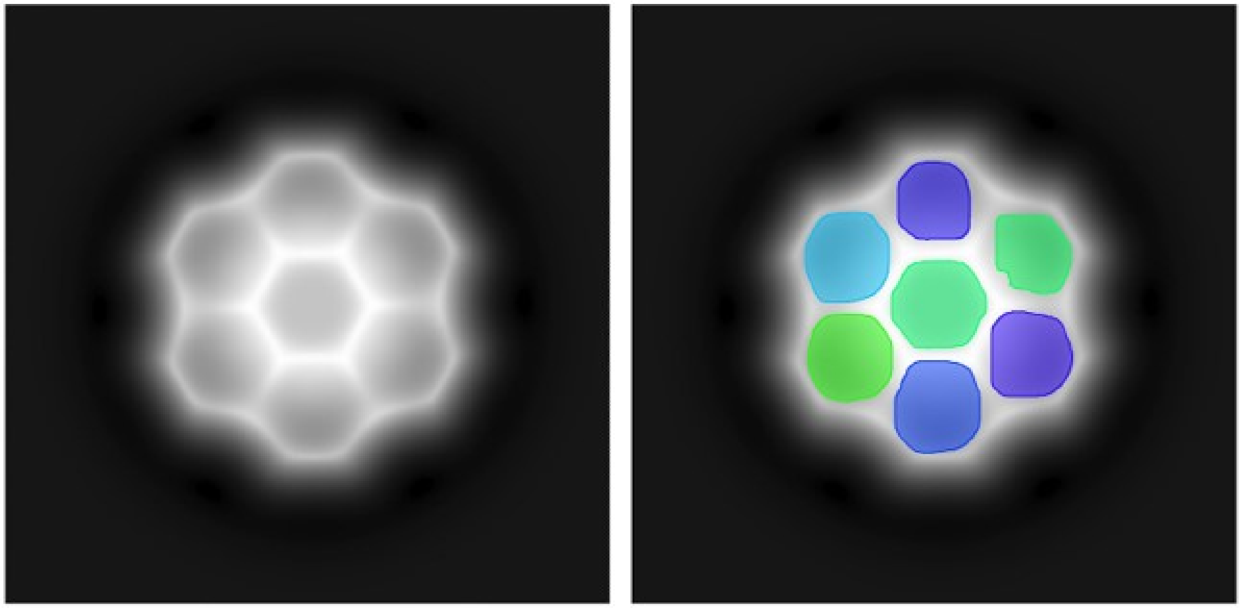


Figure 4.5: First results of using an instance segmentation model to find aromatic rings in NC-AFM images of petroleum molecules. Color scheme is random and represents a single instance, used to distinguish in case of any overlapping instances.

113]. For complex mixtures, especially for those comprised of large molecules with numerous possible structural isomers, HR-AFM provides unique insight into the identity of molecules in the mixture. This information is nearly impossible to obtain with traditional analytical techniques. The single molecule sensitivity of a CO-functionalized [43, 44] HR-AFM tip makes HR-AFM capable of directly imaging individual molecules with atomic resolution [84, 106, 108]. This advance opened the door to structural and reactivity studies of novel aromatic hydrocarbons, and was used to discover non-combustible uses of these abundant hydrocarbons [11]. HR-AFM has the potential to become a mainstream characterization technique for complex mixtures, revealing otherwise inaccessible molecular structure and statistics, but several significant experimental challenges must be overcome.

Before a single image can be obtained, the HR-AFM technique requires a significant amount of user expertise, resources, and time due to demanding experimental conditions. The experiments must be performed at extreme conditions of low temperature (5K) and ultra-high vacuum pressures of 1x10^{-10} torr provided by a liquid helium bath cryostat in ultra-high vacuum to maintain an atomically clean and mechanically extremely stable environment for extended periods of time. The user has to possess significant knowledge of physics and specific instrument training to prepare the system, samples, and probe before any of the experiments take place. HR-AFM experiments can take days, weeks, or even several months to collect a satisfactory amount of data. This process is very time-consuming because molecules need to be individually imaged and the imaging must be optimized to reveal all their details. Each image can take a few hours, even under ideal conditions, to find optimal imaging conditions for a complex molecule. For a complex molecular mixture, at least 50 molecules need to be imaged for the dataset to be statistically relevant, and compare with bulk analysis techniques such as nuclear magnetic resonance spectroscopy and mass spectroscopy. The more molecules imaged the better since many unique molecules have identical molecular weights, and so the HR-AFM imaging often reveals otherwise unknown molecular structures. The imaging conditions can change at any moment because the tip-sample junction, which is critical to the imaging, is fragile. This means the experiments require constant supervision by an expert user to optimize data collection. Any interruptions from fatigue or user mistakes cause setbacks and reduce the amount of data collected for projects with a set time frame, common on user tools. To maximize the amount of time the HR-AFM is collecting high quality data, we turn to automation using machine learning (ML) and AI.

With recent advances in ML there is major progress in automating machines to recognize patterns and perform human actions. Similar to with training self-driving cars, we can train an AI to operate an HR-AFM. Work in this field already includes automating certain aspects of scanning probe microscopy (SPM) like navigation [89], tip tuning [93, 114], and spectroscopy [91]. These AI techniques use a series of neural networks to recognize features from collected data, this information is then used by the AI which decides on the next best action to take. These elements will help to build a fully automated SPM, but a framework is needed to integrate all these individual tasks into a single machine.

Here we present Auto-HR-AFM [?], an AI script that autonomously controls the SPM to collect HR-AFM images of hydrocarbons. Auto-HR-AFM assesses collected images using a trained ML model and adjusts the probe-molecule distance to optimize the image of each molecule. The ML model we trained for Auto-HR-AFM is an instance segmentation model based on Detectron2 [100] that recognizes patterns in HR-AFM images of hydrocarbons. Our model segments the collected images into three different classes that characterize the proximity of the CO-functionalized probe to the molecule: too close, too far, or at an ideal distance for imaging molecules. The model outputs are then used by GXSM [115,116], open-source software that controls the SPM, to determine what direction the probe should move to collect a more optimal image. Once the best image is collected, the SPM proceeds to collect images of the other molecules in the experiment until a complete dataset is achieved. Our technique, which builds upon the SPM automation foundation that already exists [89, 91, 93, 114], provides an open-source platform on which specific tasks can be combined to create a fully automated SPM capable of routine characterization of molecular mixtures.

Methods

Auto-HR-AFM's Decision Making Auto-HR-AFM, similar to what an expert user would do, monitors the imaging conditions and reacts accordingly to get an optimized image. Auto-HR-AFM assesses the collected images and adjusts the probe-molecule distance to collect a more optimal image of each molecule. Auto-HR-AFM's decision making uses an instance segmentation model we trained using transfer learning starting from a Detectron2 [100] archetype that we trained to recognize features in HR-AFM images of complex molecular mixtures of petroleum pitch samples

Instance segmentation is widely used to contour unique instances of objects in image segmentation models. These models are trained to locate and label each instance of a specific object in an image and can be applied to images with multiple objects of interest, and have been recently used to find features that could signal diseases in biological samples [117], find

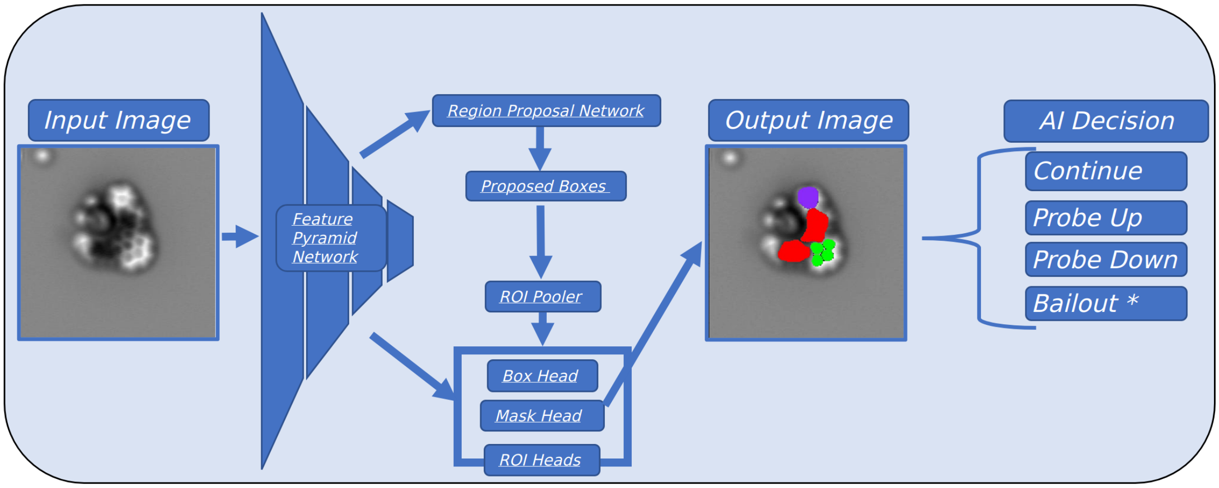


Figure 4.6: Architecture of the AI decision-making script showing all the parts of the machine learning model based on Detectron 2

structural damage in infrastructure [118], and to find population of invasive species [119]. In most applications the models are trained to recognize simple shapes experts in that field tend to look for. We apply the same technique to SPM images, since the SPM user focuses on features in SPM images. A model can be trained to recognize the features of interest of a trained SPM user.

The base model we used has a ResNet+feature pyramid network (FPN) architecture [120] and was originally trained on a collection of everyday objects featured in the COCO database [121]. The COCO database is a large-scale object detection, segmentation, and captioning dataset that contains over 200,000 labeled images.

The main architecture of the base model from Detectron2 is shown in Figure 4.6. The architecture consists of four major sections; a backbone feature pyramid network (FPN) [122], a region proposal network (RPN) [123], and two region of interest (ROI) heads [124], one for the box heads and the other for the masks.

The FPN takes in the input HR-AFM image and extracts multiple feature maps at different scales with varying receptive fields. The RPN detects regions of interests in the feature maps produced by the FPN and provides (by default) 1000 boxes with respective confidence intervals. The ROI heads take those 1000 boxes from the RPN and fine tune using a fully connected network. Our model was trained on an NVIDIA GEForce RTK 3080, running all the scripts on a Jupyter notebook.

Experimental Methods The first step in training these models is collecting the data by imaging the molecules of interest. Our dataset consists of 599 images of 160 unique molecules found in petroleum compounds. SPM measurements were performed with a Createc-based low-temperature (LT)-STM system custom upgraded with HR-AFM capability and operated using the open-source GXSM control software [116]. The UHV system base pressure was 7e-11 torr. Our system uses a qPlus sensor [22] that has a 25 μm PtIr tip wire attached. The wire was cut and sharpened by focused ion beam (FIB) milling. The sensor went through a final cleaning procedure in UHV using Ar^+ sputtering from three directions before being mounted on the SPM scanner.

Sample and tip preparation A clean Cu(111) surface was prepared using standard Ar^+ sputter anneal cycles for refreshing a previous clean crystal. A typical cleaning consists of 3 to 4 cycles of 4 to 5 μA at 1 kV Ar^+ sputtering on a 8mm diameter crystal for 15 minutes each followed by 10 to 25 minutes annealing up to 550 °C. After the Ar^+ sputter anneal cycles the crystal is loaded into the microscope and cooled to 5 K.

We directly deposited molecules on the cold surface. Pure molecules were typically sublimated, while molecular mixtures are usually "flashed" (quick one shot heat up to over 800°C) from miniature amounts of powder adhered to a silicone carrier substrate that was heated by direct current. Depending on the molecules, they may be pre-purified via a brief test sublimation process in UHV before exposing the source to our Cu(111) surface. This also allows for a simple visual rate adjustment via finding the onset of powder on the carrier starting to diminish during the sublimation process. During deposition via the cryostat door into the STM on the sample the temperature rose briefly due to radiation exposure to about 10 K.

Ultimate metal tip apex shaping was performed via controlled nano indentations into the copper metal crystal and bias pulsing.

The last preparation step always was exposing the sample to a small CO dose while at 5 K, specifically 10 s CO exposure at 2×10^{-8} mbar via the cryostat door as required for tip functionalization purpose only.

The qPlus sensor used for this work operated at a resonance of 30210 Hz with a typical Q-factor of 10'000. The CO pickup from Cu(111) was achieved simply via consecutive scanning over a CO molecule in very close proximity in STM mode with a bias of a few mV and various currents up to 50 pA.

Information on HR-AFM measurements performed using GXSM HR-AFM measurements were performed using GXSM's special constant height control mode with automated constant current (STM mode) transitions if a compliance setting (probe safety or also automated big/3D molecule lift mode) of a max allowed tunnel current is exceeded. Therefore a small bias of 20 mV was typically applied in HR-AFM-mode. For frequency detection, the custom hi-speed GXSM RedPitaya-PAC-PLL controller was used in combination with the MK3-A810 SPM-Controller [1]. Tip oscillation amplitudes were typically around 60 pm.

STS and dI/dV spectroscopy was performed using an external Lock-In Amplifier (SRS Model 7265 Dual Phase DSP Lock-in Amplifier). The bias was modulated at 299 Hz at typically 10mV or 5mV pure sine amplitude.

Data preparation All the images collected for the dataset had a 330 x 330 pixel resolution and were fed into the model as jpg files. 60 images were randomly picked from the 599 images and were simply augmented by flipping and slight shearing. These augmented 60 images were used to train our machine learning model to recognize three different classes of probe-molecule distance, which can be seen in Figure 4.11. Regions where the probe-molecule distance is ideal for imaging hydrocarbon molecules are colored in green, regions

[1] More information about GXSM can be found at http://gxsm.sf.net. See also the GXSM project page and related forums

where the probe is too far away are colored in red, and regions where the probe is too close to the molecules are colored in purple. The three classes are used to teach the AI to distinguish between the different heights the probe may have while imaging. The other 539 images were used to test the efficiency of the model.

Labeling data An essential step to train the machine learning model used by Auto-HR-AFM was to create and label a training dataset. The quality of the model depends on the quality of the annotated data. The training dataset was made up of HR-AFM images of hydrocarbons that were previously collected.

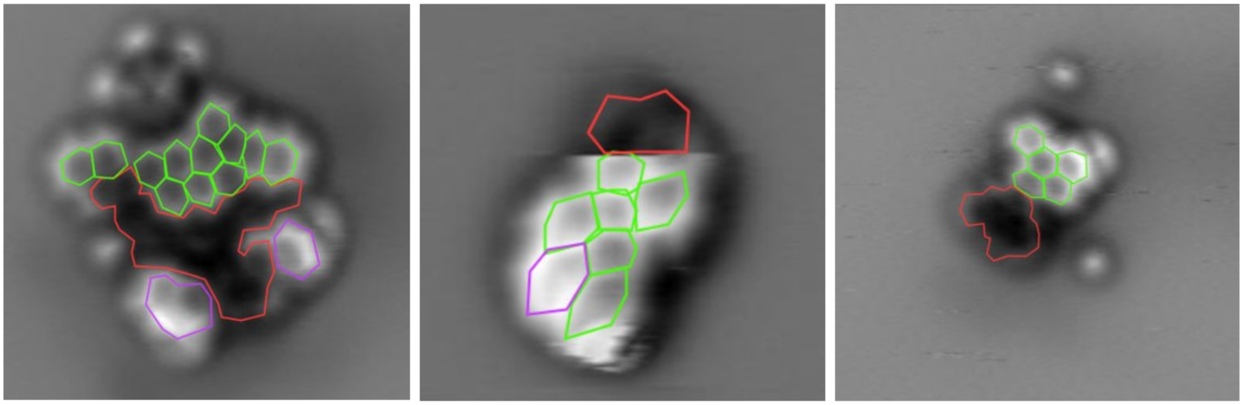


Figure 4.7: Examples of labeled data using makesense.ai.

The HR-AFM images were labeled using a web browser tool called makesense.ai [125][2]. Such labeling tool was chosen due to its simple and efficient design. Figure 4.7 shows examples of the annotations created using the labeling tool. Regions where the probe-molecule distance is ideal for imaging hydrocarbon molecules are labeled with a green polygon, regions where the probe is too far away are labeled with a red polygon, and regions where the probe is too close to the molecules are labeled with a purple polygon.

makesense.ai is capable of creating commonly used annotations: rectangles, points, lines and polygons. Figure 4.8 Rectangles are known as bounding boxes and are typically used

[2] More information about makesense.ai can be found at https://www.makesense.ai and at https://skalskip.github.io/make-sense/

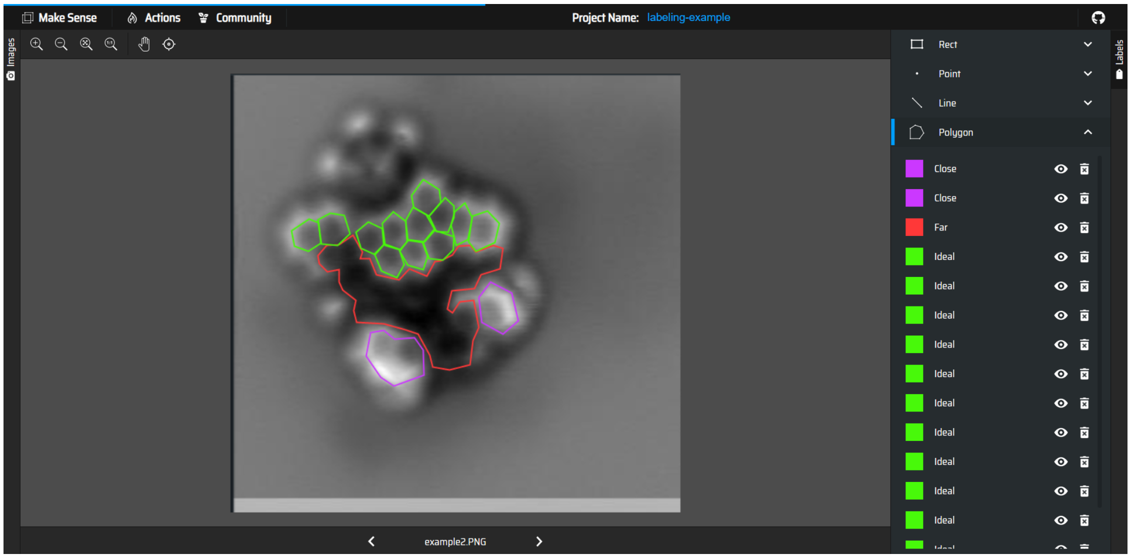


Figure 4.8: User interface of the makesense.ai labeling tool. The browser allows for new data to be uploaded and labeled. Different annotation styles can be selected in the right column. The annotations can be tracked and edited from the right column as well. Prepared labels can be downloaded in multiple formats.

for object detection and image localization machine learning models. Points are used to detect small objects and shape variations and are typically used to detect facial features, expressions, emotions, body part movements, and poses. Lines are typically used to recognize and detect lanes and outlines. Polygons are used the same way as bounding boxes but are more precise in finding the exact shape and location of the object, a reason why we chose to use these for this specific project.

The prepared labels can be downloaded in one of the supported formats. The most common formats are COCO, Pascal VOC, and YOLO. For this training dataset we downloaded the labels to json file in the COCO format [121], since that is the file type read by Detectron2. Labeling projects can be reuploaded to the makesense.ai browser tool to continue editing the labeling process which is useful especially when you want to add on more labels or expand previous datasets. The labeling process can take a significant amount of time and depends on the details of the molecules, the number of classes created, and the size of the dataset. For this training dataset, I labeled 599 images averaging around 50 to 100 images

Table 4.1: Hand-Counted vs AI Found

Hand Counted Instances	Ideal Rings Found	Far Regions Found	Close Regions Found
3193	2195	1235	876

Table 4.1: Number of hand-counted ring instances compared to the number of ring instances, far, and close regions found by the AI.

Table 4.2: Regions of Interest Percentages

ROI	Percentage
Rings Found	68.7%
Mislabeled Regions	2.95%
Flat Planar Molecules	3.84%

Table 4.2: Rings Found: The number of rings found by the AI compared to the hand-counted rings, Mislabled Regions: Percent of AI mislabeled areas that did not correspond to an area on the molecules, Flat Planar Molecules: Percent of the total images that were planar and laid flat on the surface.

per hour using the 3 classes.

Model Performance

Auto-HR-AFM outputs an instance segmentation of the input HR-AFM image using the three classes. Example outputs can be seen in Figure 4.9. All the ring instances were also counted by hand and we found 3193 instances of ideal aromatic rings throughout the 599 images in the dataset. Auto-HR-AFM found 68.75 % of these rings but was also more selective than the user to count an ideal ring. Rings that had a very faint contrast, but were still visible by a user were accounted for in the 1235 instances of far regions found. The rest of the rings that Auto-HR-AFM grouped in the 876 instances of close regions were mostly due to them having a bright feature close to them.

Out of the 599 HR-AFM images, 3.84 % were of flat planar molecules. These images had 100% of their rings identified. The more complex, non-planar molecules have a combination of two or all of the different classes. Only 2.95 % of the images had mislabeled regions that

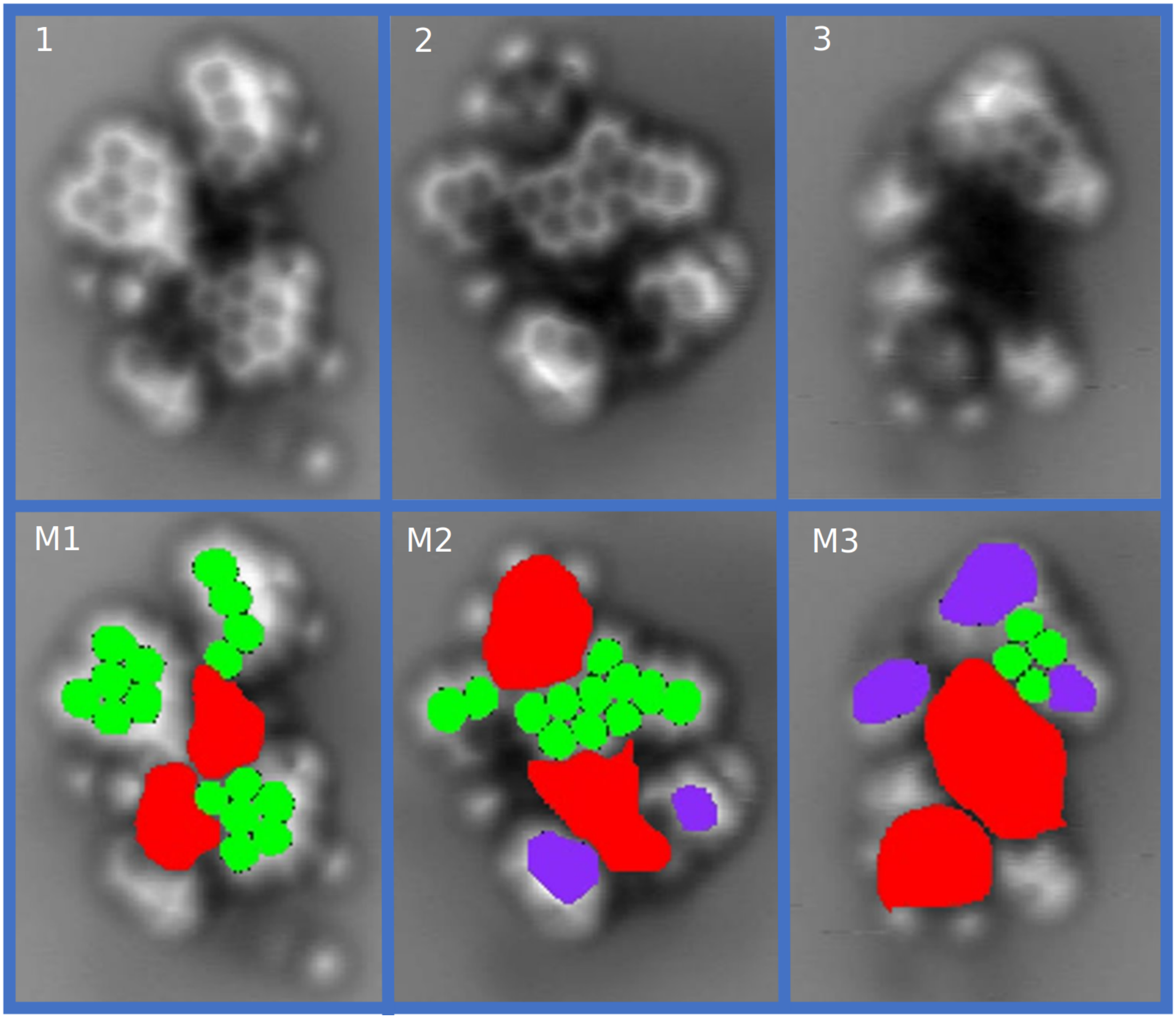


Figure 4.9: Visual representations of the outputs from the instance segmentation models. The top row are the images taken by the HR-AFM that are used as inputs for the model. The bottom row overlays how the model segments the image based on the distance between the tip and the molecule.

did not correspond to the molecule. The output of the model locates the regions of the image that correspond to the three different classes and stores that information in an array. Knowing which class is predominant in an image helps Auto-HR-AFM classify the image into either an ideal image or an image where the probe is either too far or too close to the molecule. After this classification, the AI decides whether to adjust the probe height and retake the image before proceeding to the next molecule.

Installing Detectron2 and PyTorch

Auto-HR-AFM requires Detectron2, Pytorch, and torchvision to be installed. Detailed installation instructions for these packages can be obtained at:

Detectron 2 - https://detectron2.readthedocs.io/en/latest/tutorials/install.html

Pytorch and torchvision: https://pytorch.org/

These packages have their own requirements for installation. The versions of the installed Pytorch and torchvision packages have to be compatible with the CUDA toolkit installed in the system. If previous versions of these packages are installed already installed it is best to either reinstall them or install them individually to make sure two versions are not installed at the same time.

Results and Discussion

Automating SPM tasks using AI is especially important for experiments that require collecting a large set of images in a limited time frame. Delegating tasks to the AI optimizes the time the SPM is in use. In this work, we developed and tested Auto-HR-AFM on images of a mixture of petroleum molecules deposited on Cu(111) and imaged with a CO-functionalized probe. Here we describe the steps that Auto-HR-AFM takes to automatically collect optimized data using our instance segmentation model.

Auto-HR-AFM operates in a loop composed of four parts as shown in Figure 4.10. It takes as an input an overview STM Figure 4.10 a an image containing a distribution of tagged molecules. Auto-HR-AFM then collects optimal images of these molecules in both STM (Figure 4.10 b) and HR-AFM (Figure 4.10 c,d) modes, unsupervised.

Before running Auto-HR-AFM, the user obtains a CO-functionalized tip and then takes an STM image of a region of the sample that contains a distribution of molecules. The user manually selects target molecules from the overview and then starts the Auto-HR-AFM script. Regions of interest containing target molecules can also be selected using a script that automatically finds molecules on the surface like the one used by Hellerstedt [99], but these scripts can confuse similar molecules, and their performance is affected by other features on the surface, so it is more robust to manually select the regions of interest. The order these molecules are selected will be the order the script will continue to image them while in the loop.

Next, the script moves the tip to the first target molecule and initiates a high-resolution STM image Figure 4.10 b. Depending on the time between the first overview image and the zoomed image, there could be some thermal drift that has shifted the position of the molecule. To make sure that the molecule is centered, the script calculates the center of mass of the molecule and then resets the center of the image to that location and re-images.

Next, Auto-HR-AFM determines a safe initial height value to take the first HR-AFM image by checking the current around the molecule in STM mode to estimate its height on Cu(111). Typical values are: Then Auto-HR-AFM performs an STM-to-HR-AFM transition using the SPM software and starts to collect an initial HR-AFM image. For molecules in a complex mixture, this initial image is not often ideal, especially for larger molecules whose apparent heights are not uniform.

The final part of the operation loop assesses this initial HR-AFM image and decides what action to take to collect a better image: Figure 4.10 c. Auto-HR-AFM uses a trained instance segmentation model to determine regions in the imaged molecule where the probe is too close or too far to collect a sharp HR-AFM image and regions where the probe-molecule

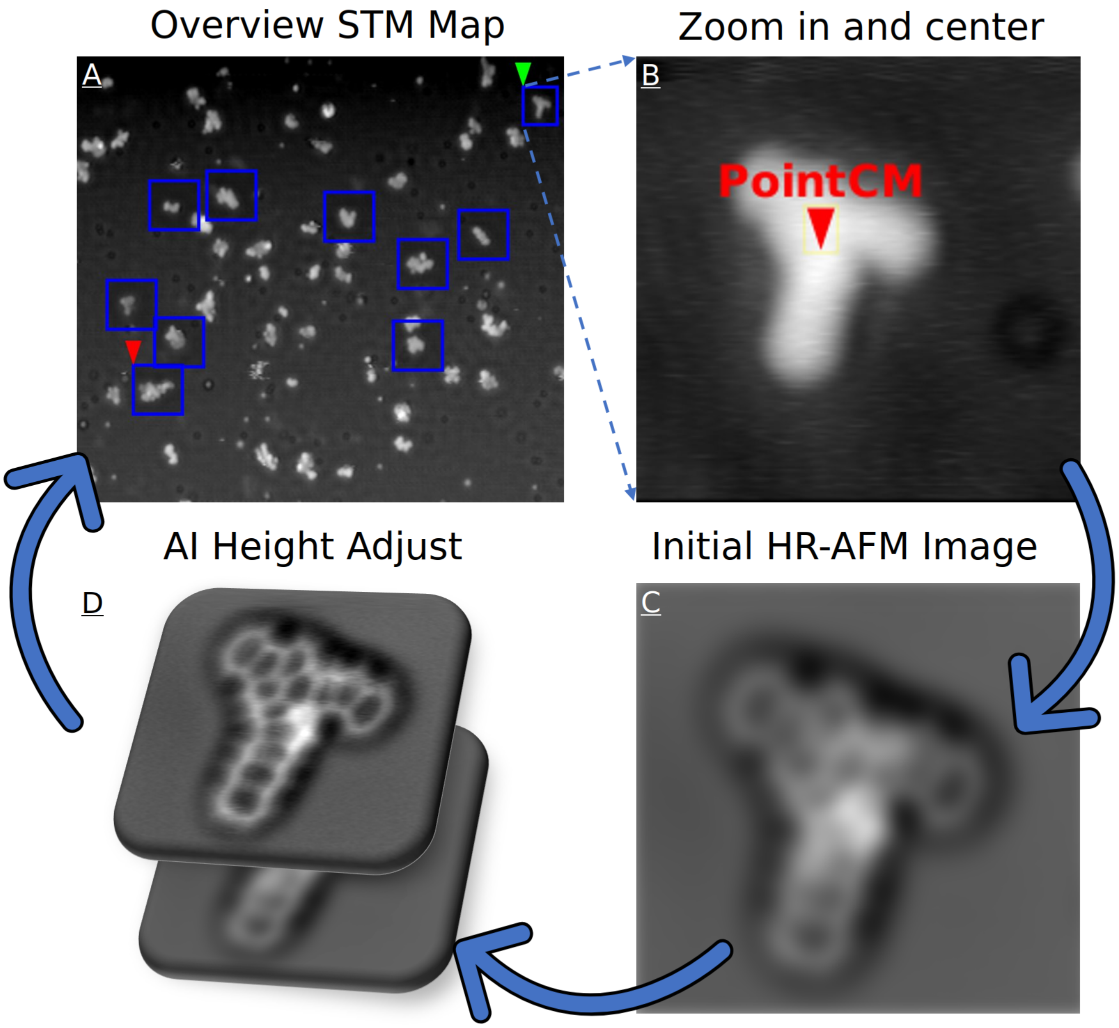


Figure 4.10: Auto-HR-AFM Script Architecture. a) Overview STM image with molecules queued to be imaged. b) Zoomed in STM image of the selected molecule, the center of mass of the molecule is used as the center coordinate to keep the molecule in frame. c) Auto-HR-AFM switches to HR-AFM mode and collects an initial image. d) The image from c) is passed through a ML algorithm to assess then optimize the quality of the imaging. Once an optimal HR-AFM is collected Auto-HR-AFM continues with the next selected molecule.

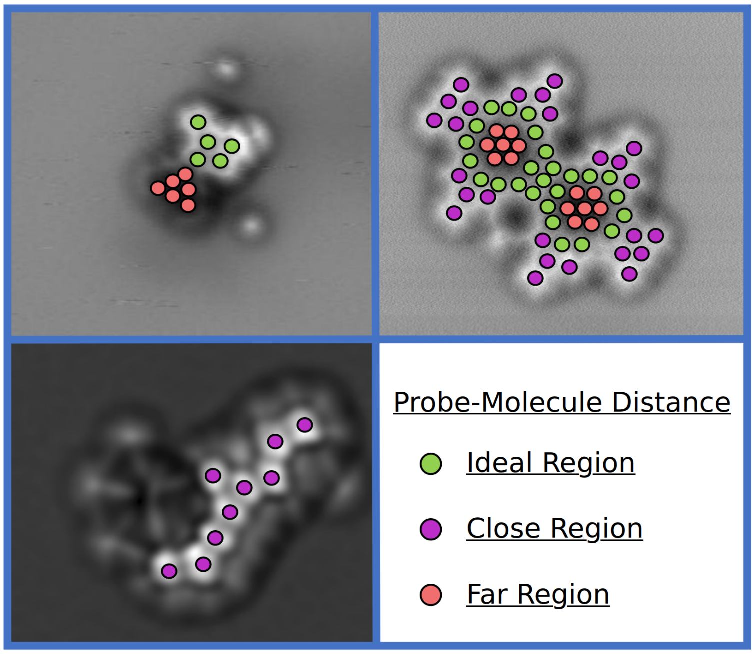


Figure 4.11: The three regions of probe-molecule distances Auto-HR-AFM is trained to detect using a diverse range of input data sources. Regions with an ideal distance for optimal HR-AFM images are seen in green. Regions that are too close or too far are seen in purple and red respectively. Left: Hydrocarbons found in petroleum mixtures, Top Right: Graphene Nanoribbon

distance is ideal [3]. These regions can be seen in Figure4.11. The instance segmentation model was trained and tested on previously collected HR-AFM images of hydrocarbons found in petroleum mixtures and can also recognize probe-molecule distances in other molecules like graphene nanoribbons as seen in Figure 4.11.

When the probe is too far or too close to the molecule, Auto-HR-AFM takes a single step of ± 0.3 Angstroms and takes another, now more optimal image. For molecules lying flat on the Cu(111) surface, the AI's decision is straightforward, either move closer, move further away, or continue on to the next molecule if the height is already optimal. For more difficult situations when the molecules are more complex and not lying flat on the surface there is a combination of the three different regions. Auto-HR-AFM segments the molecule into too far, too close or ideal imaging regions. Based on which region is predominant on the

[3] Details of these regions seen in the Data Labeling section of this Chapter

molecule, Auto-HR-AFM adjusts the probe to take a more optimal image of the molecule.

Similar to a human user, Auto-HR-AFM moves on to the next molecule once an optimal image is collected. Auto-HR-AFM's script can be customized to collect a series of images on any given molecule or to perform other SPM tasks before moving on. When the script decides to move on to the next molecule, the loop repeats. The loop ends when there are no more molecules to image in a given overview area. The user could then select a new area to image and restart the script to continue data collection. Considering that each molecule could take somewhere between thirty minutes to an hour, and there are overview scans that have greater than ten molecules of interest in them, this tool can be applied overnight or left running for days depending on how many target molecules are defined by the user.

A functionalized CO probe can be stable throughout multiple data collection runs, but there is always the possibility for the CO at the tip apex becomes lost or shifted during any run. If this happens as the script is running, the loop will continue but all the HR-AFM images will appear dark and no atomic resolution will be achieved. At this point, the user has to intervene and functionalize the probe with CO once more. Once the probe is functionalized again the user can run the script once more to continue imaging where it left off or to repeat some of the imaging while the CO was lost. A future instance of the script will be able to recognize this situation and interrupt the loop to wait for the user's intervention.

We explored and demonstrated the first practical operation of fully automated HR-AFM imaging applied to multiple distinct molecules in petroleum-based molecular mixtures. Our Auto-HR-AFM script is fully open-source and customizable, ready to be expanded for wide use. Potential next steps to improve Auto-HR-AFM would be to include other SPM ML tools like navigation and tip tuning [89, 93], and to expand the different types of molecules that the script can recognize, specifically molecules without carbon rings. Combining these ML tools is the launching point for a fully automated SPM that can handle any molecular mixture characterization project from start to finish.

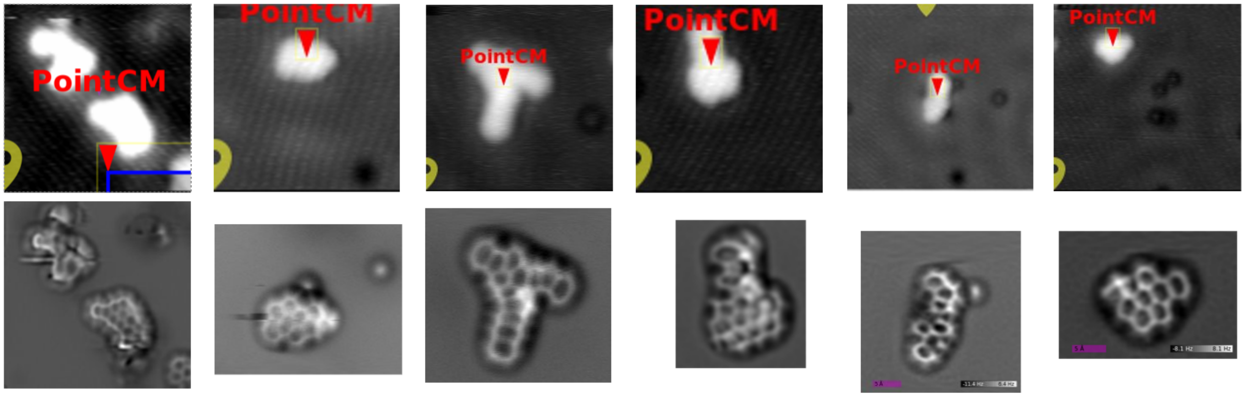


Figure 4.12: Automatically collected A) STM and B) HR-AFM of Hydrocarbons found in petroleum mixtures.

STM and HR-AFM images acquired automatically Before the incorporation of our trained machine learning models we tested the automation scripts included into Auto-HR-AFM. Figure 4.12 showcases SPM images of hydrocarbons found in a complex molecular mixture collected by the automation script before the incorporation of our final machine learning model to find the optimal distance. The top row are STM images of the molecules that show with a red marker where the center of mass of each image is located by the script. The bottom row are the HR-AFM images that were collected by the automation script while being supervised by a user in case the initial image needed to be optimized.

Some of these initial images needed user intervention[4] to optimize the imaging as seen in Figure 4.13, where a user tuned the probe-molecule distance to get more data on the molecules. Without the machine learning model incorporated in Auto-HR-AFM, the script collected SPM images but had no way to optimize them.

[4] Our script allows users to tweak parameters and intervene flow control while running.

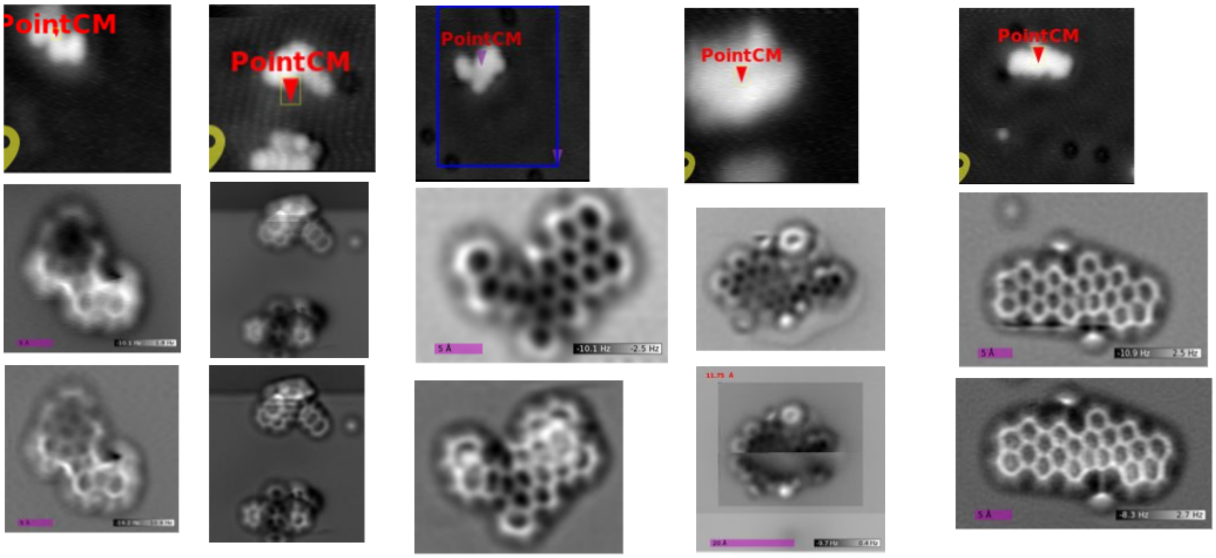


Figure 4.13: Hydrocarbons found in petroleum mixtures collected by the automation script and tuned by a supervising user. A) STM images, B) HR-AFM images, C) HR-AFM images after the user tuned the imaging parameters.

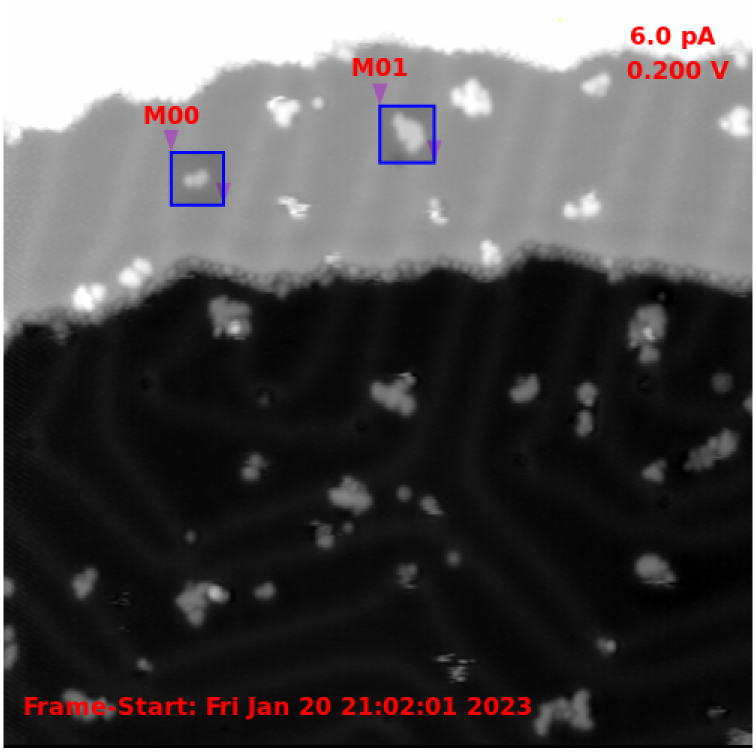
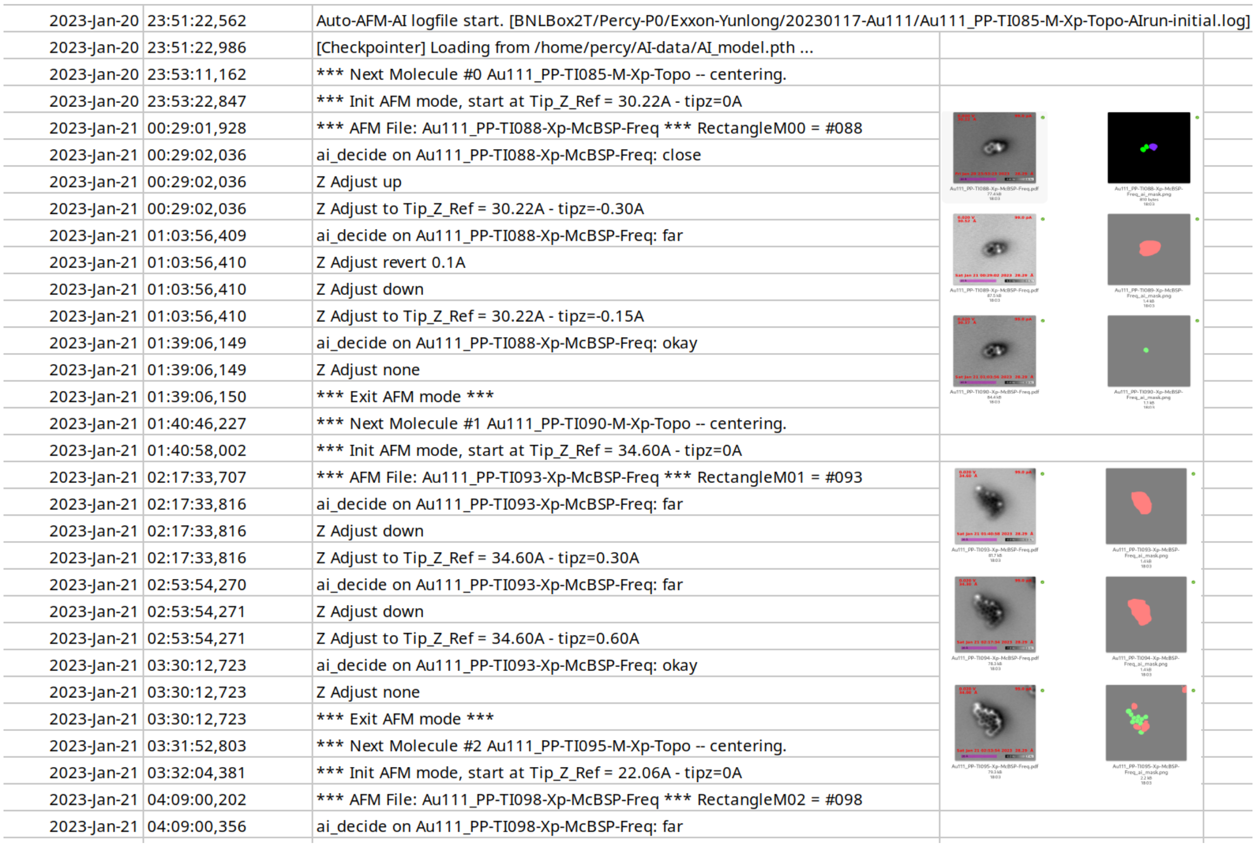


Figure 4.14: STM map for Auto-HR-AFM run with selected molecules.

Detailed transcript log of fully automated AI controlled HR-AFM molecule imaging In the latest version of Auto-HR-AFM, the HR-AFM image instance segmentation model and interpretation via AI based regional classification is fully incorporated. We added a live visualization of the AI instance segmentation model that can be used to monitor its performance.

A transcript logfile is written by the Auto-HR-AFM script to document every step taken by Auto-HR-AFM to collect an optimal HR-AFM image. Figure 4.15 shows the detailed log that keeps track of the timestamps of major actions performed by Auto-HR-AFM on the STM map shown in Figure 4.14. The log tracks the adjustments that Auto-HR-AFM performs on the probe-molecule distance based on the information provided by the instance segmentation model.

In the logfile excerpt as shown in Figure 4.15 we added related images and AI generated feature maps for far, ideal, and close regions as used for decision making and resulting



```
2023-Jan-20 23:51:22,562  Auto-AFM-AI logfile start. [BNLBox2T/Percy-P0/Exxon-Yunlong/20230117-Au111/Au111_PP-TI085-M-Xp-Topo-AIrun-initial.log]
2023-Jan-20 23:51:22,986  [Checkpointer] Loading from /home/percy/AI-data/AI_model.pth ...
2023-Jan-20 23:53:11,162  *** Next Molecule #0 Au111_PP-TI085-M-Xp-Topo -- centering.
2023-Jan-20 23:53:22,847  *** Init AFM mode, start at Tip_Z_Ref = 30.22A - tipz=0A
2023-Jan-21 00:29:01,928  *** AFM File: Au111_PP-TI088-Xp-McBSP-Freq *** RectangleM00 = #088
2023-Jan-21 00:29:02,036  ai_decide on Au111_PP-TI088-Xp-McBSP-Freq: close
2023-Jan-21 00:29:02,036  Z Adjust up
2023-Jan-21 00:29:02,036  Z Adjust to Tip_Z_Ref = 30.22A - tipz=-0.30A
2023-Jan-21 01:03:56,409  ai_decide on Au111_PP-TI088-Xp-McBSP-Freq: far
2023-Jan-21 01:03:56,410  Z Adjust revert 0.1A
2023-Jan-21 01:03:56,410  Z Adjust down
2023-Jan-21 01:03:56,410  Z Adjust to Tip_Z_Ref = 30.22A - tipz=-0.15A
2023-Jan-21 01:39:06,149  ai_decide on Au111_PP-TI088-Xp-McBSP-Freq: okay
2023-Jan-21 01:39:06,149  Z Adjust none
2023-Jan-21 01:39:06,150  *** Exit AFM mode ***
2023-Jan-21 01:40:46,227  *** Next Molecule #1 Au111_PP-TI090-M-Xp-Topo -- centering.
2023-Jan-21 01:40:58,002  *** Init AFM mode, start at Tip_Z_Ref = 34.60A - tipz=0A
2023-Jan-21 02:17:33,707  *** AFM File: Au111_PP-TI093-Xp-McBSP-Freq *** RectangleM01 = #093
2023-Jan-21 02:17:33,816  ai_decide on Au111_PP-TI093-Xp-McBSP-Freq: far
2023-Jan-21 02:17:33,816  Z Adjust down
2023-Jan-21 02:17:33,816  Z Adjust to Tip_Z_Ref = 34.60A - tipz=0.30A
2023-Jan-21 02:53:54,270  ai_decide on Au111_PP-TI093-Xp-McBSP-Freq: far
2023-Jan-21 02:53:54,271  Z Adjust down
2023-Jan-21 02:53:54,271  Z Adjust to Tip_Z_Ref = 34.60A - tipz=0.60A
2023-Jan-21 03:30:12,723  ai_decide on Au111_PP-TI093-Xp-McBSP-Freq: okay
2023-Jan-21 03:30:12,723  Z Adjust none
2023-Jan-21 03:30:12,723  *** Exit AFM mode ***
2023-Jan-21 03:31:52,803  *** Next Molecule #2 Au111_PP-TI095-M-Xp-Topo -- centering.
2023-Jan-21 03:32:04,381  *** Init AFM mode, start at Tip_Z_Ref = 22.06A - tipz=0A
2023-Jan-21 04:09:00,202  *** AFM File: Au111_PP-TI098-Xp-McBSP-Freq *** RectangleM02 = #098
2023-Jan-21 04:09:00,356  ai_decide on Au111_PP-TI098-Xp-McBSP-Freq: far
```

Figure 4.15: Excerpt for a generated transcript log that describes the actions Auto-HR-AFM takes to optimize the imaging along with collected images. Here annotated with mating, auto-generated images and generated AI feature maps as used for decision making. Note: Data of a petroleum-based molecule mixture sample flash deposited on Au(111).

actions. For the first selected molecule labeled with “RectangleM00” the initial probe-molecule distance ended up a being too close. The AI optimized the probe-molecule distance by the third image and then proceeded to the next selected molecule“M01”. This starts out more usual and a bit on the far side. After two AI-steered tweaks it accepts and proceeds.

Conclusions

The use of HR-AFM to image and study the chemical structure of individual molecules will help us understand the chemical structure of complicated molecules [126], identify the composition of complicated molecular mixtures [108], and differentiate heteroatoms [127]. Training a machine learning model to recognize the patterns commonly seen in an SPM experiment can help shift the routine work from user to AI. We created Auto-HR-AFM to automate HR-AFM image collection of molecular mixtures. Auto-HR-AFM uses an instance segmentation model we trained to recognize three different probe-molecule distances. Using this information Auto-HR-AFM adjusts the probe-molecule distance to collect the best possible image of each molecule. Auto-HR-AFM receives instructions to image certain molecules and one by one collects the most optimal image of each molecule.

Auto-HR-AFM's AI-guided, automated scanning for AFM imaging enables unsupervised imaging. Using Auto-HR-AFM will lead to more high-quality data being collected in the limited time frame SPM users are allocated per project. This is specifically useful at national labs where typical projects are only allocated one to two weeks at a time. This optimizes the use of the SPM and related resources needed to keep the system running. Most importantly, because imaging each molecule is automatically optimized, projects will achieve higher fidelity structure interpretation, and also a more diverse data pool, since images of non planar molecules are also collected.

Despite these advantages and significant improvements, future additions and improvements will be needed. First, adding more ML decision-making tools to Auto-HR-AFM will lead to a fully-automated SPM. Recent work automating SPM actions like navigation, tip-tuning, and spectroscopy [89,91,93,114] can be updated and integrated with Auto-HR-AFM to make the technique more broadly applicable.

Second, recent work using ML models to identify chemical structure and determine the nomenclature of molecules imaged by HR-AFM [92,128]. Auto-HR-AFM's script can be modified to integrate these molecular identification techniques. By collecting multiple molecules at a series of different heights, we can use these as input for molecular identification techniques and identify molecules in real-time.

Third, Auto-HR-AFM can learn to recognize more details in the HR-AFM images if we expand the three classes used in the instance segmentation model. By implementing more chemistry rules as we create more classes, the AI could recognize the number of carbon atoms in the rings of these molecules. Six-membered rings are most common and are expected for aromatic structures, five-membered rings are also possible, but three, four, seven, or eight membered-rings are rare and have not been reported. Also, all carbons need to satisfy tetravalency for most stable molecules, although other valencies are possible such as stable free radicals. Structures have to be consistent with their sources, formation conditions, or reactivities. For example, some aromatic structures are extremely reactive and unstable. Training a machine learning model to recognize these chemistry rules while scanning will facilitate the discovery of these unusual structures.

Auto-HR-AFM can recognize the common patterns and what a trained SPM user sees. Having this visual aid helps the AI decide on what the next best action is to optimize the imaging. Depending upon SPM users to make these decisions is the reason why these SPM experiments require intense supervision. Fully automated SPM frees the instrument from the user and enables maximum output of the SPM.

CHAPTER 5

Conclusions

Realistically cutting our ties with petroleum will not happen overnight. Our natural supplies will last us at least 50 to 100 years more [108]. It is important to continue researching new and efficient ways to make these reservoirs last as long as possible while attempting to produce the least amount of pollutants as we can. Designing new refining processes can be difficult since there is still much we do not know about the chemical structures of crude and refined products.

NC-AFM provides structural information on the molecules that compose crude oil, intermediates, and final products. Although the technique is complicated and time-consuming it provides the most detailed atomic scale information of the molecular structures found in complex mixtures. Since NC-AFM does not require conductive supports for their samples, it is a great tool to explore these refining processes on industrially relevant experimental systems.

The work I present in this dissertation uses NC-AFM to investigate the active catalytic sites of Mo_2 on industrially relevant supporting substrates (Chapter 3) and integrates machine learning/AI tools to automate NC-AFM data collection (Chapter 4).

In Chapter 3 the goal was to perform NC-AFM studies on the active catalytic sites of MoS_2 on top of insulating substrates. Due to limitations our resolution mostly from the RHK system setup this project was left incomplete. However, hydrotreatment processes using catalysts like MoS_2 have gained more interest recently, since there is a rise in biofuels that have a high C-O bond content. Understanding the role that catalysts like MoS_2 have on hydrodesulfurization reactions and improving those processes can help create efficient hydrodeoxygenation (HDO) processes to reduce the amount of C-O bonds in biofuels. Creating new efficient hydrotreatment processes will help create cleaner fuel sources.

If this project were to continue in the Hollen lab two main issues have to be resolved to improve the stability of the NC-AFM while imaging. The first is improving the resolution of the qPlus sensors in both STM and NC-AFM modes. The second is to reduce the number of procedures done during sample preparations to reduce surface contaminants.

To improve the resolution of the qPlus probes, I suggest taking a step back and testing the probes on a metal surface like gold or copper. The metal surfaces are a good playground to develop a tip functionalization procedure that is needed for high-resolution imaging of molecules. Imaging a CO molecule or another test molecule would be a good starting ground to test the noise limitations of the system. Once a solid understanding of using qPlus probes on metals and a tip functionalization procedure is developed, then it will be easier to transition to insulating surfaces.

The second issue involves the cleanliness of the samples used for this project. The reason that NC-AFM works so well on metals is that the probe is operating on a clean and flat surface. Contaminants on the surface cause the tip to misbehave and can potentially cause crashes that might affect the tip quality. Typical systems use STM mode as a backup to make sure the tip does not crash. When dealing with insulator surfaces, the STM backup is not an option. Exploring the high resolution capabilities of the qPlus probes on insulators is something that has not been explored much.

Limiting the contaminants on the surface will improve the stability of the probe. When creating the MoS_2 on SiO_2 samples, the exfoliation produced flakes that were clean and flat, but it was difficult to image them without having a large overview scan or a marker to pinpoint their location. Adding the gold grid helped in finding the flakes, but I believe introduced multiple contaminants near the flakes, especially from the chromium sticking layer. Increasing the size of the flakes so that they are easier to find or moving smaller flakes to a pre patterned clean surface would be a better process to create these samples. For the MoS_2 on AAO, increasing the size of the MoS_2 flakes so that we can find them easily is the best option since it would be difficult to pattern.

Creating a procedure to functionalize the qPlus tips on insulators and changing the stiffness of the tuning forks used could also improve the stability. Measuring the forces on the qPlus probes when approaching insulating surfaces under different functionalizations could be an interesting project in creating a qPlus sensor that is dedicated to a stable approach and imaging on insulators.

While NC-AFM can provide high-resolution images of chemical structures, intermediates, and final reaction products the technique requires stable and low temperatures to freeze molecular motion. These low temperatures are far from the ranges in which hydroprocessing catalysis processes take place. To study these processes near-ambient pressure microscopes have to be developed that can also achieve high resolution. These can be used similarly to the ambient pressure STM to study reactions in more relevant pressure environments.

To conclude in Chapter 3, here is a list of potential improvements for NC-AFM experiments of 2D flakes on insulating supports in the Hollen Lab:

- Improve Exfoliation techniques to produce larger flakes of MoS_2 or any other 2D material.

- Explore the capabilities of the qPlus.

  - Create a procedure to dose CO into the chamber.

  - Create a procedure to functionalize qPlus tips with a CO molecule.

  - Run NC-AFM tests with a functionalized tip on conductive test samples to get high-resolution imaging working.

  - Consider physical changes to probe design or RHK system to reduce noise

  - Explore the capabilities of imaging insulators while pursuing stabilizing the qPlus sensors.

- Dosing experiments: STM studies of the thiophene on gold or another sulfur-containing molecule. Switch to powders potentially.

In Chapter 4 I present my work to create an initial SPM automation script that trains artificial intelligence with computer vision and machine learning algorithms. The main result of the chapter is the creation of Auto-HR-AFM, an AI script that collects optimal HR-AFM images of hydrocarbons. Auto-HR-AFM is open source and customizable to integrate future SPM tasks. Eventually providing a framework for a fully automated SPM script that can control a system from tip approach to data collection.

The field of automation using machine learning and computer vision is growing rapidly in many fields, especially in computer science and robotics leading to commercial applications like self-driving cars, AI art and text generators like Dall-E and ChatGPT. Similar to these commercial applications there is an interest in utilizing these machine learning models and computer vision tools in the field of SPM. Initial applications have focused on automating specific tasks to navigate, perform spectroscopy, and to check the quality of the tip. Our work added automating the collection of HR-AFM images to that list. Future work should aim to combine all these automation tools into a single framework that is easy to customize and update.

For a single framework to automate every SPM task in a specific system, a main issue to take into consideration is compatibility. With Auto-HR-AFM we designed it to be modular so that it can be a living project that can be updated when a new tool is developed for it. This is especially useful in the field of ML where there is a fast-paced growth in new techniques. Older models and automation scripts also can be outdated easily and should be updated frequently to make sure that they are compatible with newer models created in the future.

With regards to Auto-HR-AFM specifically, there are many different avenues to continue improving the functionality of the script. To make the script more robust, more trials are needed to check for any bugs or to see where the code can be improved. Adding a larger data set with more labeled classes to teach the AI to recognize more features would be a good next step.

There is work being done to label and name the molecules seen in NC-AFM images using ML techniques [93, 129]; this can be integrated into Auto-HR-AFM to perform these techniques in real-time during the data collection. To have the AI recognize more molecules faster, more chemistry rules should be input into the training data set to categorize and create more classes. This would help the AI check for aromaticity, label carbons in rings, and label heteroatoms as well, making the process of identifying the molecules in real time much more effective.

Combining all these tools with the past ones will enable a fully automated SPM script that can control every aspect of data collection. Still there will always be more improvements to perform new experiments. This is a continuous process that has to be updated and has to evolve to continue to be useful. If achieved, this automation will make data collection much more efficient and will relieve the time users have to spend on collecting optimal data.

Improving SPM techniques with automation will help accelerate discovery of petroleum molecules and help scientists and engineers tune the refining processes used. Petroleum products and intermediates are complicated. To design better refining processes we need to understand their exact composition. NC-AFM is the best tool to collect this data, but the technique is still complicated and time-consuming. Automation will help utilize the technique more efficiently. The technique can also help us understand some of the current refining processes in more industrially relevant environments, but there are still limitations when performing NC-AFM at ambient pressures and on insulators. The continuing improvement of these techniques can help us understand the petroleum world more each day.

LIST OF REFERENCES

[1] Net zero by 2050 - a roadmap for the global energy sector.

[2] Kendall Beaton. Dr. gesner's kerosene: The start of american oil refining. 29(1):28–53. Publisher: [President and Fellows of Harvard College, Cambridge University Press].

[3] K. Andrews. Cell phones, aspirin, toilets, and 60 other byproducts of crude oil, 2022.

[4] D. B. Jack. One hundred years of aspirin. 350(9075):437–439.

[5] Refining crude oil - the refining process - u.s. energy information administration (EIA).

[6] G. Binnig and H. Rohrer. Scanning tunneling microscopy. 126(1):236–244.

[7] G. Binnig, C. F. Quate, and Ch. Gerber. Atomic force microscope. 56(9):930–933. Publisher: American Physical Society.

[8] J. V. Lauritsen, M. Nyberg, R. T. Vang, M. V. Bollinger, B. S. Clausen, H. Topsøe, K. W. Jacobsen, E. Lægsgaard, J. K. Nørskov, and F. Besenbacher. Chemistry of one-dimensional metallic edge states in MoS2 nanoclusters. 14(3):385.

[9] J.V. Lauritsen, M. Nyberg, J.K. Nørskov, B.S. Clausen, H. Topsøe, E. Lægsgaard, and F. Besenbacher. Hydrodesulfurization reaction pathways on MoS2 nanoclusters revealed by scanning tunneling microscopy. 224(1):94–106.

[10] J. V. Lauritsen and F. Besenbacher. Atom-resolved scanning tunneling microscopy investigations of molecular adsorption on MoS2 and CoMoS hydrodesulfurization catalysts. 328:49–58.

[11] Yunlong Zhang. Perspectives on a new age of materials for petroleum. 36(11):5529–5530.

[12] Bruno Schuler, Shadi Fatayer, Gerhard Meyer, Estrella Rogel, Michael Moir, Yunlong Zhang, Michael R. Harper, Andrew E. Pomerantz, Kyle D. Bake, Matthias Witt, Diego Peña, J. Douglas Kushnerick, Oliver C. Mullins, Cesar Ovalles, Frans G. A. van den Berg, and Leo Gross. Heavy oil based mixtures of different origins and treatments studied by atomic force microscopy. 31(7):6856–6861. Publisher: American Chemical Society.

[13] IBM100 - scanning tunneling microscope. Publisher: IBM Corporation.

[14] Naresh Marturi. Vision and visual servoing for nanomanipulation and nanocharacterization in scanning electron microscope.

[15] Hoffman Group. Scanning tunneling microscopy, intro. `https://hoffman.physics.harvard.edu/research/STMintro.php`. Accessed: 2023-04-27.

[16] Roland Wiesendanger. *Scanning Probe Microscopy and Spectroscopy: Methods and Applications*. Cambridge University Press.

[17] Claudia Brunner Jens Gerdelmann and Steve Pawlizak. Introduction: Scanning force microscopy (sfm). `https://home.uni-leipzig.de/pwm/web/?section=introduction&page=sfm`. Accessed: 2023-04-27.

[18] Ernst Meyer, Roland Bennewitz, and Hans J. Hug. *Scanning Probe Microscopy: The Lab on a Tip*. Graduate Texts in Physics. Springer International Publishing.

[19] T. R. Albrecht, P. Grütter, D. Horne, and D. Rugar. Frequency modulation detection using high- Q cantilevers for enhanced force microscope sensitivity. 69(2):668–673.

[20] Franz J. Giessibl. Advances in atomic force microscopy. 75(3):949–983.

[21] Franz J. Giessibl and Hartmut Bielefeldt. Physical interpretation of frequency-modulation atomic force microscopy. 61(15):9968–9971.

[22] Franz J. Giessibl. The qPlus sensor, a powerful core for the atomic force microscope. 90(1):011101. Publisher: American Institute of Physics.

[23] S. H. Ke, T. Uda, and K. Terakura. Quantity measured in frequency-shift-mode atomic-force microscopy: An analysis with a numerical model. 59(20):13267–13272.

[24] U. Dürig. Extracting interaction forces and complementary observables in dynamic probe microscopy. 76(9):1203. Publisher: American Institute of PhysicsAIP.

[25] John E. Sader and Suzanne P. Jarvis. Accurate formulas for interaction force and energy in frequency modulation force spectroscopy. 84(10):1801. Publisher: American Institute of PhysicsAIP.

[26] H. J. Hug, Th. Jung, and H.-J. Güntherodt. A high stability and low drift atomic force microscope. 63(8):3900–3904. Publisher: American Institute of Physics.

[27] T. Göddenhenrich, H. Lemke, U. Hartmann, and C. Heiden. Force microscope with capacitive displacement detection. 8(1):383–387.

[28] T. Göddenhenrich, H. Lemke, U. Hartmann, and C. Heiden. Magnetic force microscopy of domain wall stray fields on single-crystal iron whiskers. 56(25):2578–2580.

[29] Y. Martin, C. C. Williams, and H. K. Wickramasinghe. Atomic force microscope–force mapping and profiling on a sub 100-Å scale. 61(10):4723–4729.

[30] C. Schönenberger and S. F. Alvarado. A differential interferometer for force microscopy. 60(10):3131–3134.

[31] Gerhard Meyer and Nabil M. Amer. Novel optical approach to atomic force microscopy. 53(12):1045–1047.

[32] M. Tortonese, R. C. Barrett, and C. F. Quate. Atomic resolution with an atomic force microscope using piezoresistive detection. 62(8):834–836.

[33] O. N. Tufte and E. L. Stelzer. Piezoresistive properties of silicon diffused layers. 34(2):313–318.

[34] Franz J. Giessibl, Florian Pielmeier, Toyoaki Eguchi, Toshu An, and Yukio Hasegawa. Comparison of force sensors for atomic force microscopy based on quartz tuning forks and length-extensional resonators. 84(12):125409.

[35] Franz J. Giessibl. Atomic resolution of the silicon (111)-(7×7) surface by atomic force microscopy. 267(5194):68–71.

[36] Franz J. Giessibl. High-speed force sensor for force microscopy and profilometry utilizing a quartz tuning fork. 73(26):3956–3958.

[37] Julian Sterling. Non-contact atomic force microscopy.

[38] Yoshiaki Sugimoto, Masayuki Abe, Shinji Hirayama, Noriaki Oyabu, Óscar Custance, and Seizo Morita. Atom inlays performed at room temperature using atomic force microscopy. 4(2):156–159. Number: 2 Publisher: Nature Publishing Group.

[39] Uwe Kaiser, Alexander Schwarz, and Roland Wiesendanger. Magnetic exchange force microscopy with atomic resolution. 446(7135):522–525. Number: 7135 Publisher: Nature Publishing Group.

[40] Yoshiaki Sugimoto, Pablo Pou, Masayuki Abe, Pavel Jelinek, Rubén Pérez, Seizo Morita, and Óscar Custance. Chemical identification of individual surface atoms by atomic force microscopy. 446(7131):64–67. Number: 7131 Publisher: Nature Publishing Group.

[41] Markus Ternes, Christopher P. Lutz, Cyrus F. Hirjibehedin, Franz J. Giessibl, and Andreas J. Heinrich. The force needed to movean atom on a surface. *Science*, 319(5866):1066–1069, 2008.

[42] Franz J. Giessibl, Hartmut Bielefeldt, Stefan Hembacher, and Jochen Mannhart. Calculation of the optimal imaging parameters for frequency modulation atomic force microscopy. 140(3):352–357.

[43] Leo Gross, Fabian Mohn, Nikolaj Moll, Peter Liljeroth, and Gerhard Meyer. The chemical structure of a molecule resolved by atomic force microscopy. 325(5944):1110–1114.

[44] Prokop Hapala, Georgy Kichin, Christian Wagner, F. Stefan Tautz, Ruslan Temirov, and Pavel Jelínek. Mechanism of high-resolution STM/AFM imaging with functionalized tips. 90(8):085421.

[45] Jaime Carracedo-Cosme, Carlos Romero-Muñiz, Pablo Pou, and Rubén Pérez. QUAM-AFM: A free database for molecular identification by atomic force microscopy. 62(5):1214–1223.

[46] Michael Ellner, Pablo Pou, and Rubén Pérez. Molecular identification, bond order discrimination, and apparent intermolecular features in atomic force microscopy studied with a charge density based method. 13(1):786–795.

[47] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov. Electric field effect in atomically thin carbon films. 306(5696):666–669. Publisher: American Association for the Advancement of Science.

[48] Gowoon Cheon, Karel-Alexander N. Duerloo, Austin D. Sendek, Chase Porter, Yuan Chen, and Evan J. Reed. Data mining for new two- and one-dimensional weakly bonded solids and lattice-commensurate heterostructures. 17(3):1915–1923.

[49] WILLIAM R. Frensley. Chapter 1 - heterostructure and quantum well physics. In Norman G. Einspruch and William R. Frensley, editors, *VLSI Electronics Microstructure Science*, volume 24 of *Heterostructures and Quantum Devices*, pages 1–24. Elsevier.

[50] Yuanyuan Zhu, Quentin M. Ramasse, Michael Brorson, Poul G. Moses, Lars P. Hansen, Christian F. Kisielowski, and Stig Helveg. Visualizing the stoichiometry of industrial-style co-mo-s catalysts with single-atom sensitivity. 53(40):10723–10727. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201405690.

[51] Yuanyuan Zhu, Quentin M. Ramasse, Michael Brorson, Poul G. Moses, Lars P. Hansen, Henrik Topsøe, Christian F. Kisielowski, and Stig Helveg. Location of co and ni promoter atoms in multi-layer MoS2 nanocrystals for hydrotreating catalysis. 261:75–81.

[52] Hai Li, Jumiati Wu, Xiao Huang, Gang Lu, Jian Yang, Xin Lu, Qihua Xiong, and Hua Zhang. Rapid and reliable thickness identification of two-dimensional nanosheets using optical microscopy. 7(11):10344–10353.

[53] Caitlyn Meditz. Environmental factors on 2d material properties: influence of underlying au(111) vs amorphous gold on the local density of states in graphene.

[54] Sujay B. Desai, Surabhi R. Madhvapathy, Matin Amani, Daisuke Kiriya, Mark Hettick, Mahmut Tosun, Yuzhi Zhou, Madan Dubey, Joel W. Ager, Daryl Chrzan, and Ali Javey. Gold-mediated exfoliation of ultralarge optoelectronically-perfect monolayers. 28(21):4053–4058.

[55] Fang Liu, Wenjing Wu, Yusong Bai, Sang Hoon Chae, Qiuyang Li, Jue Wang, James Hone, and X.-Y. Zhu. Disassembling 2d van der waals crystals into macroscopic monolayers and reassembling into artificial lattices. 367(6480):903–906.

[56] Yuan Huang, Yu-Hao Pan, Rong Yang, Li-Hong Bao, Lei Meng, Hai-Lan Luo, Yong-Qing Cai, Guo-Dong Liu, Wen-Juan Zhao, Zhang Zhou, Liang-Mei Wu, Zhi-Li Zhu, Ming Huang, Li-Wei Liu, Lei Liu, Peng Cheng, Ke-Hui Wu, Shi-Bing Tian, Chang-Zhi Gu, You-Guo Shi, Yan-Feng Guo, Zhi Gang Cheng, Jiang-Ping Hu, Lin Zhao, Guan-Hua Yang, Eli Sutter, Peter Sutter, Ye-Liang Wang, Wei Ji, Xing-Jiang Zhou, and Hong-Jun Gao. Universal mechanical exfoliation of large-area 2d crystals. 11(1):2453.

[57] Amanda M. Larson, Jeremiah van Baren, Jeremy Kintigh, Jun Wang, Jian-Ming Tang, Percy Zahl, Glen P. Miller, and Karsten Pohl. Lateral standing of the pentacene derivative 5,6,7-trithiapentacene-13-one on gold: A combined STM, DFT, and NC-AFM study. 122(22):11938–11944.

[58] Fang Liu, Xiaoyang Zhu, Jue Wang, Kameron Hansen, and Mark Ziffer. DIRECT DETERMINATION OF BAND GAP RENORMALIZATION IN PHOTO-EXCITED MONOLAYER MOS2. In *Proceedings of the 74th International Symposium on Molecular Spectroscopy*, pages 1–1. University of Illinois at Urbana-Champaign.

[59] Henrik Topsøe and Bjerne S. Clausen. Active sites and support effects in hydrodesulfurization catalysts. 25(1):273–293.

[60] F. Besenbacher and J. V. Lauritsen. Applications of high-resolution scanning probe microscopy in hydroprocessing catalysis studies. 403:4–15.

[61] Leanna C. Giancarlo, Hongbin Fang, Seth M. Rubin, Alexa Avila Bront, and George W. Flynn. Influence of the substrate on order and image contrast for physisorbed, self-assembled molecular monolayers: STM studies of functionalized hydrocarbons on graphite and MoS $_2$. 102(50):10255–10263.

[62] J. G. Kushmerick and P. S. Weiss. Mobile promoters on anisotropic catalysts: Nickel on MoS $_2$. 102(50):10094–10097.

[63] Thomas F. Jaramillo, Kristina P. Jørgensen, Jacob Bonde, Jane H. Nielsen, Sebastian Horch, and Ib Chorkendorff. Identification of active edge sites for electrochemical h $_2$ evolution from MoS $_2$ nanocatalysts. 317(5834):100–102.

[64] Jinlong Gong. Structure and surface chemistry of gold-based model catalysts. 112(5):2987–3054. Publisher: American Chemical Society.

[65] Yafei Li, Zhen Zhou, Shengbai Zhang, and Zhongfang Chen. MoS2 nanoribbons: High stability and unusual electronic and magnetic properties. 130(49):16739–16744. Publisher: American Chemical Society.

[66] Yongwen Tan, Pan Liu, Luyang Chen, Weitao Cong, Yoshikazu Ito, Jiuhui Han, Xianwei Guo, Zheng Tang, Takeshi Fujita, Akihiko Hirata, and Mingwei W. Chen. Monolayer MoS2 films supported by 3d nanoporous metals for high-efficiency electrocatalytic hydrogen production. 26(47):8023–8028. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201403808.

[67] Anabela Nogueira, Raja Znaiguia, Denis Uzio, Pavel Afanasiev, and Gilles Berhault. Curved nanostructures of unsupported and al2o3-supported MoS2 catalysts: Synthesis and HDS catalytic properties. 429-430:92–105.

[68] Ji Hoon Lee, Woo Soon Jang, Sun Woong Han, and Hong Koo Baik. Efficient hydrogen evolution by mechanically strained MoS $_2$ nanosheets. 30(32):9866–9873.

[69] Hong Li, Charlie Tsai, Ai Leen Koh, Lili Cai, Alex W. Contryman, Alex H. Fragapane, Jiheng Zhao, Hyun Soo Han, Hari C. Manoharan, Frank Abild-Pedersen, Jens K. Nørskov, and Xiaolin Zheng. Activating and optimizing MoS2 basal planes for hydrogen evolution through the formation of strained sulphur vacancies. 15(1):48–53. Number: 1 Publisher: Nature Publishing Group.

[70] Guoqing Li, Zehua Chen, Yifan Li, Du Zhang, Weitao Yang, Yuanyue Liu, and Linyou Cao. Engineering substrate interaction to improve hydrogen evolution catalysis of monolayer MoS $_2$ films beyond pt. 14(2):1707–1714.

[71] Guoqing Li, Du Zhang, Qiao Qiao, Yifei Yu, David Peterson, Abdullah Zafar, Raj Kumar, Stefano Curtarolo, Frank Hunte, Steve Shannon, Yimei Zhu, Weitao Yang, and Linyou Cao. All the catalytic active sites of MoS2 for hydrogen evolution. 138(51):16632–16638. Publisher: American Chemical Society.

[72] James H. Gary, Glenn E. Handwerk, and Mark J. Kaiser. *Petroleum Refining: Technology and Economics, Fifth Edition.* CRC Press. Google-Books-ID: ocbLBQAAQBAJ.

[73] Myriam Perez De la Rosa, Samuel Texier, Gilles Berhault, Alejandra Camacho, Miguel José Yácaman, Apurva Mehta, Sergio Fuentes, Jorge Ascension Montoya, Florentino Murrieta, and Russell R. Chianelli. Structural studies of catalytically stabilized model and industrial-supported hydrodesulfurization catalysts. 225(2):288–299.

[74] Gilles Berhault, Myriam Perez De la Rosa, Apurva Mehta, Miguel José Yácaman, and Russell R. Chianelli. The single-layered morphology of supported MoS2-based catalysts—the role of the cobalt promoter and its effects in the hydrodesulfurization of dibenzothiophene. 345(1):80–88.

[75] J. H. Nielsen, L. Bech, K. Nielsen, Y. Tison, K. P. Jørgensen, J. L. Bonde, S. Horch, T. F. Jaramillo, and I. Chorkendorff. Combined spectroscopy and microscopy of supported MoS2 nanoparticles. 603(9):1182–1189.

[76] Signe S. Grønborg, Norberto Salazar, Albert Bruix, Jonathan Rodríguez-Fernández, Sean D. Thomsen, Bjørk Hammer, and Jeppe V. Lauritsen. Visualizing hydrogen-induced reshaping and edge activation in MoS2 and co-promoted MoS2 catalyst clusters. 9(1):2211.

[77] Rik V. Mom, Jaap N. Louwen, Joost W. M. Frenken, and Irene M. N. Groot. In situ observations of an active MoS2 model hydrodesulfurization catalyst. 10(1):2546.

[78] M. V. Bollinger, K. W. Jacobsen, and J. K. Nørskov. Atomic and electronic structure of MoS 2 nanoparticles. 67(8):085410.

[79] Yong Xu, Yin Li, Xi Chen, Chunfang Zhang, Ru Zhang, and Pengfei Lu. First-principle study of hydrogenation on monolayer MoS2. 6(7):075001.

[80] Cédric Bara, Elodie Devers, Mathieu Digne, Anne-Félicie Lamic-Humblot, Gerhard D. Pirngruber, and Xavier Carrier. Surface science approaches for the preparation of alumina-supported hydrotreating catalysts. 7(21):3422–3440. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cctc.201500436.

[81] Hongtao Yuan, Haotian Wang, and Yi Cui. Two-dimensional layered chalcogenides: From rational synthesis to property control via orbital occupation and electron filling. 48(1):81–90. Publisher: American Chemical Society.

[82] Zegao Wang, Qiang Li, Haoxiang Xu, Christian Dahl-Petersen, Qian Yang, Daojian Cheng, Dapeng Cao, Flemming Besenbacher, Jeppe V. Lauritsen, Stig Helveg, and Mingdong Dong. Controllable etching of MoS2 basal planes for enhanced hydrogen evolution through the formation of active edge sites. 49:634–643.

[83] Bruno Schuler, Yunlong Zhang, Fang Liu, Andrew E. Pomerantz, A. Ballard Andrews, Leo Gross, Vincent Pauchard, Sanjoy Banerjee, and Oliver C. Mullins. Overview of asphaltene nanostructures and thermodynamic applications. 34(12):15082–15105. Publisher: American Chemical Society.

[84] Shadi Fatayer, Nimesh B. Poddar, Sabela Quiroga, Fabian Schulz, Bruno Schuler, Subramanian V. Kalpathy, Gerhard Meyer, Dolores Pérez, Enrique Guitián, Diego Peña, Mary J. Wornat, and Leo Gross. Atomic force microscopy identifying fuel pyrolysis products and directing the synthesis of analytical standards. 140(26):8156–8161.

[85] Shadi Fatayer, Alysha I. Coppola, Fabian Schulz, Brett D. Walker, Taylor A. Broek, Gerhard Meyer, Ellen R. M. Druffel, Matthew McCarthy, and Leo Gross. Direct visualization of individual aromatic compound structures in low molecular weight marine dissolved organic carbon. 45(11):5590–5598. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2018GL077457.

[86] Eric I. Altman, Mehmet Z. Baykara, and Udo D. Schwarz. Noncontact atomic force microscopy: An emerging tool for fundamental catalysis research. 48(9):2640–2648. Publisher: American Chemical Society.

[87] S. Torbrügge. Atomic resoluiton nc-afm of si(111)-7x7.

[88] S. Torbrügge. Atomic resolution nc-afm imaging on au(111) at room temperature.

[89] A. Krull, P. Hirsch, C. Rother, A. Schiffrin, and C. Krull. Artificial-intelligence-driven scanning probe microscopy. 3(1):54.

[90] Benjamin Alldritt, Prokop Hapala, Niko Oinonen, Fedor Urtev, Ondrej Krejci, Filippo Federici Canova, Juho Kannala, Fabian Schulz, Peter Liljeroth, and Adam S. Foster. Automated structure discovery in atomic force microscopy. 6(9):eaay6913.

[91] Shenkai Wang, Junmian Zhu, Raymond Blackwell, and Felix R. Fischer. Automated tip conditioning for scanning tunneling spectroscopy. *J. Phys. Chem. A*, 125(6):1384–1390, 2021.

[92] Jaime Carracedo-Cosme, Carlos Romero-Muñiz, Pablo Pou, and Rubén Pérez. Molecular identification from AFM images using the IUPAC nomenclature and attribute multimodal recurrent neural networks.

[93] Benjamin Alldritt, Fedor Urtev, Niko Oinonen, Markus Aapro, Juho Kannala, Peter Liljeroth, and Adam S. Foster. Automated tip functionalization via machine learning in scanning probe microscopy. 273:108258.

[94] Boyuan Huang, Zhenghao Li, and Jiangyu Li. An artificial intelligence atomic force microscope enabled by machine learning. 10(45):21320–21326.

[95] Yicheng Wang, Konstantinos G. Papanikolaou, Ryan T. Hannagan, Dipna A. Patel, Tedros A. Balema, Laura A. Cramer, Paul L. Kress, Michail Stamatakis, and E. Charles H. Sykes. Surface facet dependence of competing alloying mechanisms. 153(24):244702.

[96] *Deep Learning with Python, Second Edition.*

[97] *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems.*

[98] Alana Gudinas, Jason Moscatello, and Shawna M. Hollen. Defect identification and statistics toolbox: automated defect analysis for scanning probe microscopy images. 33(4):045901. Publisher: IOP Publishing.

[99] Jack Hellerstedt, Aleš Cahlík, Martin Švec, Oleksandr Stetsovych, and Tyler Hennen. Counting molecules: Python based scheme for automated enumeration and categorization of molecules in scanning tunneling microscopy images. 12:100301.

[100] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Github repository: Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[101] Dikshant Sagar. Multiple myeloma cancer cell instance segmentation.

[102] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[103] Bruno Schuler, Gerhard Meyer, Diego Peña, Oliver C. Mullins, and Leo Gross. Unraveling the molecular structures of asphaltenes by atomic force microscopy. 137(31):9870–9876.

[104] Yunlong Zhang, Fabian Schulz, B. McKay Rytting, Clifford C. Walters, Katharina Kaiser, Jordan N. Metz, Michael R. Harper, Shamel S. Merchant, Anthony S. Mennito, Kuangnan Qian, J. Douglas Kushnerick, Peter K. Kilpatrick, and Leo Gross. Elucidating the geometric substitution of petroporphyrins by spectroscopic analysis and atomic force microscopy molecular imaging. 33(7):6088–6097. Publisher: American Chemical Society.

[105] Yunlong Zhang, Bruno Schuler, Shadi Fatayer, Leo Gross, Michael R. Harper, and J. Douglas Kushnerick. Understanding the effects of sample preparation on the chemical structures of petroleum imaged with noncontact atomic force microscopy. 57(46):15935–15941. Publisher: American Chemical Society.

[106] Pengcheng Chen, Jordan N. Metz, Anthony S. Mennito, Shamel Merchant, Stuart E. Smith, Michael Siskin, Steven P. Rucker, David C. Dankworth, J. Douglas Kushnerick, Nan Yao, and Yunlong Zhang. Petroleum pitch: Exploring a 50-year structure puzzle with real-space molecular imaging. 161:456–465.

[107] Yunlong Zhang. Similarities in diverse polycyclic aromatic hydrocarbons of asphaltenes and heavy oils revealed by noncontact atomic force microscopy: Aromaticity, bonding, and implications for reactivity. In *Chemistry Solutions to Challenges in the Petroleum Industry*, volume 1320 of *ACS Symposium Series*, pages 39–65. American Chemical Society. Section: 3.

[108] Yunlong Zhang. Applications of noncontact atomic force microscopy in petroleum characterization: Opportunities and challenges. 35(18):14422–14444.

[109] Leo Gross, Fabian Mohn, Nikolaj Moll, Gerhard Meyer, Rainer Ebel, Wael M. Abdel-Mageed, and Marcel Jaspars. Organic structure determination using atomic-resolution scanning probe microscopy. 2(10):821–825.

[110] Mario Commodo, Katharina Kaiser, Gianluigi De Falco, Patrizia Minutolo, Fabian Schulz, Andrea D'Anna, and Leo Gross. On the early stages of soot formation: Molecular structure elucidation by high-resolution atomic force microscopy. 205:154–164.

[111] Fabian Schulz, Mario Commodo, Katharina Kaiser, Gianluigi De Falco, Patrizia Minutolo, Gerhard Meyer, Andrea D'Anna, and Leo Gross. Insights into incipient soot formation by atomic force microscopy. 37(1):885–892.

[112] Katharina Kaiser, Fabian Schulz, Julien F. Maillard, Felix Hermann, Iago Pozo, Diego Peña, H. James Cleaves II, Aaron S. Burton, Gregoire Danger, Carlos Afonso, Scott Sandford, and Leo Gross. Visualization and identification of single meteoritic organic molecules by atomic force microscopy. 57(3):644–656. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/maps.13784.

[113] Fabian Schulz, Julien Maillard, Katharina Kaiser, Isabelle Schmitz-Afonso, Thomas Gautier, Carlos Afonso, Nathalie Carrasco, and Leo Gross. Imaging titan's organic haze at atomic scale. 908(1):L13. Publisher: The American Astronomical Society.

[114] Mohammad Rashidi, Jeremiah Croshaw, Kieran Mastel, Marcus Tamura, Hedieh Hosseinzadeh, and Robert A Wolkow. Deep learning-guided surface characterization for autonomous hydrogen lithography. 1(2):025001.

[115] Percy Zahl, Markus Bierkandt, Stefan Schröder, and Andreas Klust. The flexible and modern open source scanning probe microscopy software package <span style="font-variant:small-caps;">GXSM</span>. 74(3):1222–1227.

[116] P. Zahl, Thorsten Wagner, Rolf Möller, and Andreas Klust. Open source scanning probe microscopy control software package gxsm. *Journal of Vacuum Science and Technology B*, 28, 05 2010.

[117] Md Mostafa Kamal Sarker, Yasmine Makhlouf, Stephanie G. Craig, Matthew P. Humphries, Maurice Loughrey, Jacqueline A. James, Manuel Salto-Tellez, Paul O'Reilly, and Perry Maxwell. A means of assessing deep learning-based detection of ICOS protein expression in colon cancer. 13(15):3825.

[118] Zhun Fan, Yuming Wu, Jiewei Lu, and Wenji Li. Automatic pavement crack detection based on structured prediction with the convolutional neural network.

[119] Carolina Gonçalves, Pedro Santana, Tomás Brandão, and Magno Guedes. Automatic detection of acacia longifolia invasive species based on UAV-acquired aerial imagery. 9(2):276–287.

[120] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.

[121] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context.

[122] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection.

[123] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-CNN: Towards real-time object detection with region proposal networks.

[124] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. ISSN: 2380-7504.

[125] Piotr Skalski. Github repository: Make Sense. `https://github.com/SkalskiP/make-sense/`, 2019.

[126] Niko Pavliček, Anish Mistry, Zsolt Majzik, Nikolaj Moll, Gerhard Meyer, David J. Fox, and Leo Gross. Synthesis and characterization of triangulene. 12(4):308–311.

[127] Percy Zahl and Yunlong Zhang. Guide for atomic force microscopy image analysis to discriminate heteroatoms in aromatic molecules. 33(6):4775–4780.

[128] Niko Oinonen, Lauri Kurki, Alexander Ilin, and Adam S. Foster. Molecule graph reconstruction from atomic force microscope images with machine learning.

[129] Jaime Carracedo-Cosme, Carlos Romero-Muñiz, and Rubén Pérez. A deep learning approach for molecular classification based on AFM images. 11(7):1658.

APPENDIX A

Codes for Auto-HR-AFM: Training Detectron2 and Automation Script

This section contains Python code that was used to train our instance segmentation model and the SPM automation script described in Chapter 4. The first code was originally written in a Jupyter notebook and is used to train Detectron2 on a custom dataset. The second code is the code for Auto-HR-AFM written in Python. Both these are available on Github [**?**]:

https://github.com/Sarias13/Auto-HR-AFM

A.1 Training Detectron2 on your own dataset

This code was written on a Jupyter notebook and was used to train the instance segmentation model used in Chapter 4. The code also includes sections that use OpenCV 2 to help visualize how the instance segmentation model is performing through the steps. There are details in the comments for each block that is separated by a string of # signs. I recommend using a Jupyter notebook to test this code out, having each block as an input. Inputting one block at a time into a Jupyter notebook can help with the debugging process and makes the code easier to follow.

A nice tutorial for training instance segmentation models and learning about how to use Detectron2 on your own datasets can be found at in the Detectron2 documentation site [100]: https://detectron2.readthedocs.io/

```
################################################################

# Import Detectron2 should be installed in your local computer. https://
    detectron2.readthedocs.io/en/latest/tutorials/install.html
```

```python
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()

# import some common libraries
import numpy as np
import cv2
import matplotlib.pyplot as plt

# import some common detectron2 utilities
from detectron2 import model_zoo
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg
from detectron2.utils.visualizer import Visualizer
from detectron2.data import MetadataCatalog, DatasetCatalog

###################################################################

#Register the dataset and metadata for the model.
#the dataset are the files used for the training, testing, and validation used.
#Inputs for the register_coco_instances("Given_Dataset_Name", {}, "Label_info_json_file", "Path_to_files_to_be_included"

#Given_Dataset_Name: Name that you want to register this group of files as.
#{}: left blank.
#Label_info_json_file: Path where the json file is with your labeled data and metadata.
#Path_to_files_to_be_included: Path to the files you want to include in these instances.

from detectron2.data.datasets import register_coco_instances
```

```python
register_coco_instances("3class_train", {}, "./ncfiles3/labels_clasestres_2022-07-26-01-21-07.json", "./ncfiles3/files")
#register_coco_instances("3class_val", {}, "./labels_clasestres_2022-07-26-01-21-07.json", "./3class/valid")
#register_coco_instances("3class_test", {}, "./labels_clasestres_2022-07-26-01-21-07.json", "./3class/test")

###################################################################

#Check To see that the files/labels are uploaded correctlly on 3 random files.
import random
from detectron2.data import DatasetCatalog, MetadataCatalog
from detectron2.utils.visualizer import ColorMode

dataset_dicts = DatasetCatalog.get("3class_train")
microcontroller_metadata = MetadataCatalog.get("3class_train")

for d in random.sample(dataset_dicts, 3):
    img = cv2.imread(d["file_name"])
    v = Visualizer(img[:, :, ::-1], metadata=microcontroller_metadata, scale=1.3,instance_mode=ColorMode.SEGMENTATION)
    v = v.draw_dataset_dict(d)
    plt.figure(figsize = (14, 10))
    plt.imshow(cv2.cvtColor(v.get_image()[:, :, ::-1], cv2.COLOR_BGR2RGB))
    plt.show()

###################################################################

#Check the metadata to see what is labeled.
microcontroller_metadata

###################################################################
```

```

#Get Config Files and Trainer.

from detectron2.engine import DefaultTrainer
from detectron2.config import get_cfg
import os

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml")) #Pick your favorite from model zoo
cfg.DATASETS.TRAIN = ("3class_train",) # comes from the file registered above
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml")#Pick your favorite from model zoo
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 1000
cfg.SOLVER.STEPS = []        # do not decay learning rate
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 4 #Change depending on how many classes you have. Some lablers add an extra class, so double check

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)

#Uncomment below to train model. Leave commented to just load the config file

#trainer = DefaultTrainer(cfg)
#trainer.resume_or_load(resume=False)
#trainer.train()

```

```python
###################################################################

# Load weights from model and decide on Threshold. Run predictor
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5
cfg.DATASETS.TEST = ("3class_train", )
predictor = DefaultPredictor(cfg)

###################################################################

#Run model on one specific File
#Change Filename.png for the file you want to test model on.

from detectron2.utils.visualizer import ColorMode
im = cv2.imread("Filename.png")
outputs = predictor(im)
v = Visualizer(im[:, :, ::-1],
                metadata=microcontroller_metadata,
                scale=1,
                instance_mode=ColorMode.IMAGE_BW   # remove the colors of unsegmented pixels
)
v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
plt.figure(figsize = (20, 10))
plt.imshow(cv2.cvtColor(v.get_image()[:, :, ::-1], cv2.COLOR_BGR2RGB))
plt.show()

###################################################################

#Run on multiple files in a specifi directory.

from detectron2.utils.visualizer import ColorMode

```

```

microcontroller_metadata = MetadataCatalog.get("3class_train")
for filename in os.listdir("./"):
    if (filename.endswith(".jpg")):
        #text = np.load(filename)
        im = cv2.imread(filename)
        #print(im)
        outputs = predictor(im)
        v = Visualizer(im[:, :, ::-1],
                        metadata=microcontroller_metadata,
                        scale=1.2,
                        instance_mode=ColorMode.IMAGE_BW  # remove the colors of unsegmented pixels
        )
        v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
        plt.figure(figsize = (14, 10))
        plt.imshow(cv2.cvtColor(v.get_image()[:, :, ::-1], cv2.COLOR_BGR2RGB))
        plt.show()
        #cv2.imwrite(’result_’+filename+’’,v.get_image()[:, :, ::-1])  # uncomment to Save file if you want

###################################################################

#Run model on 3 random files in your dataset if registered. Typically used on validation or test. With a bigger dataset.
from detectron2.utils.visualizer import ColorMode
dataset_dicts = DatasetCatalog.get("3class_train") #Used on train here to double check.
#print(dataset_dicts)
#dataset_dicts = "Ringers"
for d in random.sample(dataset_dicts, 3):
    im = cv2.imread(d["file_name"])
    outputs = predictor(im)
    v = Visualizer(im[:, :, ::-1],
                   metadata=microcontroller_metadata,
                   scale=1,
                   instance_mode=ColorMode.IMAGE_BW   # remove the colors of unsegmented pixels
    )
    v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    plt.figure(figsize = (14, 10))
    plt.imshow(cv2.cvtColor(v.get_image()[:, :, ::-1], cv2.COLOR_BGR2RGB))
    plt.show()

#outputs all the information given by the model on a given file or the last file it ran on.
print(outputs["instances"].pred_classes)
print(outputs["instances"].pred_boxes)
print(outputs["instances"])

###################################################################

#Visualize the ouputs as a mask. Different than the one detectron2 uses.

mask_array = outputs['instances'].pred_masks.to("cpu").numpy()
num_instances = mask_array.shape[0]
scores = outputs['instances'].scores.to("cpu").numpy()
labels = outputs['instances'].pred_classes .to("cpu").numpy()
bbox   = outputs['instances'].pred_boxes.to("cpu").tensor.numpy()

mask_array = np.moveaxis(mask_array, 0, -1)

mask_array_instance = []
#img = np.zeros_like(im) #black
h = im.shape[0]
```

```
w = im.shape[1]
img_mask = np.zeros([h, w, 3], np.uint8)
for i in range(num_instances):
    if labels[i]==0:
        color = (250, 43, 138)
    elif labels[i]==1:
        color = (0, 0, 255)
    else:
        color = (0,255,0)
    img = np.zeros_like(im)
    mask_array_instance.append(mask_array[:, :, i:(i+1)])
    img = np.where(mask_array_instance[i] == True, 255, img)
    array_img = np.asarray(img)
    img_mask[np.where((array_img==[255,255,255]).all(axis=2))]=color

img_mask = np.asarray(img_mask)
output = cv2.addWeighted(im, 0.7, img_mask, 0.3, 0)

#################################################################

#See mask produced.
plt.figure(figsize = (14, 10))
plt.imshow(cv2.cvtColor(img_mask, cv2.COLOR_BGR2RGB))
plt.show()
#cv2.imwrite('result.jpg',img_mask)

#################################################################

#Remove black background from files.

import cv2
file_name = "result.jpg"

```

```
src = cv2.imread(file_name, 1)
tmp = cv2.cvtColor(src, cv2.COLOR_RGB2GRAY)
_,alpha = cv2.threshold(tmp,25,255,cv2.THRESH_BINARY)
b, g, r = cv2.split(src)
rgba = [b,g,r, alpha]
dst = cv2.merge(rgba,4)
cv2.imwrite("test.png", dst)

###################################################################

#Uncomment this section if you would like to use Tensorboard to check the progress of you model.

#Load Tensorboard if you want to see how model is working
#%load_ext tensorboard
#%tensorboard --logdir output
```

Listing A.1: Jupyter Notebook: Training Detectron2 on your own dataset and tools to help visualize results.

A.2 Auto-HR-AFM Full Script

This is a modular Python code that controls the GXSM SPM software. The code is split into commented parts and the functionality of the code is explained in Chapter 4. The code includes the instance segmentation model seed in Section A.1.

The start of the code includes triggers to turn sections of the code on or off. The instance segmentation model has to be loaded from an existing file on your computer. If using a different SPM software controller, the sections to send commands to the SPM have to be modified before using the script.

```
### MODES
### Trigger modes to check parts of the script. Set to True to use that part of the script.
### do_auto_locate: Includes a CV package to find molecules on a surface.
### do_stm: If you want to collect STM images
### do_afm: If you want to collect HR-AFM images
### do_AI_afm: If you want to run the AI automation script to collect HR-AFM images.

TEST_AI=False # Test Mode

do_auto_locate = False
do_stm = True
do_afm = True
do_AI_afm = True #False

###############################################
## INSTRUMENT
## Initial instrument parameters for GXSM

ZAV=8.0 # 8 Ang/Volt in V

```

```
### SETUP ENVIRONMENT CONFIGURATION
map_ch=6   # MAP IMAGE CHANNEL  (0....N)
map_diffs=gxsm.get_differentials(map_ch)
afm_ch=2  # HR_AFM IMAGE (0...N)

### DEFAULTS
# Up to 8 ScriptControls (sc)
# Level, I in pA
sc = dict(STM_Range=45, AFM_Range=45,  Molecule=1,  I_ref=20, CZ_Level=100.0,  Z_down=1.3, Z_start=0.0, Tip_Z=0.0,  Tip_Z_Ref=0.0, AutoAFMSpeed=1)

STM_ref_bias = 0.6 # 0.1
STM_scan_current = 0.0025 # 0.01
STM_points = 160
STM_dx=sc['STM_Range']/STM_points

AFM_points = 330
AFM_dx=sc['AFM_Range']/AFM_points
#sc['STM_Range']

#CZAFM_Iref=0.04
#CZAFM_Zoff=-0.1
#CZ_FUZZY_level =0.1




#################  Imports ###########################
import sys
sys.modules[__name__].__dict__.clear()

from string import *
```

```
import os
import logging
from string import *
import os
import datetime


import time
import random as rng
import cv2
#import netCDF4 as nc
import struct
import array
import math
import numpy as np
from skimage.color import rgb2gray
from skimage.color import gray2rgb
import itertools

import torch
x = torch.rand(5, 3)
print(x)
print(’CUDA: ’,torch.cuda.is_available())
print(’***’)

# Import Detectron2 should be installed in your local computer. https://detectron2.readthedocs.io/en/latest/tutorials/install.html
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()
# Import some common detectron2 utilities
from detectron2 import model_zoo
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg

#Register the dataset and metadata for the model.
#the dataset are the files used for the training, testing, and validation used. No need for all 3 to run.

from detectron2.data.datasets import register_coco_instances
from detectron2.data import DatasetCatalog, MetadataCatalog

## Setup logfile
full_path_name = gxsm.chfname(0).split()[0]
##
print(’Starting here, Logfile setup.’)
print(full_path_name)
ncfname = os.path.basename(full_path_name)
folder = os.path.dirname(full_path_name)
name, ext = os.path.splitext(ncfname)
logfile_name = folder+’/’+name+’-AIrun-initial.log’
print(’Logging to: ’, logfile_name)

logging.basicConfig(filename=logfile_name, encoding=’utf-8’, level=logging.DEBUG, format=’%(asctime)s %(message)s’)
logging.info(’Auto-AFM-AI logfile start. [{}]’.format(logfile_name))

#logging.debug(’This message should go to the log file’)
#logging.info(’So should this’)
#logging.warning(’And this, too’)
#logging.error(’Error’)




DatasetCatalog.clear()
```

```

#Register the data. Where is it located locally?
# download, and extract, adjust path below accordingly:  https://bnlbox.sdcc.bnl.gov/index.php/s/9xoFrTsPWmcB9Gp
AI_base_dir = '/home/percy/AI-data'
register_coco_instances("3class_train", {}, AI_base_dir+'/labels_clasestres_2022-07-26-01-21-07.json', AI_base_dir+'/files')

dataset_dicts = DatasetCatalog.get("3class_train")
microcontroller_metadata = MetadataCatalog.get("3class_train")

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml"))
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 4
# Load weights from model and decide on Threshold. Run predictor
cfg.MODEL.WEIGHTS = AI_base_dir+'/AI_model.pth'
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5
predictor = DefaultPredictor(cfg)




################################### Function Definitions ####################################################################################

# Setup SCs
def SetupSC():
  for i, e in enumerate(sc.items()):
    id='py-sc{:02d}'.format(i+1)
    print (id, e[0], e[1])
    gxsm.set_sc_label(id, e[0])
    gxsm.set(id, '{:.4f}'.format(e[1]))
```

```

SetupSC()

# Read / Update dict
def GetSC():
  for i, e in enumerate(sc.items()):
    id=’py-sc{:02d}’.format(i+1)
    #print (id, ’ => ’, e[0], e[1])
    sc[e[0]] = float(gxsm.get(id))
    #print (id, ’<=’, sc[e[0]])

# Update SCs
def SetSC():
  for i, e in enumerate(sc.items()):
    id=’py-sc{:02d}’.format(i+1)
    gxsm.set(id, ’{:.4f}’.format(e[1]))


#GetSC()
#SetSC()

gxsm.set(’script-control’,’1’)

def export_drawing(ch=0, postfix=’-dwg’):
  full_original_name = gxsm.chfname(ch).split()[0]
  print(full_original_name)
  folder = os.path.dirname(full_original_name)
  ncfname = os.path.basename(full_original_name)
  name, ext = os.path.splitext(ncfname)
  dest_name = folder+’/’+name+postfix
  print(’Exporting: ’, dest_name)
  gxsm.chmodea(ch)
  gxsm.autodisplay()
  time.sleep(1)
  gxsm.save_drawing(ch, 0,0, dest_name+’.png’)
  gxsm.save_drawing(ch, 0,0, dest_name+’.pdf’)

def export_png(ch=0, postfix=’autoexport’):
  full_original_name = gxsm.chfname(ch).split()[0]
  print(full_original_name)
  folder = os.path.dirname(full_original_name)
  ncfname = os.path.basename(full_original_name)
  name, ext = os.path.splitext(ncfname)
  dest_name = folder+’/’+name+postfix
  print(’Exporting: ’, dest_name)
  gxsm.chmodea(ch)
  gxsm.autodisplay()
  time.sleep(1)
  gxsm.save_drawing(ch, 0,0, dest_name+’.png’)


def init_force_map_ref_xy(bias=0.02, level=0.111, ref_i=0.05, zoff=0.0, xy_list=[[0,0]]):
  print("Measuring Z at ref")
  # set ref condition
  gxsm.set ("dsp-fbs-bias","0.1") # set Bias to 0.1V
  gxsm.set ("dsp-fbs-mx0-current-set","{:8.4f}".format( ref_i))   # Set Current Setpoint to reference value (nA)
  gxsm.set ("dsp-fbs-mx0-current-level","0.00")

  time.sleep(1) # NOW SHOULD ME ON TOP OF MOLECULE
  gxsm.set ("dsp-fbs-bias","0.02") # set Bias to 20mV
  gxsm.set ("dsp-fbs-mx0-current-set","0.05") # Set Current Setpoint to 50 pA
  # read Z ref and set
  svec=gxsm.rtquery ("z")
```

```
print('RTQ z',  svec[0]*ZAV)
pts=1
z=svec[0]*ZAV
zmin=zmax=z
for r in xy_list:
  gxsm.moveto_scan_xy(r[0], r[1])
  time.sleep(0.1)
  for i in range(0,5):
    svec=gxsm.rtquery ("z")
    time.sleep(0.02)
    zxy=svec[0]*ZAV
    if zmin > zxy:
      zmin=zxy
    if zmax < zxy:
      zmax=zxy
    #print(r, " => Z: ", zxy, " Min/Max: ", zmin, zmax)
    z=z+zxy
    pts=pts+1
z=z/pts + zoff  # zoff=0 for auto
time.sleep(1) # NOW SHOULD ME ON TOP OF MOLECULE

print("Setting Z-Pos/Setpoint = {:8.2f} A".format( z))
gxsm.set ("dsp-adv-dsp-zpos-ref", "{:8.2f}".format( z))
gxsm.set ("dsp-fbs-bias","%f" %bias)
gxsm.set ("dsp-adv-scan-fast-return","5")
gxsm.set ("dsp-fbs-scan-speed-scan","8")
gxsm.set ("dsp-fbs-ci","3")
gxsm.set ("dsp-fbs-cp","0")
levelreg = level*0.99
gxsm.set ("dsp-fbs-mx0-current-level","%f"%level)
gxsm.set ("dsp-fbs-mx0-current-set","%f"%levelreg)
gxsm.set ("dsp-fbs-bias","%f" %bias)
return z
```

```python

def exit_force_map(bias=0.2, current=0.02):
  gxsm.set ("dsp-adv-scan-fast-return","1")
  gxsm.set ("dsp-fbs-mx0-current-set","%f"%current)
  gxsm.set ("dsp-fbs-mx0-current-level","0.00")
  gxsm.set ("dsp-fbs-ci","35")
  gxsm.set ("dsp-fbs-cp","40")
  gxsm.set ("dsp-fbs-scan-speed-scan","250")
  gxsm.set ("dsp-fbs-bias","%f" %bias)

def process(input_list, threshold=20):
    combos = itertools.combinations(input_list, 2)
    points_to_remove = [point2 for point1, point2 in combos if math.dist(point1, point2)<=threshold]
    points_to_keep = [point for point in input_list if point not in points_to_remove]
    return points_to_keep

def auto_afm_scanspeed(y):
  ms = gxsm.get_slice(2, 0,0, y,1) # ch, v, t, yi, yn   ## AFM dFreq in CH3
  med = np.median(ms)
  dFspan = np.max(ms) - np.min(ms)
  if dFspan > 1.0:
    gxsm.set ("dsp-adv-scan-fast-return","5")
    time.sleep(1)
    gxsm.set ("dsp-fbs-scan-speed-scan","8")
  elif dFspan > 0.8:
    gxsm.set ("dsp-adv-scan-fast-return","5")
    time.sleep(1)
    gxsm.set ("dsp-fbs-scan-speed-scan","10")
  elif dFspan > 0.5:
    gxsm.set ("dsp-adv-scan-fast-return","5")
    time.sleep(1)
    gxsm.set ("dsp-fbs-scan-speed-scan","15")
  elif dFspan > 0.4:
    gxsm.set ("dsp-adv-scan-fast-return","2")
    time.sleep(1)
    gxsm.set ("dsp-fbs-scan-speed-scan","20")
  elif dFspan > 0.3:
    gxsm.set ("dsp-adv-scan-fast-return","2")
    time.sleep(1)
    gxsm.set ("dsp-fbs-scan-speed-scan","30")
  else:
    gxsm.set ("dsp-adv-scan-fast-return","1")
    time.sleep(1)
    gxsm.set ("dsp-fbs-scan-speed-scan","50")
  #print(’Median: ’, np.median(ms))
  #print(’Min: ’, np.min(ms))
  #print(’Max: ’, np.max(ms))
  #print(’Range: ’, np.max(ms) - np.min(ms))


def get_gxsm_img_bypkt(ch):
  # fetch dimensions
  dims=gxsm.get_dimensions(ch)
  #print (dims)
  geo=gxsm.get_geometry(ch)
  #print (geo)
  diffs=gxsm.get_differentials(ch)
  #print (diffs)
  m = np.zeros((dims[1],dims[0]), dtype=float)
  for y in range (0,dims[1]):
    for x in range (0, dims[0]):
      v=0
      m[y][x]=gxsm.get_data_pkt (ch, x, y, v, 0)*diffs[2]  # Z value in
```

```
   Ang now
  return m


def get_gxsm_img(ch):
  dims=gxsm.get_dimensions(ch)
  return gxsm.get_slice(ch, 0,0, 0,dims[1]) # ch, v, t, yi, yn


def get_gxsm_img_cm(ch):
  # fetch dimensions
  dims=gxsm.get_dimensions(ch)
  print (dims)
  geo=gxsm.get_geometry(ch)
  print (geo)
  diffs=gxsm.get_differentials(ch)
  print (diffs)
  m = np.zeros((dims[1],dims[0]), dtype=float)

  for y in range (0,dims[1]):
    for x in range (0, dims[0]):
      v=0
      m[y][x]=gxsm.get_data_pkt (ch, x, y, v, 0)*diffs[2] # Z value in Ang now

  cmx = 0
  cmy = 0
  csum = 0
  cmed = np.median(m)
  print (’Z base: ’, cmed)
  b=2
  for y in range (b,dims[1]-b):
    for x in range (b, dims[0]-b):
```

```python
      v=0
      m[y][x]=m[y][x] - cmed # Z value in Ang now
      if m[y][x] > 0.5:
        cmx = cmx+x*m[y][x]
        cmy = cmy+y*m[y][x]
        csum = csum + m[y][x]
  if csum > 0:
    cmx = cmx/csum
    cmy = cmy/csum
  else:
    cmx = dims[0]/2
    cmy = dims[1]/2
  print('PointCM: ', int(round(cmx)), int(round(cmy)))
  gxsm.add_marker_object(ch, 'PointCM',1, int(round(cmx)), int(round(cmy)), 1.0)
  export_drawing(ch, '-CM')
  return m, cmx, cmy

def ai_decide(ch):
  full_path_name = gxsm.chfname(ch).split()[0]
  cfname = os.path.basename(full_path_name)
  folder = os.path.dirname(full_path_name)
  name, ext = os.path.splitext(ncfname)
  print ('AI decide on: ', name)

  img = get_gxsm_img(ch) # Load image from AFM channel
  norm_img = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX, cv2.CV_8U) # Normalize the color scale of the image from 0 to 255
  rgb_img = gray2rgb(norm_img) # Turn Grayscale to RGB
  im_bgr = cv2.cvtColor(rgb_img, cv2.COLOR_RGB2BGR) # Turn RGB to BGR
  print(im_bgr.shape) #Print Shape to double check correct input. Should be (330,330,3)
  outputs = predictor(im_bgr) # Using the predictor from the model.

  # Place the outputs into arrays to use them easier.
  mask_array = outputs['instances'].pred_masks.to("cpu").numpy()
  num_instances = mask_array.shape[0]
  scores = outputs['instances'].scores.to("cpu").numpy()
  labels = outputs['instances'].pred_classes .to("cpu").numpy()
  bbox   = outputs['instances'].pred_boxes.to("cpu").tensor.numpy()

  #print ('Mask Array:')
  #print (mask_array)

  # Create a mask for the Input AFM image
  mask_array = np.moveaxis(mask_array, 0, -1)
  mask_array_instance = []
  height = im_bgr.shape[0]
  width = im_bgr.shape[1]
  img_mask = 255*np.ones([height, width, 3], np.uint8) # zeros

  for i in range(num_instances):
    if labels[i]==0:
#     color = (250, 43, 138) #Purple Color for close and distortion regions
      color = (128, 43, 250) #Purple Color for close and distortion regions
    elif labels[i]==1:
#     color = (0, 1, 255) #Red Color for far regions
      color = (255, 1, 0) #Red Color for far regions
    else:
#     color = (0,255,0) #Green color for ideal ring regions
      color = (0,255,0) #Green color for ideal ring regions
    image = np.zeros_like(im_bgr)
    mask_array_instance.append(mask_array[:, :, i:(i+1)])
    image = np.where(mask_array_instance[i] == True, 255, image)
    array_img = np.asarray(image)
    img_mask[np.where((array_img==[255,255,255]).all(axis=2))]=color
  img_mask = np.asarray(img_mask)
  #gxsm.load(9, img_mask)
  purple = np.where(img_mask[:,:,1] == 43)
  red = np.where(img_mask[:,:,1] == 1)
  green = np.where(img_mask[:,:,1] == 255)
  p = purple[0].size
  r = red[0].size
  g = green[0].size
  sum = p + r + g
  #print(purple[1].size)
  #print(red[1].size)
  #print(green[1].size)
  #print(sum)
  if ( g > p and g > r):
    print("Good")
    action = 'okay'
  elif( p > g and p > r):
    print("Close")
    action = 'close'
  else:
    print("Far")
    action = 'far'
  logging.info('ai_decide on {}: {}'.format(name, action))
  return action, img_mask


def ai_mask_to_gxsm_ch (ai_mask_img, chm):
  img_shape= ai_mask_img.shape
  rgb = np.moveaxis(ai_mask_img, 0, -1)
  rgb = np.moveaxis(rgb, 0, -1)
  n = np.ravel(rgb) # make 1-d
  mem2d = array.array(’f’, n.astype(float))
  mem2d=np.resize(mem2d, 4*img_shape[0]*img_shape[1])
  gxsm.chmodea (chm)
  gxsm.createscanf (chm, img_shape[1], img_shape[0], 4, 45, 45, mem2d, False)

  full_original_name = gxsm.chfname(afm_ch).split()[0]

  print(full_original_name)
  folder = os.path.dirname(full_original_name)
  ncfname = os.path.basename(full_original_name)
  name, ext = os.path.splitext(ncfname)
  dest_name = folder+’/’+name+’_ai_mask’
  print(dest_name)
  gxsm.save_drawing(chm, 0,0, dest_name+’.png’)


def locate_molecule_ch(ch,thresh_val):
  img = get_gxsm_img(ch)
  norm_img = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX, cv2.CV_8U)
  gray_img = rgb2gray(norm_img)
  max_thresh = 255
  thresh = thresh_val
  def thresh_callback(val):
        threshold = val
        canny_output = cv2.Canny(gray_img, threshold, threshold*2)
        contours,_ = cv2.findContours(canny_output, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
        contours_poly = [None]*len(contours)
        boundRect = [None]*len(contours)
        centers = [None]*len(contours)
        radius = [None]*len(contours)
        x_y_coord = []
        for i, c in enumerate(contours):
          contours_poly[i] = cv2.approxPolyDP(c, 3, True)
          boundRect[i] = cv2.boundingRect(contours_poly[i])
          centers[i], radius[i] = cv2.minEnclosingCircle(contours_poly[i])
          if radius[i] < 3 and radius[i] > 1:
            if centers[i] not in x_y_coord:
              x_y_coord.append(centers[i])

        return (x_y_coord)
  x_y_molecules = thresh_callback(thresh)
  return (x_y_molecules)

def do_stm_and_lock_on_center(mi):
  print(’STM: Scanning M’,mi)
  gxsm.startscan()
  time.sleep(2)
  gxsm.set ("dsp-fbs-scan-speed-scan","225")
  time.sleep(1)
  print(’waiting....’)

  l=0
  while l >= 0  and int(gxsm.get(’script-control’)) >0:
    l =gxsm.waitscan(False)
    #print (’Line=’,l)
    time.sleep(2)

  gxsm.stopscan()
  time.sleep(2)

  print(’STM completed. Centering.’)

  #print(’... cleanup old markers (make sure)’)
  #r=gxsm.marker_getobject_action(0, ’PointCM’,’REMOVE’)
  #print(r)
  time.sleep(1)
  print('calculate CM')
  m, cx, cy = get_gxsm_img_cm(0)
  print('CM:' ,cx,cy)
  time.sleep(1)
  print('Adjust Offset')
  r=gxsm.marker_getobject_action(0, 'PointCM','SET-OFFSET')
  print(r)
  time.sleep(2)
  print ('cleanup marker')
  r=gxsm.marker_getobject_action(0, 'PointCM','REMOVE')
  print(r)


def do_HR_AFM(mi, tipz):
  GetSC()
  z = sc['Tip_Z_Ref']
  sc['Tip_Z'] = -tipz
  sc['Molecule'] = mi
  SetSC()
  gxsm.set ("dsp-adv-dsp-zpos-ref", "{:8.2f}".format( z-tipz))
  print('HR-AFM: Scanning M',mi, ' at Z=', z-tipz)
  gxsm.startscan()
  time.sleep(2)
  print('waiting....')
  gxsm.set ("dsp-fbs-scan-speed-scan", "10")
  time.sleep(1)
  gxsm.set ("dsp-fbs-scan-speed-scan", "8")
  l=0
  lp=1

  next_ai_check=25

  #while gxsm.waitscan(False) >= 0 and  int(gxsm.get('script-control')) >0:
  while l >= 0  and int(gxsm.get('script-control')) >0:
    l =gxsm.waitscan(False)
    #print ('Line=',l)

    if do_AI_afm and l > next_ai_check:
      next_ai_check=l+25
      result, ai_mask_img = ai_decide(afm_ch)
      ai_mask_to_gxsm_ch (ai_mask_img, 10)

    time.sleep(5)
    GetSC()
    if l > lp and sc['AutoAFMSpeed'] > 0:
      auto_afm_scanspeed(l)
      lp=l+1

  print('HR-AFM completed, saving...')


###################################################################

max_mol = 50


if  TEST_AI:
  afm_ch=0
  mask_ch=9
  ct = datetime.datetime.now()
  print (ct, ' ** Test AI on ch', afm_ch)
  result, ai_mask_img = ai_decide(afm_ch)
  print (ct, ' ** ')
```

```python
#ai_mask_to_gxsm_ch (ai_mask_img, 10)

print ('Mask')
img_shape= ai_mask_img.shape
print (img_shape)
#rgb = ai_mask_img

afmimg = get_gxsm_img(afm_ch) # Load image from AFM channel
print ('AFM img')
print(afmimg.shape)
afm_norm_img = cv2.normalize(afmimg, None, 0, 255, cv2.NORM_MINMAX, cv2.CV_8U) # Normalize the color scale of the image from 0 to 255
afm_rgb_img = gray2rgb(afm_norm_img) # Turn Grayscale to RGB
print(afm_rgb_img.shape)

rgb = afm_rgb_img * (0.5*0.5*ai_mask_img/255)
ai_mask_img = rgb

rgb = np.moveaxis(ai_mask_img, 0, -1)
rgb = np.moveaxis(rgb, 0, -1)
print(rgb.shape)


n = np.ravel(rgb) # make 1-d
mem2d = array.array('f', n.astype(float))
mem2d=np.resize(mem2d, 4*img_shape[0]*img_shape[1])
#mem2d=np.roll(mem2d, -330*330)

#afm = get_gxsm_img(afm_ch) # Load image from AFM channel
#afmn = np.ravel(afm) # make 1-d
#mem2d_afm = array.array('f', afmn.astype(float))
#mem2d = np.concatenate ((mem2d_afm, mem2d))

```

```
  gxsm.chmodea (mask_ch)
  gxsm.createscanf (mask_ch,img_shape[1],img_shape[0],4, 45, 45, mem2d, False)
  #gxsm.add_layerinformation ("@ "+str(flv)+" Hz",10)
  #gxsm.createscanf : Create Scan float: gxsm.createscan (ch,nx,ny,nv pixels, rx,ry in A, array.array(’f’, [...]), append)

  max_mol = 0


##################################################################

print(’Map in CH’, map_ch+1)

print(’Removing all Rectangles!’)
r=gxsm.marker_getobject_action(map_ch, ’Rectangle’,’REMOVE-ALL’)
print(r)




if do_auto_locate:
  ##### Locate Molecules using OpenCV Some functions might be extra could be use for later...###
  print(’Finding Molecules’)
  #molecule_coord = locate_molecule_nc(basefile, 65)
  molecule_coord = locate_molecule_ch(map_ch, 65)
  pro_molecule_coord = process(molecule_coord)
  time.sleep(1)
  print(’Found Molecules:’,len(pro_molecule_coord))

  ##### Clean Up the channel being used################
  print(’Removing all Rectangles!’)
  r=gxsm.marker_getobject_action(map_ch, 'Rectangle','REMOVE-ALL')
  print(r)
  time.sleep(1)

  print('Cleanup Points')
  r=gxsm.marker_getobject_action(map_ch, 'Point','REMOVE-ALL')
  print(r)
  r=gxsm.marker_getobject_action(map_ch, '*Marker','REMOVE-ALL')
  print(r)
  time.sleep(1)

  #### Here add for loop to add boxes and labels to each molecule ######
  print('Marking Molecules at')
  print (pro_molecule_coord)

  for i in range(len(pro_molecule_coord)):
    gxsm.add_marker_object(map_ch, 'PointM{:02d}'.format(i),1, int(pro_molecule_coord[i][0]),int(pro_molecule_coord[i][1]), 1.0)

  time.sleep(1)

  if max_mol > 0:
    gxsm.set('script-control','2')
    print('waiting as long as sc>1')
    while int(gxsm.get('script-control')) > 1:
      time.sleep(0.5)

print('List Objects, Mark Mol, Setup Rects')
k=0
for i in range(0, max_mol): ##len(pro_molecule_coord)):
  o=gxsm.get_object (map_ch, i+k) ## adjust for inserted object -- always pre pended to list!
  print('O', i, ' => ', o)
  if o == 'None':
    break
  print('Marking M', i)
  r=gxsm.add_marker_object(map_ch, 'RectangleM{:02d}'.format(i), 0xff00fff0, round(o[1]),round(o[2]), sc['AFM_Range']/map_diffs[0])
  k=k+1 # we have not one more object prepended to the object list!
  print(r)

SetSC()
time.sleep(1)

if max_mol > 0:
  gxsm.set('script-control','2')
  print('waiting as long as sc>1 -- check configurations now')
  while int(gxsm.get('script-control')) > 1:
    time.sleep(0.5)

GetSC()

# make sure STM safe mode
exit_force_map(0.1, current=0.006)


gxsm.set('script-control','3')

for mi in range(0,max_mol):  ##len(molecule_coord)):
  full_path_name = gxsm.chfname(0).split()[0]
  ncfname = os.path.basename(full_path_name)
  folder = os.path.dirname(full_path_name)
  name, ext = os.path.splitext(ncfname)

  sc['Molecule'] = mi
  SetSC()
```

```

if int(gxsm.get(’script-control’)) < 1:
  break

time.sleep(1)
print(’selecting M’,mi)
r=gxsm.marker_getobject_action(map_ch, ’RectangleM{:02d}’.format(mi),’GET-COORDS’)
print(r)
if r != ’OK’:
  break

GetSC()
STM_points = round(sc[’STM_Range’]/STM_dx)
gxsm.set(’PointsX’, ’{}’.format(STM_points)) # readjust points
gxsm.set(’PointsY’, ’{}’.format(STM_points))

gxsm.set (’RangeX’,’{}’.format(sc[’STM_Range’])) # Readjust range to make all of them the same size.
gxsm.set (’RangeY’,’{}’.format(sc[’STM_Range’]))
time.sleep(1)

if do_stm:
  do_stm_and_lock_on_center(mi)
  time.sleep(1)

logging.info(’*** Next Molecule #{:d} {} -- centering.’.format(mi, name))

if int(gxsm.get(’script-control’)) < 1:
  break

# Setup AFM Scan
```

```
GetSC()
AFM_points = round(sc[’AFM_Range’]/AFM_dx)
gxsm.set(’PointsX’, ’{}’.format(AFM_points)) # readjust points
gxsm.set(’PointsY’, ’{}’.format(AFM_points))

gxsm.set (’RangeX’,’{}’.format(sc[’AFM_Range’])) # Readjust range to make all of them the same size.
gxsm.set (’RangeY’,’{}’.format(sc[’AFM_Range’]))
time.sleep(1)

# only do reconfigure scan geom -- todo: do not save
gxsm.startscan()
time.sleep(3)
gxsm.stopscan()
# do STM orbital scans +2V/-1.5V or so?

if do_afm:
  print(’moving tip on top of molecule to measure Z at ref conditons for HR-AFM’)

  # initial setpoint determinaion on this grid -- make better: assure on molecule!
  ds = 2.0
  c   = 0.0
  ref_xy_list = [ ]
  for i in np.arange(-1,2):
    for j in np.arange(-1,2):
      ref_xy_list.append([c+i*ds, c+j*ds])

  print(’HR-AFM transitioning...’)
  GetSC()
  z=init_force_map_ref_xy(0.02, level=sc[’CZ_Level’]*1e-3, ref_i=sc[’I_ref’]*1e-3, zoff = sc[’Z_start’], xy_list=ref_xy_list)
```

```
 sc['Tip_Z_Ref'] = z
 SetSC()
 tipz =0
 logging.info('*** Init AFM mode, start at Tip_Z_Ref = {:.2f}A - tipz=0 A'.format(z))
 z_ai_dir=0
 z_ai_step=0.3
 while int(gxsm.get('script-control')) >1 and tipz <= sc['Z_down'] and tipz < 2.0:
   logging.info('Starting AFM image')
   do_HR_AFM(mi, tipz)
   time.sleep(4)

   if tipz == 0:
     # set RectangleID/Label to McBSP Freq file name -- 1st of Z series
     full_original_name = gxsm.chfname(2).split()[0]
     ncfname = os.path.basename(full_original_name)
     bname, ext = os.path.splitext(ncfname)
     print(bname)
     ### CUSTOM FOR TOIS FILE NAMEING SCHEME ###
     filenumber = bname[11:14]
     print('File Number: ', filenumber)
     r = gxsm.marker_getobject_action(map_ch, 'RectangleM{:02d}'.format(mi),'SET-LABEL-TO:'+filenumber)
     print(r)
     logging.info('*** AFM File: {} *** RectangleM{:02d} = #{}'.format(bname, mi, filenumber))

   if do_AI_afm:
     result, ai_mask_img = ai_decide(afm_ch)
     ai_mask_to_gxsm_ch (ai_mask_img, 10)
     export_png(afm_ch, 'afm')
     if result == 'okay':
       logging.info(’Z Adjust none’)
       break  ## OK done
     elif result == ’far’:
       if z_ai_dir < 0:
         z_ai_step = z_ai_step*0.5
         if z_ai_step < 0.1:
           logging.info(’Z Adjust done. far, but been downwithin 10pm’)
           break  ## OK done
         logging.info(’Z Adjust revert {:.1f}A’.format(z_ai_step))
       tipz = tipz+z_ai_step # down a step
       z_ai_dir=1
       logging.info(’Z Adjust down’)
     else:
       if z_ai_dir > 0:
         z_ai_step = z_ai_step*0.5
         if z_ai_step < 0.1:
           logging.info(’Z Adjust done. close, but been up within 10pm’)
           break ## OK done
         logging.info(’Z Adjust revert {:.1f}A’.format(z_ai_step))
       tipz = tipz-z_ai_step # up a step
       z_ai_dir=-1
       logging.info(’Z Adjust up’)
   else:
     tipz = tipz+0.3 # simple down steps as programmed
   logging.info(’Z Adjust to Tip_Z_Ref = {:.2f}A - tipz={:.2f}A’.format(z, tipz))

 if int(gxsm.get(’script-control’)) >5:
     print(’waiting for re run as long as sc>5’)
     while int(gxsm.get(’script-control’)) > 5:
       time.sleep(0.5)

```

```
    logging.info('*** Exit AFM mode ***')
    exit_force_map(0.1, current=0.006)

logging.shutdown()
```

Listing A.2: Auto-HR-AFM: Python script as of 04/26/2023