7-2020

Nonlinear Dimensionality Reduction for the Thermodynamics of Small Clusters of Particles

Aditya Dendukuri
University of Arkansas, Fayetteville

Nonlinear Dimensionality Reduction for the Thermodynamics of Small Clusters of Particles

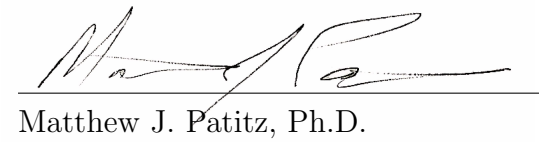
A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science

by

Aditya Dendukuri
University of Central Arkansas
Bachelor of Science in Computer Science, 2018.

July 2020
University of Arkansas

This thesis is approved for recommendation to the Graduate Council



Matthew J. Patitz, Ph.D.
Thesis Director

David M. Ford, Ph.D.
Committee Member

Lu Zhang, Ph.D.
Committee Member

Khoa Luu, Ph.D
Committee Member

ABSTRACT

This work employs tools and methods from computer science to study clusters comprising a small number N of interacting particles, which are of interest in science, engineering, and nanotechnology. Specifically, the thermodynamics of such clusters is studied using techniques from spectral graph theory (SGT) and machine learning (ML). SGT is used to define the structure of the clusters and ML is used on ensembles of cluster configurations to detect state variables that can be used to model the thermodynamic properties of the system. While the most fundamental description of a cluster is in $3N$ dimensions, *i.e.*, the Cartesian coordinates of the particles, the ML results demonstrate that sub-spaces of much lower dimension can describe the observed structural motifs. Furthermore, these sub-spaces correlate with meaningful physical variables such as radius of gyration r_g and discrete connectivity c, which can be used as state variables in thermodynamic property descriptions. The overarching theme of this thesis is to develop the practice of utilizing data-driven computational techniques to solve problems in natural sciences. Code for this project can be found at https://github.com/AdityaDendukuri/DimReductionThermodynamics.

ACKNOWLEDGEMENTS

First and foremost, I want to thank Dr. David Ford for giving me this opportunity to work on this wonderful research problem and supporting me throughout my graduate study at the University of Arkansas. He is an amazing scientist and advisor who has taught me many valuable lessons on scientific research. Thanks to him, my will to pursue academic research has strengthened significantly and I will cherish the knowledge I gained from him for the rest of my career. I also want to sincerely thank my academic advisor at the computer science department, Dr. Matthew Patitz, who introduced me to Dr. Ford and hence, in addition to his constructive critiques on my techniques, has played an instrumental role in developing my career. I want to thank Dr. Khoa Luu, who has been my de facto advisor for the deep learning aspect of this project. He has also provided me with a productive lab environment with limitless resources and bright students as my labmates. He has also given me a valuable opportunity to learn and work on projects regarding quantum computing and machine learning and hence, significantly broadening my knowledge. Finally, I want to thank my fellow graduate student Gulce Kalyoncu, who has significantly contributed and enhanced the quality of this project. Her excellent work on generating, pre-processing the datasets and developing diffusion maps with Hausdorff and Mayer f-bond distances significantly eased my transition into this project.

DEDICATION

I want to dedicate this thesis to my parents Purnachandra Rao and Anuradha Dendukuri who did an amazing job in raising me and budding my passion in sciences. I will be forever indebted to them for selflessly placing my education and career ahead of their lives. I also want to thank my uncle and aunt Dr. Ramesh and Dr. Uma Garimella for playing the role of parents far away from home and giving me strong support throughout my education in the United States. I am incredibly lucky to have such a supportive family and I would be nowhere without their selfless support.

TABLE OF CONTENTS

1 Introduction and Background

Computational science is the practice of utilizing techniques from computer science to study complex systems in engineering and sciences [1, 2, 3]. These complex systems tend to be highly nonlinear with a large parameter space, which makes an analytical study of these systems intractable. Hence, computers can be utilized to provide numerical solutions. Examples of areas in science and engineering that benefit from computational science include climate modeling [4, 5], bioinformatics [6], and molecular engineering [7, 8]. Until recently, computational science has been primarily concerned with solving highly nonlinear differential equations like the Navier-Stokes equations for fluid flow. This field is called numerical modeling and has a huge vibrant community [9, 10, 11]. The second face of computational science, which is fairly new, focuses on *data-driven* analysis using tools from statistics and data science. The dogma of this sub-field usually involves gathering data via computer simulations (or experiment) and mining fundamental properties of the system from the data set. Additionally, the recent surge in machine learning research strongly improved the prospect of advancements in studying complex systems. For example, there have been strong advances in approximating the Koopman operator, which describes the dynamic behaviour of a complex system, using machine learning [12, 13, 14]. Analysis of protein folding has also been benefited from data driven computational science [15, 16, 17].

This study focuses on the problem of *dimensionality reduction* of molecular simulation data. More specifically, a system of N interacting particles is simulated in a three-dimensional periodic box at fixed temperature using a standard Monte Carlo algorithm for

generating configurations [18, 19]. The conditions are such that configurations ranging from solid-like clusters to gas-like expanded states are observed in a single trajectory. The goal is to (1) use machine learning to identify a space of lower dimension ($<< 3N$) that captures the most important elements of the structural changes of the system and (2) correlate these lower dimensions with geometric variables that can serve as thermodynamic state variables of for the system. The motivations and specifics of this problem are in section 1.2. Key starting points of this study are the work of Bevan et. al [20], which demonstrates the utility of the dimensionality reduction technique *diffusion maps* in studying colloidal systems, and the work of Long and Ferguson [16], which provides a perspective of physical clusters as mathematical graphs. This study expands on both works by (1) exploring different distance metrics within the diffusion map framework, especially by utilizing spectral graph theory (section 2.12) to counter the performance bottleneck caused by aligning particle indices using permutations, and (2) employing an alternate dimensionality reduction technique, neural networks, to the same data sets. The rest of this section lays out the background for the tools and techniques used in this study and problem statement.

1.1 Thermodynamics

Thermodynamics is a branch of physics that describes the relationship between heat and work and provides a set of state variables, some directly observable and some abstract, that can be used to model heat and work effects on a given system [21]. Classical thermodynamics deals with bulk, *i.e.* essentially infinite, systems consisting of a large number, *e.g.* 10^{23}, of fundamental particles, which are typically atoms or molecules [22]. For a system

with a large, fixed number of particles, a fundamental equation of thermodynamics is

$$dA = -SdT - pdV, \tag{1.1}$$

where A is the Helmholtz free energy of the system (a measure of its capacity to do work), S is the entropy, T is the temperature, p is the pressure, and V is the volume. Note that if the function $A(T, V)$ is known for a given material, other useful quantities may be computed from it, for example the pressure may be computed as $p = -\left(\frac{\partial A}{\partial V}\right)_T$.

There are situations, such as a vapor bubble in a liquid or a liquid droplet in a vapor, when a system may no longer be considered as infinite in extent. As long as the system is still large enough to have a well-defined bulk region in its interior, the thermodynamic description can be adjusted by adding a term to eq. 1.1 that accounts for effects at the boundary [23, 24]. For example, for a gas bubble or liquid droplet, one may write

$$dA = -SdT - pdV + \gamma d\mathcal{A}, \tag{1.2}$$

where $\mathcal{A}$ is the surface area of the object and γ is the conjugate surface tension. This approach causes some mathematical ambiguity, because the variable $\mathcal{A}$ is not extensive (first-order homogeneous) in the system size like the variable V, but it is still practically useful in computing heat and work effects [24].

In this study, we are concerned with the situation where the system becomes so small that the definition of variables like volume and surface area become ambiguous. Generally speaking, this will happen when the number of particles N in the system is $\mathcal{O}(100)$ or less,

which is relevant to some areas of nanotechnology. In such situations, one might say that the problem should enter the realm of statistical thermodynamics [25] that invokes a model of the system as a set of interacting particles rather than a continuum. The free energy could then be computed as

$$A = -k_B T \ln Q, \tag{1.3}$$

where

$$Q = \sum_i e^{-U_i/k_B T}, \tag{1.4}$$

with k_B the Boltzmann constant, U_i the potential energy of microstate i, and the sum running over all microstates. The details of the computation of the partition function Q would vary depending on the nature of the model chosen. The basic elements of such a computation would be (1) a model for the potential energy as a function of the particle positions, $U(r_1, r_2, ...r_N)$, and (2) a method for enumerating and generating the microstates. While this approach would certainly give an answer for a specific model system, it would likely be very computationally expensive, and it would need to be repeated for each choice of N and each different type of particle that might be of interest.

1.2 Problem statement

We propose a different approach in this thesis. Using a rather generic potential model and two specific system sizes with $N < 100$, we generate a representative (but not complete) set of microstates (particle configurations) using a standard Monte Carlo simulation method. We then apply machine learning techniques to these sets to search for low-dimensional spaces in the context of pattern recognition to identify state variables that might be generally

appropriate for such systems. Such variables would replace V and $\mathcal{A}$ in an equation like eq. 1.2. From the viewpoint of physical science, this is a study of the thermodynamics of small clusters [20]. From the viewpoint of computer science and machine learning, this is an application of dimensionality reduction. The fundamental set of microstate variables for the model system is the set of Cartesian coordinates of its constituent particles, defined as

$$\mathcal{X} : \{r_1, r_2, \ldots, r_N\} \in \mathbb{R}^{3N}. \tag{1.5}$$

The dimensionality of this state space is $3N$, and the **primary goals** of this project are to (1) detect subspaces in $\mathcal{X}$ that capture the most important physical features of the systems and (2) find a set of readily computable functions of $\mathcal{X}$ that correlate with these subspaces and thus can be used as thermodynamic state variables, *e.g.* as 'replacements' for volume V and surface area $\mathcal{A}$ in eq 1.2.

1.3 Literature Review

This section summarizes the existing literature regarding model reduction for molecular simulation. Diffusion Maps (DMaps) [26], have been previously used to identify reduced dimensions in small systems of various types [27, 17, 28, 29, 30, 31, 32, 33]. One common model type that has benefited largely from these techniques is protein models [26, 8]. Protein models usually have a unique label for every residue, making abstract low dimensional representations of these systems somewhat less challenging. We focus on systems where all the atoms are identical [20], which make existing state-space representations using coordinates difficult as one must consider the structural symmetries caused by index permutations. Long

and Ferguson [34], provided a robust technique for aligning structures that addressed the expensive process of checking particle index permutations based on the idea of representing clusters using graphs via adjacency matrices. However, this technique becomes quickly intractable for clusters with larger sizes than handfuls of atoms due to a step which requires the calculation of similarity between all possible index permutations. This issue motivated us to look for representations which are *permutation invariant* and led us to rely on using spectral based techniques [35, 36] discussed in sections 2.12 and 3.

Deep learning [37], provides us with a relatively modern set of tools which can, in a sense, automate the process of pattern recognition by defining parametrized models to represent a system, and tuning the parameters based on an optimization rule. The promise of deep learning is the ability to define models which do not need to be posed in an ad-hoc fashion. Simplified potentials have been derived using deep learning based physical approximations whenever the direct application of energy landscapes has been too demanding [38, 39, 40, 41, 42]. Ballard et. al [43], introduced *deep potential*, which is a molecular dynamics scheme which demonstrates the usage of potential energy models defined via carefully crafted deep neural networks. Carrasquilla and Melko [44], used *convolutional neural networks* to scan the spin configurations of semi conductors. The results of this work were especially promising as the network learned the properties behind phase changes, without any previous knowledge about the Hamiltonian of the system. Although the approaches proposed by Ballard et. al (2017) [43] and Carrasquilla and Melko [44] are based on a classification problem, this study employs a regression model. The experimental motivation for tackling this problem is the ability to control direced assembly of small clusters of colloidal particles into crystalline states [45, 46, 47, 20, 48, 49]. The reduced model parameters de-

duced via techniques above can be used to direct the assembly process [20] in an experimental environment.

2 Model Systems and Computational Methods

This chapter describes the two main model systems, namely clusters of 3 and 13 particles interacting via the Lennard-Jones potential, and the simulation methods used to generate representative configurations that comprise the dataset. Following the model systems, the mathematical and compuational techniques used in this study are laid out. The primary contribution of this study is the introduction of a spectral distance metric (see section 2.13.3) to define a Markov process over molecular simulation data of LJ clusters or diffusion maps (unsupervised learning) and the usage of a regression based neural network optimization problem for learning low dimensional representations for the potential energy distributions (supervised learning).

2.1 Lennard-Jones (LJ) interaction potential

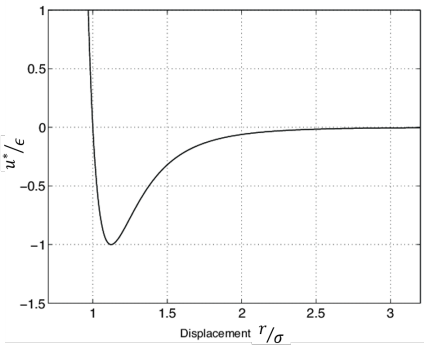


Figure 2.1: LJ potential. Potential energy as a function of distance between atoms.

The Lennard-Jones (LJ) potential (fig 2.1) is a commonly used [25] model that captures the energetic interaction between a pair of atoms due to the London dispersion forces

arising from instantaneous dipole fluctuations. As such, it is a very good model for the interaction between neutral, nonpolar, spherical atoms like noble gases (*e.g.*, argon). As can be seen in the figure, the model includes an attraction (negative potential energy) at moderate distances that gradually approaches zero at infinite separation and a repulsion (positive potential energy) at short distances that rapidly approaches infinity at zero separation. The potential has two parameters; ϵ characterizes the attraction and sets the depth of the potential well, and σ characterizes the particle size and sets the location of the transition from net repulsion to net attraction. Beyond the specific application to noble gases, the LJ potential is a useful general model in statistical thermodynamics because it is a continuous function that contains both short-range repulsion and moderate-range attraction.

The mathematical model for the LJ potential between atom i and atom j is

$$u_{ij} = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{2.1}$$

$$r_{ij} = ||r_i - r_j||. \tag{2.2}$$

The LJ model is a pairwise interaction model. If we have a system with N atoms, we can compute the total potential energy U of a configuration by summing all the pairwise contributions u_{ij} as

$$U = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} u_{ij}. \tag{2.3}$$

The pairwise potential u_{ij} is minimized when the pair of atoms is at a separation $r_{min} = 2^{1/6}\sigma$, but finding the minimum total potential energy U for N atoms is more complicated, and in fact the minimization objective of equation 2.3 is a popular test case for optimization

algorithms [50]. Various low-energy LJ structures for given N have been computed and tabulated [50].

2.2 Structural Variables of LJ Clusters

A useful concept to add for our study is the definition of a physical bond between two LJ atoms. We somewhat arbitrarily define a bond cutoff distance as $\eta = 1.2\sigma$, which is the point where $u_{ij} = -0.891\epsilon$ is too weak to be considered as a connection. With this in mind, we define a pairwise function

$$f_{ij}(r_{ij};\eta) = \begin{cases} 1, & \text{if } r_{ij} < \eta \\ 0, & \text{else} \end{cases} \tag{2.4}$$

to describe whether a bond exists between atoms i and j in a particular configuration. We define the number of bonds (n_b) of a cluster as

$$n_b = \sum_{i \sim j} f(r_{ij};\eta) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} f(r_{ij};\eta). \tag{2.5}$$

Connectivity (c), is the mean of the number of bonds each particle defined as

$$c = \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N} f(r_{ij};\eta).$$

Radius of gyration (r_g), is a continuous variable that fundamentally is the mean over the distance of each particle from the center of mass of collected particles. For molecular systems,

r_g can be formulated as a function of pairwise euclidean distances as

$$r_g = \frac{1}{2N}\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ij}}. \tag{2.6}$$

2.3 Monte Carlo simulation methods

With N particles and three dimensions, this is a $3N$ dimensional system. The dataset was generated using canonical Monte Carlo simulations [18], in which particles are moved based on a probability distribution following $e^{-\beta\Delta U}$, where ΔU is the change in potential energy and $\beta = \frac{1}{k_B T}$, where k_B is the Boltzmann constant [51]. We study LJ clusters in canonical ensembles in a three-dimensional periodic boundary box [18]. The temperature in dimensionless form is

$$T^* = \frac{k_B T}{\epsilon},$$

where ϵ is the LJ energy parameter characterizing the well depth or 'bond strength'. The box dimensions are 10σ on all sides, where σ is the LJ particle diameter. The potential energy in dimensionless form is $u^* = \frac{U}{3\epsilon}$, where U is the total potential energy arising from the sum of the N pairwise contributions (equation 2.3). The simulation ran for 10^7 steps and 5000 configurations were sampled for analysis. Before going forward with the analysis, it is helpful to process the dataset by centering around the origin and aligning to a single axis of rotation. This process is done by first calculating the center of mass of each cluster and translating all the atoms by moving the center of mass to the origin and aligning the clusters to a single axis of rotation. This step is not necessary for graphical methods as only pairwise distances are used. However, neural net optimization benefits greatly from

this step as it removes the ambiguities caused by the transnational and rotational degrees of freedom of the clusters. In addition to that, this also proves handy to visualize the high dimensional state space before doing any analysis as shown in figures 2.3 and 2.4. The idea is that when centered, as closely packed clusters have lower energy, we can observe a loose pattern where points closer to the origin tend to have lower energy and the energy spectrum can be observed as we move away from the origin.

2.4 Model LJ3: the 3-particle LJ cluster

The first system we employ has $N = 3$. This is perhaps the simplest model system that has some basic features of changing shape and connectivity of a cluster of atoms. With three particles and three dimensions, this is a nine-dimensional system. The dataset was generating using the canonical Monte Carlo simulations, as described above, with $T^* = \frac{k_B T}{\epsilon} = 0.18$. The simulation ran for 10^7 steps and 5000 configurations were sampled for analysis. In the results below, the potential energy is reported in dimensionless form as $u^* = \frac{U}{3\epsilon}$, where U is the total potential energy arising from the sum of the three pairwise contributions (equation 2.3).

Fig. 2.2a shows examples of the different configurational states this system sampled, varying from tightly clustered to completely broken. Fig. 2.2b is a histogram of the potential energies of the states in the dataset. The system sampled the range of possible values, since a tightly clustered state will have three pairwise bonds, each contributing -ϵ, for a value of $u^* = -1$, while a completely dissociated state will have no bonds for a value of $u^* = 0$. $T^* = 0.18$ was a good choice of temperature in this respect, as all structural motifs and potential energies were observed in a single trajectory. It is tempting to think of the peaks

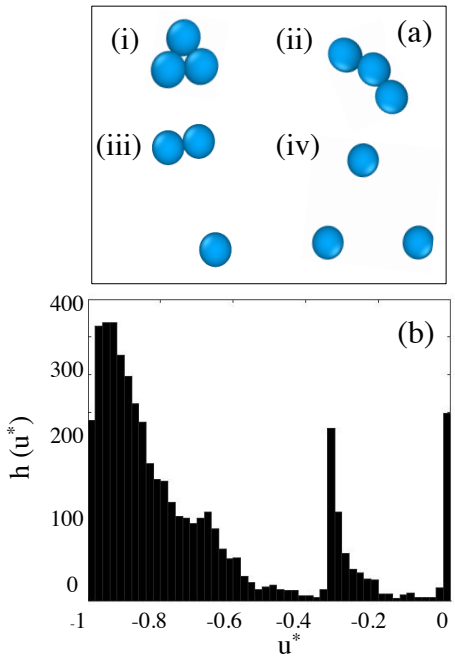


Figure 2.2: Results from the Monte Carlo simulation. (a) Representative snapshots showing the four major observed structural motifs: (i) tightly clustered, (ii) linear, (iii) partially dissociated, and (iv) completely dissociated. (b) Histogram of dimensionless potential energy u*.

in the histogram as corresponding to specific structural motifs, but such assignment is not straightforward, as can be seen from the ML results below.

Fig. 2.3a shows the raw data for the 5000 configurations [52]. Each configuration was mean-centered and aligned by its principal axes of rotation before analysis to remove collective rotational and translational degrees of freedom (Fig. 2.3b). The pattern where tightly packed clusters demonstrated a lower energy can be observed clearly after center-aligning the clusters.

2.5 Model LJ13: the 13-particle LJ cluster

LJ13 is a complicated system with many possible states (see section 2.6) and large dimensionality (13 particles $\times$ 3 coordinates = 39 dimensions). Since, there is no temperature

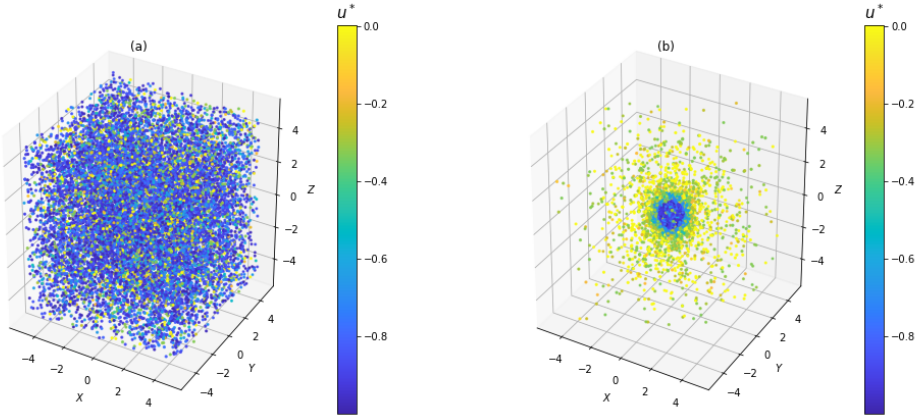


Figure 2.3: LJ3 state space at $T^* = 0.18$. (a) pre-processing (b) post-processing.

that can capture all the three phases (solid, liquid and gas), we used two set of datasets ($T^* = 0.28$ and 0.4). $T^* = 0.28$ mostly consists of solid and liquid phases while 0.4 captures liquid and gaseous phases. The collective state space and visualization are visualized in figure 2.4 and graphical snapshots of individual configurations are visualized in figure 2.6. Although the behaviour of these systems is similar to LJ3, the potential energy values seem to have a unimodal distribution instead of having a bimodal distribution.

2.6 Graphical Representation of Lennard Jones Molecules

Consider a Lennard Jones (LJ) cluster consisting of N particles and potential energy u^*. Since the coordinate space for the constituent atoms $\mathcal{X} \in \mathbb{R}^{3N}$ is not invariant under translation, rotation and index permutation, a graphical representation is very effective to represent the particle clusters [34]. The LJ cluster can be represented as an undirected graph $\mathcal{G}$ defined via an adjacency matrix $\mathbf{A}_{ij}$ that encodes local bond structure within the atoms. We define two adjacency matrices, one being the Eucledian distance $\mathbf{R}_{ij} = ||x_i - x_j||^2$ representation and a binary distance $G_{ij} = f(\mathbf{R}_{ij}; \eta)$, where $f(\mathbf{R}_{ij}; \eta) = 1$ if $R_{ij} \geq \eta$. Both

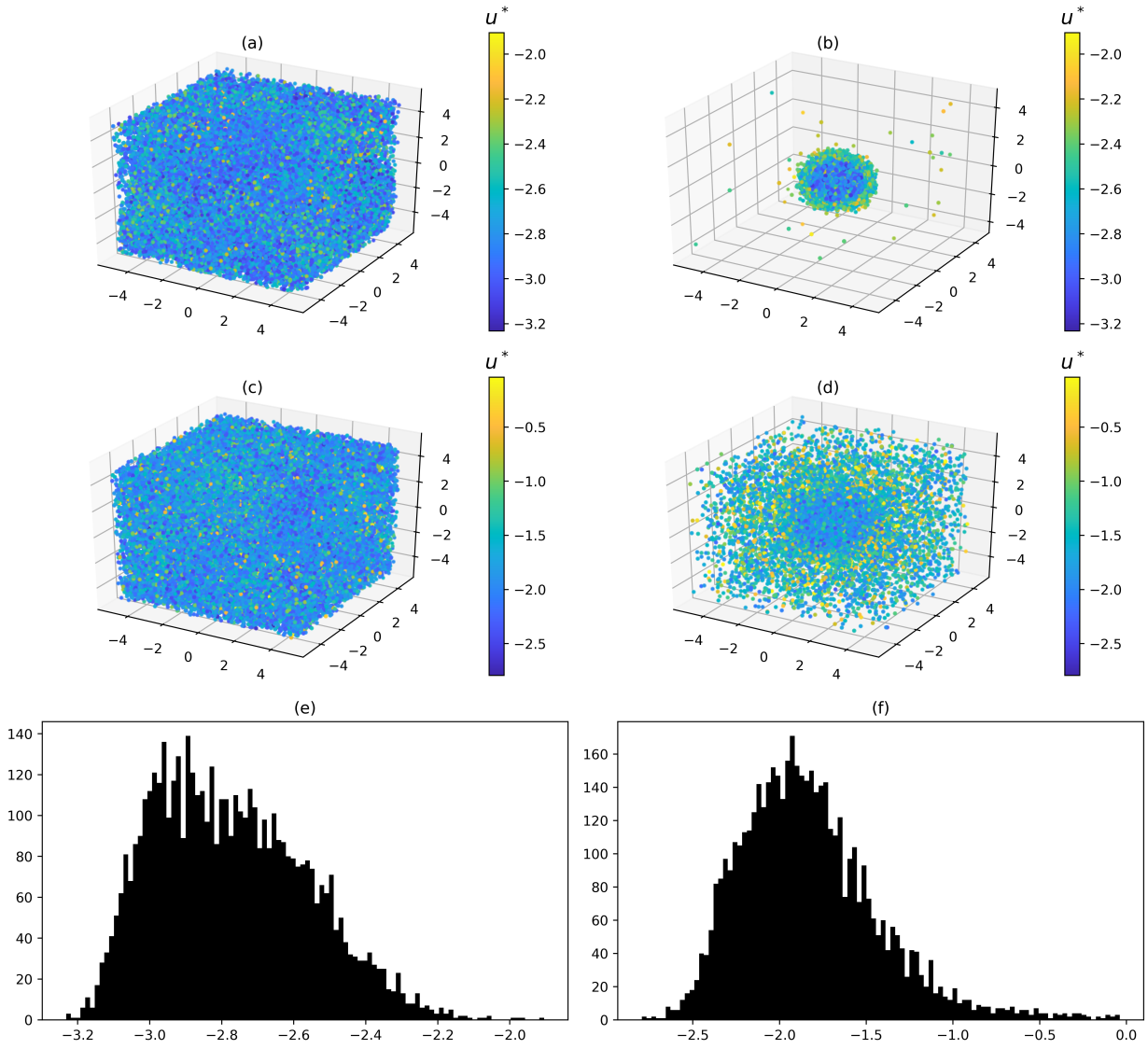


Figure 2.4: LJ13 at $T^* = 0.4$ and $T^* = 0.28$ model systems. The 39d space is collapsed into 3d space colored by potential energy by plotting 3d location of each particle in each cluster and coloring by the potential energy u^*. (a) $T^* = 0.28$ state space (pre-processed) (b) $T^* = 0.28$ state space (post-processed) (c) $T^* = 0.4$ state space (pre-processed) (d) $T^* = 0.4$ state space (post-processed) (e) $T^* = 0.28$ potential energy histogram (f) $T^* = 0.28$ potential energy histogram.

of these matrices are symmetric as the graph is undirected. These matrices can be used to compute some important quantities describing the atomic cluster. The primary variable is the degree vector $\boldsymbol{d} \in \mathbb{R}^N$, which holds the degree or number of connections for each node

and is computed by summing over the rows of G as

$$d_i^{(G)} = \sum_{j=1}^{N} G_{ij}. \tag{2.7}$$

We can use this degree vector notification to calculate the physical variables introduced in section 2.2. We can compute connectivity from d as

$$c = \frac{1}{2N} \sum_{i=1}^{N} d_i = \sum_{j=1}^{N} f(r_{ij}; \eta). \tag{2.8}$$

Radius of gyration (r_g) can be also calculated from the continuous adjacency matrix R and degree $d_i^{(R)} = \sum_{j=1}^{N} r_{ij}$ in the squared form,

$$r_g^2 = \frac{1}{2N} \sum_{i} d_i^{(R)}. \tag{2.9}$$

For larger clusters, n_b would not be a good metric to distinguish clusters in a macro scale. Hence we used number of clusters n_c, which is tricky to determine purely from looking at coordinate space $\mathcal{X}$. Therefore, one must calculate it from the graph adjacency representation with binary distance $\mathbf{G}$. We use a recursive algorithm to determine the number of clusters. The backbone algorithm is graph traversing using depth first search. The algorithm starts by starting a depth first search walk from a random node, adding all of the visited nodes in a list keeping track of this specific cluster. Once all the particles in this cluster are visited, we pick another random node which is not visited and repeat the same process until all nodes are visited. The number of clusters (n_c) would be the number of times we had to restart the depth first search.

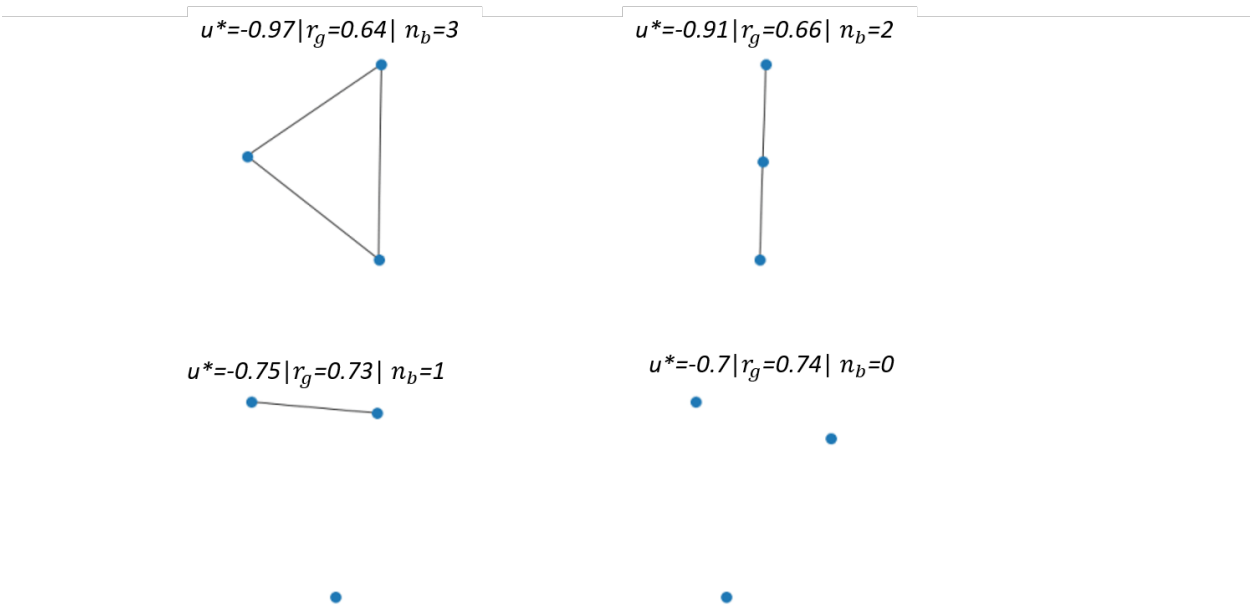


Figure 2.5: LJ3 graphical snapshots. These snapshots are not in scale and are only meant to visualize the structure.

We can visualize how these variables represent the structure of the LJ clusters in figure 2.5 for LJ3 and 2.6 for LJ13. These representations are different from purely plotting them in 3d space as this representation is purely connectivity information in a simpler 2d space. We can get a loose sense of how these structures look like in a macro level. For LJ3 we used n_b to distinguish the clusters in macro scale and for LJ13, we used number of clusters n_c. These figures have a label attached to them showing their respective structural variables.

2.7 Spectral Graph Theory

Spectral graph theory [53] is the study of spectral information of computational graph $\mathcal{G}$ and its properties. The spectral information of a graph corresponds to the eigenvalues and eigenvectors of the graph's Laplacian matrix $\mathbf{L}$, which is computed from the adjacency

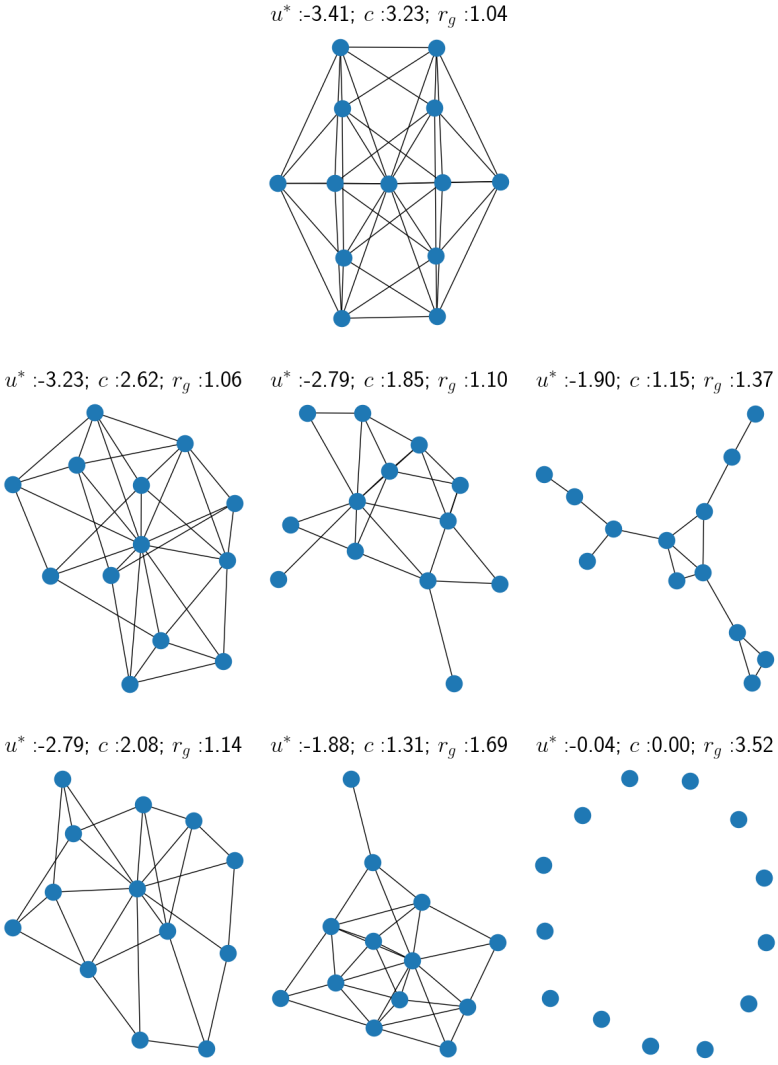


Figure 2.6: LJ13 configurations. These snapshots are two-dimensional representations of three-dimensional structures and are not to scale. Lines represent the existence of pairwise bonds. Each snapshot is labeled with the value of u^*, c, and r_g as defined in the text. (a) Configuration with global minimum of u^*; (b) Snapshot with lowest u^* value observed in the $T^* = 0.28$ data set; (c) Snapshot with median u^* value observed in the $T^* = 0.28$ data set; (d) Snapshot with highest u^* value observed in the $T^* = 0.28$ data set; (e)-(f) is the same as (b)-(d) but for the $T^* = 0.4$ data set.

matrix $\mathbf{A}$. The adjacency matrix $\mathbf{A}$ is a symmetric matrix that maps vertices to each other based on a function. For a physical cluster (a set of N particles with known locations in Cartesian space), a corresponding graph can be built with N vertices and a single edge joining each pair of vertices. The elements of the corresponding adjacency matrix would logically be some function of the Eucledian distance between the particle pairs, which would have a larger value when the distance was shorter and vice versa. That function might be continuous or it might be discrete, *i.e.* based on the definition of a bond between the two particles as discussed in section 2.1, in which case the adjacency representation would be binary as $\mathbf{G}_b \in \{0, 1\}$.

There are several advantages to an approach that employs mathematical graphs to model physical clusters. The graph representation is translation- and rotation-invariant, as the space consists only of pairwise distances between the particles. Spectral graph theory allows us to compute representations that are invariant under index permutation, which relaxes the NP-complete problem of aligning indices [54, 55]. The established theory has a variety of connections between the properties of a graph and its Laplacian eigenvalues/vectors. For example, in the context of random walks on graphs, a Gaussian process over the spectral space of the graph Laplacian is shown to be related to the cover times. *Laplacian embedding* is used to encode the graphical structure in an Eucledian space. Diffusion maps [55] utilize Laplacian embedding to encode the random walk on a lower dimensional Eucledian space.The purpose for the rest of this chapter is to lay out the theory behind spectral graph representations.

2.7.1 The Graph Laplacian

For a continuous function f, the *Laplace operator* (Δ) computes the divergence of its gradient as

$$\Delta f = \nabla \cdot \nabla f. \tag{2.10}$$

Physically, the operator describes the local density of the flux of the gradient in f; a large positive value implies that the location is acting like a source, with a large outward flux of the gradient vector. This idea can be extended to discrete systems like graphs using matrices.

For a graph $\mathcal{G}(V, E)$ with vertices V and edges E, the flux of the gradient of a function f at vertex u can be estimated by a properly weighted sum over the differences in f at all connected vertices, as

$$\Delta f(u) = \sum_{v \sim u} w_{ij}(f(u) - f(v)), \tag{2.11}$$

where the sum is over all vertices v that are connected to u, and w_{uv} is the value associated with the edge connecting u and v. Here Δ is the *discrete Laplace operator* corresponding to the continuous version defined in eq. 2.10. Equation 2.11 can be expressed in matrix-vector notation as

$$\Delta f \rightarrow \mathbf{Lf}, \tag{2.12}$$

where $\mathbf{f}$ is a vector containing the values of f at the vertices, and $\mathbf{L}$ is a square matrix with elements defined by

$$L_{ij} = d_i - w_{ij}. \tag{2.13}$$

with d_i being the degree of veretx i, defined as the sum of the values of the edges connected

to that vertex, $d_i = \sum_j w_{ij}$.

If we identify the elements w_{ij} as defining an adjacency matrix $\mathbf{W}$, then the discrete Laplacian in quadratic form for a vector input (x) can be written as

$$\mathbf{x}^t\mathbf{L}\mathbf{x} = \sum_{i\sim j} w_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2 \tag{2.14}$$

$$= \sum_{i\sim j} w_{ij}\mathbf{x}_i^2 + \sum_{i\sim j} w_{ij}\mathbf{x}_j^2 - \sum_{i\sim j} 2w_{ij}\mathbf{x}_i\mathbf{x}_j \tag{2.15}$$

$$= \sum_{i\sim j} \mathbf{d}_j\mathbf{x}_i^2 + \sum_{i\sim j} \mathbf{d}_i\mathbf{x}_j^2 - 2\sum_{i\sim j} w_{ij}\mathbf{x}_i\mathbf{x}_j \tag{2.16}$$

$$= \frac{1}{2}(\mathbf{x}^t\mathbf{D}\mathbf{x} + \mathbf{x}^t\mathbf{D}\mathbf{x} - 2\mathbf{x}^t\mathbf{W}\mathbf{x}) \tag{2.17}$$

$$= \mathbf{x}^t(\mathbf{D} - \mathbf{W})\mathbf{x}. \tag{2.18}$$

Where $\mathbf{D}$ is the degree diagonal matrix and $\mathbf{W}$ is a matrix encoding w_{ij} values. The expression is divided by half in the thrid step as $\sum_{i\sim j} = \frac{1}{2}\sum_i \sum_j$, so we do not count repeated maps in the permutation sum ($\because \mathbf{D}_{ij} = \mathbf{D}_{ji}$). The previous derivation implies the following well-known relationship for the *Laplacian matrix*

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{2.19}$$

where $\mathbf{D}$ is the diagonal degree matrix [53]. The graph Laplacian $\mathbf{L}$ has many useful mathematical and descriptive properties. It is a symmetric matrix, which means the eigenvectors and eigenvalues are real numbers according to the spectral theorem.

2.7.2 Laplacian Embedding

The generalized notion of embedding the structural information of an input on an Eucledian space through the Laplacian operator is termed as *Laplacian embedding*. Consider a graph $\mathcal{G} = (V, E)$ with the adjacency matrix $\mathbf{A}$ and Laplacian matrix L, the objective is to embed $\mathcal{G}$ on a real number space $\mathcal{X} \in \mathbb{R}^k$ for some dimension k, such that the transformed coordinates of the connected nodes ($\boldsymbol{x}_i \in \mathcal{X}$) are close to each other and disconnected nodes placed further. For nodes, this objective can be set up as a minimization problem as in equation 2.20 shown below

$$\underset{\substack{\boldsymbol{x}\in\mathcal{X}\\|\boldsymbol{x}|=1}}{\arg\min} \sum_{i\sim j} A_{ij}(x_i - x_j)^2 = \underset{\substack{\boldsymbol{x}\in\mathcal{X}\\|\boldsymbol{x}|=1}}{\arg\min}\, \boldsymbol{x}^t\mathbf{L}\boldsymbol{x}. \tag{2.20}$$

The left hand side of equation 2.20 is the *Dirchlet Sum* of the L. If nodes i and j are connected, the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ should be minimized. There is a normalization constraint to account for any arbitrary scaling [56]. This function can be rewritten as the right hand side of the equation 2.20 in terms of the Laplacian $\mathbf{L}$. We can solve this optimization problem in a number of ways but a simple way is to use Lagrange multipliers [56]. If we consider a multiplier ϕ, we can formulate equation 2.20 as a lagrangian with normalization constraint for input $\boldsymbol{x}^t\boldsymbol{x} = 1$ as

$$G(\boldsymbol{x}; \phi) = \boldsymbol{x}^t\mathbf{L}\boldsymbol{x} - \phi(\boldsymbol{x}^t\boldsymbol{x} - 1) \tag{2.21}$$

If we minimize this expression with respect to x we get

$$\frac{\partial G(x;\phi)}{\partial x} = 2\mathbf{L}\boldsymbol{x} - 2\phi\boldsymbol{x} = 0 \implies \mathbf{L}\boldsymbol{x} = \phi\boldsymbol{x}. \tag{2.22}$$

Hence, if we expand out the Laplacian form of the Dirichlet sum 2.20 with input as an eigenvector ψ with eigenvalue λ, we see the form $\boldsymbol{x}^t\mathbf{L}\boldsymbol{x} = \boldsymbol{x}^t\lambda\boldsymbol{x} = \lambda\boldsymbol{x}^t\boldsymbol{x} = \lambda$. As the eigenvalues of the discrete Laplacian $\mathbf{L}$ are always positive with the smallest eigenvalue being 0 [53], we pick the lowest eigenvalue which is not zero. Since, this eigenvalue minimizes equation 2.20, this eigenvector *embeds* the graph $\mathcal{G}$ in these eigen-vectors of the Laplacian $\mathbf{L}$, hence the term Laplacian embedding. For example, the points in a graph could be 'ordered' in one dimension using the projected value of each point into the second eigenvector.

2.8 Diffusion Maps

Diffusion maps, developed by Coifman and Lafon [27], is a dimensionality reduction technique based on Laplacian embedding discussed in previous sections. Consider a high dimensional dataset $\mathcal{X}$ representing a complex system. In this study, $\mathcal{H} \in \mathbb{R}^{3N}$, is a set of n coordinate points defined in equation 1.5. The goal is to detect a subspace $\mathcal{L} \in \mathbb{R}$ that correlates with perturbations in $\mathcal{H}$ and $dim(\mathcal{L}) << dim(\mathcal{H})$. The starting point of diffusion maps is the diffusion equation (or the heat equation) defined as,

$$\frac{\partial f}{\partial t} = \alpha\Delta f, \tag{2.23}$$

where α is the diffusion constant and Δ is the Laplace operator. The key idea is to treat all of the data points as a computational graph and use tools from spectral graph theory to model a random walk using the diffusion equation. If f is a distribution over all points in the dataset holding the stationary probability of each state, $\frac{\partial f}{\partial t}$ would define the walk with the topology of the data described by the Laplace operator Δ. This Laplace operator is the key connection to the spectral graph theory techniques we will be utilizing for dimensionality reduction.

2.8.1 Stochastic process over dataset

Consider a stochastic process over the points in $\mathcal{H}$, defined as a random walk problem on a graph. The random walk is based on transition probabilities, encoded in a right stochastic matrix $\mathbf{M}$. The specifics of the random walk operation are discussed in section 2.8.3. The idea is that transitioning to points that are *closer*, imply a higher transition probability. First, we define a matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ mapping N samples of points in $\mathcal{H}$ based on a *distance* metric which is meant to capture differences in properties that are inherent to $\mathcal{H}$. For the model system of Lennard Jones clusters which is the focus of this study, the distance should capture the states in $\mathcal{X}$ caused by the statistical mechanics of these clusters. The distance metrics used for this study are discussed in the following sections. For now, we denote an arbitrary distance matrix $\mathbf{D}$, mapping points in a set $\mathcal{H}$ based on a distance function d as

$$\mathbf{D}_{ij} = d(\mathcal{H}_i, \mathcal{H}_j). \tag{2.24}$$

To convert this distance matrix into a stochastic process, first we convert $\mathbf{D}$ into a gaussian kernel $\mathbf{K}$, with bandwidth ϵ as

$$\mathbf{K}_{ij} = e^{\frac{-\mathbf{D}_{ij}^2}{2\epsilon^2}}. \tag{2.25}$$

This kernel matrix is normalized to make a *Markov matrix* $\mathbf{M}$. Consider a diagonal matrix $\tilde{K} = diag\left(\sum_j K_{ij}\right)$, which holds the sum of each row in the corresponding diagonal element. The kernel matrix $\mathbf{K}$ is row normalized as

$$\mathbf{M} = \tilde{\mathbf{K}}^{-1}\mathbf{K} \tag{2.26}$$

This matrix is *positive*, *symmetric* and *normalized*, which can be interpreted as a stochastic operator encoding transition probabilities. In a sense, this is also in the form of a graph adjacency matrix, allowing us to interpret the dataset as a graphical structure and by using ideas from spectral graph theory, we can extract geometrical properties of the dataset and project them on a lower dimensional Eucledian space.

2.8.2 Spectral analysis of stochastic operator

The Markov matrix $\mathbf{M}$ defined in the previous section is positive and symmetric, which mean this matrix is *positive semi definite*. This means that the eigenvalues of this matrix are real and positive and the eigenvectors assosiated with these eigenvalues are also in real number space. Since, we are interpreting this Markov matrix as an adjacency matrix of a graph embedding transition probabilities, we can use Laplacian embedding from section 2.7.1 to embed the information in $\mathbf{M}$ in an Eucledian space such that closer nodes with higher transition probability are placed closer. One important thing to note is that the

degree vector of $\mathbf{M}$ is the $\mathbb{1}$ vector as the Markov matrix is normalized. This means that we can expand the eigenvalue problem of Laplacian $\mathbf{L}$ of $\mathbf{M}$ as

$$\mathbf{L}\psi = \lambda\psi$$

$$(\mathbf{D} - \mathbf{M})\psi = \lambda\psi$$

$$\mathbb{1}\psi - \mathbf{M}\psi = \lambda\psi$$

$$\mathbf{M}\psi = (1 - \lambda)\psi.$$

$\mathbf{M}$ is a *positive semi-definite* matrix with positive eigenvalues. Based on the derivation above, we know that the maximum eigenvalue of $\mathbf{M}$ is 1 as the smallest value of λ is 0, making $(1 - \lambda = 1)$. Hence, the eigenvalues of $\mathbf{M}$ are in the range $\{0, 1\}$. This property can be exploited for reducing dimensions for the stochastic system represented by the Markov matrix $\mathbf{M}$ as shown in the following section.

2.8.3 Dimensionality Reduction using Diffusion Maps

This section lays out the dimensionality reduction process by using diffusion maps. Consider ρ to be the stationary distribution over a high dimensional input $\mathcal{H}$, the distribution can be updated for next step according to diffusion equation as

$$\rho_{t+1} = \mathbf{M}\rho_t.$$

Notice that if the distribution was the first eigenvector, we would have the trivial eigenvalue of $\mathbf{M}$ as 1. We can see that the physical meaning of this eigenvalue can be seen here

as $\psi_{t+1} = \mathbf{M}\psi_t = \psi_t$, which is implying transitioning to the same state and is virtually meaningless to define a walk. This is why we discard the first trivial eigenvector. We can look at $\mathbf{M}$ as a linear transformation with a basis eigenspace Ψ. Hence, the stationary distribution can be represented as a linear combination of the eigenvectors as

$$\rho = \sum_i \alpha_i \psi^{(i)}.$$

If we plug this in the random walk, we get

$$\mathbf{M}\rho_t = \mathbf{M} \sum_i \lambda_i \psi^{(i)}.$$

Since, the eigenvalues are in the range of $0 \leq \lambda \leq 1$, we can approximate the original distribution ρ, by considering enough eigenvectors in the sum that converge to ρ. This speaks to the dimensionality reduction step where we need to only sample a few eigenvectors which converge to the original distribution of the data. Once a Markov transition matrix is defined for $\mathcal{H}$, the reduced coordinates can be detected by observing the spectral decay $\{\lambda_1 = 1 > \lambda_2 > \lambda_3 \ldots \lambda_n\}$ and picking the top k eigenvectors corresponding with the non trivial eigenvalues with the least gap. These eigenvectors are also referred to as the *diffusion coordinates* of $\mathcal{H}$. The distance metrics implemented in this study are best described in section 2.12 that best describe the transition probabilities among the states in $\mathcal{H}$.

2.9 Hausdorff Distance

To compute Hausdorff distance, the Euclidean distance from each individual particle in one configuration to all the particles in the other configuration is computed, and the shortest of these distances is determined. The greatest of these distances gives the Hausdorff distance between the two configurations. Mathematically, this can be represented as ([27])

$$d_H = \max\{\max_{1\leq i\leq N}\min_{1\leq j\leq N}||\mathcal{X}_1^i - \mathcal{X}_2^j||, \max_{1\leq i\leq N}\min_{1\leq j\leq N}||\mathcal{X}_2^j - \mathcal{X}_1^i||\} \tag{2.27}$$

where, S_1^i and S_2^j are the positions of i^{th} and j^{th} particles in the configurations S_1 and S_2, respectively.

2.10 Mayer f-Bond Distance

The second distance metric is difference in the Mayer f-bond from statistical mechanics [57]. The mayer f-bond distance is defined as

$$F = 1 - exp(-u^*), \tag{2.28}$$

and the distance between two configurations was defined as the absolute difference between their respective F values as

$$d_M = |F_1 - F_2|. \tag{2.29}$$

2.11 IsoRank Distance

IsoRank [55] is an algorithm to detect the best alignment between two protein networks. IsoRank was first used as a distance metric for DMap calculations for molecular clusters by Long and Ferguson [34], to counter the lack of an invariant basis in the dataset when the particles are indistinguishable. This distance metric uses two adjacency matrices. The first matrix ($\mathbf{R}$) holds the Euclidean distance $\mathbf{R}_{ij} = ||x_i - x_j||^2$ between particles i and j while the second matrix ($\mathbf{G}$) is a binary connectivity matrix representing the existence of a bond between the two particles which can be defined based on a cutoff distance. IsoRank realizes the best alignment by re-arranging the particle indices to address the index permutation symmetries. For snapshots i and j, let $\mathbf{R}_i^*$, $\mathbf{G}_i^*, \mathbf{R}_j^*$, $\mathbf{G}_j^*$ be the best alignment detected by IsoRank. Then, the distance kernel for building the diffusion mapping is calculated by the sum of the absolute differences between the inter-particle distances as

$$d_I(ij) = \sum_{p=1}^{N} \sum_{q=p+1}^{N} |\mathbf{R}_i^*(p,q) \circ \mathbf{G}_i^*(p,q) - \mathbf{R}_j(p,q) \circ \mathbf{G}_j(p,q)| \tag{2.30}$$

With $\circ$ being the element-wise (hadamard) product. The algorithm clips away particles that are far away by multiplying the distance with the binary bond value. This also tends to generate a discrete subspace due to the sparse nature of $\mathbf{G}$.

2.12 Spectral Distance

This section lays out a distance metric for quantifying structural similarity of atomic clusters. The distance should capture the structural similarity between two clusters and remain invariant to translation, rotation and index permutation. The distance is then com-

puted as the absolute difference between the eigenvalue matrices as defined in Eq. (2.31). Since the result will be a diagonal matrix, the trace would be the summation of all the elements. The graphical representation is already translational and rotational invariant but not index-permutation invariant. We propose the use of spectral distance for the diffusion map algorithm. For an adjacency matrix $\mathbf{A}$, the node indices can be swapped using permutation operator ($\mathcal{P}$) as $\mathcal{P}\{\mathbf{A}\} \longrightarrow \mathbf{PAP}^{-1}$, where $\mathbf{P}$ is a permutation matrix that switches graph indices. For an adjacency matrix $\mathbf{A}$, the eigenvalues are the solution of the equation $det(\lambda\mathbf{I} - \mathbf{A}) = 0$. If we subject $\mathbf{A}$ to a transformation which swaps particle indexes using permutation matrices, it can be seen that the eigenvalues are invariant to the permutation transformation.

$$\begin{aligned}
\det(\lambda\mathbf{I} - \mathcal{P}\{\mathbf{A}\}) &= \det(\lambda\mathbf{I} - \mathbf{PAP}^{-1}) \\
&= \det(\mathbf{P}(\lambda\mathbf{I} - \mathbf{A})\mathbf{P}^{-1}) \\
&= \det(\mathbf{P})\det(\lambda\mathbf{I} - \mathbf{A})\det(\mathbf{P}^{-1}) \\
&= \det(\lambda\mathbf{I} - \mathbf{A}).
\end{aligned}$$

In addition to the structural implications of the eigenvalues $\{\lambda_1 > \lambda_2 > \lambda_3 \ldots \lambda_n\}$ and eigenvectors, this invariant nature makes them a good variable for graph similarity measure. Consider two clusters with adjacency matrices A_1 and A_2. The eigen decomposition of the two vectors yield $\Psi_1\Lambda_1\Psi_1^t$ and $\Psi_2\Lambda_2\Psi_2^t$, where Λ is a diagonal matrix holding the eigenvalues and Ψ is the orthogonal matrix holding the respective eigen-vectors. We define the

eigenvector distance as

$$d_\Lambda = \frac{1}{N}\sum_{i=1}^{n}|\Lambda_1 - \Lambda_2|. \tag{2.31}$$

Although the eigenvalues are invariant, the eigenvectors can change their sign. Let $A = \Psi_1\Lambda\Psi_1^{-1}$ and $B = \mathcal{P}\{A\} = \Psi_2\Lambda\Psi_2^{-1}$ be two isomorphic graphs. This can be expanded as,

$$B = \Psi_2\Lambda\Psi_2^{-1} = (\mathbf{P}\Psi_1)\Lambda(\Psi_1^{-1}\mathbf{P}^{-1}). \tag{2.32}$$

For any eigenvectors ψ and $\phi = \mathcal{P}\{\psi\}$,

$$\psi\lambda\psi^{-1} = (\mathbf{P}\phi)\lambda(\phi^{-1}\mathbf{P}^{-1}) \implies \psi = \pm\mathbf{P}\phi.$$

This indicates that the Laplacian eigenvectors of co-spectral graphs can have a different sign. Hence, we take the absolute value before computing the distance. We use the eigenvector distance in L2 norm form.

$$d_\Psi = \frac{1}{2N}\sum_{i=1}^{n}(|\Psi_i| - |\Phi_i|)(|\Psi_i| - |\Phi_i|)^t \tag{2.33}$$

For this study, we employ eigenvalue and eigenvector distance for both R and G representations of the graph to get both discrete and continuous representations of the structure in the data.

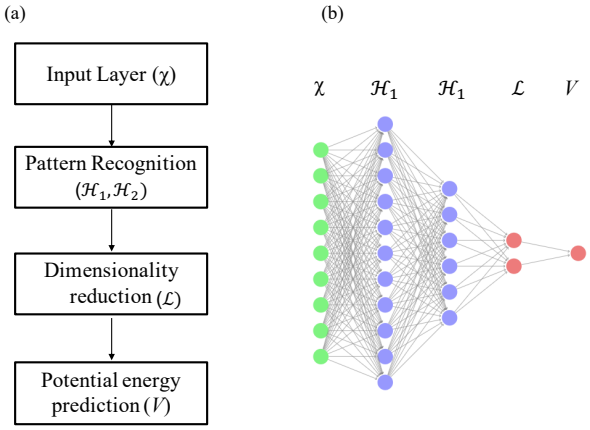


Figure 2.7: The structure of the neural network. $\mathcal{X}$ holds the coordinates of the three particles. $\mathcal{H}_1$ and $\mathcal{H}_2$ are hidden layers for pattern recognition. $\mathcal{L}$ is the layer that represents the reduced dimension which is used to predict the total potential energy (u^*) of the state. Since this is a case of supervised learning, the dimensionality reduction mapping ($\mathcal{X} \rightarrow \mathcal{H}_1 \rightarrow \mathcal{H}_2 \rightarrow \mathcal{L}$) mapping should steer towards a space representing the coordinates and preserving mapping to potential energy at the same time.

2.13 Deep Learning

Deep Learning [37] refers to a set of techniques that involve defining parametrized composite functions, and subjecting them to an optimization objective to learn a distribution over the dataset. The composite functions act as layers of this model and the large number of these functions give meaning to the term *deep*, as each function is composed of many other parametrized functions. These functions are called multi layerd perceptrons or deep neural networks.

2.13.1 Classical neural networks

Classical neural networks [58] are defined as k-partite graphs which represent non-linear transformations. The nodes of the graph may or may not be fully connected. The first layer is the input to the network. To propagate through the network, we transform the input based on the bond strength between the nodes (weights). Let the input to the

network be represented as $\vec{x}$, and the transformation matrix $\hat{W}$ hold the weights between the connections. The transformation for a layer L can be modelled as:

$$L_{\hat{W}}(\vec{x}) = f_\sigma(\hat{W}\vec{x} + \vec{b}) \tag{2.34}$$

The bias vector ($\vec{b}$) acts as the intercept of the linear model. The model's output is passed into a logistic function,

$$f_\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{2.35}$$

for non-linearity. If there are multiple layers, this output will be the input to the next transformation and hence, a neural network can be represented as:

$$N(\vec{x}) = L_n \circ L_{n-1} \circ .....L_2 \circ L_1(\vec{x}) \tag{2.36}$$

This function can be used to model a variety of regression and classification problems. The weights and the biases act as tunable parameters which can be adjusted to compute the desired result. This fine-tuning process is termed as *training* the network and is set up as an optimization problem. The training is carried out by an algorithm called *backpropogation*. The backpropogation algorithm minimizes the divergence (also called *loss*) between the predicted and original values. The loss function can be thought of a measure of accuracy of the model. Mean-squared error, defined as follows, is a common choice for loss function.

$$\mathcal{L} = ||N_{W,b}(\vec{x}) - y(\vec{x})||^2 \tag{2.37}$$

This loss is subjected to an optimization problem as

$$\arg\min_{W,b} \mathcal{L}(N_{W,b}(\vec{x}), y(\vec{x})) \longrightarrow \arg\min_{W,b} ||N_{W,b}(\vec{x}) - y(\vec{x})||^2 \tag{2.38}$$

and the training is carried out by updating the weights and biases such that Eq. (2.37) is minimized according to

$$\hat{W} = \hat{W} - \eta \frac{\partial \mathcal{L}}{\partial \hat{W}}; \quad \vec{b} = \vec{b} - \eta \frac{\partial \mathcal{L}}{\partial \vec{b}} \tag{2.39}$$

The derivatives are calculated via the chain rule as it is a sequence of composite functions or layers.

2.13.2 Convolutional Neural Networks (CNNs)

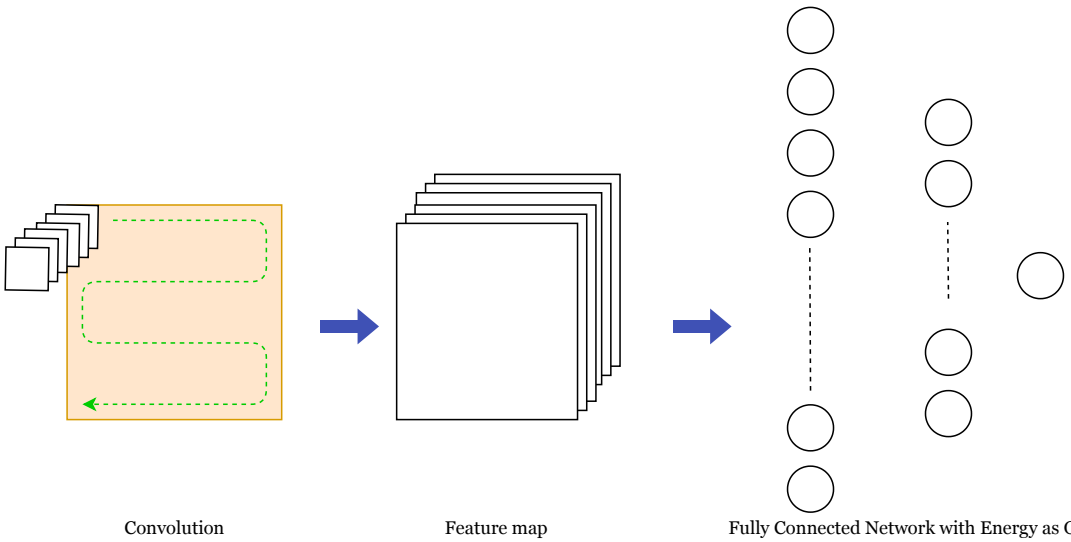


Figure 2.8: Visualizing the Convolutional Neural Network architecture. The input adjacency matrix is subject to a convolution operation using 16 filters resulting in 16 feature maps which are then flattened and passed trough a fully connected neural network.

Convolutional Neural Networks (CNN's) [37] are primarily used for feature extraction in images and signals. The idea is to scan the input locally using convolution filters and

generating a new feature map that can be used for pattern recognition. This technique has been very successful in extracting local features in images such as facial recognition, image segmenting and speech recognition in audio signals. The feature extraction approach using convolution operation can be also applied on the adjacency matrices representing atomic clusters. Carrasquilla and Melko, [44] used a similar idea for scanning spin configurations of atoms and classifying the input to two phases of matter. For this case, the scanning is done on the connectivity matrix and the output is the potential energy which makes this a regression problem. Since the adjacency matrices explicitly encode the graphical structure of the cluster, the convolution operation should recognise local connectivity patterns within the cluster. For an N particle cluster, the input to the CNN would be the $N \times N$ adjacency matrix and the convolution operation would scan the input with stride (step size) as 1. After the feature map is built, the output can be flattened to a 1D vector and then passed through classic fully connected layers (section 2.13.1) outputting the predicted energy. Considering the input A, the first pass of the CNN would be the convolution operation using a weight kernel w_k, which is a square matrix of size n_w. For non-linearity, the output is subject to a logistic sigmoid function ($\sigma(x) = \frac{1}{1+e^{-x}}$). Collectively the first step is shown as below.

$$A \longrightarrow \sigma(A * w_k) = \sigma\left(\sum_{k=1}^{h}\sum_{l=1}^{n_k}(\mathbf{A}_{i+k-1,j+l-1} \cdot \mathbf{w}_{k,l})\right) \tag{2.40}$$

The output of the convolution pass is a matrix (also called a feature map). If the input is of size $N \times N$, kernel size being K, size of stride (scanning step size) being S and padding magnitude being p, then the size of an output tensor is

$$N_{\text{out}} = \left\lfloor \frac{N - K + 2P}{S} \right\rfloor + 1. \tag{2.41}$$

For this work 16 convolution filters were used and hence the output is a set of 16 feature maps. After the convolution step, a *max-pooling* operation is applied where a filter is applied to reduce the dimensionality of the feature map. The max pooling step scans the input with a pooling kernel (say size $k \times k$), with the output being the maximum input value within the respective feature space the kernel is acting on. The output is then collectively flattened to a 1D vector and the successive layers are simply fully connected.

2.13.3 Dimensionality Reduction Using Deep Learning

We define the input to the network as our higher dimensional space $\mathcal{H}$. We define the predecessor layer of the final layer to be the reduced space $\mathcal{R}$. The mappling between $\mathcal{R}$ and U_{pred} is linear as follows

$$U_{pred} = \theta_1 r_1 + \theta_2 r_2 + \theta_3 r_3. \tag{2.42}$$

Where $\{r_1, r_2, r_3\} \in \mathcal{R}$ are the reduced coordinates with $\{\theta_1, \theta_2, \theta_3\}$ being the respective parameters. The training objective is set up as minimizing the divergence between the distributions between predicted energies and ground-truth energies. This study employs the L_2 norm as the loss objective

$$\mathcal{L} = ||U_{pred}(\vec{x}) - U_{true}||^2. \tag{2.43}$$

Where U_{true} are the true energy values taken from the dataset. This loss is subjected to an optimization problem as

$$\arg\min_{W,b} \mathcal{L}(U_{pred}(\vec{x}), U_{true}) \longrightarrow \arg\min_{W,b} ||U_{pred}(\vec{x}) - U_{true}||^2 \tag{2.44}$$

This network has a deep architecture with hidden layers. If the network converges, $\mathcal{R}$ will embedd the following mapping

$$\forall x \in \mathcal{X}, \exists r \in \mathcal{R} \quad \textit{such that } u^*(x) = u^*(r)$$

Hence, the variables in $\mathcal{R}$ can be defined as a lower dimensional embedding of $\mathcal{X}$.

3 Experiments, Results and Discussion

This chapter presents and discusses results for the methods for model reduction introduced in Chapter 2. First, we briefly review the nature of the experimental data and summarize the diffusion map (DMap) and neural net (NN) algorithms, and then present the results with discussion. The results are organized based on the two model systems, LJ3 and LJ13, with the latter system studied at two different temperatures. For LJ3 we use data generated at only one temperature, $T^* = 0.18$, as all structural motifs of this model are captured in a single Monte Carlo trajectory at this temperature, as discussed in section 2.4. For LJ13 we study data sets from two independent Monte Carlo trajectories. One set, generated at temperature $T^* = 0.28$ (Fig. 2.4a), contains structures that are primarily single solid- and liquid-like clusters, with relatively few *broken* configurations. The other set, generated at $T = 0.4$ (Fig. 2.4b) contains more *broken* clusters and even some vapor-like configurations. For both models, the embedding of the data in the subspaces generated by DMaps and NNs are presented. To define the markov matrix for diffusion maps, we use 4 distance metrics described in sections 2.12. For spectral distance, we use two adjacency matrix representations (based on matrix $\mathbf{R}$ or $\mathbf{G}$), and for each of these, two distance metrics (d_Λ, d_ψ) are used and are denoted as ($d_\Lambda^{(G)}, d_\Lambda^{(R)}, d_\psi^{(G)}$ and $d_\psi^{(R)}$), as described in section 2.12. We finally examine the distance kernels to see how the information encoded in the kernels translates to the information in the reduced subspaces to provide a better insight towards how these subspaces are generated.

3.1 Algorithm Summary

This section lays out the algorithms used to implement the main ideas presented in chapter 2. The two main dimensionality techniques used in this study are diffusion maps and neural network optimization. Since we are using three kinds of input (coordinate space $\mathcal{X}$, $\mathbf{G}$ and $\mathbf{R}$), we arbitrarily denote these high dimensional inputs $\mathcal{H}$ unless specified.

3.1.1 DMap Workflow

The diffusion maps algorithm is shown in Algorithm 1 with the theory backing it presented in section 2.8. Consider n samples of points in $\mathcal{H}$, which is derived from the coordinate space for clusters in $\mathcal{X} \in \mathbb{R}^{3N}$ with N being the number of particles in the cluster.

Algorithm 1: Diffusion Maps Algorithm

Input: Set of n points in the model space $\mathcal{H}$.

Result: Subspace $\mathcal{L} \in \mathbb{R}^k$ where $k << 3N$.

- Calculate distance matrix $\mathbf{d} \in \mathbb{R}^{n \times n}$ mapping pairs of points in $\mathcal{X}$.
- Choose kernel bandwidth δ from a $(ln(\delta), ln(\sum K))$ plot (see Fig. 3.1).
- Subject $\mathbf{d}$ to a kernel $\mathbf{K}$ such that $K_{ij} = e^{\frac{-d_{ij}^2}{2\delta}}$.
- Row normalize $\mathbf{K}$ to generate a stochastic markov matrix $\mathbf{M}$.
- Eigen-decompse $\mathbf{M} = \Psi^t \Lambda \Psi$, $\{\lambda_1 = 1 \geq \lambda_2 \ldots \lambda_n \in \Lambda\}$.
- Identify k principal eigenvalues excluding λ_1.
- Return $\mathcal{L} = \Psi_{2:k}$, where $\Psi_{2:k}$ refers to collecting 2^{nd} to k^{th} eigenvectors.

Before defining the Markov matrix, a value must be chosen for the kernel bandwidth parameter δ. Fig. 3.1 shows the log-log plot of the sum of the all the elements of the Gaussian kernel as a function of δ for the LJ3 data using the IsoRank distance metric. (All DMap analyses in this work, regardless of data set and distance metric employed, yielded plots qualitatively similar to this figure, so this is the only example shown.) The plot shows a sigmoidal behaviour that can be interpreted as follows. For low δ values, the scaled distances between pairs of data points are relatively large; therefore no points are well-connected and the kernel sum is small. Conversely, for high δ values, the scaled distances are relatively short and all points are well-connected in a pairwise sense, resulting in a large kernel sum. The linear region that bridges those two extremes is the region of interest, as it captures the

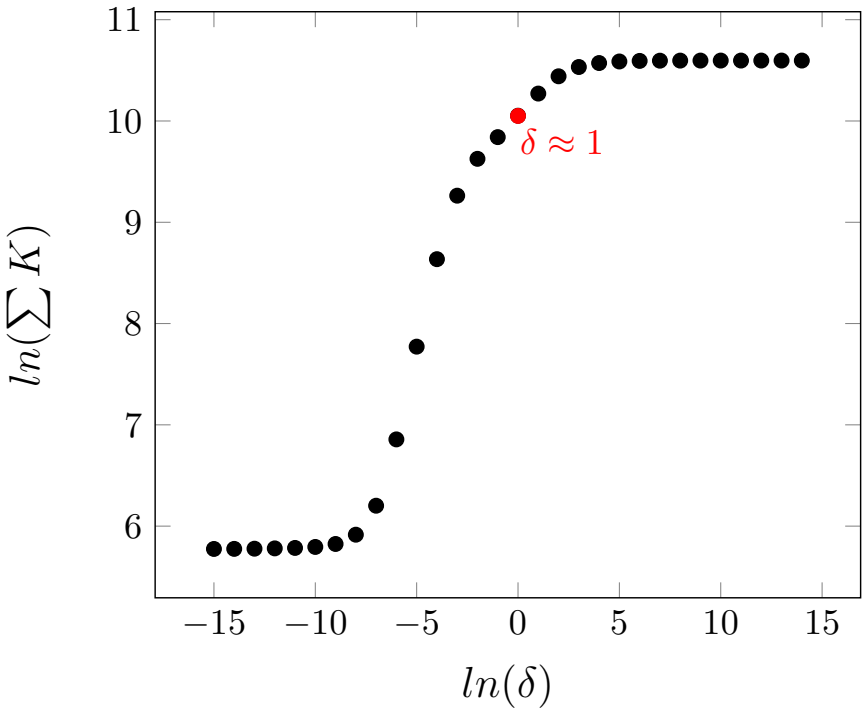


Figure 3.1: Identification of the kernel bandwidth for the IsoRank-based distance metric. The selected value of δ is highlighted.

natural connectivity of the data at intermediate scales, and the value of δ is selected from this region. After generating the subpsaces, we investigate what order parameters these spaces encode by coloring the subspace eigenvectors with structural variables (order parameters) discussed in Chapter 2. If these spaces exist, we should see a strong correlation between the diffusion spaces and the physical variables.

3.1.2 Neural Network Workflow

If we denote the input to a neural network as $\mathcal{H}$, the penultimate layer defined as the reduced layer $\mathcal{L}$, the potential energy prediction step is the output of the neural network model ($\mathcal{N}$) as shown in algorithm 2

Algorithm 2: Potential Energy prediction using NN models.

Input: Coordinate space $\mathcal{X}$.

Result: $\mathcal{L}$ and U_{pred}.

Initialize $y = \mathcal{X}$;

Initialize $\mathcal{L}$;

foreach *layer* l *in* $\mathcal{N}$ **do**

    $y = l(y)$;

    if l *is penultimate* **then**

        $\mathcal{L} = y$;

    end

end

$U_{pred} = y$;

return $\{\mathcal{L}, U_{pred}\}$

Where $l(\cdot)$ can be any type of layer model function with corresponding set of parameters listed in section 2.13. Algorithm 2 assumes variables to be mutable *i.e* be able to take up any shape or form of the assignment for simplicity. The following algorithm 3, describes the steps to train $\mathcal{N}$ and learn the lower representation $\mathcal{L}$ based on a cost function C. Assume the parameters for each layer are stored in Θ.

Algorithm 3: ADAM algorithm [59] for training neural networks.

Input: Coordinate space $\mathcal{X} \in \mathbb{R}^{3N}$ and potential energies $U \in \mathbb{R}$. β_1 and $\beta_2 \in [0, 1)$ are exponential decay rates for momentum estimates.

Initialize $U_{pred} = \mathcal{X}$;

Initialize $\mathcal{L}$;

foreach *point* x *in* $\mathcal{X}$ **do**

$\mathcal{L}, U_{pred} = \mathcal{N}(x)$;

$\nabla C = \left[\frac{\partial C(U_{pred}, U_x)}{\partial \theta}\right]_{\theta \in \Theta}$;

$m = \beta_1 \cdot m + (1 - \beta_1) \cdot \nabla C$;

$v = \beta_2 \cdot v + (1 - \beta_2) \cdot \nabla C^2$;

$\widehat{m} = m / (1 - \beta_1)$;

$\widehat{v} = v / (1 - \beta_2)$;

$\Theta = \Theta - \eta \cdot \widehat{m} / (\sqrt{\widehat{v}} + \epsilon)$;

end

return $\mathcal{L}$;

Algorithm 3, is the *Adam* optimizer [59], which has a controlled approach for updating weights by assigning momentum values for the learning rate η. Adam optimization systematically adjusts η in the training process based on the the gradients, which greatly stabilizes optimization [59]. Details behind calculating gradients for layers are shown in section 2.44. Once, the network learns a representation (*i.e* the model converges and stops updating weights with zero gradients), the parameters Θ would be adjusted to calculate $\mathcal{L}$ from $\mathcal{X}$ based on the constraint in statement 2.13.3.

3.2 Results for LJ3

This section presents subspaces generated by DMap and NN analysis for the three-particle LJ system at dimensionless temperature 0.18.

3.2.1 Diffusion Maps

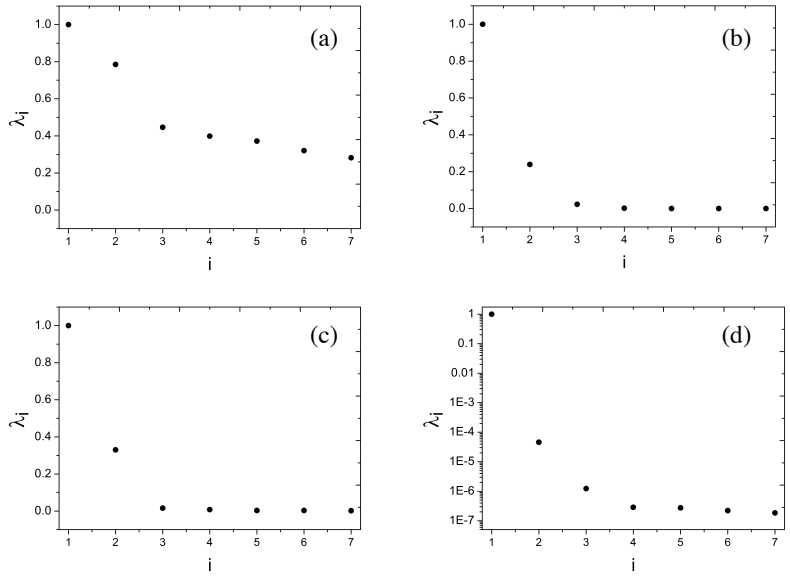


Figure 3.2: Eigenvalue spectra for the four distance metrics considered: (a) d_H, (b) d_M, (c) d_I, and (d) $d_\Lambda^{(R)}$.

The eigenvalue spectra for the four distance metrics used in this study are presented in Fig. 3.2. One way to determine the existence of a low-dimensional manifold and identify the number of dimensions in that manifold is to look for a break in the eigenvalue spectrum and count the number of eignevalues above the gap, ignoring the first eigenvalue. Following this approach, we see that the low-dimensional space for our system has at most two dimensions. Ferguson et. al [17] reported an alternative method by which the dimensionality is estimated by computing the slope of the linear region of the plot in Fig. 3.1 and doubling its value. Following this approach, we found that the dimensionality of our system should be at most

2.6. In summary, then, all results and interpretations indicate that no more than three dimensions should be required to adequately describe our system.

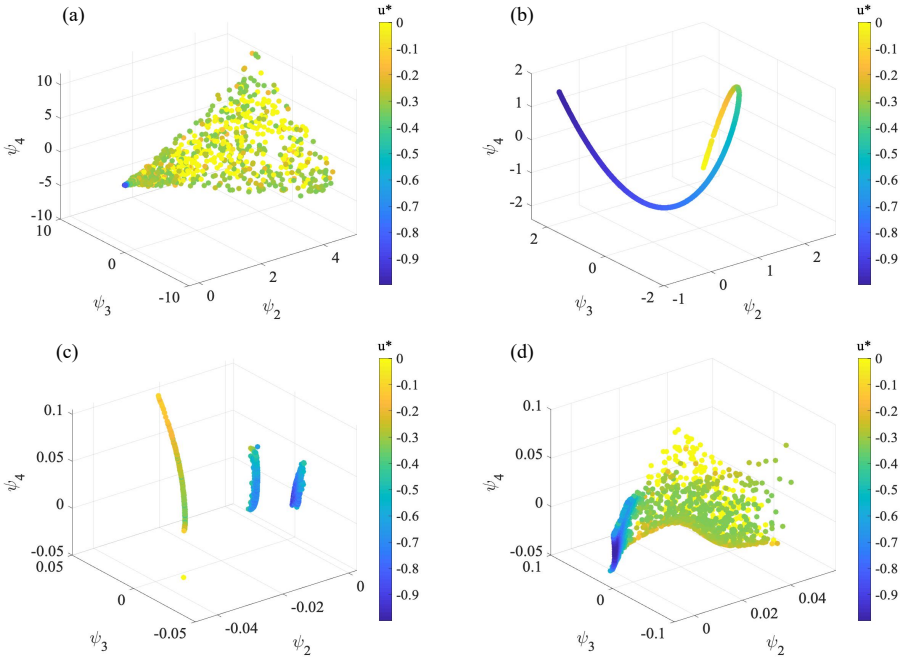


Figure 3.3: Subspaces generated by Diffusion Maps in three diffusion coordinates with distance metrics as (a) d_H (b) d_M (c) d_I (d) $d_\Lambda^{(R)}$. Data are colored by the potential energy for LJ3.

.

Subspaces generated by diffusion maps for LJ3 are shown in Fig. 3.3 in three dimensions. Fig. 3.3a, is the subspace generated with d_H, showing that first nontrivial eigenvector, ψ_2, segregates the configurations that have lower potential energy values (blue) from the ones with moderate (green) and high (yellow) values. The 2d surface is plotted in Fig. 3.4a. Fig. 3.3b, which is the space generated by using d_M distance metric, indicates that only one dimension is needed to describe the data and that ψ_2 is strongly correlated with potential energy, which is perhaps not surprising since the distance metric is based directly on the potential energy function. This can be seen better in 2 dimensions as shown in Fig. 3.4b. Although this distance metric creates a compact and smooth subspace for the four different

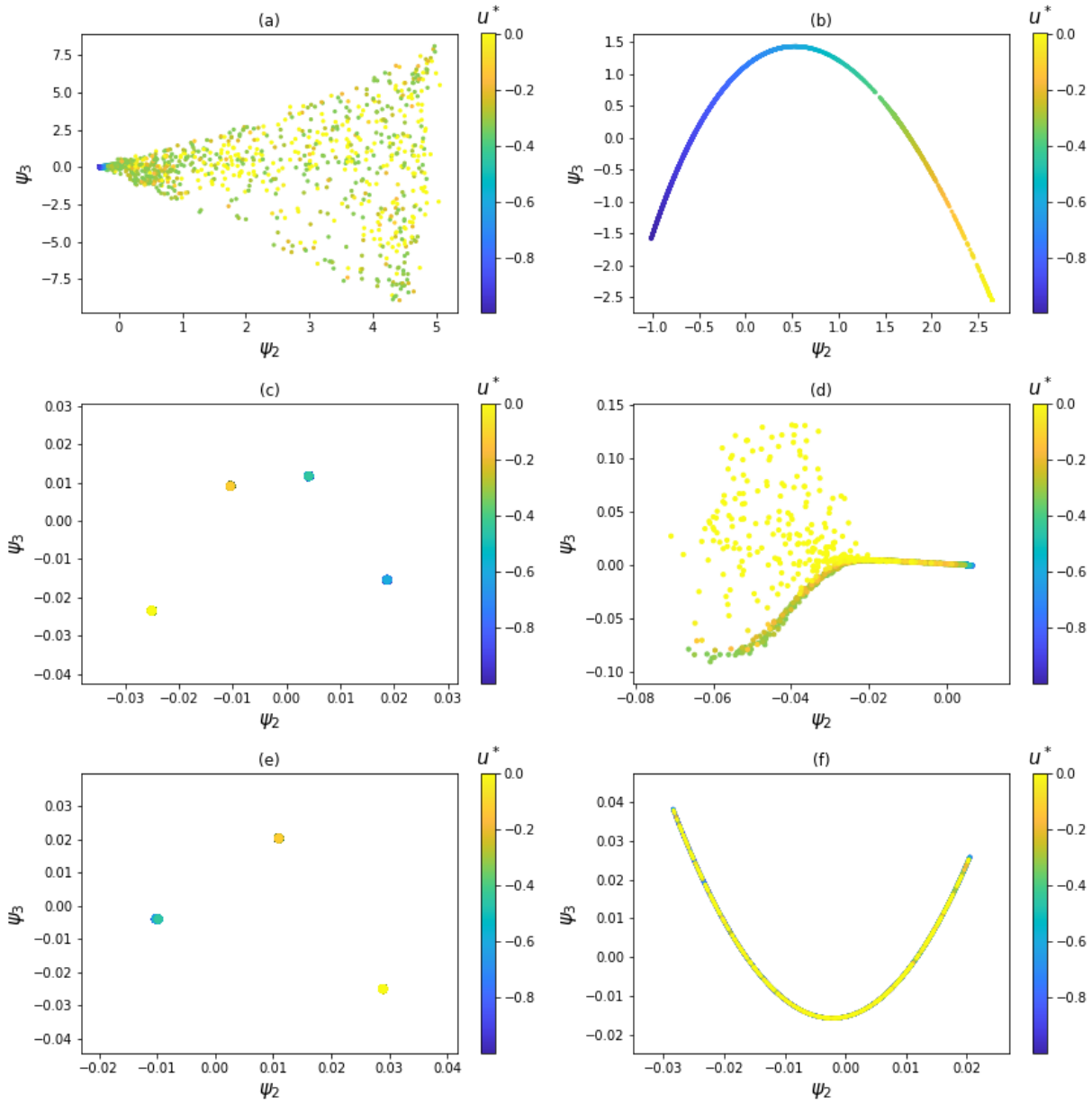


Figure 3.4: 2d Diffusion coordinates for LJ3 ($T^* = 0.18$) generated by (a) d_H (b) d_M (c) $d_\Lambda^{(G)}$ (d) $d_\Lambda^{(R)}$ (e) $d_\psi^{(G)}$ (f) $d_\psi^{(R)}$. Data colored by potential energy u^*.

structural motifs, the requirement of computing the potential energy is a major weakness, since the underlying potential energy function may not generally be known for a given empirical data set. Fig. 3.3c shows that employing d_I yields a result in which the data are separated into four distinct clusters, which correspond to to the four structural motifs (three, two, one, or no pairwise bonds) as shown in Fig. 3.6b. Interestingly, the correlation of these

structures with the potential energy values is much weaker than we originally anticipated as seen in Fig. 3.6a. Another fact to note is that diffusion coordinates with d_I reduced the fully disconnected configurations to one point as unbonded particles are clipped away by multiplication with zero, resulting in fully broken clusters being indistinguishable. Figure 3.3d is the three dimensional subspace generated by $d_\Lambda^{(R)}$, which shows the potential energy encoded in a manifold. The energy speration in higher energy values can be seen much better in two diffusion coordinates as shown in figure 3.5d.

Now we turn our attention to discuss specifically the effect of the family of spectral distance metrics introduced in this study. Fig. 3.5c and d are subspaces generated from the eigenvalue distances $d_\Lambda^{(G)}$ and $d_\Lambda^{(R)}$ respectively. The diffusion space generated by $d_\Lambda^{(G)}$ encoded number of bonds (n_b) very well in ψ_2 as seen in Fig. 3.5c but does not capture any variations in one of the four motiffs as it reduces all the variations to a single point. However, using $d_\Lambda^{(R)}$, yielded a continuous space which encoded r_g^2 very well as seen in Fig. 3.5d. Since u^* is not equivalent to r_g^2 or n_b, the energy encoded is not as clean in these coordinates as seen in figures 3.4c and d.

Figures 3.5e and f are subspaces generated from the eigenvector distances $d_\psi^{(G)}$ and $d_\psi^{(R)}$ respectively. The diffusion space generated by $d_\psi^{(G)}$ collapsed into three points correlating with the number of clusters as seen in Fig. 3.5e. One thing to note is that fully connected clusters and clusters with one broken bond are still singular clusters and it is interesting that $d_\psi^{(G)}$ segregated the data based on n_c instead of n_b. Although, $d_\psi^{(R)}$ embedded the data into a smooth one-dimensional curve, none of the physical variables correlated very well with psi_2, so the data here are shown colored by n_c.

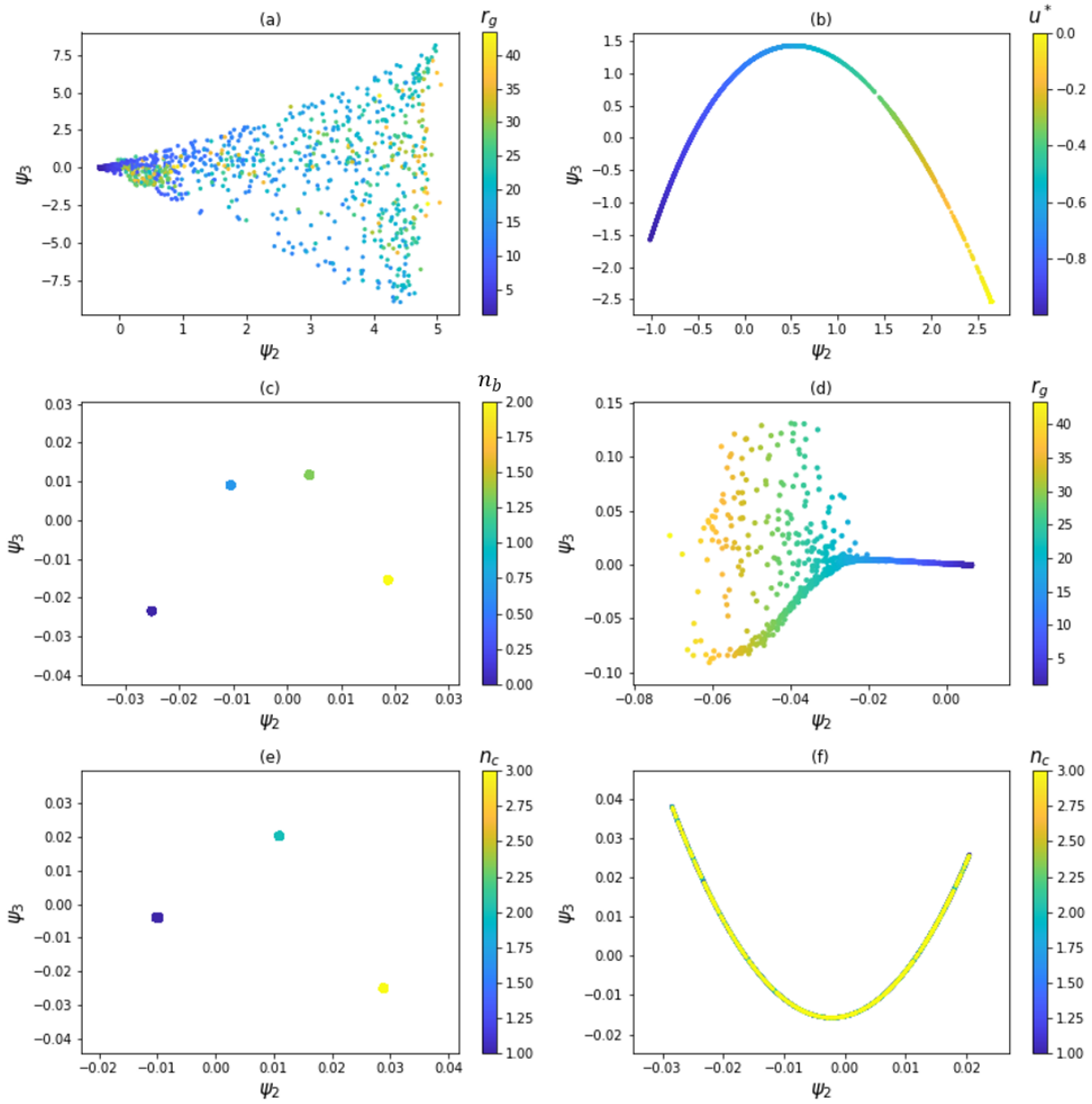


Figure 3.5: 2d Diffusion coordinates for LJ3 ($T^* = 0.18$) generated by (a) d_H (b) d_M (c) $d_\Lambda^{(G)}$ (d) $d_\Lambda^{(R)}$ (e) $d_\psi^{(G)}$ (f) $d_\psi^{(R)}$. Data colored by physical variable that seem to correlate best with the respective space.

3.2.2 Neural Networks

The dataset is split 60% for training, 30% for testing and 10% for validation. The network converged in about 10,000 epochs and encoded $\mathcal{H}$ in a two-dimensional linear space $\mathcal{L}$ to represent u^*, as seen in Fig. 3.7. The first variable, $\mathcal{L}_1$, captures most of the changes

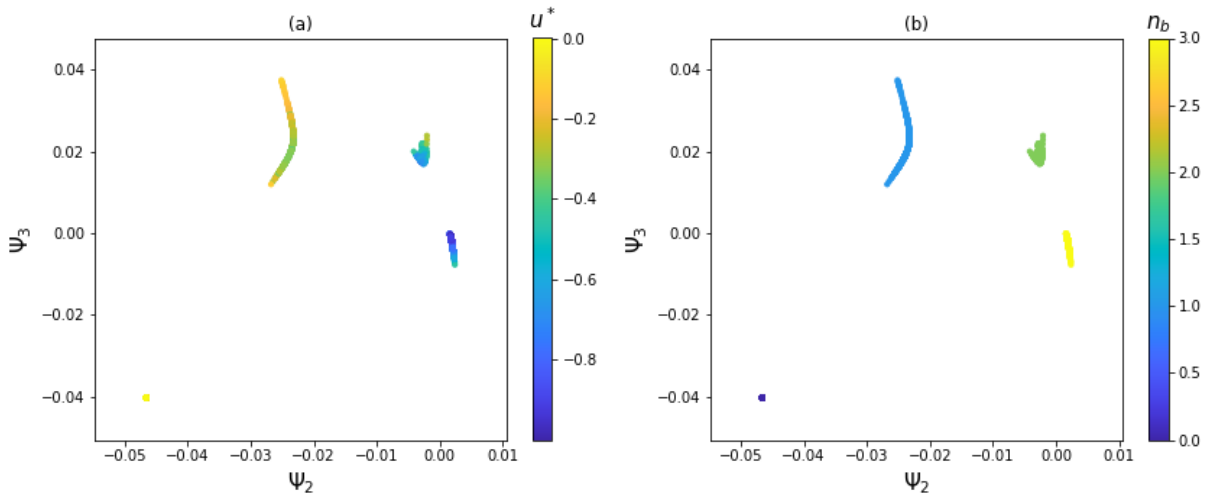


Figure 3.6: 2d Diffusion Coordinates generated using IsoRank based distance metric d_I introduced by Long and Ferguson [34].This space is colored by (a) u^* (b) n_b. We can see that the four point groups correspond to the four possible motifs.

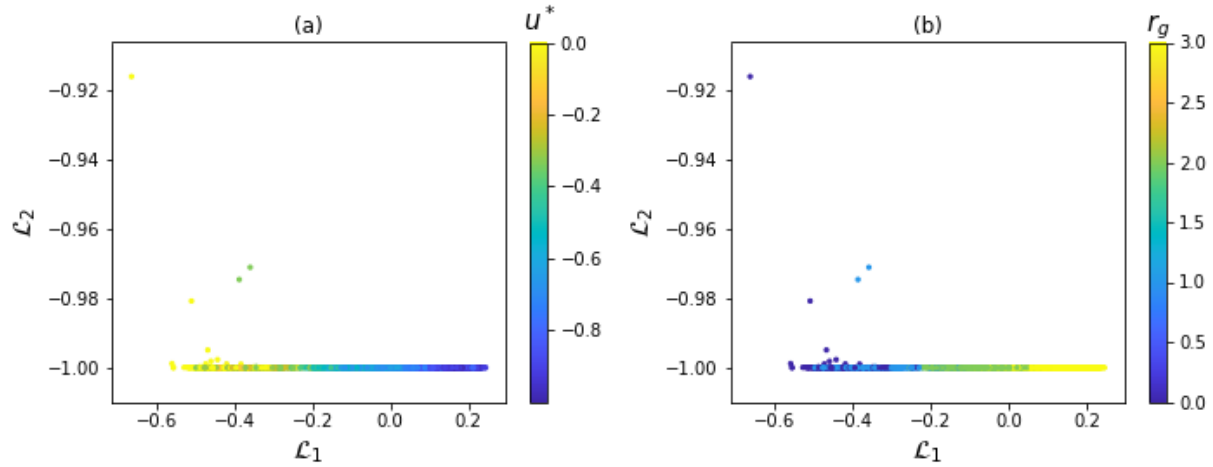


Figure 3.7: Reduced subspace space $(\mathcal{L}_1, \mathcal{L}_2)$ generated by the neural network colored by (a) u^* (b) r_g

in the low-energy structures with the second variable, $\mathcal{L}_2$, becoming important only for the higher-energy, broken configurations. The strong weakness of this technique is that this requires a key variable representing each point in the input for the network to learn. For our case, we use u^* as a constraint to control the dimensionality reduction step, but many data-sets may not have such information available.

3.3 Results for LJ13

This section presents subspaces generated by DMap and CNNs for the 13-particle LJ system sampled from two dimensionless temperatures, $T^* = 0.40$ and 0.28. For larger clusters, instead of n_b we use an equivalent variable connectivity c to avoid scaling to large numbers.

3.3.1 Diffusion Maps

The initial analysis (not shown) on this data set suggested a dimensionality of two or less for each distance metric, therefore we will focus on two-dimensional representations of the data. The energy distribution in the diffusion coordinates is visualized in Fig. 3.8 for $T^* = 0.4$ and 3.10 for $T^* = 0.28$. Fig. 3.8a is the subspace generated by mapping points with d_H which resulted in the same space as LJ3 with the first principle axis correlating best with r_g^2 as seen in 3.9a. Figure 3.8b, generated by d_M correlated with potential energy as seen in Fig. 3.9b which, as mentioned before is expected because mayer f-bond is an explicit function of u^*.

Figures 3.8c and d are the diffusion coordinate spaces generated by the eigenvalue distance (equation 2.31), correlating with connectivity and radius of gyration respectively similar to LJ3 as seen in figures 3.9 c and d respectively. Since connectivity is a discrete value, we can see gaps in the space similar to LJ3. Since the metric $d_Lambda^{(G)}$ is still based on discrete values, the results for that metric again show some discrete character, although not to the extent seen for LJ3 where the results collapsed to only four branches.. Although the subspace generated by $d_\Lambda^{(R)}$ for LJ13 as seen in Fig. 3.9d showed a similar space as LJ3,

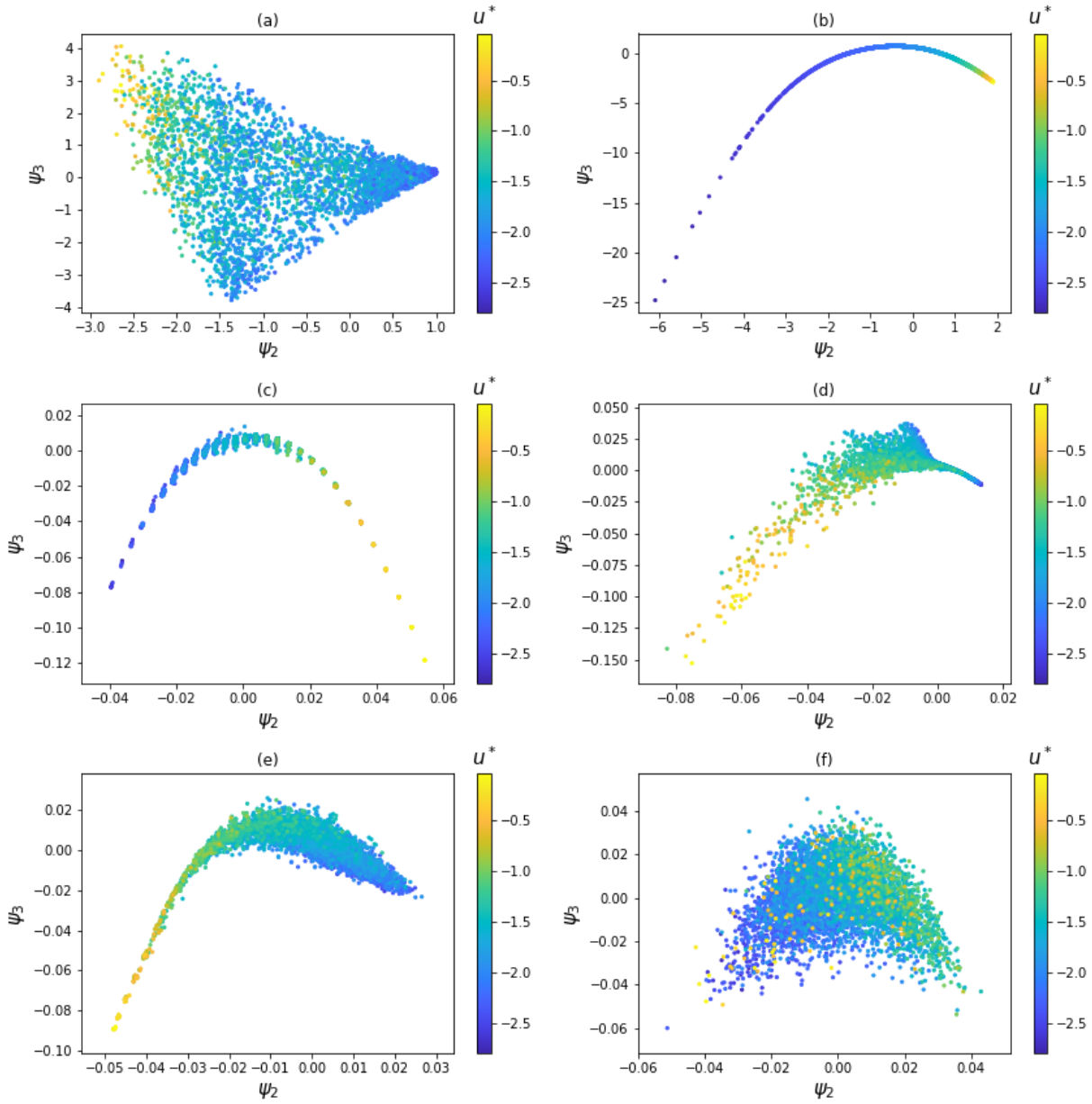


Figure 3.8: 2d Diffusion Coordinates generated using (a) d_H (b) d_M (c) $d_\Lambda^{(G)}$ (d) $d_\Lambda^{(R)}$ (e) $d_\psi^{(G)}$ (f) $d_\psi^{(R)}$. Colored by potential energy u^* with trajectories sampled from $T^* = 0.4$.

the third eigenvector ψ_3 did not show any symmetry. However, if we look at Fig. 3.9d, we can see that within the manifold itself, ψ_3 encodes the spread of u^* which is interesting as according to these results for $d_\Lambda^{(G)}$, ψ_2 encodes the physical variable with ψ_3 encoding the energy distribution at a specific value of r_g. These results are consistent with $T^* = 0.28$ as

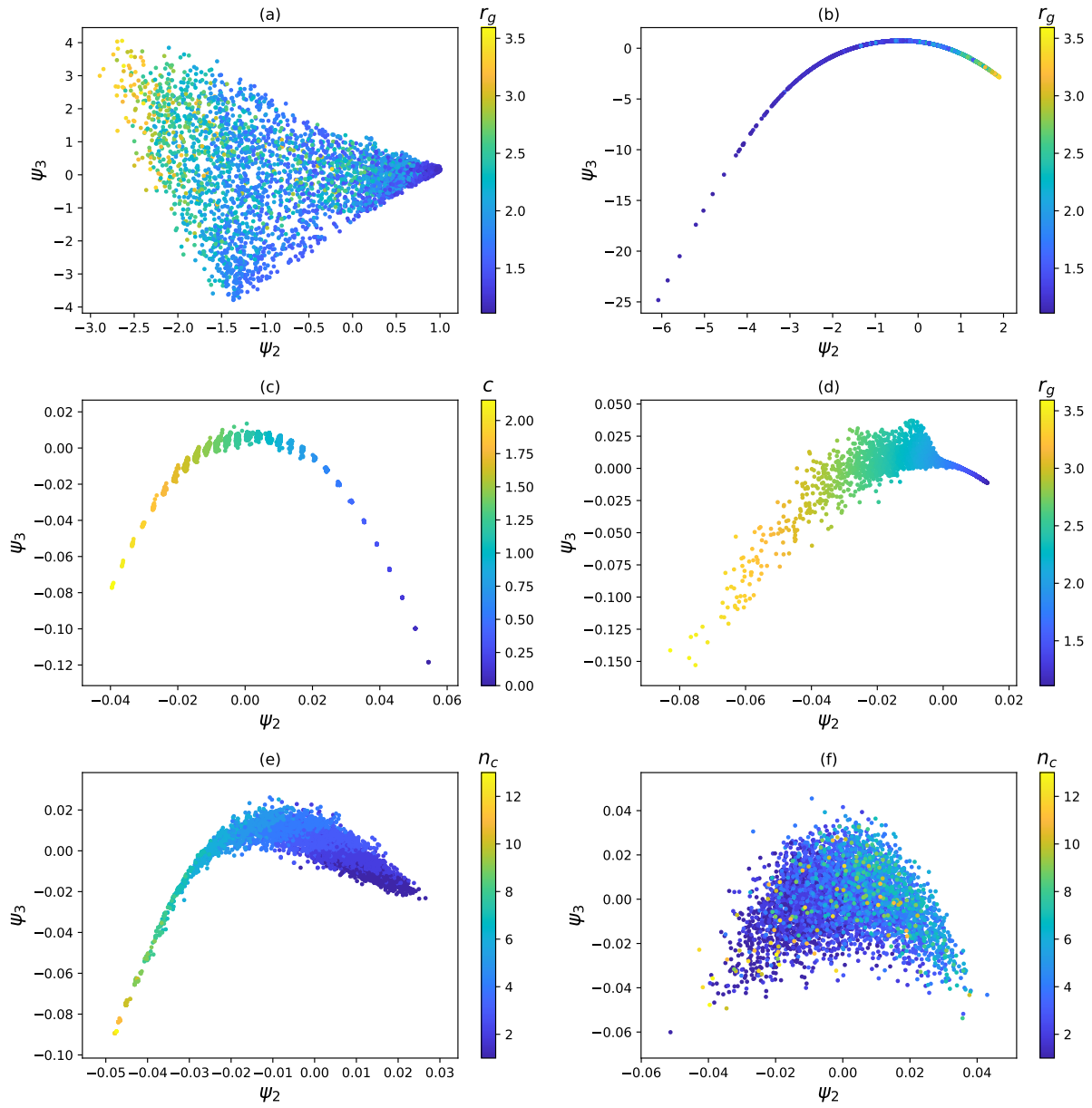


Figure 3.9: 2d Diffusion Coordinates generated using (a) d_H (b) d_M (c) $d_\Lambda^{(G)}$ (d) $d_\Lambda^{(R)}$ (e) $d_\psi^{(G)}$ (f) $d_\psi^{(R)}$. Colored by potential energy u^*. Colored with best correlated structural variable for trajectories sampled from $T^* = 0.4$.

seen in Fig. 3.11 but the shape of the spaces are significantly different. This is due to the fact that $T^* = 0.28$ primarily consisted of tightly packed clusters and very few cases of loose and broken clusters. One can infer that the probability of states at more *dense* regions of these diffusion coordinates is very high in given temperature. This makes sense as clusters

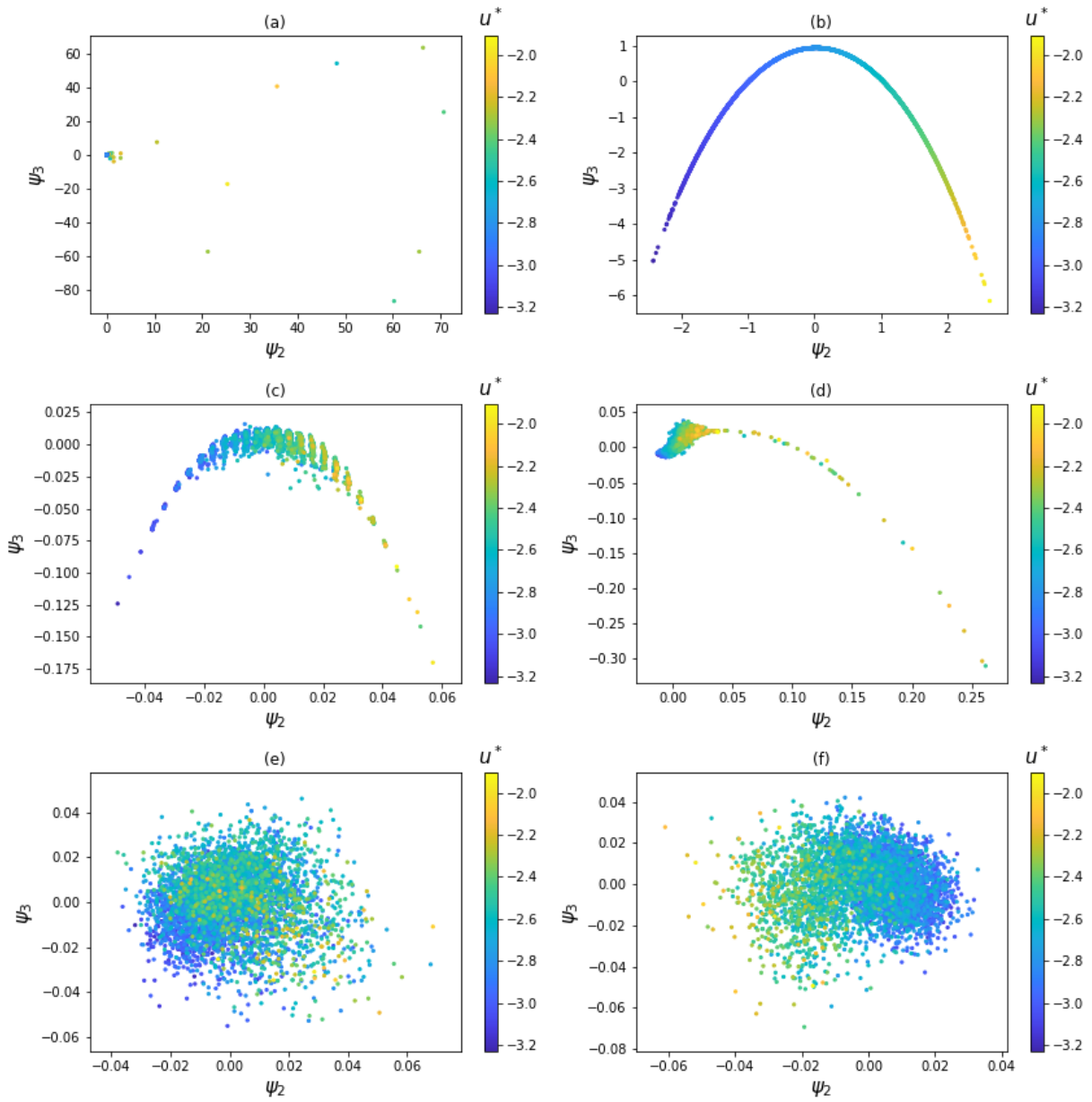


Figure 3.10: 2d Diffusion Coordinates generated using (a) d_H (b) d_M (c) $d_\Lambda^{(G)}$ (d) $d_\Lambda^{(R)}$ (e) $d_\psi^{(G)}$ (f) $d_\psi^{(R)}$. Colored by potential energy u^* with trajectories sampled from $T^* = 0.28$.

at lesser temperatures tend to be more stable with less fluctuations and that is why we see a more concentrated region for tightly packed (low r_g) clusters instead of a more *spread out* space as $T^* = 0.4$. This is most likely why Fig. 3.10a looks so sparse as compared to Fig. 3.8a. We can also infer from $d_\Lambda^{(G)}$ that even though states are tightly packed, there can exist

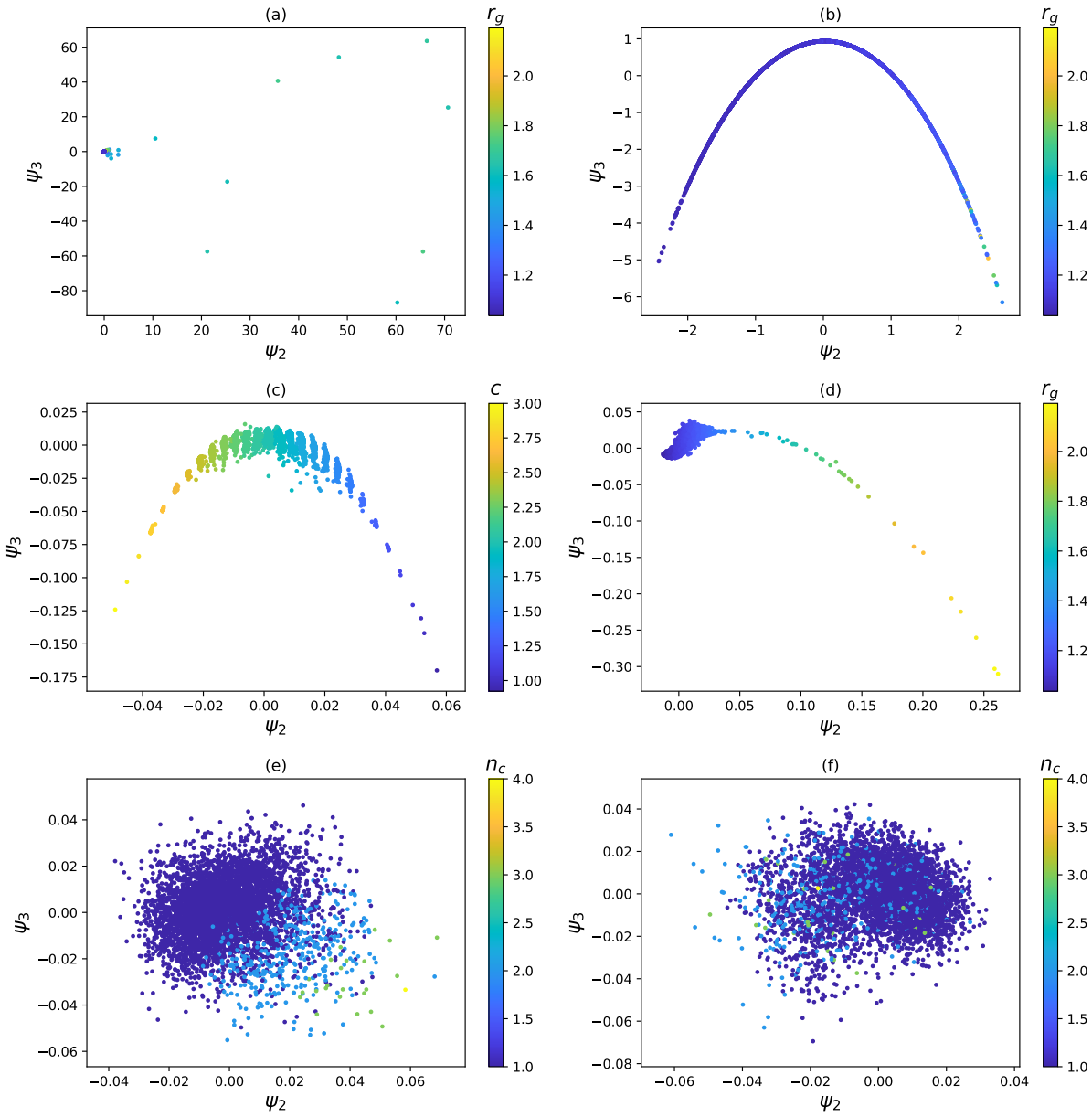


Figure 3.11: 2d Diffusion Coordinates generated using (a) d_H (b) d_M (c) $d_\Lambda^{(G)}$ (d) $d_\Lambda^{(R)}$ (e) $d_\psi^{(G)}$ (f) $d_\psi^{(R)}$. Colored by potential energy u^* with trajectories sampled from $T^* = 0.28$.

a variation of connectivity c values in these states. This might not be obvious as intuitively, one might expect that in a tighly packed cluster, all points are connected to each other but it is not the case for specific crystalline shapes like an icosahedron, which is the minimum potential energy structure.

Although the diffusion space generated by $d_{\psi}^{(G)}$ correlated best with n_c for the LJ3 (Fig. 3.5e), this correlation was not as seamless for LJ13. However, n_c was the best encoded variable as shown in (Fig. 3.9e) with ψ_3. For $T^* = 0.28$, the space was mostly filled with $n_c = 1$, which is understandable as $T^* = 0.28$ primarily holds solid/liquid phases with $n_c = 1$. Similar to previous cases, $d_{\psi}^{(R)}$ did not yield any much meaningful results, but we can see a loose pattern in ψ_2.

3.3.2 Convolutional Neural Networks

We split our dataset 60% for training, 30% for testing and 10% for validation. Using the gradient descent optimizer, the model converged within 1000 steps and was able to predict the potential energy of any graphical snapshot purely by scanning the adjacency matrix. The subspace generated $\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3\}$ shown in Fig. 3.12, explicitly indicated a one dimensional variable, where $\mathcal{L}$ correlates with connectivity. This variable correlated best with connectivity with some noise. For our case, we interpret this variable as purely structural which is a function of the underlying energy landscape of the molecular system without the knowledge about any of the thermodynamic constants such as the Lennard Jones bandwidth. One disadvantage of this technique is that it not naturally permutation invariant unlike the spectral distance metrics we used for diffusion maps. However, existence of more than one kernel starting out as random variables can naturally discover patterns under permutation. The network performed poorly on $T = 0.28$ as the dataset has low variance due to the stability of low energy clusters.

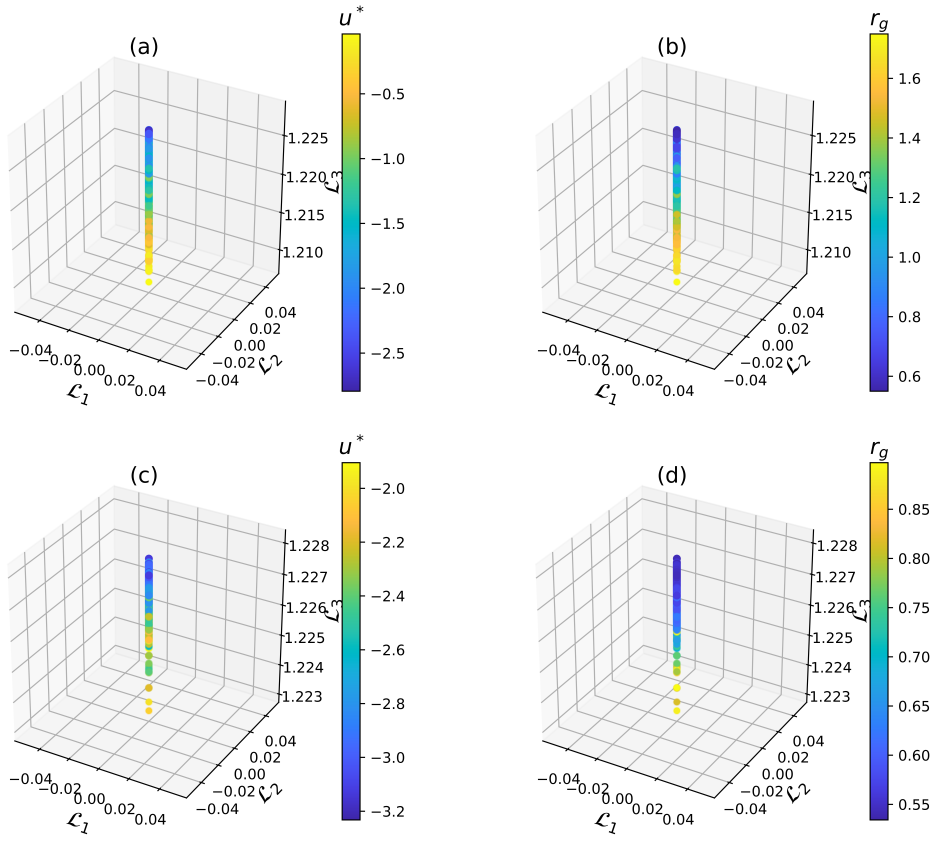


Figure 3.12: Subspaces of neural network for (a) $T^* = 0.4$ colored by u^* (b) $T^* = 0.4$ colored by r_g (c) $T^* = 0.28$ colored by u^* (d) $T^* = 0.28$ colored by r_g

.

3.4 Analytic study of distance metrics

We employ an analytic study of the distance metric to gain insight towards why the distance metrics captured the variables the way they did. We need to establish the relationship between the eigenvalues and the structural variables to determine what variables these subspaces represent. We relate the eigenvalue distance to the difference in connectivity

using the reverse triangle inequality of absolute values,

$$d_\Lambda : \frac{1}{N}\sum_{i=0}^{N} |\Lambda_{1i} - \Lambda_{2i}| \leq \left| \frac{1}{N}\sum_{i=0}^{n} \Lambda_{1i} - \frac{1}{N}\sum_{i=0}^{n} \Lambda_{2i} \right|. \tag{3.1}$$

We ignore the modulus operator on individual eigenvalues because the laplacian matrix is positive semidefinite (all eigenvalues are positive). We know that the trace of a matrix is equal to the sum of its eigenvalues. Therefore for a binary laplacian matrix L_G with a degree vector d,

$$\frac{1}{N}\sum_{i=0}^{N} \Lambda_i = \frac{1}{N} Tr(\mathbf{L_G}) = \frac{1}{N}\sum_{i=0}^{n} d_i = c. \tag{3.2}$$

Hence plugging (3.2) in (3.1), we get

$$d_\Lambda : \frac{1}{N}\sum_{i=0}^{n} |\Lambda_{1i} - \Lambda_{2i}| \geq |c_1 - c_2|. \tag{3.3}$$

If we color the diffusion space with connectivity, we can observe perfect correlation with diffusion coordinate Ψ_2 in fig 3.9a. For a weighted graph representation with eucledian distance adjacency $\mathbf{R}$, the eigenvalue sum is related to the squared radius of gyration r_g^2 as shown

$$\frac{1}{N}\sum_{i=0}^{n} |\Lambda_i| = \frac{1}{N} Tr(L_R) = \frac{1}{N}\sum_{j=0}^{n}\sum_{i=0}^{n} r_{ij} = \frac{1}{2N} r_g^2. \tag{3.4}$$

If we plug this in (3.1), we get ,

$$d_\Lambda : \frac{1}{N}\sum_{i=0}^{n} |\Lambda_{1i} - \Lambda_{2i}| \geq \frac{1}{2N} |r_{g1}^2 - r_{g2}^2|. \tag{3.5}$$

The diffusion coordinate Ψ_2 for the eucledian distance representation indeed correlates strongly with radius of gyration as shown in Fig. 3.9 b. Fig. 3.13 visualizes the distance kernels when the data is sorted based on a variable. This way, we can see how these distances generate the subspaces correlating with corresponding variables. We look at the simple case of LJ3 with distances $d_\Lambda^{(\mathbf{G})}$ and $d_\Lambda^{(\mathbf{R})}$. We have already established that the detected variable for $d_\Lambda^{(\mathbf{G})}$ is connectivity and radius of gyration r_g^2 for $d_\Lambda^{(\mathbf{R})}$. Fig. 3.13a is the distance kernels when configurations are sorted based on energy. We can see a loose pattern in distances but when we arrange based on connectivity c, we can see four distance values which correspond to the four classes as seen in Fig. 3.5c. One interesting observation is that it does not matter what way these configurations are arranged as the *permutation invariant* property of spectral decomposition will capture these patterns regardless. We can see the similar pattern for d_Λ over $\mathbf{R}$. Although, the spread is less gradual for tightly packed clusters. Fig. 3.13c is distance kernel for d_Λ when configurations are sorted based on energy and Fig. 3.13d is the kernel when energies are arranged based on radius of gyration.

3.5 Order Parameters

Results discussed in previous sections suggest a combination of r_g^2 and n_b as a sufficient set of order parameters, or small-system thermodynamic variables. For LJ13, r_g^2 and c were identified as good variables with c correlating with the principal diffusion coordinate ψ_2 generated with distance $d_\Lambda^{(G)}$ and r_g correlating with ψ_2 generated from $d_\Lambda^{(R)}$. The number of clusters n_c correlated with ψ_2 generated from $d_\psi^{(G)}$ very well for LJ3, but it was not as strong for LJ13 as an extra principle eigenvector ψ_3 was needed to get a loose correlation. The number of clusters might be useful for larger clusters, but for small clusters like LJ3 this

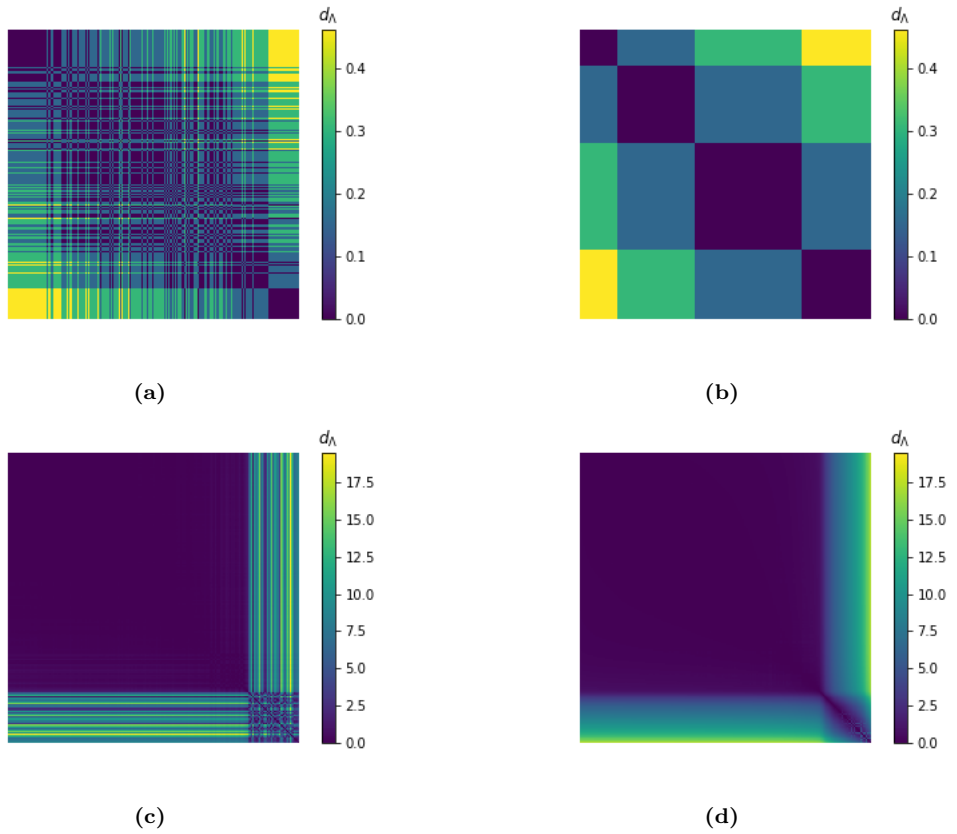


Figure 3.13: Distance kernels displayed as images. (a) $d_\Lambda^{(G)}$ when configurations arranged based on energy. (b) $d_\Lambda^{(G)}$ when configurations arranged based on connectivity. (c) $d_\Lambda^{(R)}$ when configurations arranged based on energy. (d) E$d_\Lambda^{(R)}$ when configurations arranged based on radius of gyration.

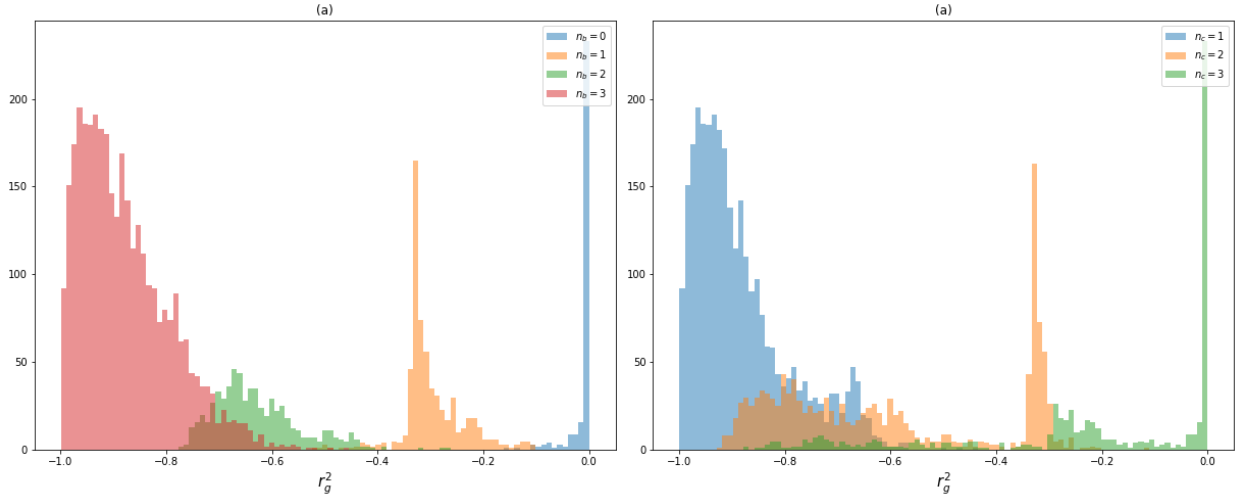


Figure 3.14: Comparing energy distributions in macrostates of LJ3 with respect to the variable (a) n_b (b) n_c.

measure of structure similarity might be too coarse. For example, clusters with two bonds and three bonds are indistinguishable using n_c. Fig. 3.14 visualizes energy distributions in all the possible values for n_b and n_c. There is a significant amount of merging for energies in case of n_c, making it a poor descriptor of energy. However if we look at distributions in n_b, we can clearly see a decent seperation in higher energy clusters but there is a merge in lower energies. Hence, we can see that n_b is a better discrete variable to represent the LJ cluster with r_g being the continuous variable.

Fig. 3.15 shows the potential energy landscape of configurations plotted in reduced parameter space for LJ3 and Fig. 3.16 for LJ13. Fig 3.17 shows how these landscapes look when looked at from a perspective of a specific physical variable. n_b separates the data into the four structural motifs exhibited by the model and r_g^2 catches the modes of motion (shape changes) within a given different motif. Note that the $n_b = 3$ state is the stiffest, with large changes in u^* resulting from small changes in r_g^2, while $n_b = 0$ is the least stiff. With the exception of some broadening in the $n_b = 1$ state, which may have to do with modes of motion that cannot be captured purely by r_g^2, this representation does an excellent job of collapsing the data. Fig 3.16, is the potential energy landscape for LJ13 in reduced space. The variables detected are the same as LJ3 but as explained before in section 2.1, we use connectivity c instead of number of bonds n_b to avoid scaling for large values of n_b. These variables can be used to describe the state of a cluster instead of individual particle coordinates and can be used for applications such as controlling colloidal particle assembly [20].

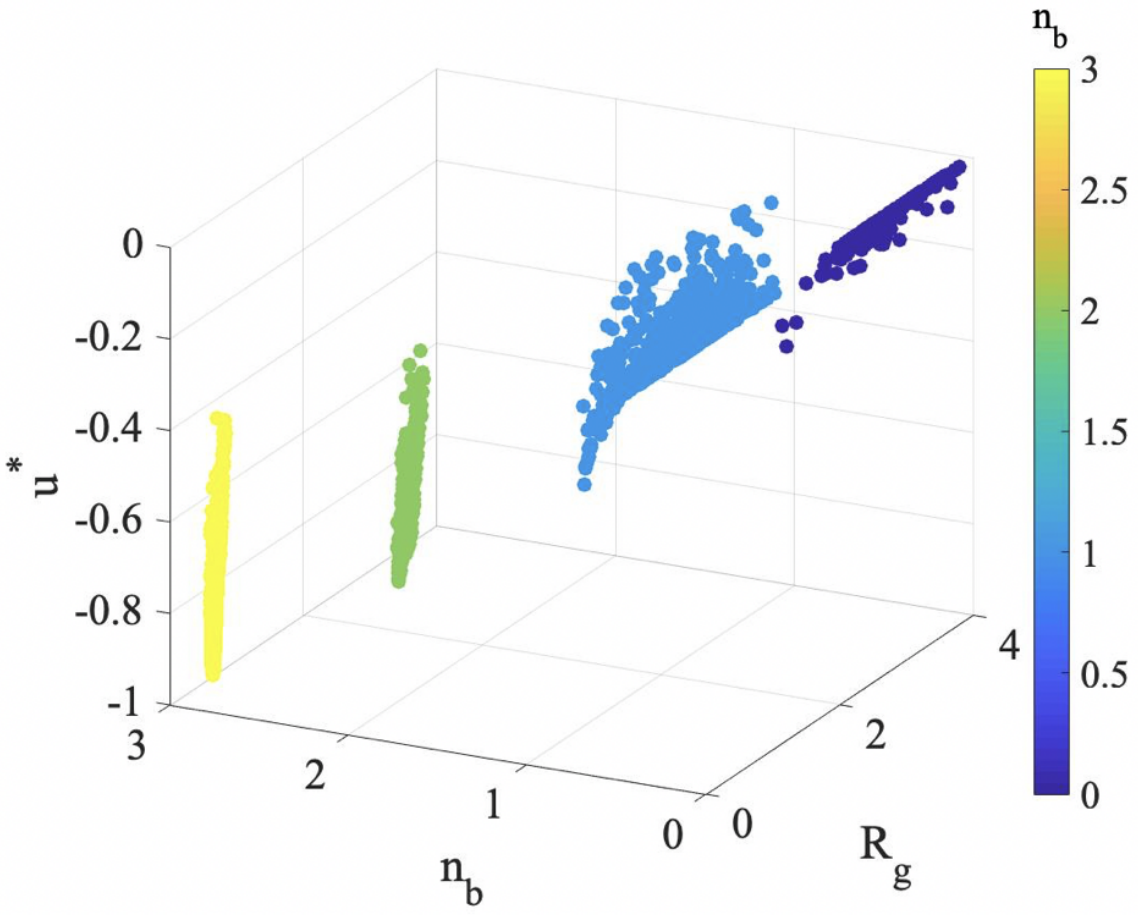


Figure 3.15: Energy Landscape for LJ3 with reduced parameters.

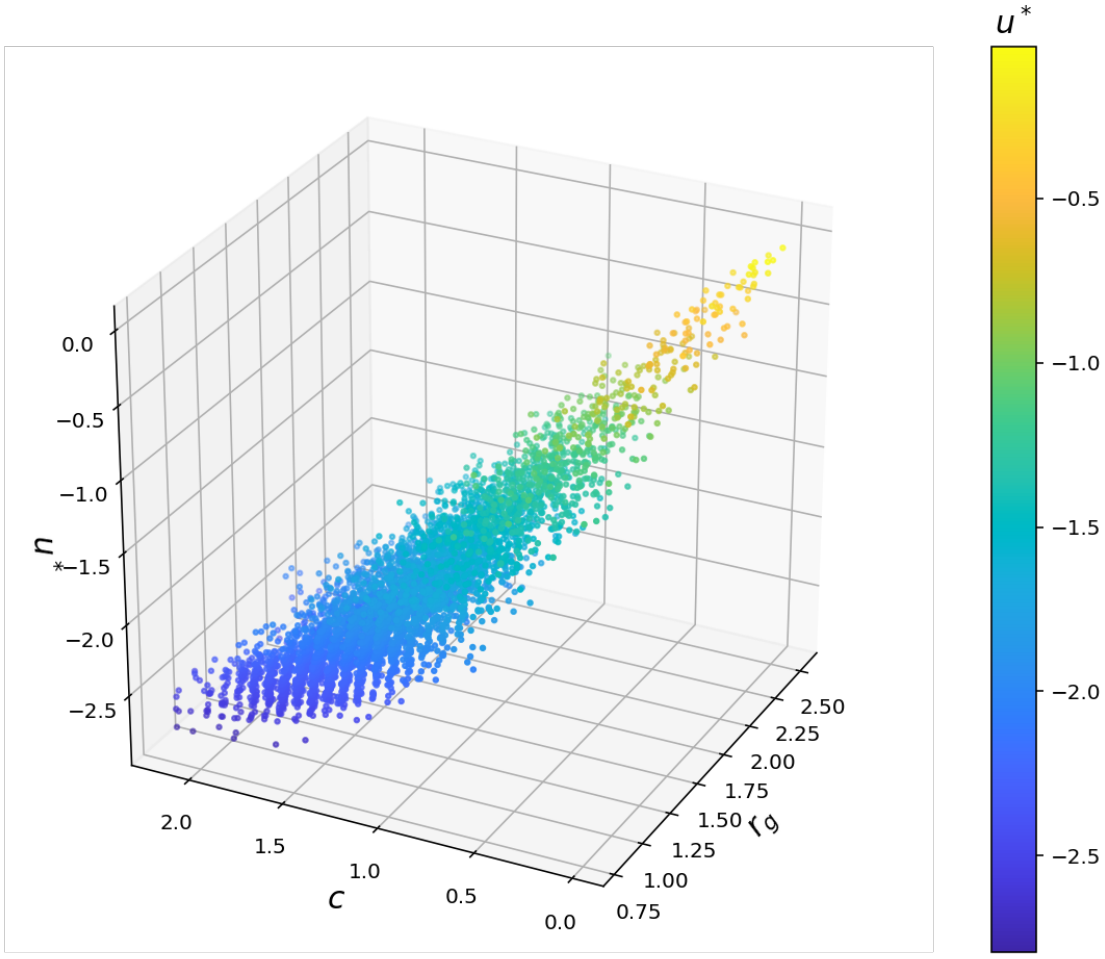


Figure 3.16: Energy Landscape for LJ13 with reduced parameters.

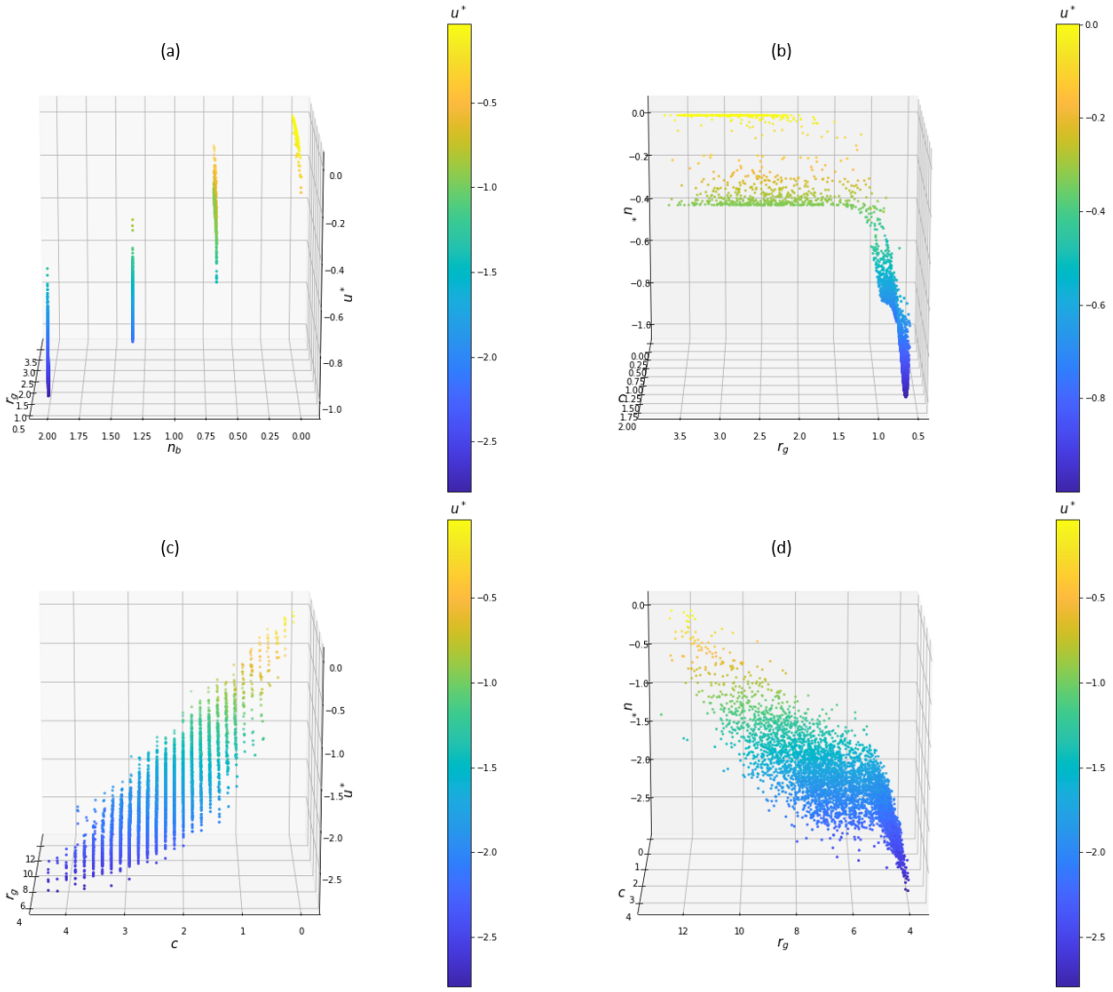


Figure 3.17: Energy landscape in reduced space with different perspectives. (a) LJ3 space in perspective of n_b (b) LJ3 space in perspective of r_g (c) LJ13 space in perspective of c (d) LJ3 space in perspective of r_g.

4 Conclusion

This study demonstrates the utility of spectral graph theory and machine learning in studying complex systems. These techniques recognised patterns in a noisy, high dimensional system of Lennard Jones clusters and detected variables to behave as thermodynamics state variables on which an energy landscape can be constructed. The variables detected were connectivity c and radius of gyration r_g (see section 3). This study also introduced the use of spectral distance metrics to capture structural differences between atomic clusters. These distance metrics are permutation invariant, which relaxes the expensive step of handling index symmetries. Finally, an explicit relation between these distance metrics and structural differences was also derived (see section 3.4). Diffusion Maps proved to be a very powerful technique by detecting variables in noisy simulation data. The second main method used for this study is to set up a neural network optimization problem and mining reduced variables from the latent spaces. The neural net approach resulted in the same variables as diffusion maps. These reduced representations can be used to simplify experimental control of colloidal molecules using a simplified set of parameters. Due to the abstract nature of these techniques, these ideas can easily be implemented for various other complex systems. Overall, this was a very fruitful project which brought together multiple fields like graph theory, machine learning and statistical thermodynamics.

Bibliography

[1] B. P. Zeigler, T. G. Kim, and H. Praehofer, *Theory of modeling and simulation*. Academic press, 2000.

[2] G. H. Golub and J. M. Ortega, *Scientific computing: an introduction with parallel computing*. Elsevier, 2014.

[3] F. E. Cellier and J. Greifeneder, *Continuous system modeling*. Springer Science & Business Media, 2013.

[4] P. Lynch, "The origins of computer weather prediction and climate modeling," *Journal of Computational Physics*, vol. 227, no. 7, pp. 3431–3444, 2008.

[5] S. Weart, "The development of general circulation models of climate," *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, vol. 41, no. 3, pp. 208–217, 2010.

[6] M. S. Waterman, *Introduction to computational biology: maps, sequences and genomes*. CRC Press, 1995.

[7] S. V. Hoover and R. F. Perry, *Simulation: a problem-solving approach*. Addison-Wesley Reading, MA, 1989.

[8] C. Lemmen and T. Lengauer, "Computational methods for the structural alignment of molecules," *Journal of Computer-Aided Molecular Design*, vol. 14, no. 3, pp. 215–232, 2000.

[9] R. J. LeVeque and R. J. Leveque, *Numerical methods for conservation laws*. Springer, 1992, vol. 3.

[10] S. C. Chapra, R. P. Canale *et al.*, *Numerical methods for engineers*. Boston: McGraw-Hill Higher Education,, 2010.

[11] R. Hamming, *Numerical methods for scientists and engineers*. Courier Corporation, 2012.

[12] J. N. Kutz, J. L. Proctor, and S. L. Brunton, "Koopman theory for partial differential equations," *arXiv preprint arXiv:1607.07076*, 2016.

[13] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of koopman eigenfunctions for control," *arXiv preprint arXiv:1707.01146*, 2017.

[14] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, 2016.

[15] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 113–127, 2006.

[16] A. L. Ferguson, A. Z. Panagiotopoulos, I. G. Kevrekidis, and P. G. Debenedetti, "Nonlinear dimensionality reduction in molecular simulation: The diffusion map approach," *Chemical Physics Letters*, vol. 509, no. 1-3, pp. 1–11, 2011.

[17] A. L. Ferguson, A. Z. Panagiotopoulos, P. G. Debenedetti, and I. G. Kevrekidis, "Systematic determination of order parameters for chain dynamics using diffusion maps," *Proceedings of the National Academy of Sciences*, vol. 107, no. 31, pp. 13 597–13 602, 2010.

[18] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids*. Oxford university press, 2017.

[19] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2001, vol. 1.

[20] M. A. Bevan, D. M. Ford, M. A. Grover, B. Shapiro, D. Maroudas, Y. Yang, R. Thyagarajan, X. Tang, and R. M. Sehgal, "Controlling assembly of colloidal particles into structured objects: Basic strategy and a case study," *Journal of Process Control*, vol. 27, pp. 64–75, 2015.

[21] K. G. Denbigh and K. G. Denbigh, *The principles of chemical equilibrium: with applications in chemistry and chemical engineering*. Cambridge University Press, 1981.

[22] D. V. Schroeder, "An introduction to thermal physics," 1999.

[23] T. L. Hill, "Perspective: Nanothermodynamics," *Nano Letters*, vol. 1, no. 3, pp. 111–112, 2001.

[24] M. Turmine, A. Mayaffre, and P. Letellier, "Nonextensive approach to thermodynamics: Analysis and suggestions, and application to chemical reactivity," *The Journal of Physical Chemistry B*, vol. 108, no. 49, pp. 18 980–18 987, 2004.

[25] D. A. McQuarrie, *Statistical Mechanics*, ser. Harper's Chemistry Series. New York: HarperCollins Publishing, Inc., 1976.

[26] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the national academy of sciences*, vol. 102, no. 21, pp. 7426–7431, 2005.

[27] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler, "Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems," *Multiscale Modeling & Simulation*, vol. 7, no. 2, pp. 842–864, 2008.

[28] M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi, "Determination of reaction coordinates via locally scaled diffusion map," *The Journal of chemical physics*, vol. 134, no. 12, p. 03B624, 2011.

[29] Y. Xue, P. J. Ludovice, M. A. Grover, L. V. Nedialkova, C. J. Dsilva, and I. G. Kevrekidis, "State reduction in molecular simulations," *Computers & Chemical Engineering*, vol. 51, pp. 102–110, 2013.

[30] L. V. Nedialkova, M. A. Amat, I. G. Kevrekidis, and G. Hummer, "Diffusion maps, clustering and fuzzy markov modeling in peptide folding transitions," *The Journal of chemical physics*, vol. 141, no. 11, p. 09B611_1, 2014.

[31] R. A. Mansbach and A. L. Ferguson, "Machine learning of single molecule free energy surfaces and the impact of chemistry and environment upon structure and dynamics," *The Journal of chemical physics*, vol. 142, no. 10, p. 03B607_1, 2015.

[32] C. J. Dsilva, R. Talmon, C. W. Gear, R. R. Coifman, and I. G. Kevrekidis, "Data-driven reduction for multiscale stochastic dynamical systems," *arXiv preprint arXiv:1501.05195*, 2015.

[33] W. F. Reinhart, A. W. Long, M. P. Howard, A. L. Ferguson, and A. Z. Panagiotopoulos, "Machine learning for autonomous crystal structure identification," *Soft Matter*, vol. 13, no. 27, pp. 4733–4745, 2017.

[34] A. W. Long and A. L. Ferguson, "Nonlinear machine learning of patchy colloid self-assembly pathways and mechanisms," *The Journal of Physical Chemistry B*, vol. 118, no. 15, pp. 4228–4244, 2014.

[35] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 32, no. 5, pp. 922–923, 1976.

[36] P. K. Redington, "Molfit: A computer program for molecular superposition," *Computers & chemistry*, vol. 16, no. 3, pp. 217–222, 1992.

[37] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[38] J. Behler, "Perspective: Machine learning potentials for atomistic simulations," *The Journal of chemical physics*, vol. 145, no. 17, p. 170901, 2016.

[39] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," *Physical review letters*, vol. 98, no. 14, p. 146401, 2007.

[40] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Physical review letters*, vol. 108, no. 5, p. 058301, 2012.

[41] J. S. Smith, O. Isayev, and A. E. Roitberg, "Ani-1: an extensible neural network potential with dft accuracy at force field computational cost," *Chemical science*, vol. 8, no. 4, pp. 3192–3203, 2017.

[42] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature communications*, vol. 8, no. 1, pp. 1–8, 2017.

[43] A. J. Ballard, R. Das, S. Martiniani, D. Mehta, L. Sagun, J. D. Stevenson, and D. J. Wales, "Energy landscapes for machine learning," *Physical Chemistry Chemical Physics*, vol. 19, no. 20, pp. 12 585–12 603, 2017.

[44] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, no. 5, p. 431, 2017.

[45] D. J. Beltran-Villegas, R. M. Sehgal, D. Maroudas, D. M. Ford, and M. A. Bevan, "A smoluchowski model of crystallization dynamics of small colloidal clusters," *The Journal of chemical physics*, vol. 135, no. 15, p. 154506, 2011.

[46] Beltran-Villegas, D. J, R. M. Sehgal, D. Maroudas, D. M. Ford, and M. A. Bevan, "Colloidal cluster crystallization dynamics," *The Journal of chemical physics*, vol. 137, no. 13, p. 134901, 2012.

[47] R. M. Sehgal, J. G. Cogan, D. M. Ford, and D. Maroudas, "Onset of the crystalline phase in small assemblies of colloidal particles," *Applied Physics Letters*, vol. 102, no. 20, p. 201905, 2013.

[48] R. M. Sehgal and D. Maroudas, "Equilibrium shape of colloidal crystals," *Langmuir*, vol. 31, no. 42, pp. 11 428–11 437, 2015.

[49] Y. Yang, R. Thyagarajan, D. M. Ford, and M. A. Bevan, "Dynamic colloidal assembly pathways via low dimensional models," *The Journal of chemical physics*, vol. 144, no. 20, p. 204904, 2016.

[50] D. Wales *et al.*, *Energy landscapes: Applications to clusters, biomolecules and glasses*. Cambridge University Press, 2003.

[51] J. C. Chen and A. S. Kim, "Brownian dynamics, molecular dynamics, and monte carlo modeling of colloidal systems," *Advances in colloid and interface science*, vol. 112, no. 1-3, pp. 159–173, 2004.

[52] D. M. Ford, A. Dendukuri, G. Kalyoncu, K. Luu, and M. J. Patitz, "Machine learning to identify variables in thermodynamically small systems," *Computers & Chemical Engineering*, p. 106989, 2020.

[53] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[54] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International journal of pattern recognition and artificial intelligence*, vol. 18, no. 03, pp. 265–298, 2004.

[55] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," *Proceedings of the National Academy of Sciences*, vol. 105, no. 35, pp. 12 763–12 768, 2008.

[56] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[57] J. E. Mayer, "Contribution to statistical mechanics," *The Journal of Chemical Physics*, vol. 10, no. 10, pp. 629–643, 1942.

[58] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[60] A. Dendukuri, G. Kalyoncu, K. Luu, and D. M. Ford, "Detecting subspaces for small lennard-jones clusters using diffusion maps," *Journal of Physical Chemistry*, 2020, under review.

Appendix: All Publications Published, Submitted, and Planned

D. M. Ford, A. Dendukuri, G. Kalyoncu, K. Luu, and M. J. Patitz, "Machine learning to identify variables in thermodynamically small systems," *Computers & Chemical Engineering*, p. 106989, 2020

A. Dendukuri, G. Kalyoncu, K. Luu, and D. M. Ford, "Detecting subspaces for small lennard-jones clusters using diffusion maps," *Journal of Physical Chemistry*, 2020, under review