UNIVERSITY OF CALIFORNIA,
IRVINE

Using Machine Learning to Construct and Categorize Density Functionals

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Chemistry

by

Bhupalee Kalita

Dissertation Committee:
Professor Kieron Burke, Chair
Professor Craig Martens
Professor Filipp Furche

2022

DEDICATION

In memory of Pijush Kanti Tahbildar (1994 - 2009)

The friend I cherished,
The brother I lost,
The young scientist who inspired my journey and whose dreams I must fulfill.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Kieron Burke, for pushing me outside my comfort zone in every aspect and helping me find strength in my weaknesses. His efforts and encouragement led me to learn concepts and skills that would be valuable for the rest of my life. With his guidance and support, I could gain enough confidence to picture myself as a researcher and explore new exciting research territories.

Next, I sincerely thank the Burke group members for providing an excellent and supportive learning environment. I would especially like to thank my fellow graduate student and collaborator, Ryan Pederson, for his help and guidance in code implementation and calculations. I am grateful to my collaborator, Dr. Li Li, for assisting us with the codes and concepts for differentiable DFT programming. As a collaborator and a group member, I will remain indebted to Dr. Ryan J. McCarty for his guidance and inspiration in every aspect of life and research. I thank Dr. Suhwan Song for running and teaching me to run the DC-DFT calculations and Dr. Stefan Vuckovic and Prof. Eunji Sim for all the DC-DFT-related inputs for categorizing density functionals.

I would also like to thank my defense committee members, Prof. Filipp Furche and Prof. Craig Martens, for their helpful comments, questions, and suggestions for the thesis. In addition, I would like to acknowledge the support and assistance from the staff members at the UCI Department of Chemistry and the UCI International Center.

A few great friends made life at Irvine memorable; I would like to thank them all. Mainly, I thank Ankita Biswas, a classmate, and a neighbor, who has always had my back for the past five years. Remembering pre-pandemic life, I am grateful to Dr. Sree Ganesh Balasubramani and Dr. Saswata Roy for sharing their valuable insights into research and life.

Last but not least, I would like to express my sincere gratitude to my parents and siblings for understanding my dreams and aspirations. Their constant encouragement and blind faith in me kept me motivated. I would never be able to make up for missing out on several sad and happy moments of their lives in the past five years. This thesis and my adventure in science are owed to their hopes, compromise, and dedication.

I am grateful to the National Science Foundation for funding our projects (Grant No. CHE-1856165 and CHE-2154371). I am also thankful to Machine Learning and Physical Sciences (MAPS), the UCI graduate training program funded by the National Science Foundation (Grant No. DGE-1633631), for supporting my research for a year. I would also like to thank Prof. Stephan M. Mandt, my co-advisor in this program from the Department of Computer Science and Statistics, for his helpful comments.

The Physical Sciences Machine Learning Nexus program also briefly supported my endeavors in machine learning. I would like to thank retired Prof. Roger McWilliams, the person behind the initiative, for putting his faith in me as a Machine Learning Nexus Fellow and giving me the opportunity to learn from the excellent machine learning community at UCI.

Finally, I would like to acknowledge the publishers for allowing me to use the following work in this thesis:

⟨Ch. 1⟩ R. Pederson, B. Kalita, and K Burke. Machine learning and density functional theory. *Nature Reviews Physics*, 4(6):357-358, 2022. Reproduced by permission of the Nature Publishing Group.

⟨Ch. 2⟩ B. Kalita, L. Li, R. J. McCarty, and K. Burke. Learning to approximate density functionals. *Accounts of Chemical Research*, 54(4):818-826, 2021. Reproduced by permission of the American Chemical Society.

⟨Ch. 4⟩ B. Kalita, R. Pederson, J. Chen, L. Li, and K. Burke. How well does Kohn-Sham Regularizer work for weakly correlated systems? *The Journal of Physical Chemistry Letters*, 13(11):2540-2547, 2022. Reproduced by permission of the American Chemical Society.

VITA

Bhupalee Kalita

EDUCATION

2022 Doctor of Philosophy in Theoretical Chemistry, University of California Irvine
2016 Master of Science in Chemistry, University of Hyderabad, India
2014 Bachelor of Science in Chemistry, Gauhati University, India

RESEARCH EXPERIENCE

2018–2022 Graduate Student Researcher, University of California Irvine
Ph.D. Advisor: Prof. Kieron Burke
Machine learning density functional theory

2016–2017 Research Assistant, Jawaharlal Nehru Center for Advanced Scientific Research, India
Descriptor-based microkinetic mapping of catalytic trends

2016 Summer Intern, Institute of Advanced Studies in Science and Technology, Guwahati, India
Mechanistic study on co-encapsulation of chloramphenicol

2016 Master's Thesis Research, School of Chemistry, University of Hyderabad, India
A new variational principle for the double well potential density matrices

2015 Indian Academy of Sciences Summer Research Fellow
Jawaharlal Nehru Center for Advanced Scientific Research, India
Gas-phase site charge calculation mixed ionic liquid clusters

TEACHING EXPERIENCE

2017–2020 School of Physical Sciences, University of California Irvine

2020 Chem 254-Machine Learning: Course Co-Creator
2020 Canvas course on Machine Learning: Teaching Assistant
2019 Chem 1LC-General Chemistry Lab, Teaching Assistant
2018 Chem 132C-Molecular Structure: Teaching Assistant
2018 Chem 1LE-General Chemistry Lab: Teaching Assistant
2017 Chem 5-Scientific Computing Skills: Teaching Assistant

JOURNAL PUBLICATIONS

1. S. Crisostomo, R. Pederson, J. Kozlowski, **B. Kalita**, A. C. Cancio, K. Datchev, A. Wasserman, S. Song, and K. Burke. Seven useful questions in density functional theory. Submitted to *Letters in Mathematical Physics*, 2022. *arXiv:2207.05794*

2. R. Pederson, **B. Kalita**, and K Burke. Machine learning and density functional theory. *Nature Reviews Physics*, 4(6):357-358, 2022.

3. **B. Kalita**, R. Pederson, J. Chen, L. Li, and K. Burke. How well does Kohn-Sham Regularizer work for weakly correlated systems? *The Journal of Physical Chemistry Letters*, 13(11):2540-2547, 2022.

4. **B. Kalita** and K. Burke. Using machine learning to find new density functionals, A section in Roadmap on Machine Learning in Electronic Structure by Kulik et al. *Accepted manuscript, Electronic Structure*.

5. **B. Kalita**, L. Li, R. J. McCarty, and K. Burke. Learning to approximate density functionals. *Accounts of Chemical Research*, 54(4):818-826, 2021.

CONFERENCE PUBLICATIONS

1. **B. Kalita**, R. Pederson, L. Li, and K. Burke. Generalizability of density functionals learned from differentiable programming on weakly correlated spin-polarized systems, *NeurIPS Differentiable Programming Workshop*, 2021.

CONTRIBUTED PRESENTATIONS

2022 Oral Presentation at the APS March Meeting
Kohn-Sham regularizer in the bond-dissociation limit

2021 Oral Presentation at the (TD)DFT Student Seminar Series
Machine learning density functionals: Testing the Kohn-Sham regularizer

AWARDS AND HONORS

2020–2021 NSF Fellow, Machine Learning and Physical Sciences (MAPS)
National Science Foundation Research Traineeship (NRT) Program

2020 Physical Sciences Machine Learning Nexus Data Science Fellow
School of Physical Sciences, University of California Irvine

2019–2020 Honorary Fellow, Machine Learning and Physical Sciences (MAPS)
National Science Foundation Research Traineeship (NRT) Program

2016 Prof. V. V. Sharma Memorial Award
School of Chemistry, University of Hyderabad

2011-2016 INSPIRE Scholarship for Higher Education (SHE)
Department of Science and Technology, Government of India

ABSTRACT OF THE DISSERTATION

Using Machine Learning to Construct and Categorize Density Functionals

By

Bhupalee Kalita

Doctor of Philosophy in Chemistry

University of California, Irvine, 2022

Professor Kieron Burke, Chair

Density functional theory (DFT), combined with standard exchange-correlation approximations, is a usefully accurate and efficient tool to generate computational predictions in chemistry and material sciences. In the past decade, machine learning has been used extensively to build density functional approximations that concur with human-defined standards. This thesis details the effort to construct and characterize exchange-correlation approximations in DFT with physics-informed machine learning.

The Kohn-Sham regularizer (KSR) is a differentiable approach for making machine-learned density functionals. It allows approximating the exchange-correlation functional while self-consistently solving the Kohn-Sham equations. It was initially formulated to generate accurate predictions for strongly-correlated molecules. Here I discuss a spin-adapted extension of the KSR that machine-learns the exchange-correlation energy densities as a functional of the spin densities and substantially improves generalizability for weakly correlated molecules. With a neural network approximation that accounts for nonlocal interactions, a training set of just five atoms and ions in 1D can predict the ground-state properties of several molecules with near chemical accuracy. The differentiability of spin-adapted KSR ensures a fast convergence during training and yields accurate predictions of the exchange-correlation potentials and other properties, often complying with known exact behaviors.

While this serves as a proof of concept for what machine learning can achieve, such methods, in principle, can add to the complexity of the existing diverse approaches for designing exchange-correlation approximations, further deluding the existence of a unified scheme for systematic improvement of density functionals. On the other hand, machine learning, especially unsupervised learning algorithms, can help categorize different exchange-correlation approximations without introducing human bias or considering any absolute errors. To answer the question of how several exchange-correlation functionals are similar or different from each other, we propose a novel approach to group these functionals based on statistical learning tools. This approach does not use any exact information, accounts for density-driven differences in approximations based on the theory of density-corrected DFT, and avoids any form of bias between empirical, partially-empirical, and non-empirical approximations. It sorts exchange-correlation functionals based on similarities predicted using a novel, parameter-free unsupervised clustering algorithm. For 33 popular exchange-correlation functionals and the MGAE109 dataset, this scheme generates categories of functionals that somewhat mimic the popular Jacob's ladder categorization while depicting that Minnesota functionals of recent vintage might have strayed far from the path of typical functional development.

Part I

Introduction

Chapter 1

Motivation and Section Summaries

Part of this chapter is written with Ryan Pederson and Kieron Burke. Published in *Nature Reviews Physics*, 4(6):357-358, 2022.

1.1 Density Functional Theory Everywhere

In physical sciences, density functional theory (DFT) is often the go-to computational method for solving electronic structure problems. DFT provides fully quantum solutions at a fraction of the cost of solving the Schrödinger equation directly by mapping the coupled many-body problem to a single-particle problem. The electronic energy is considered as a functional (a function of a function) of the electron (probability) density, with only a small portion, the exchange-correlation energy, being approximated.

It is staggering to see just how important DFT calculations have become. Each year, tens of thousands of papers report useful predictions from DFT calculations, and today about one-third of the National Energy Research Scientific Computing Center supercomputing resources use DFT [40]. John Perdew, who developed many of the formulas in current use, is one of the most cited

physicists of all time.

1.2 The GIGO Principle in DFT

The GIGO principle is an adage in computing, standing for garbage in, garbage out. In DFT, this means that a calculation is only as good as the approximate functional used. Humans have worked at this for almost a century, and hundreds of different approximations are in use nowadays. Some build in well-studied limits, such as the uniform electron gas, and satisfy many known physical constraints of the exact functional, while others are tuned and fitted to reference datasets. Regardless, general failures have been identified over the years. A decade-old review [24] focused on the struggle to describe strongly correlated systems. This most grievous failure can be understood from the perspective of fractional charges (systems with noninteger total charge) and fractional spins (systems with noninteger spin magnetization). The exact energy is a linear interpolation of the energy of the adjacent integer systems, but approximations miss this, producing embarrassingly large systematic errors in strongly correlated systems as simple as stretched H_2. Overcoming such fundamental DFT challenges is essential to expanding its applicability and reliability in condensed matter physics.

1.3 Machine Learning DFT

A proof of principle for ML-DFT appeared ten years ago. For a simple problem, the kinetic energy of non-interacting fermions in a 1D box, an ML method (kernel ridge regression) could be used to find an approximation of the functional by training on examples from accurate numerical calculations [153]. The resulting functional was far more accurate than anything ever designed by humans but only useful for simple model systems such as those it was trained on. The associated learning efficiency was also low, as hundreds of training examples were needed to reach high

accuracy for rather compact chemical space.

Later, the density was machine-learned directly from the external potential [11]. This demonstrated the practical usefulness of ML in DFT through realistic examples, such as proton transfer in an ML molecular dynamics simulation of malonaldehyde. However, unlike traditional DFT approximations, such ML models rarely generalize across elements.

Since then, there have been many attempts to bring the promise of ML to practical, generalizable functional construction. These efforts can be divided into two categories: those starting from traditional approximate forms suggested by humans (which are biased toward local and semilocal approximations) and those that use the entire density (that is, a nonlocal approximation) in some hard-to-understand way. Such nonlocal functionals can have poor generalizability, as the input feature space becomes vastly more complicated than local and semilocal forms, which depend only on the density and its gradient at each point.

As described in Ref. [109], a neural network (NN) functional was trained on accurate densities as well as energies of just three molecules, producing semilocal ML approximations that worked as well as human-designed functionals for 150 test molecules, generalizing very well. A similar approach was used in Ref. [93], but nonlocal forms based on convolution NNs were also used to learn an entire dissociation curve within chemical accuracy, including the strongly correlated region, with only two training examples. The model also generalized well for other new (but similar) strongly correlated molecules that were not encountered in training. During training, an end-to-end differentiable DFT code (where all components are differentiable) was used to obtain gradient information by backpropagation through the entire self-consistent calculation. Such robust gradient-based training results in impressive generalization of functional approximations.

However, the most recent exciting development comes once again from DeepMind [85]. Using vast computational resources, a bevy of 17 researchers revived an old human-designed suggestion, a local hybrid functional [27], that had been difficult to control. Their new NN-based func-

tional, DM21, was trained by evaluating the energy non-self-consistently using approximate densities. The regression loss consisted of an energy loss plus an explicit gradient regularization term, thereby making this training approach substantially cheaper than Ref. [93]. DM21 was trained on thousands of molecular systems, orders of magnitude more than previous ML training sets, and outperforms most other hybrid functionals on standard molecular benchmarks with impressive generalization. This ML functional can be used for main-group chemistry calculations, like most human-designed functionals. By including training on simple systems with fractional charges and spins, DM21 appears to perform significantly better than earlier approaches for strongly correlated systems. For instance, DM21 correctly dissociates systems such as H_2, H_2^+, and N_2, meeting the long-standing DFT challenge of strong correlation in molecular systems.

Researchers all over the world are currently trying out DM21, testing many different aspects to see if it lives up to its promise. The world of DFT applications is far too vast for DM21 developers to run even a fraction of useful tests in their original paper. Many promising approximations run into unexpected difficulties when tried in practice. The community will examine computational cost, accuracy, and transferability when testing DM21.

1.4 Overview of the Dissertation

This thesis details efforts to make new density functional approximations and categorize existing human-designed exchange-correlation approximations using machine learning. First, in Chapter 2, the development of a few of the machine learning applications for the kinetic energy and exchange-correlation energy functionals and their performances are reviewed. Then, Chapter 3 and Chapter 4 cover two approaches for constructing machine-learned density functionals. Finally, Chapters 5 and Chapter 6 explain the concept of unsupervised learning and how we can use it to categorize different exchange-correlation approximations. An overview of these chapters is given below.

1.4.1 Chapter 2: Learning to Approximate Density Functionals

This chapter introduces the preliminary works in machine learning DFT. The first attempt in making an orbital-free machine learned kinetic energy functional approximation was for several fermions in a box [153]. This study used a simple kernel ridge regression technique but could only produce accurate densities with principal component analysis. Next, the chapter discusses the extension of this work for describing bond-breaking [152] and improving accuracy with exact conditions [67]. Instead of learning the kinetic energy functional from the density, it is also possible to learn the density from the potential and the total energy from the learned density in a different mapping. This approach is also discussed for real molecules for molecular dynamics simulations of malonaldehyde [11]. By training on ab-initio examples instead of Kohn-Sham DFT results, this orbital-free approach can yield accurate molecular dynamics trajectories for molecules [9]. Then with a change of direction, the discussion shifts to modeling the universal part of the functional with kernel ridge regression, especially for strongly correlated molecules [91]. However, all these studies do not address one primary drawback of machine learning approximations - their limited generalizability for data that are too different from the training set. Finally, the chapter briefly describes the differentiable DFT approach for neural network exchange-correlation functionals [93] which tries to address the generalizability issue for strongly correlated molecules. This approach is later covered in detail in Chapter 4.

1.4.2 Chapter 3: Machine Learned Density Functionals with Legendre Transformation

Here, the idea of a novel approach is presented that proposes using DFT theoretical construction as the machine learning model. This chapter describes the construction of the machine-learned density functional as an approximation to the Lieb functional [90] in an orbital-free manner. This method utilizes a finite set of potentials and the property of the concavity of the ground-state en-

ergy to generate an approximation. Results have been discussed for a one-dimensional two-site Hubbard model at different interaction strengths and one-dimensional non-interacting harmonic oscillator, exponential, and delta-function potentials. A modification of the crude approximation has also been proposed for the two-site Hubbard model, which by incorporating Hermite interpolated ground-state energy, accurately reproduces the exact density. The chapter ends with a discussion of the possibility of constructing a more generalized machine-learned density functional that is transferable across different systems.

1.4.3 Chapter 4: Howe Well Does Kohn-Sham Regularizer Work for Weakly Correlated Systems?

Kohn-Sham regularizer (KSR) is an end-to-end differentiable machine learning approach for optimizing a physics-informed exchange-correlation functional within a differentiable Kohn-Sham DFT framework. In a proof of principle for 1D systems, KSR was shown to generalize well from weakly correlated molecules to stretched molecules of the same type [93]. This chapter covers the modified KSR approach with the incorporation of spin and discusses its generalizability for training on atomic systems and testing on molecules at equilibrium. Physics-informed neural networks were designed to use as local, semilocal or nonlocal approximations for exchange-correlation. While the atoms-to-molecules generalization error for the semilocal approximation was comparable to the original KSR, the modified nonlocal approximation with spin-adapted KSR could predict the ground state energy of several 1D molecules with near-chemical accuracy. This chapter also briefly discusses the finite generalization of the nonlocal approximation for strong correlation. Finally, it was shown that the KSR approach could, in principle, learn any human-designed approximation and, due to the differentiability of the program, can also yield qualitatively correct exchange-correlation potentials and other derivative quantities without explicitly training on them.

1.4.4 Chapter 5: Unsupervised Learning

This chapter serves as an introduction for the next chapter, where categorizing density functional approximations is discussed. Most often, machine learning tasks refer to the supervised learning task where we have some observational pairs in the training set and try to learn the relationship between them. The machine learning or the neural network techniques discussed in the previous chapters are all supervised learning examples. Hence, a thorough introduction to unsupervised learning is necessary. Chapter 5 describes what unsupervised learning is and what are some examples of unsupervised learning tasks. Then, it covers the concepts of dimensionality reduction, density estimation, and clustering and briefly describes several algorithms available to solve these problems. Special attention is paid to clustering while analyzing what clustering method may be suitable for dealing with small datasets with non-globular clusters. Since clustering is a relative evaluation of the relationship between data points, the chapter also briefly discusses methods for evaluating clustering quality which can help determine undetermined parameters associated with clustering.

1.4.5 Chapter 6: Categorizing Density Functionals with Unsupervised Learning

Chapter 6 details our effort towards answering whether recent density functionals are straying from the path towards the exact functionals or whether we can classify density functionals into a few simple categories. The tool of choice was unsupervised learning. However, finding the best descriptors to describe the functionals is essential to cluster a dataset meaningfully. We define descriptors that can account for both functional-driven and density-driven differences (unlike most other approaches) based on the theory of density-corrected DFT [150]. This chapter also describes a novel parameter-free clustering algorithm that can categorize the density-based functional fingerprints without human-introduced bias. For 33 different exchange-correlation functionals, including

DM21, we calculate the descriptors based on the MGAE109 dataset and perform parameter-free clustering. The generated functional groups somewhat mimic the well-known Jacob's ladder[121] categorization. However, most Minnesota functionals deviate from Jacob's ladder and form a separate cluster. Two other clustering methods suitable for small datasets are also explored in detail. The chapter ends with the discussion of suitable dimensionality reduction methods for visualizing the functional clusters.

A few chapters are direct copies of papers published for diverse audiences. Hence, there are inconsistencies in notations, definitions, and acronyms between chapters. Therefore, readers should refer to the definitions given within each chapter to prevent confusion.

Part II

Machine Learning Density Functional Theory

Chapter 2

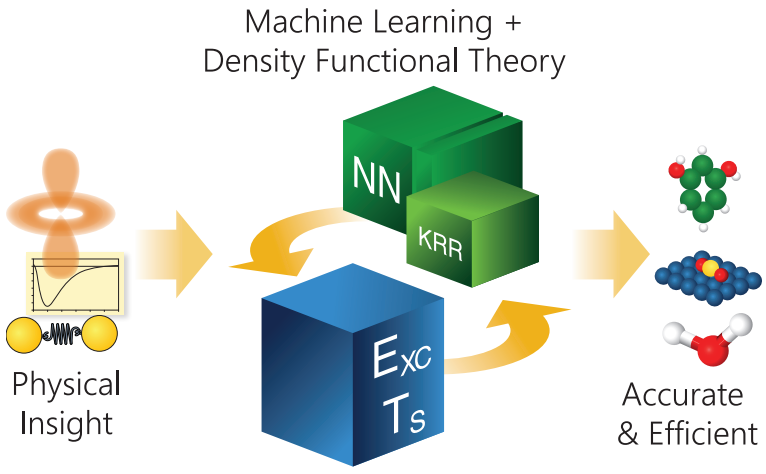
Learning to Approximate Density Functionals

written with Li Li, Ryan J. McCarty, and Kieron Burke. Published in *Accounts of Chemical Research*, 54(4):818-826, 2021.

Abstract: *Density functional theory (DFT) calculations are used in over 40,000 scientific papers each year in chemistry, materials science, and far beyond. DFT is beneficial because it is computationally much less expensive than ab-initio electronic structure methods and allows systems of considerably larger size to be treated. However, the accuracy of any Kohn-Sham DFT calculation is limited by the approximation chosen for the exchange-correlation (XC) energy. For more than half a century, humans have developed the art of such approximations, using general principles, empirical data, or a combination of both, typically yielding useful results but with errors well above the chemical accuracy limit (1 kcal/mol). Over the last 15 years, machine learning (ML) has made significant breakthroughs in many applications and is now applied to electronic structure calculations. This recent rise of ML begs the question: Can ML propose or improve density functional approximations? Success could significantly enhance the accuracy and usefulness of*

DFT calculations without increasing the cost.

In this work, we detail efforts in this direction, beginning with an elementary proof of principle from 2012, namely finding the kinetic energy of several fermions in a box using kernel ridge regression. This is an example of orbital-free DFT, for which a successful general-purpose scheme could make even DFT calculations run much faster. We trace the development of that work to state-of-the-art molecular dynamics simulations of resorcinol with chemical accuracy. By training on ab-initio examples, one bypasses the need to find the XC functional explicitly. We also discuss how the exchange-correlation energy can be modeled with such methods, especially for strongly correlated materials. Finally, we show how deep neural networks with differentiable programming can be used to construct accurate density functionals from very few data points by using the Kohn-Sham equations as a regularizer. All these cases show that ML can create approximations of greater accuracy than humans and can find approximations that can deal with complex cases such as strong correlation. However, such ML-designed functionals have not been implemented in standard codes because of one last great challenge: generalization. We discuss how effortlessly human-designed functionals can be applied to a wide range of situations and how difficult that is for ML.



Key references

- Snyder, J. C.; Rupp, M.; Hansen, K.; Müller, K.-R.; Burke, K. Finding density functionals with machine learning. *Phys. Rev. Lett.* **2012**, *108*, 253002. [153] *In a proof of principle, kernel ridge regression was used to approximate the kinetic energy of noninteracting fermions, and highly accurate self-consistent densities were obtained using projected functional derivatives.*

- Brockherde, F.; Vogt, L.; Li, L.; Tuckerman, M. E.; Burke, K.; Müller, K.-R. Bypassing the Kohn-Sham equations with machine learning. *Nature Communications* **2017**, *8*, 872. [11] *The density-potential and the energy-density maps were learned directly using machine learning. A molecular dynamics simulation of malonaldehyde using machine-learned functionals could capture the intramolecular proton transfer process.*

- Li, L.; Baker, T. E.; White, S. R.; Burke, K. Pure density functional for strong correlation and the thermodynamic limit from machine learning. *Phys.Rev. B* **2016**, *94*, 245129. [91] *By training a machine learning model for exchange-correlation, with data from a density matrix renormalization group calculation, chemically accurate results were obtained for atomic chains, even when strongly correlated, and extrapolated to the thermodynamic limit.*

- Li, L.; Hoyer, S.; Pederson, R.; Sun, R.; Cubuk, E. D.; Riley, P.; Burke, K. Kohn-Sham equations as regularizer: building prior knowledge into machine-learned physics. *Phys. Rev. Lett.* **2021**, *126*, 036401. [93] *A deep neural network was trained using the Kohn-Sham equation as an implicit regularizer. A diatomic dissociation curve was reproduced within the chemical accuracy limit with just two training molecules.*

2.1 Introduction

Direct solution of the Schrödinger equation for electrons (traditionally designated as *ab initio* in quantum chemistry) yields chemically accurate energies (errors below 1 kcal/mol). However, computational costs scale poorly with system size, limiting its routine applicability to smaller molecules. On the other hand, density functional theory (DFT) calculations typically scale much more favorably, allowing routine calculation of molecules with hundreds of atoms. This increased applicability comes at a cost: The effective noninteracting Kohn-Sham equations that, in principle, yield exact ground-state energies and densities, in practice, require a small fraction of the total energy (called the exchange-correlation (XC) energy) to be approximated in an uncontrolled way.

Presently, there are hundreds of distinct approximations to the XC energy [13], all of which are available in common electronic-structure codes. Some have been designed from general physics principles, without reference to any specific molecular or material system [115]. Others have been fitted and tested on an ever-growing population of databases of distinct molecular systems and properties, and these yield higher accuracies on those systems [53]. However, almost all use essential ingredients, such as the density, gradient, and a fraction of Hartree-Fock (HF) exchange, and are inspired by physical or chemical insight.

In the past decade, machine learning (ML) has seen some remarkable successes in various applications, including image recognition, language translation [62], and even playing curling [179]. ML is also increasingly being applied to problems in physical sciences, where it can help with, for example, extraction of salient features from microscopy images [107] or climate modelling [70]. It can also be used to speed up purely computational tasks. In electronic structure theory, there has been much success in designing new force fields using ML, creating far more accurate force fields than previous human-designed attempts [169]. ML force-fields can reproduce results from DFT or any *ab-initio* methods at a fraction of the computational cost, simply by training on carefully chosen examples, and are already available in useful codes [73].

A different and arguably more difficult task is to use ML to design new density functional approximations or to improve existing ones. This is simply a regression problem, i.e., fitting a function of many variables. But regression in DFT involves fitting a functional, which can be considered a function of infinitely many variables, and that complicates the task.

There are several distinct approaches to using ML to make functionals. If the goal is to make DFT calculations run faster, one such problem is approximating the KS kinetic energy functional, i.e., the kinetic energy of the noninteracting KS orbitals ($T_S[n]$), thereby bypassing the need to solve the KS equations, the most expensive step in most DFT implementations. If T_S could be computed rapidly, it could revolutionize all DFT calculations by making them run much faster [36]. This is called orbital-free DFT (OF-DFT) [153, 152, 94, 11]. Moreover, training data is abundant as every self-consistent cycle of every DFT calculation ever performed yields a set of orbitals (and hence density) and their T_S. However, the path to success is not smooth. To determine the density in OF-DFT, one must solve an Euler equation [15] requiring an accurate and well-behaved functional derivative of T_S. Due to limited information in direct training, ML-designed interpolating functionals that are extremely accurate for the energy almost necessarily yield poor functional derivatives.

The more traditional problem is to improve the accuracy of DFT, either by modifying existing XC approximations or creating completely new forms [109, 34, 93]. Usually (but not always [83]), the functional derivative of the XC energy is somewhat unimportant to the energy. However, unlike the orbital-free approach, the amount of accessible, accurate training data from higher levels of theories is limited and is primarily available for relatively small systems. Nonetheless, promising ML ideas developed for OF-DFT can also be applied to the XC case. Combining both can improve accuracy and computational cost simultaneously [91].

Another essential objective is to find new forms that overcome the drawbacks of traditional human-designed XC approximations. For instance, most molecules and many materials in their equilibrium state are considered weakly correlated where ingredients used in the past work reasonably

well and can be borrowed to design ML functionals. However, most XC approximations fail to break bonds correctly because they fail when a bond is stretched, and electrons localize on specific sites. Thus the complete binding energy curves of even H_2^+ and H_2 represent paradigm difficult problems for standard DFT [23]. A stretched bond is an example of a strong correlation that provides a good test for ML-designed functionals. Fig. 2.1 shows an ML-functional reproducing an entire binding energy curve from training on only two bond lengths [93]. Such bonds are even more difficult for OF-DFT if semi-local approximations (terms that depend on only the density and its gradient) [115, 7] are used, as the same considerations apply even more strongly to $T_S[n]$.

In principle, ML-designed functionals need not be limited by human imagination and intuition as ML can use the density everywhere to find the energy contribution at a point (a fully non-local functional) [94]. This is an ambitious goal. Humans have an almost 100-year head-start on this task [13], and it may be a while before an ML functional becomes as useful and practical as B3LYP [7] in chemistry. In current studies, many simplifications are made for efficient data generation and more straightforward implementation, simply to see if a new ML approach *can* work before building more realistic or general applications.

Thus, several of the examples discussed here are for one-dimensional analogs of true electronic structure problems [153, 152, 94, 91, 93]. For the noninteracting problem, a practical code can be written in a few minutes for solving the Schrödinger equation and training data generated within hours on a single core. For interacting systems, highly accurate solutions can be obtained very efficiently in one dimension, using a method called the density matrix renormalization group (DMRG) [178]. DMRG is a very powerful quantum solver, using matrix product states, with many applications to strongly correlated model systems relevant to condensed matter physics [57] and also in quantum chemistry [180]. Recently, considerable effort was made to create a one-dimensional analog of molecular systems using DMRG to handle strongly correlated effects [160], making data generation much more manageable. Such simplicity ensures maximum flexibility and ease in interfacing with existing ML codes, which often come in prepackaged routines.

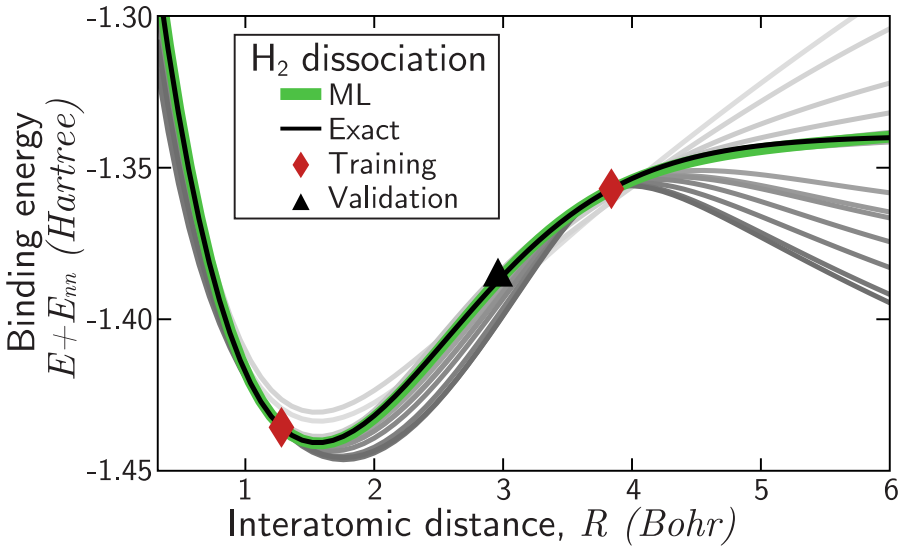


Figure 2.1: The dissociation curve of a one-dimensional H_2 molecule, created using the ML XC approximation of Ref. [93] by training with DMRG data at just two configurations. Darkening shades of grey show predictions from underfitting to overfitting but distributed around the exact curve due to the physics prior knowledge built into the model. The optimal green curve, found by validating the model at a single configuration produces chemically accurate results. E_{nn} is the nucleus-nucleus repulsion energy. See Fig. 2.11 for details.

A helpful introduction to ML for chemical scientists can be found in Ref. [135] with a glossary of terms. Here, we simply distinguish between kernel methods and deep neural networks, the two methods used in the key references. The fundamental problem is that of regression with many parameters, where some regularization method is required to avoid overfitting. Regularization is any procedure that allows one to control how smooth the fit is. Ridge regularization penalizes overfitting with the sum of the squares of the fitting coefficients. The kernel trick maps a low-dimensional space to a higher one to create a function that is more straightforward to fit [61], which is especially relevant in our case. Kernel ridge regression (KRR) remains a standard tool in ML today.

However, many of the most impressive gains in ML have recently come from neural networks (NN). These are characterized by the graph of differentiable operations, architectures with various inductive biases, and scalability on hardware accelerators [69]. Their performance can usually be continuously improved by increasing the model capacity, with copious addition of data, whereas

more traditional methods can saturate or become too expensive to train [156]. The first application of ML to density functional design was using NN [168]. This pioneering work used exact energies and XC potentials to fit an XC functional that remains relevant. In this article, we discuss the chronological developments of ML density functionals focusing only on the work of our research group, but comprehensive reviews are available elsewhere [101].

2.2 Prototype

Here we review the most elementary application of ML to create an approximate OF-DFT functional [153]. The simplest problem imaginable is considering the energy levels of a 1D potential between infinite walls. It is trivial to solve such box problems numerically, filling the levels with same-spin fermions so that there is one particle per level. For N fermions in the box, the KRR kinetic energy functional is:

$$T^{\mathrm{ML}}[n] = \sum_{j=1}^{N_T} \alpha_j k(n, n_j), \tag{2.1}$$

where N_T is the number of training densities, α_j are the weights and k is a Gaussian kernel of the form

$$k(n, n_j) = \exp(-\int d^3r (n(\mathbf{r}) - n_j(\mathbf{r}))^2 / 2\sigma^2). \tag{2.2}$$

The weights α_j are found by minimizing the mean squared error of $T^{\mathrm{ML}}[n]$ for all training data plus a regularization penalty, while σ can be determined by cross-validation. Each data point adds an integral over the entire density inside the Gaussian kernel; hence, the resulting functional is completely non-local.

Three Gaussian potential dips were placed randomly inside the box to generate data. For $N = 1$, with as few as 80 training densities, chemically accurate (error less than 1 kcal/mol) predictions

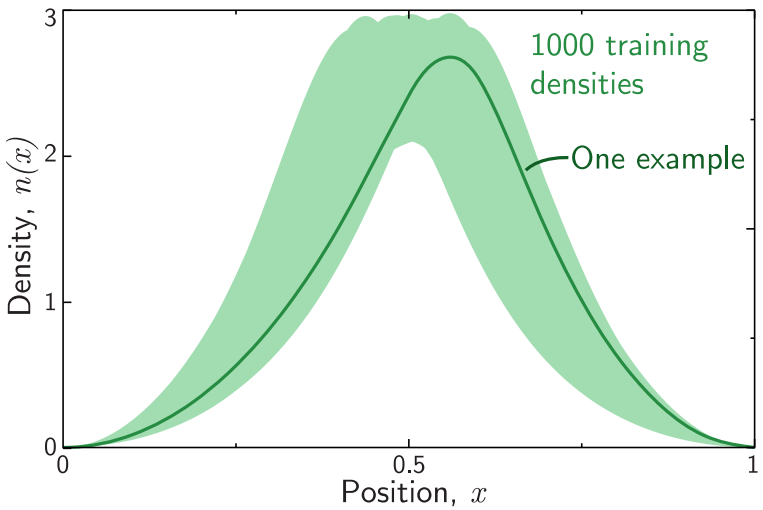


Figure 2.2: The range of variation within the data set of 1000 training densities for $N = 1$ for the box problem (green). These densities can be accurately reproduced using the projection method discussed in Ref. [153]. Adapted with permission from Ref. [153]. Copyright 2012 American Physical Society.

were made for the kinetic energies of a test set drawn from the same distribution, and shown in Fig. 2.2. This was a huge improvement compared to semi-local XC approximations (error = 160 kcal/mol). However, to be useful, an approximate T_S must also have an accurate derivative, so that the Euler equation yields an accurate density [15]. The functional derivative of KRR $T^{\mathrm{ML}}[n]$ has the form,

$$\frac{\delta T^{\mathrm{ML}}}{\delta n(x)} = \sigma^{-2} \sum_{j=1}^{N_T} \alpha_j \left(n_j(x) - n(x) k(n, n_j) \right). \tag{2.3}$$

This derivative is shown in Fig. 2.3. It oscillates wildly relative to the exact curve. This is expected as the exact functional derivative describes the change in the functional in every direction in the infinite-dimensional space of densities, but with KRR, one could only expect it to be accurate in the very few directions in which it has training data.

To overcome this problem, a constraint was added to the minimization, $\delta(E[n] - \zeta g[n]) = 0$, where the functional $g[n] = 0$ defines the manifold of training densities. The specific $g[n]$ can be determined using principal component analysis (PCA) [135]. The cartoon in Fig. 2.4 illustrates this

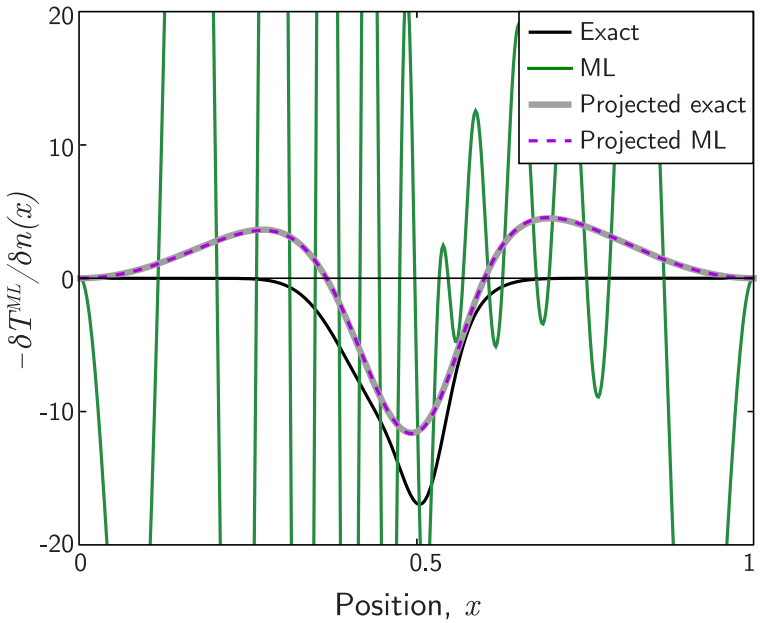


Figure 2.3: Functional derivative of $-T^{\mathrm{ML}}[n]$, the exact derivative, $v(x)$, and their projections on the data-manifold for $N_T = 100$. Adapted with permission from Ref. [94]. Copyright 2015 John Wiley and Sons.

process. One first calculates the usual functional derivative and then projects it onto the local principal components with the greatest variations among the nearby training densities. This leads downhill on the training manifold, and since the optimal density should be within that manifold, it finds a density very close to the exact minimizer. Although the projected derivative is very accurate, as in Fig. 2.3, the error of the functional evaluated on this projected ML density, $T^{\mathrm{ML}}[n^{\mathrm{ML}}]$, is substantially larger than that of T^{ML} on the exact density, chemical accuracy is still achieved with 150 training samples for one particle.

Six alternative kernels were tried, of which three had comparable performance, including the Gaussian used here. A detailed account of all the KRR implementations is given in Li et al. [94]. The details of how the projection method works are also explained, discussing the relative contributions of the energy and density to the error. An analysis of the functional found, and the hyperparameter landscape, is available in Ref. [173].

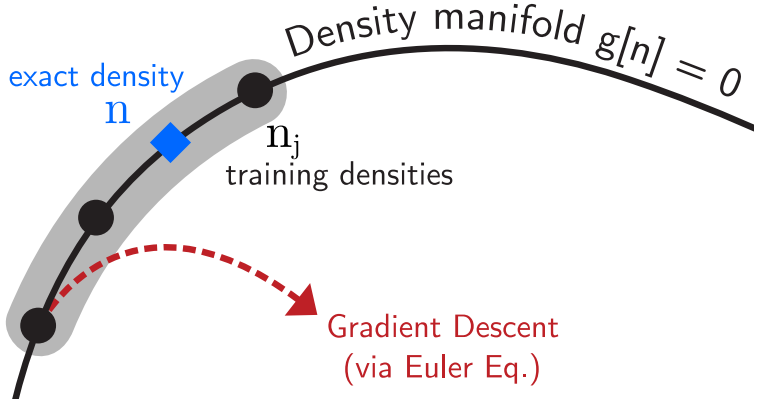


Figure 2.4: The training densities and the exact density are on the density manifold defined by $g[n] = 0$. The solution of the Euler equation via simple gradient descent becomes unstable (red dashed curve) and leaves the shaded region.

2.3 Orbital-Free DFT

Inspired by the proof of principle from Ref. [153], many questions arise as one works toward chemical realism.

2.3.1 Bond breaking

Orbital-free semi-local approximations to $T_S[n]$ fail worse than those for XC when a chemical bond is stretched. An implementation of KRR to correctly describe the stretched bond limit can be found in Snyder et al. [152]. They trained $T_S^{ML}[n]$ with data from KS-DFT along the bond distance of several prototype 1D diatomic molecules and tested if the non-local ML approximation, similar to the one in the box problem, could remain accurate all along the dissociation curve. To tackle the highly curved density manifold, a technique called nonlinear gradient denoising (NLGD) was also proposed. By utilizing kernel principal component analysis (kPCA) [141] to capture the low-dimensionality, this method improves the accuracy of the projected gradient descent with even fewer training densities compared to standard PCA in Ref. [153].

For both H_2 and LiH, the relative error in $T_S^{ML}[n]$ evaluated on the projected density with NLGD

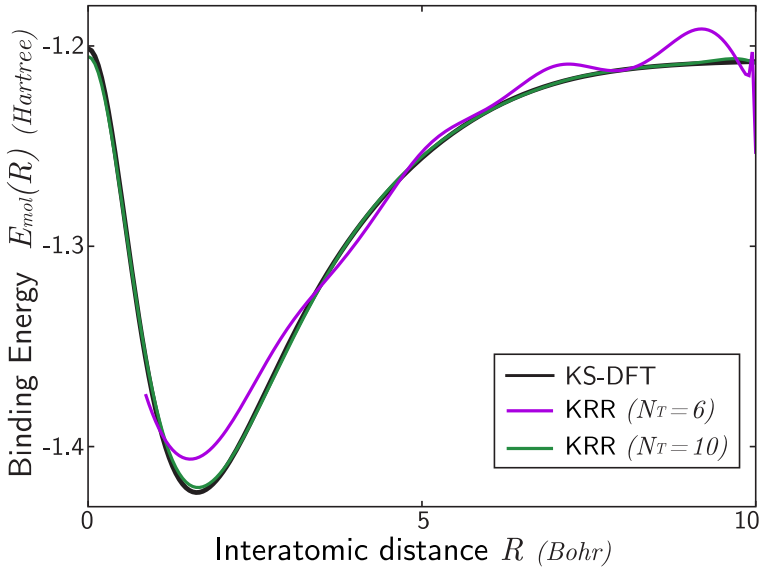


Figure 2.5: The molecular binding energy curve obtained with constrained optimal densities (KRR-NLGD) for 1D model of H_2. Adapted with permission from Ref. [152]. Copyright 2013 AIP Publishing.

was less than 1 kcal/mol with just $N_T = 10$. By increasing the training set size to 20, the bond dissociation energy, equilibrium bond length, and the zero-point vibrational frequency could be determined to be within 1%. Fig. 2.5 depicts how accurately the ML algorithm reproduces the exact binding energy curve of H_2 obtained from a DFT calculation.

The NLGD algorithm is further illustrated in Ref. [154] for the 1D box problem. A 3D expansion of a similar OF-DFT mapping can be found in Ref. [183] where a convolutional neural network predicts the potential energy surface for hydrocarbon chains with accuracy comparable to those of human-designed functionals. Examples of improvements made in human-designed functionals for the same problem can be found in Seino et al. [147] and Golub et al. [54], who trained neural networks for $T^{\mathrm{ML}}[n]$ that included up to third-order and fourth-order gradients of the density.

2.3.2 Exact conditions

In DFT, known theoretical properties (exact conditions) are used to constrain the form of approximate functionals [7, 115, 161]. However, the ML models above cannot be analyzed by checking for such conditions. The weights in the KRR functional are large and alternate in sign, suggesting the possibility of predicting unphysical negative kinetic energy. However, all test densities considered had accurate positive ML kinetic energies, i.e., throughout the training density manifold.

In order to make these KRR functionals less system-specific and to enable easier training, a later study [67] incorporated one of the elementary exact conditions of DFT, the coordinate scaling, within the KRR optimization,

$$T_S[n_\gamma] = \gamma^2 T_S[n],\ n_\gamma(\mathbf{r}) = \gamma^3 n(\gamma\mathbf{r}),\ \gamma > 0. \tag{2.4}$$

Two 1D systems were studied separately- the exactly solvable Hooke's atom, and the H_2 molecule with accurate DMRG energies and densities. After training the KRR model on scaled density n_γ, it was evaluated on a test set of 50 densities for the two systems. Fig. 2.6 shows that in Hooke's atom, the scaled kinetic energy functional was much more accurate than its unscaled counterpart, but not for H_2.

Scaling makes the densities of different configurations of Hooke's atom look similar to one another. However, that is not so for H_2. Hence no improvement is seen in its kinetic energy. This results from the enormous changes in density as you move within the training manifold. Would scaling improve learning if several molecules at different bond distances were simultaneously trained?

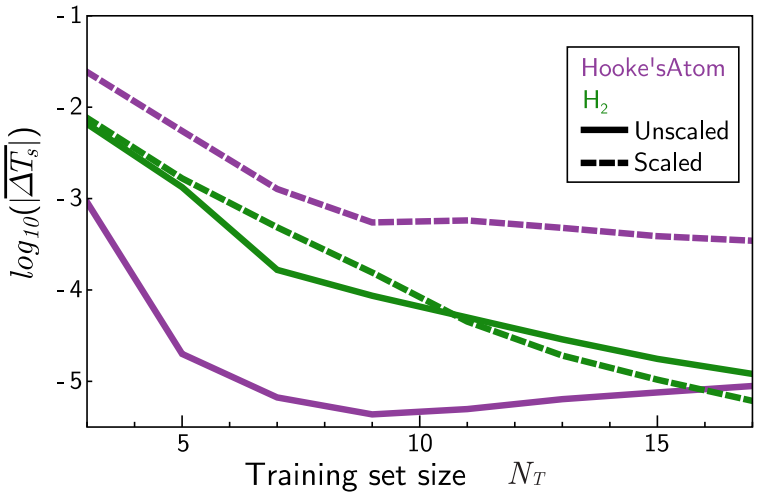


Figure 2.6: The error in the kinetic energy functional trained on scaled and unscaled densities for both 1D Hooke's atom and 1D H_2 molecule. Adapted with permission from Ref. [67]. Copyright 2018 AIP Publishing.

2.3.3 Molecular dynamics of single molecules

New complications arise when ML is applied to chemically realistic problems. Brockherde et al. [11] tried incorporating these methods in realistic 3D electronic structure codes, but as the number of degrees of freedom increased, the cost of the projection method to determine the density became prohibitive. A relatively simple workaround is to learn the density directly as a functional of the potential and bypass the need to solve either the KS equations or the Euler equation. The KRR density and energy models in Ref. [11] were capable of running molecular dynamics (MD) with a standard XC approximation (PBE) for a small organic molecule, malonaldehyde. Training sets were generated by running classical MD simulations at higher temperatures, e.g., 500 K (to ensure sampling of higher energy regions of the potential energy surface), and then performing DFT calculations at snapshots of such simulations. With sufficient training, the errors in the density map became much smaller than density differences due to different XC approximations.

The performance of this ML density functional along the MD trajectory is shown in Fig. 2.7. The error is most prominent in the region where the proton transfer occurs because these configurations are not included in the training set. One could quickly run a KS calculation for this particular

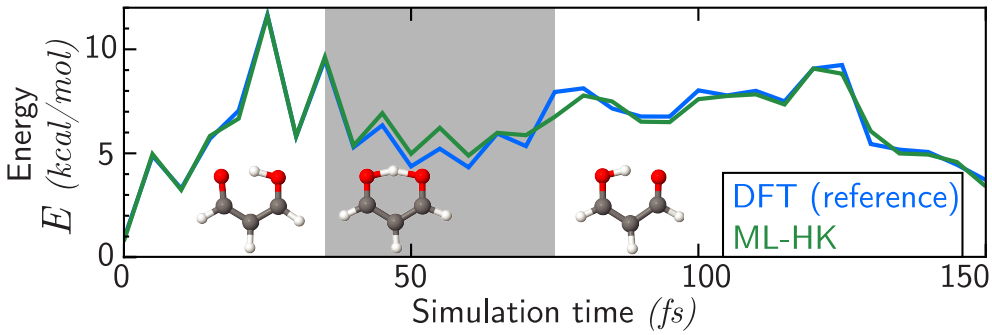


Figure 2.7: Predicted ML energy along a 0.15 ps MD trajectory of malonaldehyde showing the transfer of a proton between oxygen atoms. Adapted from Fig. 5 of Ref. [11]. Copyright 2017 Nature Research, licensed under Creative Commons Attribution 4.0 International.

configuration and retrain including that data point to reduce this error. Standard KS-MD does not yield accurate proton transfer rates, as nuclear tunneling plays an important role and requires more sophisticated approaches [103].

2.3.4 Δ-DFT and chemical accuracy

Although the training data used for malonaldehyde were generated from approximate DFT, in principle, the ML functional could also be trained on energies and densities from higher-level *ab-initio* theories, such as coupled-cluster, i.e. to bypass the KS equations as if they had been solved with chemical accuracy.

In practice, it is difficult to extract accurate densities for training from a CCSD(T) calculation [130], but one can simply learn accurate energies as a functional of the density of a standard DFT calculation. This leads to several different energy functionals that ML can produce: the *ab-initio* energy, the DFT energy, and the difference in the two (Δ-DFT), which is much easier to learn (i.e., converges much more rapidly with training data) because the error in a DFT calculation is a very smooth function of the nuclear coordinates. All this was done in a recent work by Bogojeski, Vogt-Maranto, et al. [9]. Of many different situations studied, the highlight is ML-MD simulations, in which a rotation barrier in resorcinol was probed. A semi-local XC functional makes a substantial

error in the rotation barrier, and Fig. 2.8 shows how the DFT trajectory bifurcates from the accurate trajectory. The KRR-DFT energy on the ML density yields almost perfect agreement with a full DFT MD simulation. Self-consistent DFT corrected with Δ-DFT calculated on the ML density yields trajectories with errors less than 0.2 kcal/mol. Using the ML density with the CCSD(T) energy without performing DFT calculations at each step usually gives a good trajectory but with substantial energy errors. Moreover, directions can appear in a wholly unphysical trajectory taking the molecule outside the manifold on which the density functional works.

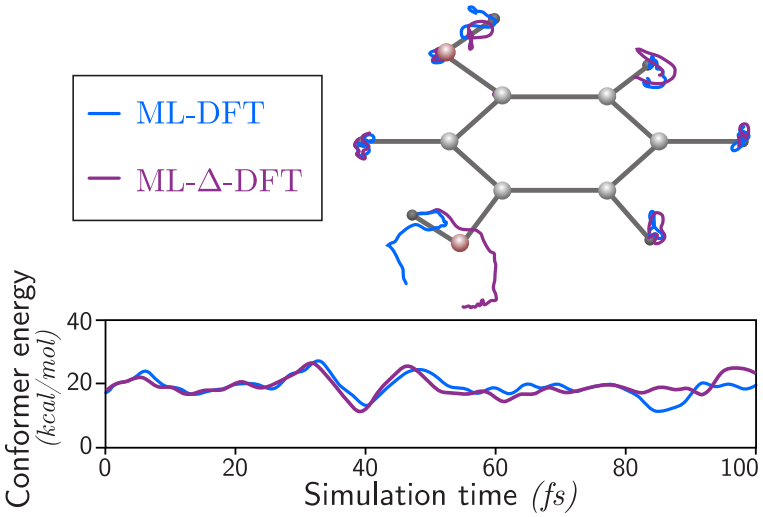


Figure 2.8: Positions and energy of the resorcinol conformer switch predicted using standard DFT alone (blue), and after correction with Δ−DFT trained on CCSD(T) energies (purple). Adapted from Fig. 3 of Ref. [9]. Copyright 2020 Nature Research, licensed under Creative Commons Attribution 4.0 International.

Unfortunately, it is difficult to generalize these methods to other systems or to strong correlation. A similar machine-learned correcting functional was also defined in Dick et al. [33] for liquid water, which used an NN to predict accurate ground-state properties by approximating the difference in energies and forces from the DFT densities. Later, an approximation for XC was also constructed with this method [34].

2.4 Exchange-Correlation:

We turn now to models for XC. Much work in the literature applies to weakly correlated systems. We focus on creating fully non-local ML approximations to handle strong correlation. Because highly accurate densities and energies are cumbersome and expensive to generate for training, we return to the simpler 1D world for testing these ideas.

2.4.1 Strong correlation and thermodynamic limit

For materials applications, the actual strong correlation is even worse than in stretched H_2. For example, for stretched H_4, semi-local XC approximations create four broken spin-symmetry solutions, not two. Ultimately, for solid-state applications, one should be able to handle the infinite chain, or in other words, the thermodynamic limit [160].

In Li et al. [91], the task was to learn both T_S and XC and their derivatives for 1D H-atom chains of fixed separation varying from equilibrium to very stretched and chains varying from two to twenty atoms to extrapolate to the thermodynamic limit accurately. Contrary to Ref. [152], the KRR machinery was applied to DMRG energies and densities to approximate both $T_S[n]$ and $E_{XC}[n]$ in one shot. This was an extremely ambitious goal given the requirement of accurate functional derivatives and the enormous size of the kinetic and Hartree energies. The NLGD method described in the previous sections [152] yields an extremely accurate 1D H_2 dissociation curve. However, this method becomes too costly for longer chains as the number of grid points in the density increases. Without accurate derivatives, one can still easily learn energies but not calculate accurate densities.

The key was the representation of the density. There is too much freedom when it is simply a function of the large grids needed to represent the system. Many alternative representations were tried, but the ultimate winner was the simple atoms-in-molecules partitioning of Hirshfeld [65]. A molecular density of an N-atom chain was decomposed into a weighted sum of distorted atomic

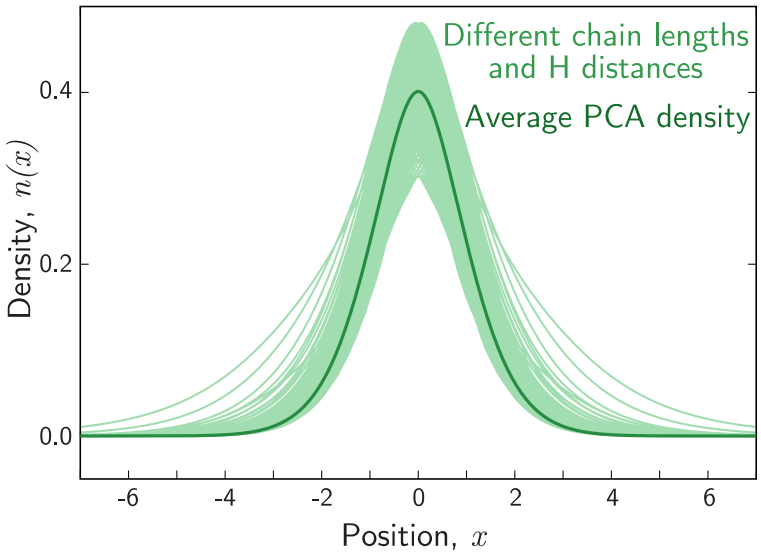


Figure 2.9: Individual Hydrogen partition densities for every interatomic separation *R* within the training set for a chain of length *N* and the base density found using PCA. Note similarity to Fig.2.2. Adapted with permission from Ref. [91]. Copyright 2016 American Physical Society.

densities. After collecting and centering all these atomic densities, PCA was used to create a data-driven basis for the allowed density variations shown in Fig. 2.9. This reduced the time needed to calculate the optimizing densities by several orders of magnitude while retaining chemical accuracy. The infinite-chain limit of 1D H-atoms could then be found with chemical accuracy, treating all aspects of the DFT calculation with KRR on a PCA basis, learned from atoms-in-molecules. DMRG results for both extrapolation of finite chains and periodic systems agreed with each other and with the ML result to within 1 kcal/mol (Fig. 2.10).

On reflection, it would have been much easier to approximate the XC energy alone with ML methods in this calculation and use the KS procedure to produce accurate densities. This seems a worthwhile test for future work and might also have been helpful in Ref. [152].

Other studies have also tried to address strong correlation with ML-DFT on model systems [29, 111, 140]. However, developments are more prominent for weakly correlated systems [89, 99, 46].

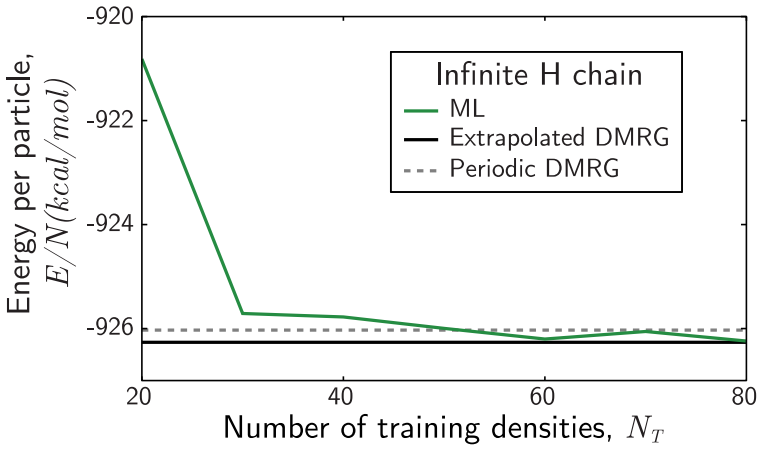


Figure 2.10: Energy of the infinite H-chain with a uniform interatomic spacing of 2.08 Bohr trained using extrapolated DMRG chain densities and energies. Adapted with permission from Ref. [91]. Copyright 2016 American Physical Society.

2.4.2 Kohn-Sham regularizer (KSR)

Here, we look again at full binding energy curves obtained with DMRG to find XC approximations that correctly break bonds, but now within the KS framework. A pioneering study [109] showed that by including density errors in the loss function of a feed-forward NN, one could achieve performance comparable to human-designed functionals for an actual molecule by training on just three or four molecules. This is because density is the functional derivative of the energy with respect to external potential. By training with densities, one simultaneously improves the energy and all possible linear responses to changes in the potential. This greatly enhances the possibilities of generalization.

There are several other efforts to build a transferable ML-DFT model with different approaches [137, 136, 72, 140]. The most recent work by Li et al. [93] pushes the inspiration from Ref. [109] forward in two significant respects. The first is to see if an entire dissociation curve can be found with minimal training on a few examples. The second is a theme of deep learning in general, namely the importance of differentiable programming (DP). DP keeps rigorous components where we have essential physics prior knowledge and well-established numerical methods. By using DP, one can automatically apply gradient-based approaches to optimization, unlike earlier work.

NNs often have many more parameters than training examples and must be regularized. Prior knowledge is usually included via constraints on the network, physics-informed loss functions, or feature preprocessing [67, 142]. Ref. [93] treats the procedure of solving the KS equations as a differentiable program and trains an XC functional using a loss function of density and energy. By backpropagating, the KS equations work as an implicit regularizer for the model. It learns to sample and generate a trajectory from the initial guess density to the exact density during the self-consistent cycle. This improves generalization compared to direct ML models without the KS scheme, such as the KRR models described above, as these models use only the final step results for training and have little information about initial densities.

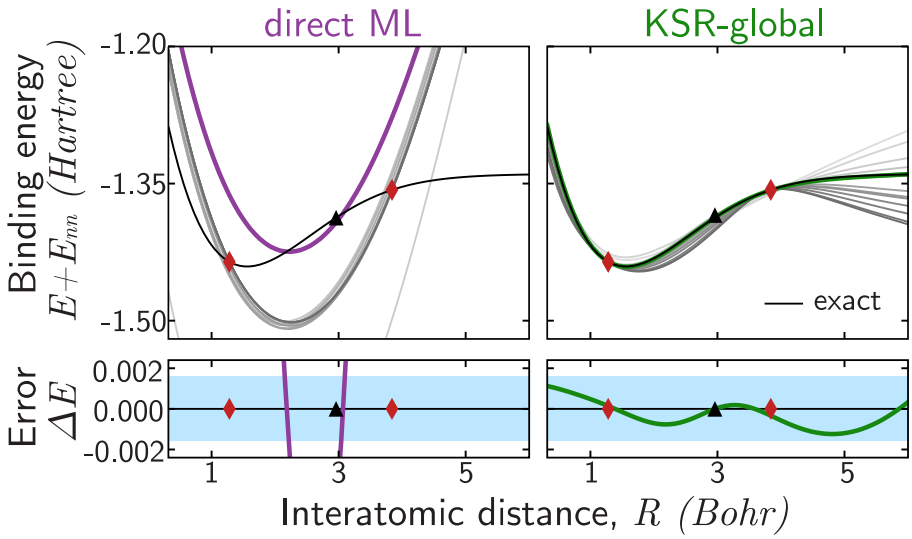


Figure 2.11: One-dimensional H_2 dissociation curve, similar to Fig. 2.5 but with DMRG data instead of KS-DFT. The colored curves are the optimal models trained on two configurations (red diamonds) and validated on $R = 3$ (black triangle). An ML model directly predicting E from geometries overfits the training data. However, the global KSR functional improves with each iteration of the KS equations (grey lines). The lower panel shows that the KSR predictions are within the chemical accuracy limit (light blue region). Adapted from Fig. 1 of Ref. [93]. Copyright 2021 American Physical Society, licensed under Creative Commons Attribution 4.0 International.

The success of the KSR model is apparent from the high accuracy achieved for stretched systems. In Fig. 2.11, the entire dissociation curve of the H_2 molecule is reproduced with chemical accuracy by training at just two separations. A similar performance was reported for H_4. Inclusion of the density loss term generates a much better prediction for the density and the XC potential compared

to energy loss alone. The KSR is transferrable in the sense that it could also predict energies for H_2^+ or two H_2 molecules, even though the model was never exposed to those molecules. A successful extrapolation of this method for 3D real molecules may hold the key for a generalizable practical ML density functional, which can surpass the accuracy of any human-designed functional.

2.5 Outlook

In the arena of OF-DFT, a natural question has arisen. If we can find sufficiently accurate force fields by training on DFT (or better) data, why do we need orbital-free DFT? Won't a force field always be much faster (even if slower than simpler force fields)? The current answer is: maybe. For some specific but significant limited cases, ML force fields are both faster and do not run into difficulties. However, there are problematic configurations that current force fields cannot resolve [128]. Moreover, a DFT calculation can be performed for any combination of any atoms in any configuration, whereas most force fields are designed to explore materials configuration space with one or two elements or chemical compound space with about a dozen elements relevant to medicinal chemistry. A few DFT runs on new combinations of elements and configurations would be cheaper than the cost of new training. Between these two extremes, there is likely room for orbital-free ML-DFT.

However, the main focus is to improve XC approximations. Here, there are two distinct areas. For the weakly correlated systems most often encountered in chemistry and many materials, substantial improvements in accuracy would be incredibly useful and might be achievable by finding better combinations of the many approximate functionals already suggested. For strongly correlated systems (including complete dissociation curves of molecules), going beyond the usual semi-local starting points is likely a requirement, and here, the advantage of ML to create entirely non-local functionals is clear.

Possibly the greatest challenge to creating fully non-local functionals is that of generalizability. We need approximations that can be applied to systems of effectively arbitrary size and boundary conditions (open or periodic). A functional that uses the entire density throughout the system is so sophisticated that training on densities of one molecule is unlikely to yield great accuracy on another and must be retrained for every case. Nevertheless, the simplest and oldest XC approximation, local exchange [35], generalizes perfectly by using only the density at each point to determine its contribution to the XC energy. An ML functional that uses the density within a given radius of the point might improve accuracies for weakly correlated systems but is unlikely to avoid catastrophic failures for strong correlation. The search for the elusive XC functional will continue but now includes machine learning alternatives to human designs.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. DGE 1633631 (B.K.) and CHE 1856165 (R.J.M, K.B.).

Part III

Using Machine Learning to Construct Density Functionals

Chapter 3

Machine Learned Density Functionals with Legendre Transformation

3.1 Introduction

Data-intensive machine learning (ML) algorithms have recently gained attention in quantum and classical computation. ML methods have been successfully utilized in predicting properties of molecules and materials from large databases of Kohn-Sham (KS) density functional theory (DFT) calculations [129, 104, 43], finding potential energy surfaces within molecular dynamics (MD) simulations [81, 112, 145], and in the construction of density functional approximations [153, 152, 94, 91, 183, 177]. To develop a data-driven approach, we must have an underlying pattern in the data. In DFT, such a pattern is confirmed by the Hohenberg-Kohn (HK) theorem [66] which states that the density uniquely determines all ground-state properties of a system. For a non-relativistic many-body problem, it is possible to determine the ground-state energy by splitting the variational principle into two steps via Levy-Lieb constrained search approach [90, 97]. First, the universal part of the functional, $F[n]$, is determined, and a second minimization yields the ground-state

energy:

$$
\begin{aligned}
F[n] &= \min_{\Psi \to n} \langle \Psi | \hat{T} + \hat{V}_{ee} | \Psi \rangle \\
E &= \min_{n} \left\{ F[n] + \int n(\mathbf{r}) v(\mathbf{r}) d^3 r \right\},
\end{aligned}
\tag{3.1}
$$

where $n(\mathbf{r})$ is the one-particle density, normalized over N particles and $v(\mathbf{r})$ is the external potential. By definition, the kinetic energy, $\hat{T}$, and the electron-electron repulsion energy, $\hat{V}_{ee}$, are evaluated over all normalized antisymmetric wavefunctions, Ψ. Almost all practical DFT calculations use the KS scheme to estimate the unknown $F[n]$ [86]. It defines an auxiliary system of noninteracting electrons that has the same $n(\mathbf{r})$ as the interacting system and calculates F in terms of three different contributions,

$$
F[n] = T_{\mathrm{S}}[n] + U[n] + E_{\mathrm{XC}}[n]. \tag{3.2}
$$

The Hartree electrostatic self-repulsion energy, $U[n]$, is exact, and the noninteracting kinetic energy, $T_{\mathrm{S}}[n]$, is given by the self-consistent solution of a one-body Schrödinger equation of the KS system. Hence only a tiny fraction of $F[n]$, the exchange-correlation (XC) energy, $E_{\mathrm{XC}}[n]$, needs to be approximated as a functional of the electron-spin densities [15].

KS-DFT can produce usefully accurate results, with the calculation cost from the diagonalization of the KS eigenvalue equation scaling as $O(N^3)$, for an N-electron system [77]. On the other hand, in the orbital-free counterpart (OF-DFT), the noninteracting kinetic energy is approximated directly as a functional of $n(\mathbf{r})$. The ground-state $n(\mathbf{r})$ is determined self-consistently via Euler-Lagrange constrained minimization,

$$
\frac{\delta T_{\mathrm{S}}[n]}{\delta n(\mathbf{r})} = \mu - v_{\mathrm{S}}(\mathbf{r}), \tag{3.3}
$$

where μ is the chemical potential, and v_{S} is the KS potential. Suppose we can accurately formulate the kinetic energy of the KS electrons. In that case, $T_{\mathrm{S}}[n]$, as a functional of the ground-state density

$n(\mathbf{r})$, OF-DFT can produce results as accurate as KS-DFT with only linearly scaled computational cost [77]. However, since $T_S[n]$ is typically comparable to the system's total energy [36], the relative accuracy of the kinetic energy functional must be much higher than that of XC functionals. Also, the functional derivative needs to be sufficiently accurate to solve the Euler equation to find the self-consistent ground-state density.

The current efforts concerning the construction of machine-learned density functionals (MLDF) aim to provide a computational advantage over KS-DFT through this orbital-free scheme for large-scale molecular calculations. We can define MLDFs as functionals obtained by fitting on a bunch of input data (training set) that can predict the value of the functional on new data (test set). It is relatively easy to obtain data for training since every iteration of every solution of the KS equation yields an exact $T_S[n]$.

The kernelized and regularized form of linear regression, KRR, has been used extensively in the previous ML-DFT studies due to its effectiveness in high-dimensional spaces. It is a method of interpolation constructed by piecing together weighted non-linear kernel functions. There exist several approaches to constructing MLDF:

3.1.1 Orbital-free map (ML-OF)

The non-interacting kinetic energy functional machine-learned with KRR as a functional of n, $T^{ML}[n]$, has the form given in Eq. 2.1 [153, 152, 94],

$$T^{ML}[n] = \sum_{j=1}^{N_T} \alpha_j k(n, n_j).$$

Here N_T is the number of training densities, α_j are the weights, and k is the kernel. For $T^{ML}[n]$, the measurement of similarities between densities is often approximated using a Gaussian kernel,

$$k[n,n_j] = \exp\left(-\int d^3r(n(\mathbf{r})-n'(\mathbf{r}))^2/2\sigma^2\right). \tag{3.4}$$

The weights α_j are found by minimization of the cost function,

$$C(\alpha) = \sum_{j=1}^{N_T}\left(T^{ML}[n_j]-T[n_j]\right)^2 + \lambda\alpha^T\mathbf{K}\alpha, \tag{3.5}$$

where, $\mathbf{K}$ is the kernel matrix, $K_{ij} = k[n_i,n_j]$. The regularization strength, λ, and scale of the Gaussian, σ, can be determined from cross-validation.

The performance of KRR-based ML functionals is driven by the chosen hyperparameters and training set size [61]. One of the main difficulties associated with these approaches is extracting a sufficiently accurate functional derivative of the noninteracting kinetic energy to find the self-consistent density. The functional derivative is expected to be accurate only in directions staying within the manifold spanned by the training set. We can overcome this difficulty with a locally linear projection using principal component analysis (PCA). When an optional density for a system is found in one of these ways, the errors are typically an order of magnitude larger than those evaluated on the exact densities. So the number of training data must be increased to achieve the same level of accuracy as $T^{ML}[n]$.

3.1.2 Hohenberg-Kohn map (ML-HK)

This alternative approach involves direct learning of the KS density from the potential using KRR as a linear combination of Fourier basis functions, $\phi_l(x)$, with coefficients $u^{(l)}[v]$ [11],

$$n^{ML}[v](x) = \sum_{l=1}^{L} u^{(l)}[v]\phi_l(x). \tag{3.6}$$

We can successfully implement this method for molecular energy calculations in 3D with a standard quantum-chemical code. However, although this method can yield exact densities, learning kinetic or total energy requires a second KRR mapping.

3.1.3 Extension to many-body problem

While both ML-OF and ML-HK maps can be much faster than traditional KS-DFT, the accuracy of these ML functionals depends on the exchange-correlation functional used to generate the training set. In an attempt to approximate the universal part of the functional, $F[n]$, a Δ-learning approach based on the ML-HK map has been applied to learn the correction to standard PBE energies with respect to coupled-cluster calculations based on PBE exchange-correlation densities [9]. It was found that machine learning the difference between the coupled-cluster energy and the energy obtained with certain exchange-correlation functional is much more efficient than learning either of them separately. While this is only applicable for weakly correlated systems, Li et al. [91] constructed a more accurate DMRG-trained KRR-based MLDF for $F[n]$ for a variety of one-dimensional hydrogen atom chains that are strongly correlated. An MLDF which contains $T_S[n]$ and $E_{XC}[n]$ in it does not suffer from any of the disadvantages of KS-DFT and can perform well even in the strongly correlated regime.

3.1.4 MLDF with exact conditions

Traditional density functionals usually start from a local or semilocal form. They only include some fraction of exact exchange, using no or just a few empirical parameters (e.g., PBE [115], or B3LYP [7]). So far, machine-learned density functionals have been approximated in an entirely nonlocal fashion requiring many thousands of non-unique parameters [153], and they are not expected to satisfy any of the exact conditions in DFT. These ML functionals are system-specific but do not suffer from some drawbacks of standard functionals that start from a local approximation.

What happens when one introduces exact conditions in constructing these MLDFs? Hollingsworth et al. [67] partially addressed this question in their study involving 1D Hooke's atom and 1D hydrogen molecule. They explored the effect of one of the most straightforward exact conditions in KS-DFT- the uniform coordinate scaling relation. It was found that enforcing the exact condition dramatically improves the learning curve for the 1D Hooke's atom, but not for H_2.

Taking these developments one step further, we have undertaken a different approach to construct a new class of MLDF for the universal part of functional, $F[n]$, which has DFT hard-wired within it through Levy-Lieb constrained search approach [90, 97]. While this MLDF is more compliant with the basic structure of DFT than the ML-HK and the ML-OF maps, it can still offer computational efficiency similar to the orbital-free method. This effort combines the benefits of the traditional DFT and the ML world while simultaneously addressing their drawbacks. Next, in the research progress section, we will discuss the applicability of this approach for a few simple, precisely solvable 1D systems.

3.2 Learning the Universal Part of the Functional with Legendre Transformation

Lieb defines $F[n]$ as a supremum over one-electron potentials [97], so that

$$T[n]+V_{ee}[n] \equiv F[n] \equiv \sup_{v(\mathbf{r})} \left\{ E[v(\mathbf{r});N] - \int n(\mathbf{r})v(\mathbf{r})d^3r \right\}, \tag{3.7}$$

$E[v(\mathbf{r});N]$ being the electronic energy which is a functional of the external potential $v(\mathbf{r})$ and a function of the number of electrons, $N = \int n(\mathbf{r})d^3r$. This Lieb functional can be derived from the variational principle for the energy and a Legendre transformation from the external potential to the electron density. Since $F[n]$ is convex (for $0 \leq \alpha \leq 1$, $F[\alpha n_1 + (1-\alpha)n_2] \leq \alpha F[n_1] + (1-\alpha)F[n_2]$) [97], the exact ground-state energy of the system with N-electrons is obtained by

minimizing the energy functional,

$$E_v[n] \equiv F[n] + \int n(\mathbf{r})v(\mathbf{r})d^3r. \tag{3.8}$$

Lieb maximization is exact in principle. However, if we consider only a finite set of potentials, that is, if we evaluate the Lieb functional as a supremum over only a few $v(\mathbf{r})$, how accurately can we determine $F[n(\mathbf{r})]$ at a different $v(\mathbf{r})$? Contrary to $F[n]$, $E[v(\mathbf{r})]$ is concave in $v(\mathbf{r})$ [97]. For example, if we know the ground-state energy at two potentials v_1 and v_2, for another potential $v = \alpha v_1 + (1-\alpha)v_2, 0 \leq \alpha \leq 1$, the continuous functional $E[v]$ is given by the inequality

$$E[v] \geq \alpha E[v_1] + (1-\alpha)E[v_2]. \tag{3.9}$$

Substituting Eq. (3.9) into the Lieb functional gives,

$$\begin{aligned}
F[n] &\equiv \sup_v \left(E[v] - \int n(\mathbf{r})v(\mathbf{r})d^3r \right) \\
&\geq \sup_\alpha \left(E[v] - \int n(\mathbf{r})(\alpha v_1 + (1-\alpha)v_2)d^3r \right) \equiv \tilde{F}[n] \\
\tilde{F}[n] &\geq \sup_\alpha \left(\alpha E[v_1] + (1-\alpha)E[v_2] - \int n(\mathbf{r})(\alpha v_1 + (1-\alpha)v_2)d^3r \right) \equiv F^{ML}[n] \\
F^{ML}[n] &= \sup_\alpha \left(E[v_2] - \int n(\mathbf{r})v_2 d^3r + \alpha \left(E[v_1] - \int n(\mathbf{r})v_1 d^3r - E[v_2] + \int n(\mathbf{r})v_2 d^3r \right) \right) \\
&= \sup_\alpha \left(F_{v_2}[n] + \alpha(F_{v_1}[n] - F_{v_2}[n]) \right).
\end{aligned} \tag{3.10}$$

We expect to get the exact $F_{v_1}[n]$ when $\alpha = 1$ and $F_{v_2}[n]$ when $\alpha = 0$. This scheme can be generalized to any number of potentials, $v(\mathbf{r})$, and it automatically satisfies the relation $F[n] \geq F^{ML}[n]$. Thus, without external constraints, an in-built exact condition is defined in this MLDF. Characteristics of this approximation were studied with the 1D Hubbard model first.

3.2.1 The Hubbard dimer

The analytically solvable asymmetric two-site Hubbard model in 1D serves as a simple but excellent test system to check the viability of our MLDF. It also presents the opportunity to explore the adaptation of our model in a strongly correlated regime. For a two-site Hubbard model with open boundaries, the Hubbard Hamiltonian is given by [18],

$$\hat{H} = -t\sum_{\sigma}\left(\hat{c}^{\dagger}_{1\sigma}\hat{c}_{2\sigma} + h.c.\right) + U\sum_{i}\hat{n}_{i\uparrow}\hat{n}_{i\downarrow} + \sum_{i}\upsilon_i\hat{n}_i, \tag{3.11}$$

where $t_{12} = t_{21}* = t$ is the hopping integral, U is the Coulomb integral, υ_i is the on-site potential and $\upsilon_1 + \upsilon_2 = 0$. We can find the analytical solution of the model for an integer occupation N [18]. For the noninteracting case, we get the simple tight-binding result for the ground state energy and density,

$$E = -\sqrt{(2t)^2 + \Delta\upsilon^2},\ \Delta n = -2\Delta\upsilon/\sqrt{(2t)^2 + \Delta\upsilon^2}, \tag{3.12}$$

where $\Delta\upsilon$ is the difference in the onsite potential, and Δn is the occupation difference. We initially defined $\Delta\upsilon_i = \frac{y_i}{\sqrt{1-y_i^2}}$ and selected random y-values, $0 \leq y \leq 1$, to generate a sparse training set and calculated $F^{ML}[n]$ using Eq. (3.10). This was named Legendre transformed density functional (LTDF). The self-consistent density can be calculated from the self-consistent ground-state energy by numerical differentiation according to the Euler equation. The approximated kinetic energy and the ground-state density for the tight-binding case are shown in Fig. 3.1.

The accuracy of our MLDF can be improved further with increasing training set size, but it approximates the self-consistent density rather poorly. Thus, by implementing LTDF on one of the most straightforward model systems, we inferred that our MLDF needs further modification to generate accurate predictions for more complex problems.

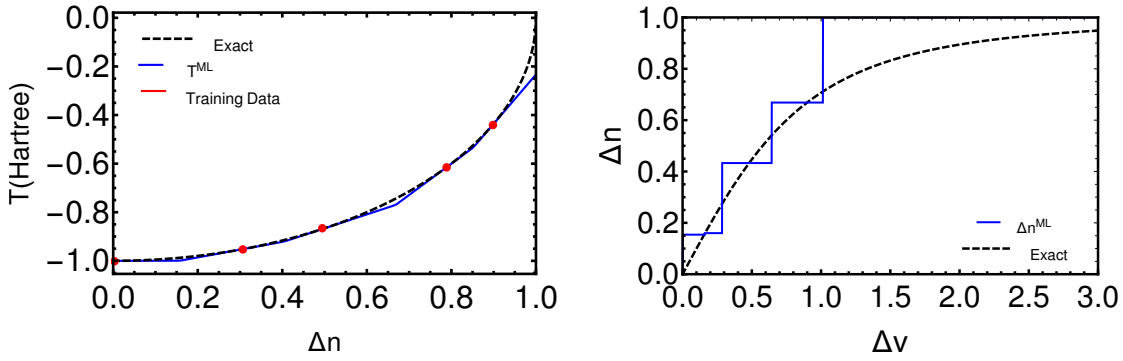


Figure 3.1: $T^{ML}(\Delta n)$ (LT-DF) and Δn^{ML} of the Hubbard dimer for $U = 0$ and $2t = 1$ with five training potentials. $T(\Delta n)$ corresponding to the training potentials are shown in red.

3.2.2 Legendre transformation with Hermite interpolation

In order to generate more accurate density predictions, we tried to modify this simple approximation without improving the training set size. The ground-state energy $E^{ML}(\Delta\upsilon)$ was approximated with a polynomial interpolation method first and then $E^{ML}(\Delta\upsilon)$ was used to get $F^{ML}(\Delta n)$ with Lieb maximization and Δn^{ML} by solving Euler equation. One way to secure a continuous derivative for the Hubbard dimer is through Hermite interpolation (HI). In general, $E^{ML}(\Delta\upsilon)$ can be expressed as a sum of localized Hermite approximations,

$$E^{ML}(\Delta\upsilon) = \sum_{j=1}^{N_T-1}\sum_{i=1}^{M}\left(E_i^j\phi_i^j + \frac{dE_i^j}{d\Delta\upsilon_i^j}\psi_i^j\right), \tag{3.13}$$

where M is the total number of nodes within each element, ϕ_i^j is the Hermite basis function associated with energy E_i at $\Delta\upsilon_i$ within element j and ψ_i^j is the Hermite basis function associated with the corresponding first derivative, $\frac{dE_i^j}{d\Delta\upsilon_i^j} = -\Delta n_i^j$ (for the two-site Hubbard model). We considered $M = 0$, so the second sum vanishes, giving rise to the piecewise cubic Hermite interpolation.

Cubic Hermite interpolation approximates the ground-state energy of the tight-binding model quite accurately, as shown in Fig.(3.2) with an absolute minimum error (MAE) of 0.0017 Hartree. When this energy was used to calculate $T^{ML}(\Delta n)$ with Lieb maximization and Δn^{ML} using the Euler

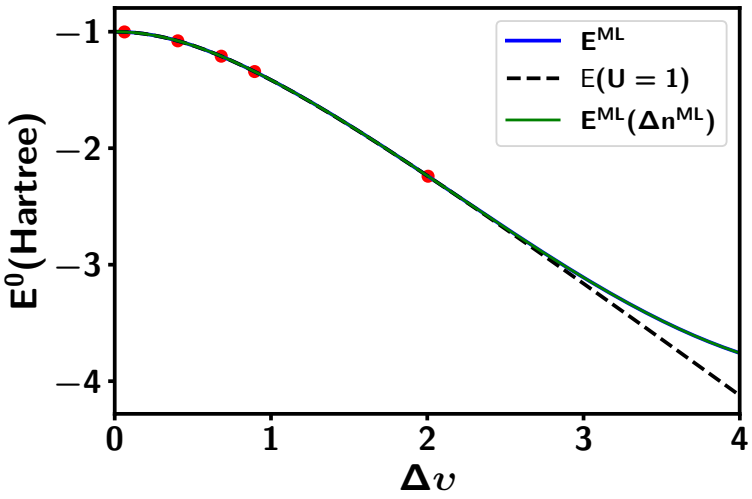


Figure 3.2: $E^{ML}(\Delta\upsilon)$ of the non-interacting Hubbard dimer obtained using piecewise cubic Hermite interpolation (HI-LTDF) with $N_T = 5$. The green line coincides with the blue line and corresponds to the self-consistent energy calculated using Δn^{ML}.

equation with only five training potentials and finite grid-points, the previously observed discontinuity vanishes. The extrapolation error in kinetic energy becomes negligible, and the overall MAE($T(\Delta n)$) is close to 0.0006 Hartree, while the MAE in the predicted density is around 0.0004 Hartree.

Next, we checked the performance of Hermite interpolated LTDF in the weak and strong correlation limit for $U = 0.2, 1, 2, 5$ & 10. Fig. 3.3 shows the predicted density and the kinetic energy for $U = 1$ and 10. As the correlation strength increases, the training set size has to be increased to generate a correct prediction for Δn. However, $F^{ML}(\Delta n)$ is transferable without any extrapolation error, mostly due to U-dependent interpolation. To quantify these errors further, the functional-driven error, $\Delta E_F = E^{ML}(\Delta n) - E(\Delta n)$ and the density-driven error, $\Delta E_D = E^{ML}(\Delta n^{ML}) - E^{ML}(\Delta n)$ were calculated. As shown in Fig. 3.4, the density-driven error starts to dominate over the functional-driven error with increasing correlation strength. With $N_T = 20$, we can still achieve chemical accuracy (1kcal/mol) for $U = 10$. Thus, our MLDF could provide accurate descriptions of the simplest strongly correlated model system, and predictions are systematically improvable with enhanced training.

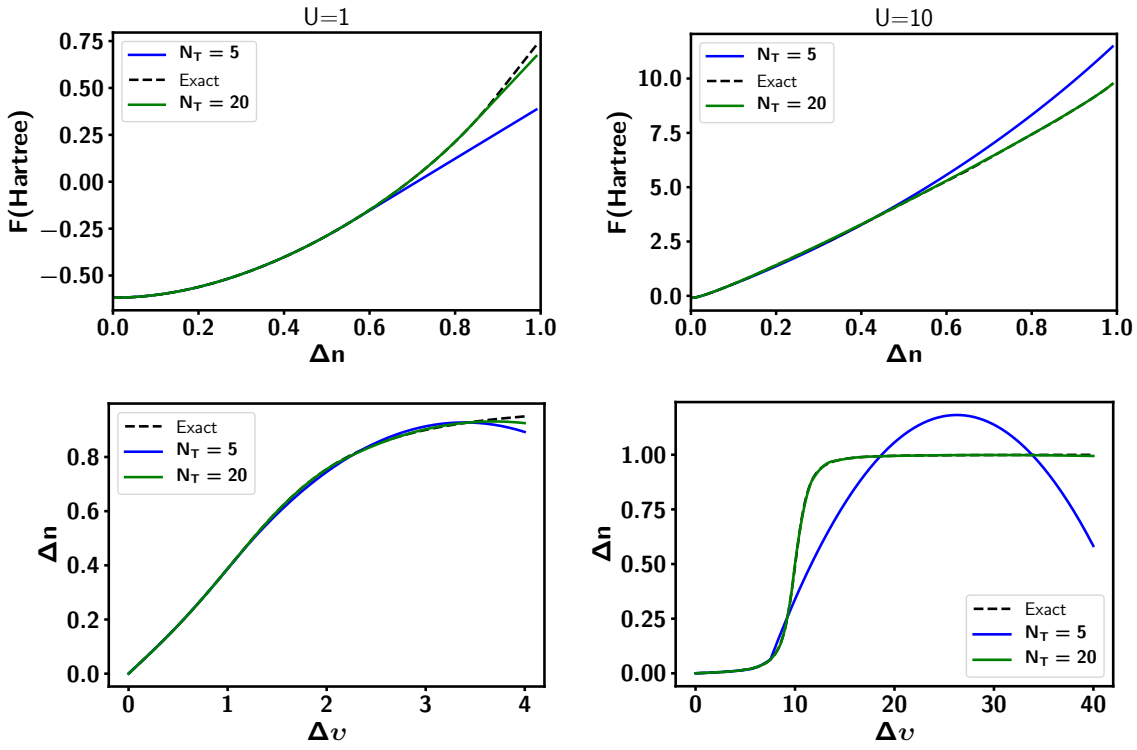


Figure 3.3: $F^{ML}(\Delta n)$ and Δn^{ML} of the Hubbard dimer for $U = 1$ and $U = 10$ at $2t = 1$ with five and twenty training potentials. The training potentials were U-dependent ($\Delta v_i' = \Delta v_i * U$ for $U > 1$).

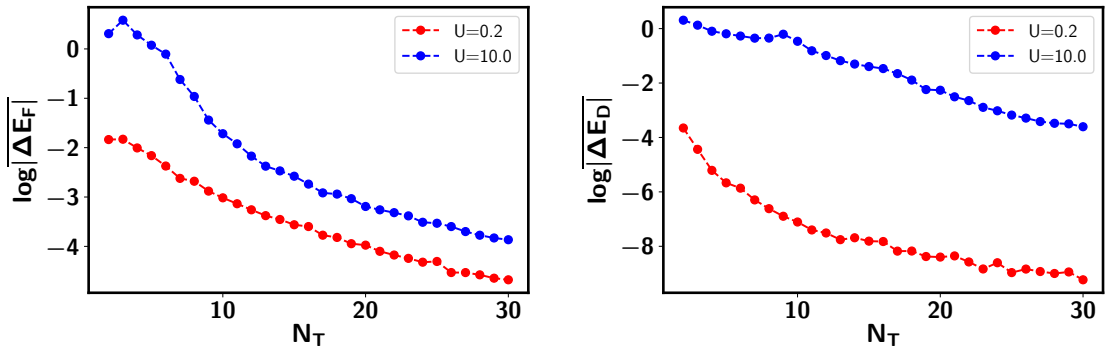


Figure 3.4: Functional-driven and density-driven errors averaged over 100 randomly selected test sets with $N_T = 100$ for the interacting Hubbard dimer at $U = 0.2$ and $U = 10$. Energy values are calculated in Hartree.

3.2.3 Legendre transformation in the real space

Taking another step towards practical applications of our MLDF, we extended our observations further to real simple systems. However, the situation is more complicated in real space even in

1D since $F[n(x)]$ is a functional of the density $n(x)$. We checked the validity of our model for the exactly solvable noninteracting harmonic oscillator case first by defining our kinetic energy functional as,

$$T^{ML}[n(k')] = \sup_{v(k_i,x)} \left(E[v(k_i,x)] - \int n(k',x)\frac{1}{2}k_i x^2 \right), \tag{3.14}$$

k_i refers to the different training potentials, and the kinetic energy is evaluated on a test density, $n(k',x)$. As shown in Fig. 3.5, by performing Lieb maximization over only 4 k_i's, for a test set of densities, $0 \leq k' \leq 3$, we could get a qualitative agreement with the exact kinetic energy for one electron within the interpolation region.

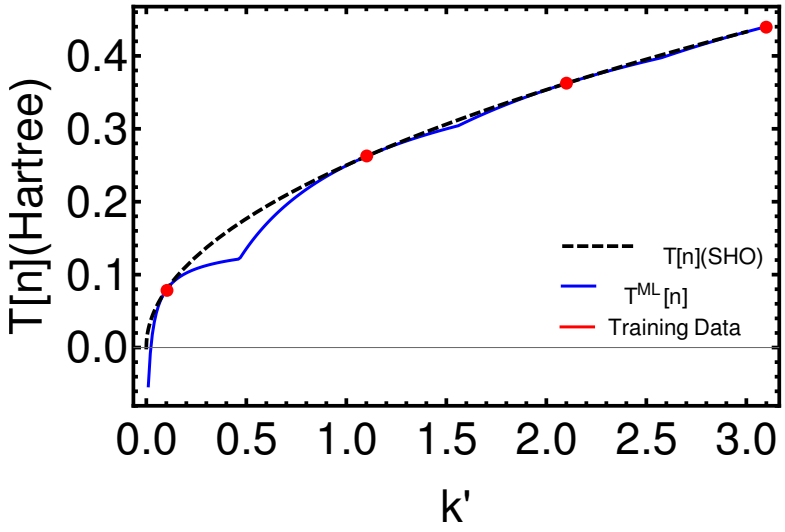


Figure 3.5: Kinetic energy of the one-electron harmonic oscillator in 1D using Legendre transformation with four training potentials. The negative value of $T^{ML}[n]$ near $k' = 0$ is associated with the lack of training for $k_i < 0.1$.

Other two test cases were the one-electron exponential potential, $v(\kappa,x) = -\exp(-\kappa|x|)$ and the delta-function potential, $v(x) = -\alpha\delta(x)$. For the exponential potential, the Schrödinger equation can be converted to the Bessel equation, and the eigenvalues are obtained from spatial symmetry [4]. In both these two cases, with only 4-5 training potentials, our MLDF was able to closely approximate the exact kinetic energy (shown in Fig. 3.6). The reproducibility of the ground-state

density is another issue that has not been addressed yet.

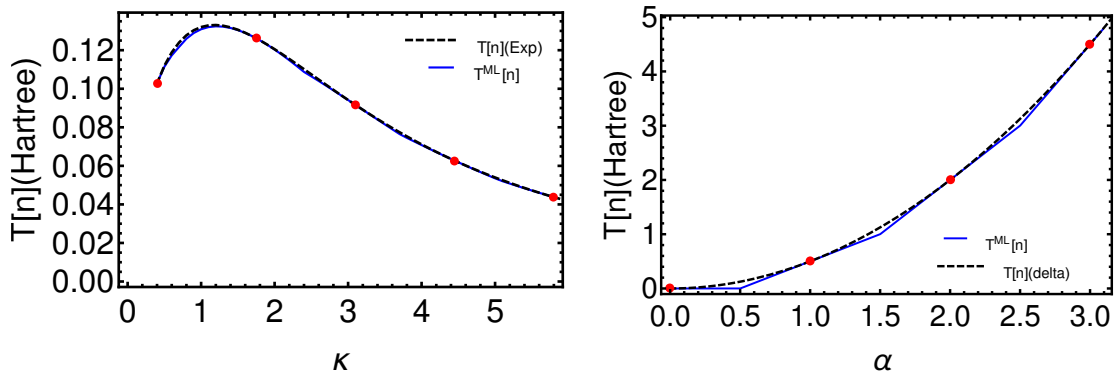


Figure 3.6: $T^{ML}[n]$ associated with the one-electron exponential and the $\delta-$function potentials obtained from Legendre transformation with $N_T = 5$ and $N_T = 4$ respectively. For the exponential case, interpolation was performed within $0.4 \leq \kappa \leq 6$.

Combining these three real space examples can provide further insights into the possibility of constructing a more generalized density functional. However, for any machine-learning model, the essential factors of criticism include lack of transferability and system-specificity. Furthermore, ML generally only works for test cases similar to the training data, and each time we deal with a different problem, we need to train the algorithm first. To address these issues, we extended our observations further by training our MLDF on one type of potential, e.g., delta function potential, and testing it with the density of another type of potential, e.g., harmonic oscillator or exponential potential.

Fig. 3.7 shows that with our MLDF, the delta function or the harmonic oscillator eigenvalues can produce a qualitative approximation to the exponential potential kinetic energy. However, each of them individually underestimates the exact kinetic energy. This underestimation is more apparent for the harmonic oscillator case when delta function potential or the exponential case was used for the training. Nevertheless, regardless of the potential we use, we can reproduce the overall trend of the exact noninteracting kinetic energy $T[n(x)]$ for a given density $n(x)$. This signifies that the current MLDF complies with the fundamental theorem of DFT, and future works should be directed towards overcoming the quantitative difference in such approximations.

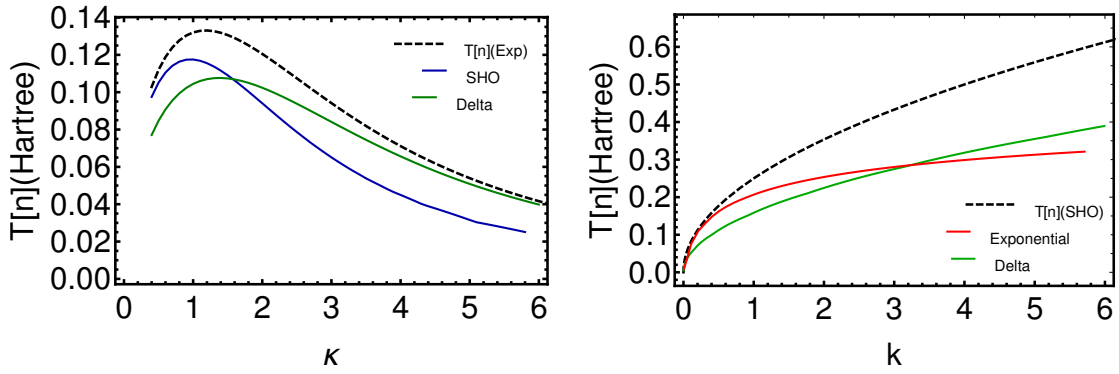


Figure 3.7: $T[n]$ prediction generated with Legendre transformation for the exponential potential kinetic energy using the harmonic oscillator or the $\delta-$function potential with $N_T = 60$. The second figure depicts the same for the harmonic oscillator using exponential or the $\delta-$function potential with $N_T = 55$.

The transferability of the proposed MLDF across different 1D model problems presents us with an opportunity to construct a more generalized MLDF. Current attempts are directed toward improving the undervalued $T[n]$ predictions. If we look at Fig. 3.7, the kinetic energy of the harmonic oscillator is well-reproduced when we train with energies and potentials of the exponential problem for $k < 3$, and at larger k, the delta-function potential performs better. As first step, both the harmonic and the delta function potentials were included at similar training locations and Legendre transformation was performed as an approximation to the kinetic energy of the exponential case, $T^{ML}[n^{exp}(\kappa,x)] = sup_{v(k_i/\alpha_i,x)}\,(E[v(k_i/\alpha_i,x) - \int n^{exp}(\kappa,x)v(k_i/\alpha_i,x)dx)$. The same was done for the simple harmonic oscillator using the exponential potential and the delta-function MLDFs. These two cases are shown in Fig. 3.8. For the harmonic oscillator, the kinetic energy can be reproduced accurately at lower k, but our approach has to be modified further to simulate the correct behavior at large k. On the other hand, for the exponential case, the error is prominent for $0.5 \leq \kappa \leq 5$. Both plots exhibit a discontinuity corresponding to the shift in the potential of choice.

In addition to generalizing the functional, we aim to improve the current approach for the real space systems to determine the self-consistent density with limited training by incorporating the information about the derivative of the energy. While traditional interpolation methods like Hermite

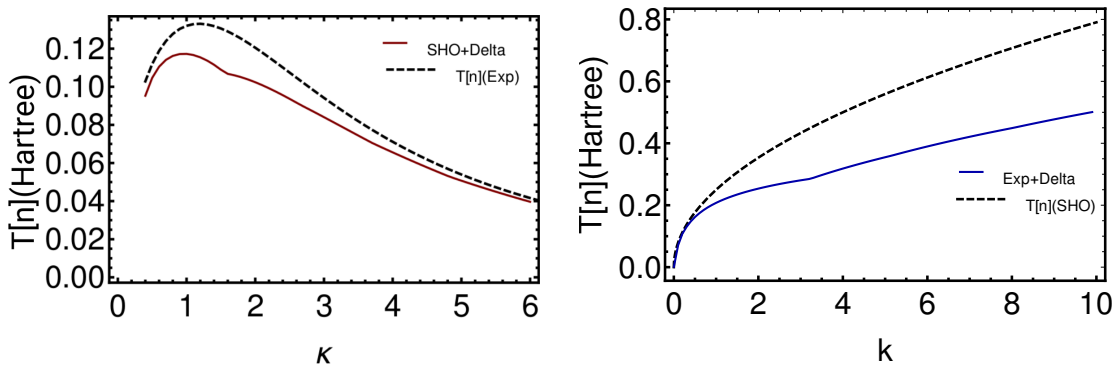


Figure 3.8: Approximated $T[n]$ of the exponential case with a combination of simple harmonic and $\delta-$function potential ($N_T = 60$) and that of the harmonic oscillator obtained from combining the exponential and $\delta-$function potential ($N_T = 55$) respectively.

interpolation can be explored further, introducing other state-of-the-art machine-learning methods might be proven viable for complex 3D systems. However, to proceed to this step and decide which method will be more workable, we need to build a solid foundation of the underlying theory for constructing the MLDF in 1D. We plan on testing our theory for 1D molecules in weak and strong correlation limits using DMRG-generated training sets. If this MLDF can be validated for several 1D model systems- both interacting and noninteracting, with few fixes, we can expand our work to small molecules in 3D.

All the observations so far have been performed with precisely solvable models, and training data was accessible. However, generating training data to learn the universal functional, $F[n]$ using highly accurate couple-cluster calculations or Monte-Carlo simulations will be computationally expensive for actual molecules. Therefore, it is essential to check the validity of our scheme with specific $E_{\mathrm{XC}}[n]$ approximations within KS-DFT. We can also develop a Legendre transform interpolated $\Delta-$learning approach by creating an approximation to the correction to the KS-DFT energies with respect to coupled-cluster calculations. We tried constructing LTDF for the LDA kinetic energy functional with a 1D noninteracting harmonic oscillator as a model system. Fig. 3.9 depicts that although LTDF seems to be compatible with $T^{LDA}[n]$, $|T^{LDA} - T^{ML-LDA}|$ oscillates between positive and negative values, while in the exact case, our approximation consistently undermines

the actual kinetic energy.

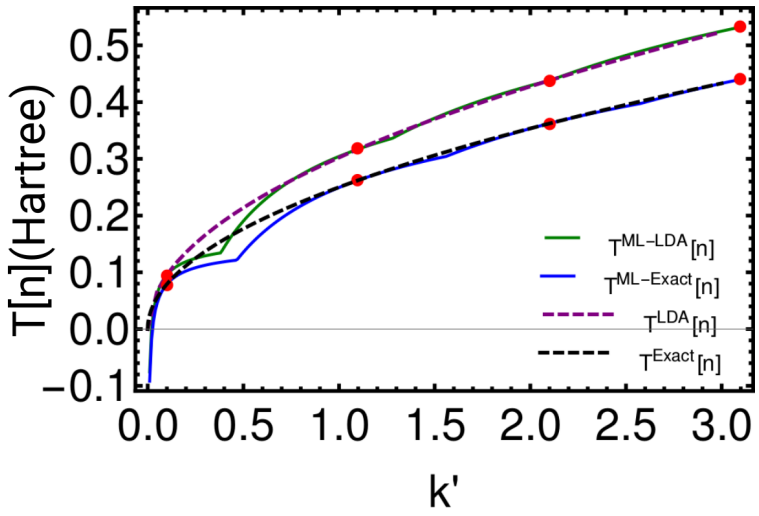


Figure 3.9: $T^{ML}[n]$ approximations generated from Legendre transformation for the exact and the LDA kinetic energies of 1D one electron harmonic oscillator with four training potentials. $T^{ML}[n]$ exhibits similar extrapolation errors near $k' = 0$ for both LDA and the exact cases.

3.3 Conclusion

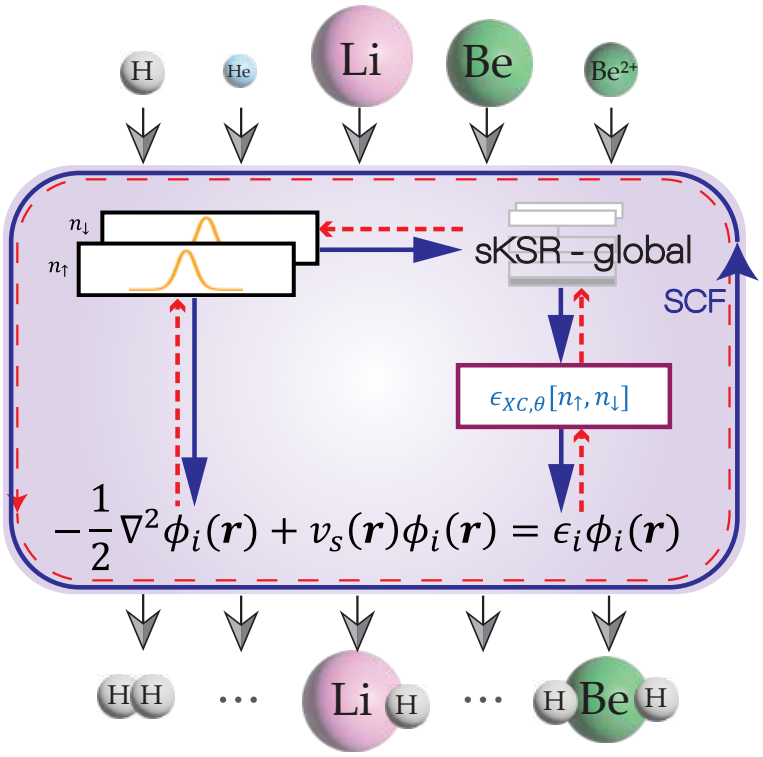
The current MLDF has been defined as an approximation for the Lieb's functional, and hence the fundamental theorem of DFT is incorporated within it. Thus, it should not suffer from the drawbacks we encounter in KS-DFT calculations due to the local and semilocal approximations of the exchange-correlation energy. Although determining if exact conditions are satisfied is less evident in ML-DFT, we can say that our MLDF is also different from all the previous MLDFs, which were approximated in an entirely nonlocal fashion. Thus, as we aimed, the current approach paves a road towards a successful combination of the merits of both the traditional DFT and the ML field, and future works in this direction can potentially give rise to a functional that would be able to produce accurate energies and densities with higher computational efficiency compared to other concurrent KS-DFT methods.

Chapter 4

How Well Does Kohn–Sham Regularizer Work for Weakly Correlated Systems?

written with Ryan Pederson, Jielun Chen, Li Li, and Kieron Burke. Published in *The Journal of Physical Chemistry Letters*, 13(11):2540-2547, 2022.

Abstract: *Kohn-Sham regularizer (KSR) is a differentiable machine learning approach to finding the exchange-correlation functional in Kohn-Sham density functional theory (DFT) that works for strongly correlated systems. Here we test KSR for weak correlation. We propose spin-adapted KSR (sKSR) with trainable local, semilocal, and nonlocal approximations found by minimizing density and total energy loss. We assess the atoms-to-molecules generalizability by training on one-dimensional (1D) H, He, Li, Be, Be^{++} and testing on 1D hydrogen chains, LiH, BeH_2, and helium hydride complexes. The generalization error from our semilocal approximation is comparable to other differentiable approaches, but our nonlocal functional outperforms any existing machine learning functionals, predicting ground-state energies of test systems with a mean absolute error of 2.7 milli-Hartrees.*



4.1 Introduction

Determining the ground-state properties of many-electron systems is fundamental to molecular modeling problems in chemical and material sciences. However, solving the Schrödinger equation explicitly for more than a few hundred electrons is computationally intractable. Among several methods of approximation, Kohn-Sham density functional theory (KS-DFT or simply DFT) [66, 86], a method based on the electron density distribution rather than the many-electron wave function, provides chemically useful results with $O(N^3)$ scaling for an N-electron system [36]. DFT is formally exact, but the exchange-correlation (XC) energy, resulting from the quantum-mechanical interaction between electrons, must be approximated in practice. Hundreds of XC energy functional approximations have been formulated in the past few decades [102]. Functionals can be designed non-empirically, for example using physics and chemical-based intuition and satisfying known exact constraints [115], or can involve some fitting to reference data [190]. However, in any approach, these functional approximations do not yield chemical accuracy in general, that is,

with errors less than 1.6 milli-Hartrees (mH) in atomic units (or 1 kcal/mol). Improving the accuracy of XC functional approximations often incurs additional computational cost in the practical DFT calculation [14]. However, there is no systematic way in general to develop and improve XC functional approximations.

In recent years, machine learning (ML) has been used to find better DFT approximations. Attempts have been made to enhance either the speed or accuracy of DFT. Some used ML techniques to boost computational efficiency by approximating the non-interacting kinetic energy without solving the KS equations [153, 95, 11, 76]. In an effort to improve the accuracy of ML-DFT, a significant leap was achieved by Nagai et al. [109], who used a neural network (NN) model to approximate the XC functional and trained it with high accuracy coupled cluster (CCSD(T)) energies and densities of just three small molecules, while self-consistently solving the KS equations. This functional impressively generalized to 148 small molecules [28] to predict their energies and densities with accuracies comparable to human-designed functionals. However, the test set atomization energies were not chemically accurate. Also, they didn't have access to gradient information and were therefore limited to a gradient-free optimization scheme, which is inherently slow, often suffers poor convergence issues, and is difficult to scale to more complex NN models.

In DFT, many useful properties are extracted from the density, although an XC functional approximation need not produce accurate densities along with accurate energies [82]. In KS-DFT, we calculate the density self-consistently, and there is a nonlinear dependence of the XC functional on the density. Learning this relationship requires not only the ground truth mapping of the functional inputs to outputs but also how the functional performs in the underlying process. Hence the use of differentiable programming [5] becomes more intuitive [71]. With differentiable programming, conditioning the networks with physical insights becomes much simpler, and it can further help to ease the process of training.

Recently, Li et al. [93] made a valuable step in this direction by considering the entire DFT self-consistent calculation as a differentiable program. They implemented an end-to-end differentiable

DFT code for 1-dimensional (1D) systems using JAX [10], a library that provides differentiation, vectorization, just-in-time compilation, and other composable transformations of Python and NumPy programs [58]. They parameterized the XC functional with an NN which incorporated non-local information about the density, along with known physical constraints. The self-consistent KS calculations were embedded into the training process by backpropagating the gradients through the KS iterations. It was dubbed the Kohn-Sham regularizer (KSR). It could yield chemically accurate energies for uniformly separated 1D hydrogen chains at any separation by training on highly accurate energies and densities from only a few separations.

Following a similar approach, Kasim and Vinko [78] implemented an end-to-end differentiable DFT code in 3D for Gaussian-type orbitals and trained local and semi-local NN-based XC functional approximations, evaluating performance on small molecules. In another work, Dick et al. [146] constructed a semilocal XC functional that was carefully curated to account for several known exact conditions and pretrained to match SCAN, a popular meta-GGA functional [161]. While both of these works explore the generalizability of ML approximations for weakly correlated molecules with differentiable DFT codes, they do not incorporate global information, and their accuracy is limited to that of human-designed semilocal functionals. A slightly different approach involves introducing an ML correction term to a nonempirical or partially-empirical XC functional within a KS-DFT self-consistent framework [34, 22]. In such an approach, only a portion of the XC energy is approximated using ML and the functionals retain the characteristics of the baseline XC functional used. The recently proposed ML local hybrid functional, DM21 [85], addresses spin-symmetry breaking and delocalization error in DFT functionals. Consequently, it performs well on several main-group benchmark datasets and also correctly dissociates molecules. Unlike KSR, this functional is trained on large datasets of highly accurate reaction energies (not densities) in the loss function without explicitly supervising the self-consistent iterations.

[C3] Li et al. [93] explored the generalizability of KSR for a few strongly correlated systems with stretched bonds which is a completely different domain from most chemical applications of DFT.

The aim there was to generate accurate binding energy curves (all the way to the dissociation limit) using the entire density (for the nonlocal approximation called global-KSR), using inputs at only two separations, for unpolarized hydrogen chains. The generalizability was in finding the entire bond-dissociation energy curve of these chains. Moreover, only the total density was used and not the spin densities.

In the present work, we propose spin-polarized versions of local, semilocal, and nonlocal XC functional approximations within a differentiable spin-DFT implementation of KSR. We modify these approximations to predict XC energy densities using spin-densities as feature vectors while optimizing the NN parameters using total density and energy loss. Contrary to Ref. [93], we test the KSR approach in the domain of routine DFT calculations in chemistry, namely in and around equilibrium bond lengths. We find the remarkable result that training on energies and densities of a few atoms (and ions) alone produces accurate ground-state energies for equilibrium molecules (very reminiscent of the use of appropriate norms while avoiding using any covalent bond energies). We train and test on a variety of different elements, to obtain the generalizability relevant to chemistry. Almost all previous work in the chemical domain tests various approximate functional forms employing the standard ingredients locally [109, 146, 78]. Our work achieves high accuracy using the total density and is not limited to a specific set of human-chosen features.

4.2 The Spin-Adapted Kohn-Sham Regularizer

The practical implementation of DFT involves solving the Kohn-Sham (KS) equations to calculate the ground-state electron density,

$$\left\{-\frac{1}{2}\nabla^2 + v_{\mathrm{S}}[n](\mathbf{r})\right\}\phi_i(\mathbf{r}) = \varepsilon_i\phi_i(\mathbf{r}). \tag{4.1}$$

The electron density, $n(\mathbf{r})$, is the sum of the probability density over all occupied one-electron KS orbitals, $n(\mathbf{r}) = \sum_i |\phi_i(\mathbf{r})|^2$. The KS potential, $v_{\mathrm{S}}[n](\mathbf{r})$, contains the external one-body potential, the Hartree potential, and the XC potentials,

$$v_{\mathrm{S}}[n](\mathbf{r}) = v(\mathbf{r}) + v_{\mathrm{H}}[n](\mathbf{r}) + v_{\mathrm{XC}}[n](\mathbf{r}). \tag{4.2}$$

The XC potential is the functional derivative of the XC energy, $E_{\mathrm{XC}}[n]$, with respect to the electron density [86], $v_{\mathrm{XC}}[n](\mathbf{r}) = \delta E_{\mathrm{XC}}[n](\mathbf{r})/\delta n(\mathbf{r})$. We can express $E_{\mathrm{XC}}[n]$ in terms of an XC energy density per electron, $\varepsilon_{\mathrm{XC}}[n](\mathbf{r})$:

$$E_{\mathrm{XC}}[n] = \int d^3r\, \varepsilon_{\mathrm{XC}}[n](\mathbf{r})\, n(\mathbf{r}). \tag{4.3}$$

The ground-state energy is calculated from the self-consistent density by summing the non-interacting kinetic energy, T_{S}, the external potential energy, V, the Hartree energy, U, and the XC energy,

$$E_0 = T_{\mathrm{S}}[n] + V[n] + U[n] + E_{\mathrm{XC}}[n]. \tag{4.4}$$

The computational efficiency is also affected by the level of approximation used for the XC functional [120].

Density matrix renormalization group (DMRG) [178] can be used to efficiently generate highly accurate benchmark energies and densities for these 1D analog systems. We can address such systems using 1D KS-DFT calculations as well with suitable XC energy functional approximations, such as the 1D local spin-density approximation (LSDA) which was constructed in Ref. [4] from the 1D exponentially repelling uniform electron gas.

In essence, KSR is a ML-DFT regularization technique that utilizes a differentiable analog of the standard self-consistent DFT computational flow during training to train a suitable parameterized model for $E_{\mathrm{XC}}[n] = E_{\mathrm{XC},\theta}[n]$, where θ are trainable parameters [93]. In this work, we consider

NN-based (neural) XC models, but KSR as a regularization technique can apply more broadly to any differentiable model choice. Knowledge of physical properties and constraints in the exact XC functional can help guide the construction of a neural XC approximation. The NN that parameterizes the XC functional in KSR is carefully curated to account for a few of the expected behaviors of the exact XC functional. Nonlocality is facilitated by adding a global convolution layer in $\varepsilon_{\mathrm{XC},\theta}[n]$ to help capture long-range interactions. The sigmoid linear unit (SiLU or Swish) [38, 131] activation function is used throughout because of its infinite differentiability. The KSR network is also complemented with a self-interaction gate (SIG) that partially cancels the self-interaction error by mixing in a portion of Hartree energy density to $\varepsilon_{\mathrm{XC}}$.

In Ref. [93] several neural XC functional models were proposed: a local functional which only depends on the density at each point (KSR-LDA), a semi-local functional that uses local and gradient information about each point (KSR-GGA), and a global functional which utilized the global convolution layer and the SIG described above (KSR-global).

A main deficiency of the KSR technique in Ref. [93] is that it does not explicitly account for spin, and so may not generalize well for spin-polarized systems. Extending this technique and associated NN models to spin DFT requires a differentiable framework that can backpropagate through resulting spin densities. Spin is often incorporated in the neural XC functional using relative polarization, ζ, as a feature [109]. For up and down spin densities, $\{n_\uparrow, n_\downarrow\}$, $\zeta = (n_\uparrow - n_\downarrow)/n$. While ζ can be introduced as an additional input channel to KSR neural $\varepsilon_{\mathrm{XC}}$, its scale can be very different relative to n in general. Instead, we use up and down spin densities as input features, which have similar scales. The usual models and concepts for KSR can be extended to obtain a spin-adapted KSR (sKSR).

In sKSR-global, we have a global convolution layer that takes spin densities as inputs, and the kernel takes the form:

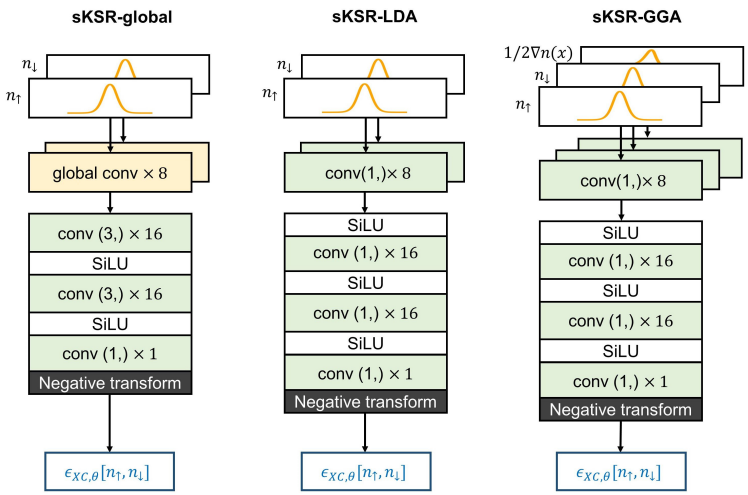
$$G(n_\sigma(x), \xi_p) = \frac{1}{2}\xi_p \int dx' n_\sigma(x') e^{-|x-x'|/\xi_p}, \tag{4.5}$$

where $\sigma \in \{\uparrow,\downarrow\}$ and ξ_p is a trainable parameter that represents an interaction scale. To keep the number of parameters comparable with KSR-global, we input each spin density to a global convolution layer consisting of 8 channels. We then concatenate the output on the channel dimension and input it to the latter convolution layers. For weakly correlated systems and greater generalizability, this approximation does not include any SIG. The rest of the network architecture is kept unchanged. sKSR-LDA and sKSR-GGA approximations to XC are devoid of global information. For sKSR-LDA, two convolution layers with filter size one and 8 channels map the spin-density to $\varepsilon_{\mathrm{XC}}$ at the same spatial point x. In sKSR-GGA, we specify the total density gradient explicitly as an additional input channel along with the spin-densities. Instead of using one convolution layer with filter size three, we use three convolution layers with filter size one and 8 channels each. The rest of the sKSR-LDA and sKSR-GGA architectures are also similar to KSR-LDA and KSR-GGA. Fig. 4.1(a) shows the comparative network structures for all three types of approximations. In all cases, the resulting $\varepsilon_{\mathrm{XC}}$ is symmetrized with respect to the input of the up and down densities:

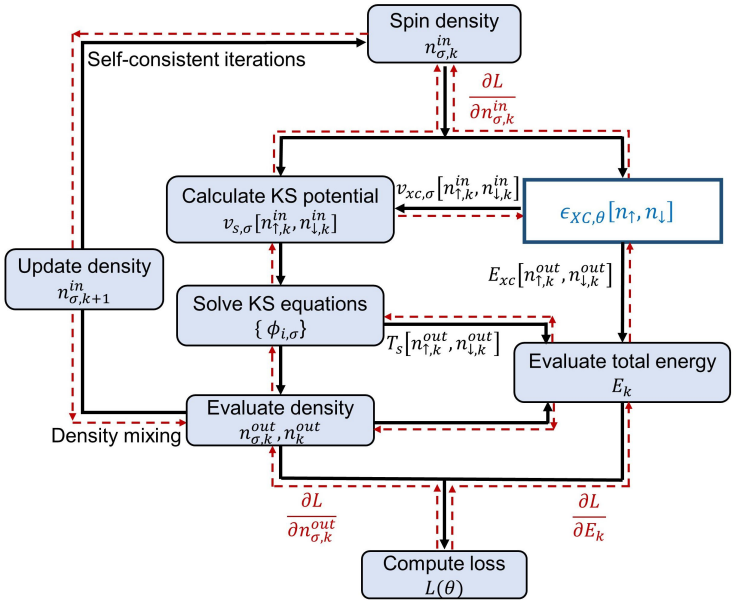
$$\varepsilon_{\mathrm{XC}}^{\mathrm{symm}}[n_\uparrow, n_\downarrow] = \frac{1}{2}\left[\varepsilon_{\mathrm{XC}}[n_\uparrow, n_\downarrow] + \varepsilon_{\mathrm{XC}}[n_\downarrow, n_\uparrow]\right]. \tag{4.6}$$

Our approximation replaces the $\varepsilon_{\mathrm{XC}}$ in a spin-polarized self-consistent KS-DFT framework. For spin-polarized systems we perform the above spin-unrestricted KS-DFT procedure, however for unpolarized systems we use spin-restricted KS-DFT to preserve spin-symmetry. Fig. 4.1(b) shows the conventional computational flow and the flow of the gradients during the self-consistent optimization. To train the neural XC functional, we use the following loss function:

$$\begin{aligned} L(\theta) = \quad & \underbrace{\mathbb{E}_{\mathrm{train}}\left[(E^{\mathrm{sKSR}} - E^{\mathrm{DMRG}})^2/N_e\right]}_{\text{energy loss } L_E} \\ & + \underbrace{\mathbb{E}_{\mathrm{train}}\left[\int dx\,(n^{\mathrm{sKSR}} - n^{\mathrm{DMRG}})^2/N_e\right]}_{\text{density loss } L_n}, \end{aligned} \tag{4.7}$$



(a)



(b)

Figure 4.1: (a) sKSR-global, sKSR-LDA and sKSR-GGA architectures to calculate ϵ_{XC} from spin-densities. (b) sKSR – differentiable KS-DFT with spin-polarization. Black arrows refer to the conventional computational flow. The gradients flow along red-dashed arrows to minimize the loss during training.

where E^{sKSR} and n^{sKSR} are the converged total energy and total density obtained from the neural XC functional approximations, and E^{DMRG} and n^{DMRG} are the exact ground-state electronic energy and total density for each of the test systems. The total loss is evaluated as an expectation over training examples, where N_e is the number of electrons for a given training example. All quantities

are in atomic units. We only consider the converged energy in the energy loss term rather than the energy trajectory throughout KS iterations, which was explored in Ref. [93]. In this work we find that the self-consistent calculations converge quickly for the small atoms and ions used in training, and incorporating energy loss from each KS iteration minimally affects the efficiency of the optimization process. The gradients are calculated based on the total loss with respect to the parameters, θ, through automatic differentiation. They are back-propagated across the self-consistent cycles and the parameters of the neural XC functionals are updated until the total loss is minimized.

4.3 Results

4.3.1 Learning a human-designed functional

As a simple consistency test, we pose the question: can KSR learn human-designed functionals from their observable results? Here we specifically investigate whether sKSR-LDA can learn the relatively simple but general human-designed 1D LSDA XC functional. Since our sKSR-LDA model utilizes hundreds of parameters, it is unclear whether training on just a few LSDA generated DFT results will yield a neural XC model that matches LSDA. We find that by training sKSR-LDA on LSDA-generated He and Li^{++}, we recover the LSDA XC functional almost exactly for unpolarized and fully polarized systems, see Fig. 4.2. The sKSR-LDA model deviates at the high-density limit (low r_s limit) due to the limitation that our training densities only consist of $r_s > 0.5$.

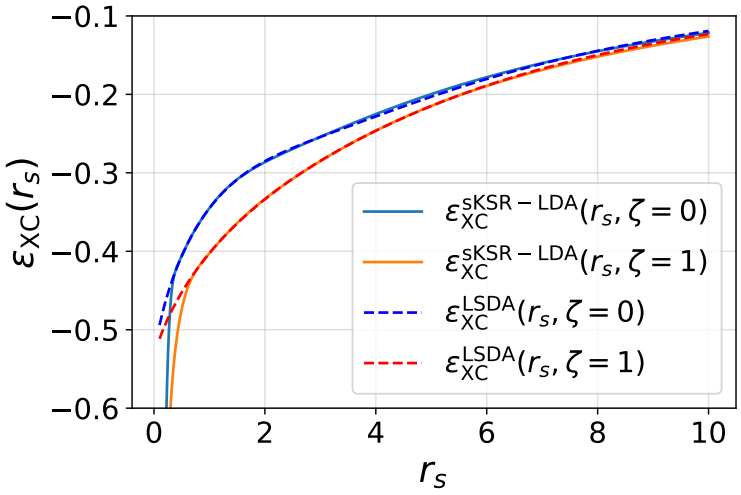


Figure 4.2: sKSR-LDA trained on 1D LSDA-calculated Li^{++} and He energies and densities. Here $r_s = 1/2n$ and ε_{XC}^{unif} corresponds to the XC energy density of the 1D uniform electron gas [4].

4.3.2 Generalizing from atoms to molecules

Next, we assess generalizability by training sKSR models using a few 1D atomic systems and testing on unseen 1D molecular systems. We trained all three models on DMRG energies and densities of H, He, Li, Be, and Be^{++} and validated on Be^{+}. For training and validation details, see Appendix. The trained model was later used to calculate the properties of several molecules in their equilibrium ground-state or relaxed form (see Table 4.1). The errors in total energies, ionization, and atomization energies, as well as the average density losses for all three neural XC functional approximations, are reported in Table 4.2. Compared to LSDA, the mean absolute error (MAE) in sKSR-LDA calculated energies is reduced by a factor of three. On the other hand, sKSR-global is an order of magnitude higher in accuracy and yields total energies with an MAE of 2.7 mH, not so far from the chemical accuracy limit of 1.6 mH. The cumulative MAEs for the training, validation and test datasets are reported in Appendix.

The importance of spin in sKSR can be seen by comparing results with the original KSR-global model from Ref. [93]. For a valid comparison, we consider KSR-global without the SIG and train it with the sets from Table. 4.1, without adding the energy trajectory loss. The MAE in KSR-global

Table 4.1: Training, validation and test sets for generalizability experiment. The molecules in the test set refer to the relaxed structures.

Training	**Validation**	**Testing**
H, He, Li	Be^{+}	H_2, H_3, H_4, H_2^+, H_3^+
Be, Be^{++}		LiH, BeH_2, HeH^+
		H-He-He-H^{2+}
		He-H-H-He^{2+}

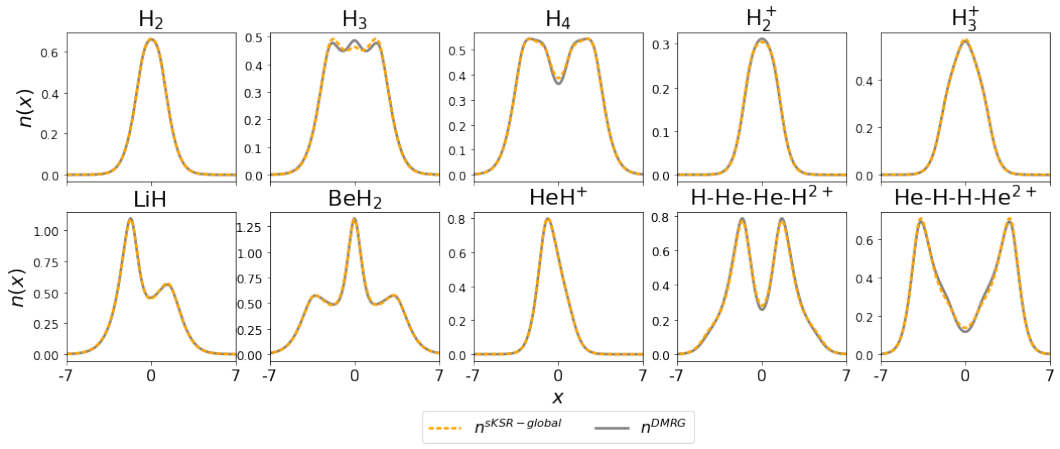
Table 4.2: Total energy errors (in mH), density losses (in 10^{-4} Bohr^{-1}), and errors in ionization potentials for atoms and atomization energies in molecules (in mH) calculated using uniform gas LSDA [4], sKSR-LDA, sKSR-GGA, and sKSR-global respectively, for the training, validation, and test sets in Table 4.1.

Dataset	**Symbol**	**LSDA**			**sKSR-LDA**			**sKSR-GGA**			**sKSR-global**		
		ΔE	L_n	ΔIP	ΔE	L_n	ΔIP	ΔE	L_n	ΔIP	ΔE	L_n	ΔIP
Training	H	26.6	5.35	-26.6	4.51	0.55	-4.50	4.49	0.31	-4.49	0.85	0.33	-0.85
	He	41.4	2.89	-8.46	20.2	0.63	-21.3	7.49	0.24	-10.2	-0.69	0.03	0.62
	Li	33.7	5.02	16.6	-11.5	0.40	37.4	-12.0	1.37	20.2	-2.37	0.12	2.79
	Be	24.5	1.18	21.4	-23.5	1.03	12.1	-2.70	0.65	-5.29	1.16	0.07	-1.23
	Be^{++}	55.3	0.75	-18.1	29.2	0.16	-46.1	6.55	0.49	-34.1	0.41	0.02	-1.43
	MAE	36.3	3.04	18.3	17.8	0.56	24.3	6.65	0.16	14.8	1.10	0.12	1.38
Validation	Be^{+}	46.0	1.95	9.37	-11.3	0.12	40.5	-7.99	0.61	14.5	-0.07	0.03	0.49
				ΔAE			ΔAE			ΔAE			ΔAE
Test	H_2	34.04	1.82	19.2	19.5	0.35	-10.5	6.83	1.99	2.14	-0.73	0.07	2.43
	H_3	35.6	1.93	44.3	0.45	0.21	13.1	-3.07	5.57	16.5	-3.56	3.22	6.11
	H_4	32.3	3.82	74.3	7.66	1.59	10.4	-9.34	4.18	27.3	2.87	1.46	0.53
	H_2^+	19.6	6.68	7.09	2.78	0.71	1.73	1.68	1.71	2.81	-1.94	1.04	2.79
	H_3^+	31.2	0.78	22.1	20.6	1.87	-11.6	15.4	11.5	-6.44	-0.40	0.47	2.09
	LiH	30.9	3.72	29.5	-8.55	2.47	1.53	-16.6	3.86	9.14	-4.38	0.66	2.86
	BeH_2	32.8	7.49	45.0	-27.8	5.5	13.4	-34.6	3.09	40.9	-5.07	1.29	7.93
	HeH^+	37.3	1.71	4.18	18.8	0.17	1.40	5.18	0.59	2.31	-1.60	0.13	0.91
	H-He-He-H^{2+}	36.7	14.7	46.1	5.00	6.00	35.5	-9.04	2.50	24.0	5.39	4.52	-6.77
	He-H-H-He^{2+}	46.1	7.40	36.7	19.9	6.48	20.6	4.35	4.75	10.6	0.79	5.47	-2.18
	MAE	33.6	5.00	32.9	13.1	2.53	12.0	10.6	3.98	14.2	2.67	1.83	3.46

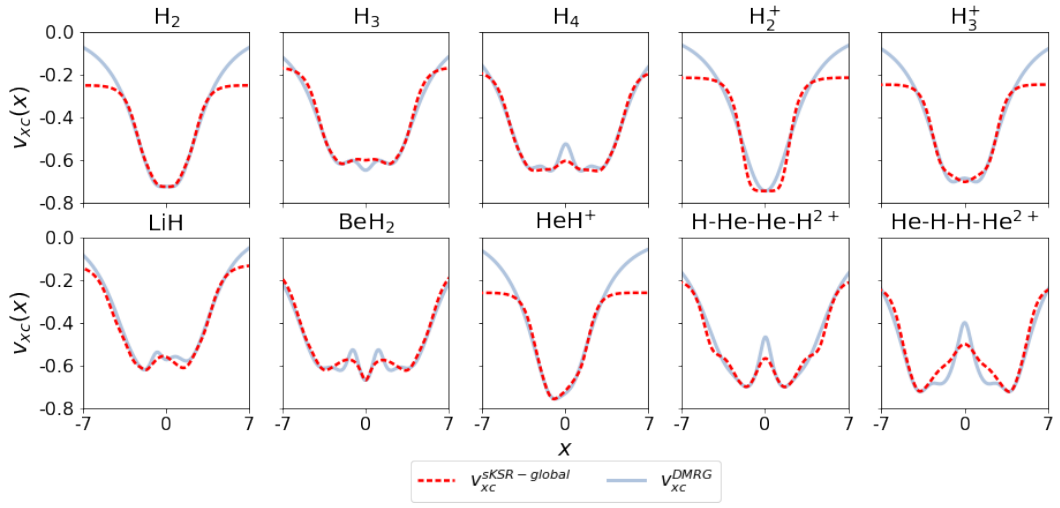
predictions for total energies of the test molecules is 10.02 mH, comparable to sKSR-GGA, but much worse than sKSR-global (see Table. A2 in Appendix). sKSR-global also converges more quickly than KSR-global, reaching lower training losses with fewer training steps (see Fig. A7 in Appendix).

The size of our dataset is practically limited by the chemical space provided by 1D and the associated exponential interaction. Even though we are dealing with a much smaller dataset, we trained the sKSR models on the ground-state energies and densities of 5 atomic systems only and did not include any molecules, contrary to results in Ref. [109] and Ref. [78] which train on derived quantities, such as atomization and ionization energies, and include molecules in training.

Using sKSR-global, the predicted densities of each molecule have little noticeable error, see Fig. 4.3(a). The corresponding XC potentials are shown in Fig. 4.3(b). For all unpolarized systems, we run restricted KS calculations, and the up and down XC potentials match, while for polarized systems (Li, Be^+, H_2^+, and H_3 only) we run unrestricted KS calculations. The sKSR-LDA and sKSR-GGA total densities and XC potentials for the test set are included in the Appendix. The



(a)



(b)

Figure 4.3: (a) The densities obtained using sKSR-global (orange dashes) and the exact ground-state densities (gray), (b) average XC potentials calculated from sKSR-global approximation (red dashes) to ε_{XC} and their exact counterparts calculated with DMRG (light blue) for the test molecules in Table. 4.1 at equilibrium separations. The sKSR potentials are shifted by a constant for a better comparison with the exact XC potentials. sKSR-global was trained on H, He, Li, Be, and Be^{++} and validated on Be^+. Note that, in general, these 1D densities and XC potentials can differ even qualitatively from their 3D analogs.

comparison to exact XC potentials is not expected to be as precise as potentials are extremely sensitive to densities. However, for each of these examples, we see that the sKSR-global XC potential closely mimics the exact XC potential, even though we did not include XC potentials in the training. Furthermore, seemingly large deviations in the XC potentials can result in similar resulting densities. For example, this can be seen in the case of BeH_2 where the XC potentials are noticeably different but the resulting densities are very similar. The KS potentials are reasonably accurate for the test set (see Appendix). Note that similar to the exact XC potentials, the sKSR-global XC potentials are smooth, due to the use of a smooth activation function.

We can use these potentials to validate the known theoretical properties of the exact XC potentials for different test systems, compare with other XC approximations, and utilize them to introduce corrections to existing local and semilocal approximations. Similarly, sKSR-global can also produce quite accurate spin-densities even though we did not incorporate spin-densities in the loss function while training the XC functionals (see Fig. A2 in Appendix).

4.3.3 Generalizing to strong correlation

A very interesting question is: how does our weakly-correlated sKSR behave for strongly-correlated systems? We answer this by studying the paradigm case of the H_2 binding curve in Fig. 4.4, where the sKSR-global curve remains highly accurate up to at least 3 Bohr. Just as with all single-particle methods, the restricted calculation yields energy that is far too high in the dissociated limit. On the other hand, an unrestricted calculation, which breaks spin-symmetry beyond about 4 Bohr, does dissociate correctly, but at the price of poor spin densities and a kink in the binding energy curve. Fig. A6 in Appendix shows analogous features for sKSR-LDA and sKSR-GGA, and also shows the accuracy of the total density of the unrestricted solutions at large separations. Fig. 4.4 also shows the result of a KSR-global calculation (i.e., total density only), but trained just on atoms. While it naturally dissociates correctly, it is much less accurate. Of course, the sKSR-global of

Ref. [93] is chemically accurate for the entire curve because its training included a stretched bond.

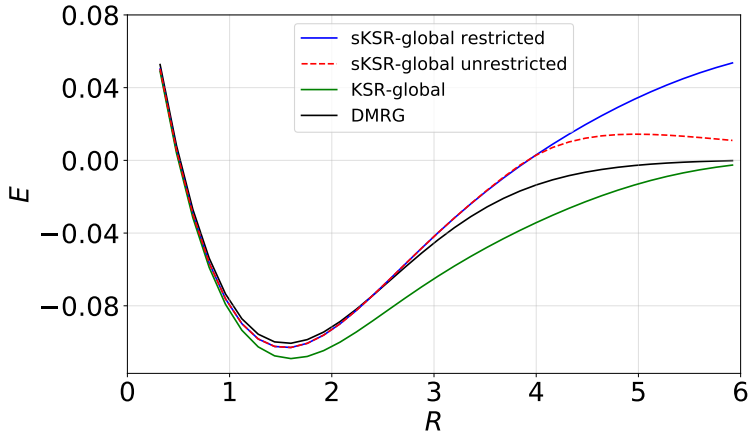


Figure 4.4: The binding energy curve of H_2 molecule calculated based on the total energy prediction for H_2 molecule and the energy of the individual H atoms. sKSR-global was evaluated using restricted KS (blue) and unrestricted KS (red dashes) scheme. The DMRG (black) and KSR-global (green) results are also shown. All the neural approximations, with and without spin, are trained on the dataset given in Table. 4.1.

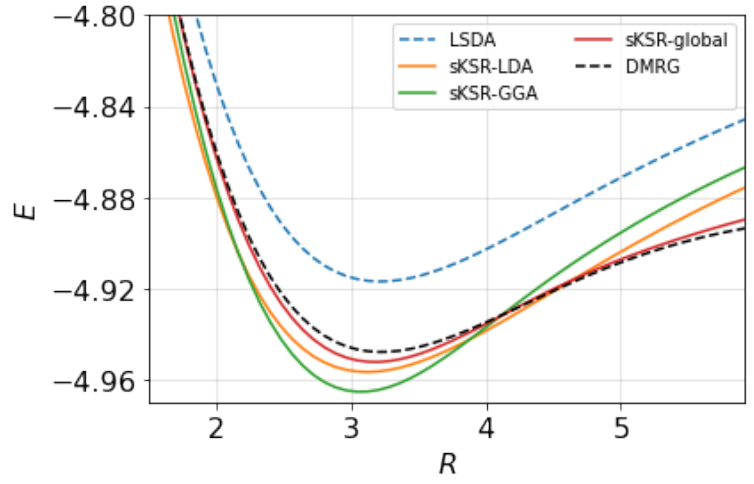
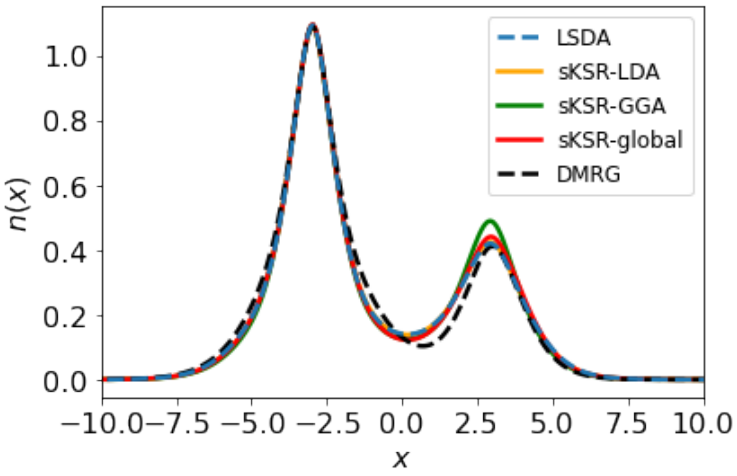


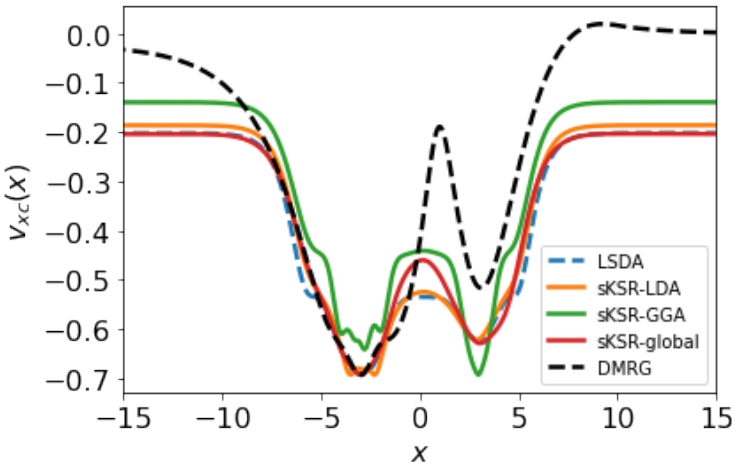
Figure 4.5: The complete dissociation energy curve of LiH molecule generated with sKSR-LDA (orange), sKSR-GGA (green) and sKSR-global(red). The DMRG (black dashes) and the uniform gas LSDA (blue dashes) results are also shown. The neural XC functional approximations were trained and validated on atoms and ions given in Table. 4.1.

In many cases, the predictability of sKSR can extend well beyond the equilibrium bond distance.

Fig. 4.5 shows the complete dissociation energy curve of LiH obtained from restricted calculation.



(a)



(b)

Figure 4.6: (a) The total density and (b) the average XC potentials of LiH at a bond-distance of 5.92 Bohr calculated with the three neural XC functionals as well as uniform-gas LSDA. The exact (DMRG) average XC potentials are included for comparison.

Near equilibrium, sKSR-LDA and sKSR-GGA underestimate the binding energy but perform better than LSDA. As the bond is stretched, sKSR-GGA and sKSR-LDA quickly deviate from the expected trajectory. However, sKSR-global performs well throughout, extending its predictive accuracy well beyond the equilibrium bond distance. We show the total density and the XC potential of stretched LiH at 5.92 Bohr in Fig. 4.6. LSDA largely overestimates the total energy of the stretched molecule, but its density remains reasonably accurate. The XC potentials calculated from neural XC functional approximations are comparable, with sKSR-global closely approximating the exact behavior. A comparison of the sKSR-global and the exact total density and XC potential of stretched LiH with respect to the atomic contributions from Lithium and Hydrogen is included in the Appendix.

The approximate total energy of a molecule can have two types of error contributions: the error due to the approximate functional and the error arising from the self-consistent density [174]. For most XC functionals, the total density calculated from the self-consistent solution of the KS equations works as an excellent approximation to the exact density for most systems. Hence, the density-driven error is often negligible. However, some approximations can have significant density-driven errors [84]. For our test molecules, the errors in the self-consistent densities were trivial and consequently had minimal impacts on the atomization energy errors. The functional and density-driven errors in our neural XC functional approximations are reported for the hydrogen molecule in the supplementary information section.

4.4 Conclusion

We found that sKSR-global achieves remarkable accuracy and generalization for 1D systems in a very data-efficient manner by including the self-consistent KS equations into the training. sKSR-global predicts the ground-state energy of ten unseen 1D molecules in equilibrium with a mean absolute error of 2.7 mH ($\sim$1.7 kcal/mol) when trained with just five atomic and ionic systems.

Hence, a nonlocal XC functional approximation trained on atomic energies and densities has the potential to generate chemically-accurate predictions for most 1D weakly-correlated molecules. An extension of the nonlocal approximation to real systems can lead to an ML functional that is applicable across a broad chemical spectrum without using an exceedingly large training set. The end-to-end differentiable implementation also ensures smooth and reasonable XC potentials. In addition, sKSR-global trained on atoms can adequately describe a molecule with a stretched bond. Combining differentiable programming with inherent physical intuition thus takes us one step closer to a generalizable, chemically accurate ML XC functional.

The application of the current sKSR algorithm is limited to 1D systems and our test set does not include real 3D molecules. However, the methods presented are transferable to 3D and we anticipate that the characteristic performance is not unique to 1D systems, as these systems tend to mimic their 3D analogs [176]. The low-dimensional examples are useful for quick and rigorous assessment of the quality of an approximation. Besides, the predictions from the local and semilocal approximation explored in our study are consistent with the 3D differentiable formulations in Ref [78] and Ref. [146].

Acknowledgement

This work is supported by National Science Foundation, grant no. DGE-1633631 (B. K.), CHE-1856165 (B. K., K. B.), and Department of Energy, grant no. DE-SC0008696 (R. P.).

Data Availability Statement

The training and testing data and the one-dimensional density functional theory solver used for the uniform electron-gas LDA calculations are available at https://github.com/pedersor/DFT_1d. The

ML models and the JAX version of the DFT code are available from the corresponding author upon reasonable request.

4.5 Appendix

4.5.1 Calculation details

4.5.1.1 Data generation

All training data are generated from 1D DMRG calculations [178] with exponential approximation. We choose the electron-electron interaction to be exponential,

$$v_{\text{exp}}(x) = A\exp(-\kappa|x|), \tag{A1}$$

where the parameters, $A = 1.071295$ and $\kappa^{-1} = 2.385345$ are adjusted to mimic soft-Coulomb interaction [4]. Similarly, the external potentials for a 1D molecular system are expressed as

$$v(x) = -\sum_j Z_j v_{\text{exp}}(x - x_j), \tag{A2}$$

where Z_j is the nuclear charge and x_j is the position of the j^{th} nucleus. This allows us to create 1D analogs of atomic systems and linear molecules, such as BeH_2. The extended Hubbard-like Hamiltonian [4] for 1D systems is solved in real space on a grid of 513 points within the range $x \in \{-20.48, ..., 20.48\}$ with a separation distance of 0.08 Bohr and center at $x = 0$. Calculations are done using the ITensor library [42] with an energy convergence threshold of 10^{-7} Hartree.

Exact KS potentials and XC potentials were generated for DMRG-calculated spin densities using a modified version of the KS-inversion algorithm outlined in Ref. [41]. The code used to perform KS-inversion is publicly available at [1].

4.5.1.2 1D KS calculations

The 1D KS-DFT code is also implemented with the external potential given in Eq. A2. Same real space grids are considered for solving the KS equations (Eq. 4.1). In the initial KS iteration, we use initial spin-densities corresponding to those in KS potential, $v_{\mathrm{s},\sigma}(\mathbf{r}) = v(\mathbf{r})$. The XC energy densities are calculated for spin-densities and the spin-polarized XC potentials ($v_{\mathrm{XC},\uparrow}, v_{\mathrm{XC},\downarrow}$) are extracted from the integrated XC energy. JAX [10] can be used in practice to obtain functional derivatives using automatic differentiation,

$$v_{\mathrm{XC},\sigma}(x) = \frac{\delta E_{\mathrm{XC}}[n_\uparrow, n_\downarrow]}{\delta n_\sigma(x)} = \frac{\delta \int dx' \, n(x') \, \varepsilon_{\mathrm{XC}}[n_\uparrow, n_\downarrow](x')}{\delta n_\sigma(x)}, \tag{A3}$$

where $\varepsilon_{\mathrm{XC}}$ is calculated from the density using one of the three functional approximations shown in Fig. 4.1. The resulting KS potential is then used to solve the KS eigenvalue equation. We use the solutions to calculate the output spin densities and the total density, $n = n_\uparrow + n_\downarrow$. By summing KS kinetic energy, Hartree energy and XC energy, we get the total electronic energy. Before the next KS cycle, the spin-densities are updated through linear spin-density mixing with an exponentially decaying mixing factor α [93],

$$n^{in}_{\sigma,k+1} = n^{out}_{\sigma,k} + \alpha(n^{out}_{\sigma,k} - n^{in}_{\sigma,k}). \tag{A4}$$

We repeat this process until the integrated absolute difference in the input and output densities becomes negligibly small (of the order of 10^{-6}). No symmetry conditions are enforced in our calculations as our training and test set contain asymmetric examples.

The LSDA approximation is implemented in 1D with the uniform gas exchange energy for the exponential interaction given in Ref. [4] and an accurate parameterized model for the correlation energy.

4.5.1.3 Training, validation and test

For the training set and validation set containing atoms and ions, we used a fixed number of iterations during the training process for all three XC functionals. Based on the convergence of standard 1D KS-DFT calculations with local density approximation for these systems, the number of KS iterations was fixed at 10.

We repeated the training process for sKSR-global, sKSR-LDA, and sKSR-GGA with 30 random seeds. The model was trained with L-BFGS algorithm [98]. Parameters checkpoints were saved at the interval of 10 steps until L-BFGS was converged. The optimal checkpoint for each seed was determined as the checkpoint that predicted the total energy and density loss of the validation set with the lowest mean absolute error (MAE). Then the analysis was repeated for all seeds to determine the best set of parameters.

For the test system, the number of KS iterations required for convergence varies based on the complexity of the system. While running the 1D KS code with the parameterized neural XC for these systems, we fixed the number of KS iterations at 30, sufficient for the largest molecule in the test set.

4.5.1.4 Computational resources

We generalized the codes available in the open-sourced JAX-DFT library [92] to build the spin-adapted Kohn-Sham regularizer. Training and testing can be accomplished on NVidia GPUs or conventional CPU nodes.

4.5.2 Optimizing NN architectures

The number of convolution layers for each one of the three KSR networks at different levels of approximation is set according to the proposed architectures in Ref. [93]. All the results reported in this paper used 8 channels in the global convolution layer of sKSR-global as well as the first convolution layer in the sKSR-LDA and sKSR-GGA architectures. Increasing the number of channels from 8 to 16 for both up and down densities does not affect the final energy and density predictions, but increases the cost of the calculation. Adding the self-interaction gate to sKSR-global also does not improve generalization for weak correlation.

4.5.3 Experimental details

The MAE in energies for N examples in the test set was calculated as,

$$\mathrm{MAE} = \sum_{i=1}^{N} |E^{\mathrm{KSR}} - E^{\mathrm{DMRG}}|/N, \tag{A5}$$

and for the density, the average density loss L_n was calculated from the test set density losses (see Eq. 4.7).

The cumulative MAEs in total energies and density losses for the training, validation, and test datasets with all the XC functional approximations examined in the main text are reported in Table A1. The MAEs in the ionization potentials for the 6 atomic systems in the training and validation sets and the MAEs in atomization energies for the molecules in the test set are also included.

Fig. A2 has the up and down spin-densities calculated with the sKSR-global approximation to the XC energy density for the ten test molecules. Fig. A1 shows the corresponding KS potentials for these molecules. Total densities and spin-up and spin-down XC potentials calculated with sKSR-LDA and sKSR-GGA are shown in Fig. A3 and Fig. A4, respectively.

We compare the atomic contributions of Li and H to the total density and the XC potentials of stretched LiH at 5.92 Bohr bond distance in Fig. A5. The exact densities of Li and H visibly add up to the stretched total density of LiH. The exact hydrogen XC potential is shifted vertically by 0.27 Hartree to match the hydrogen peak in LiH. At the dissociation limit, this value should approach the ionization potential difference of H and Li (0.35 Hartree).

Fig. A6 shows the binding energy curves of H_2 molecule calculated using uniform gas LSDA, sKSR-LDA, sKSR-GGA, and sKSR-global. Results from both restricted and unrestricted KS calculations are shown. It also includes the calculated total densities at 4.96 Bohr from each of the methods.

Table A1: MAE for total energies, ionization potentials (IP), and atomization energies (AE), and average density losses ($\times 10^{-4}$ Bohr^{-1}) with each KSR XC functional approximations for all the atoms, ions and molecules in all datasets in Table. 4.1. All energies are in mH. For all KSR models, we used the same training and validation sets from Table. 4.1. LSDA corresponds to the reference 1D uniform gas XC functional [4].

Method	ΔE	L_n	**IP**	**AE**
LSDA	35.3	4.29	16.8	32.9
sKSR-LDA	14.5	1.76	27.0	11.9
sKSR-GGA	9.21	2.58	14.8	14.2
sKSR-global	2.02	1.18	1.24	3.46

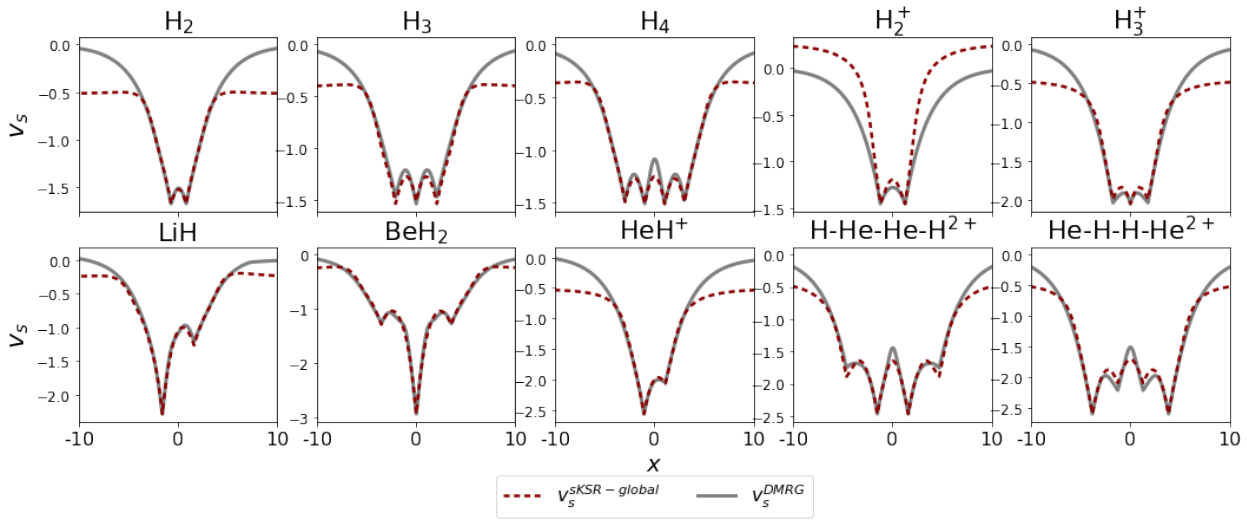
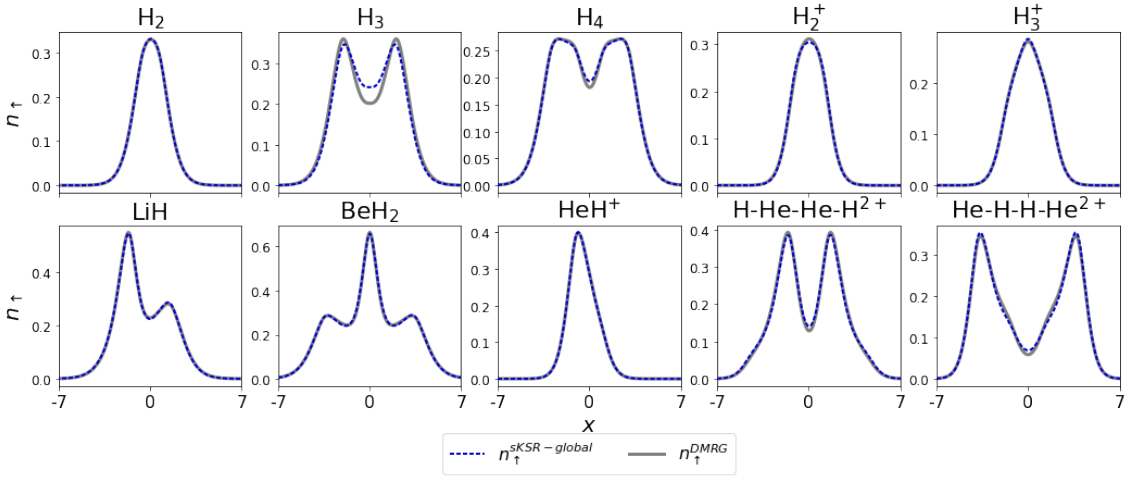
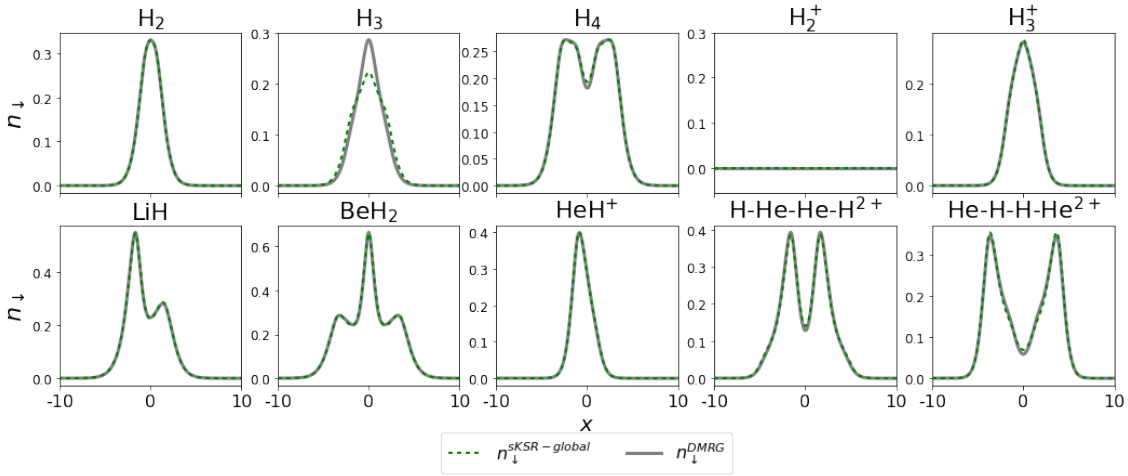


Figure A1: The exact ground-state KS potentials (gray) and the KS potentials obtained using sKSR-global (red dashes) for the test molecules in Table. 4.1 at equilibrium separations.

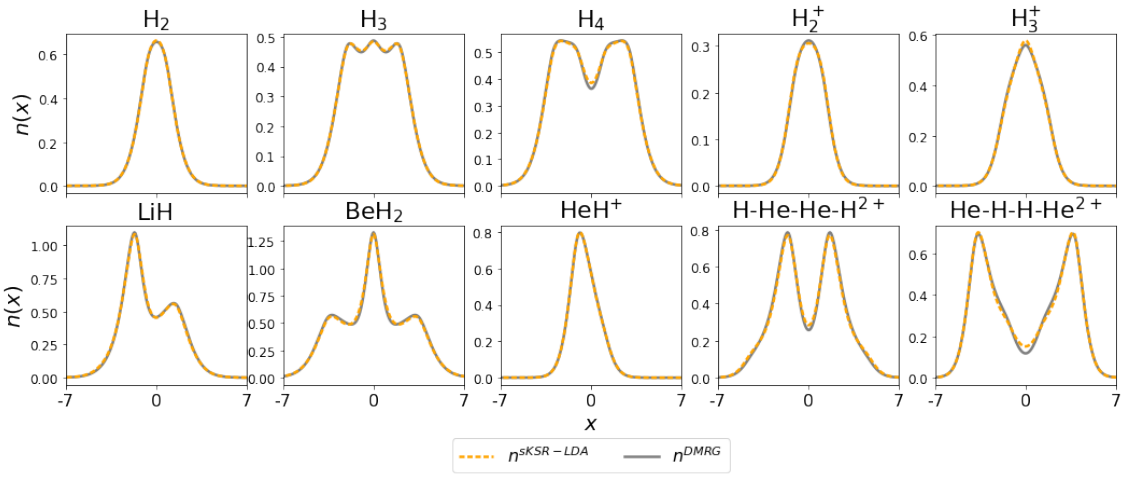


(a)

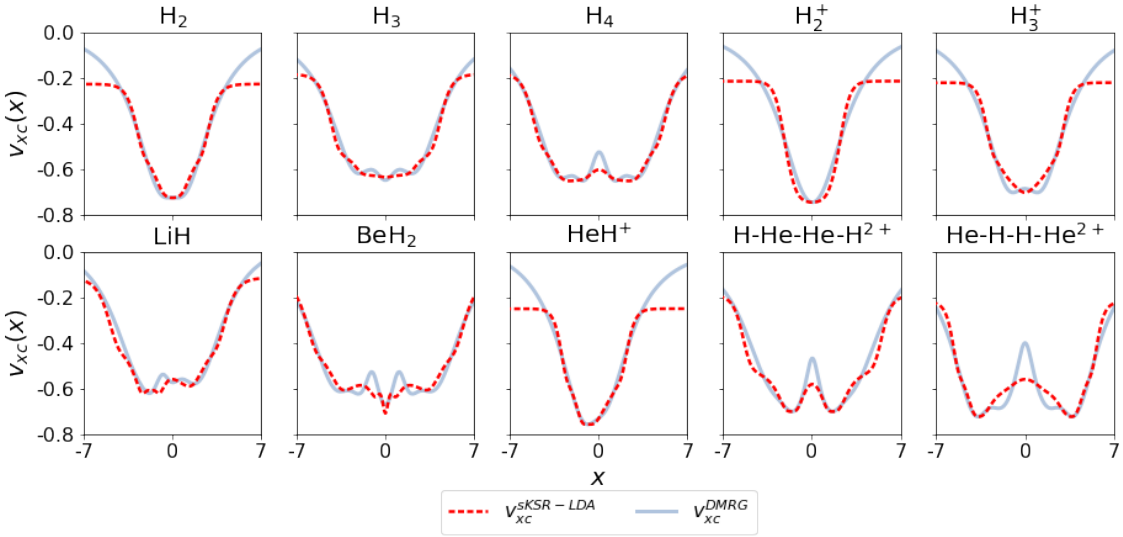


(b)

Figure A2: (a)sKSR-global spin-up (blue dashes) and (b) spin-down (green dashes) densities compared with the DMRG spin-up and spin-down (gray) densities for the test molecules in Table. 4.1 at equilibrium separations.

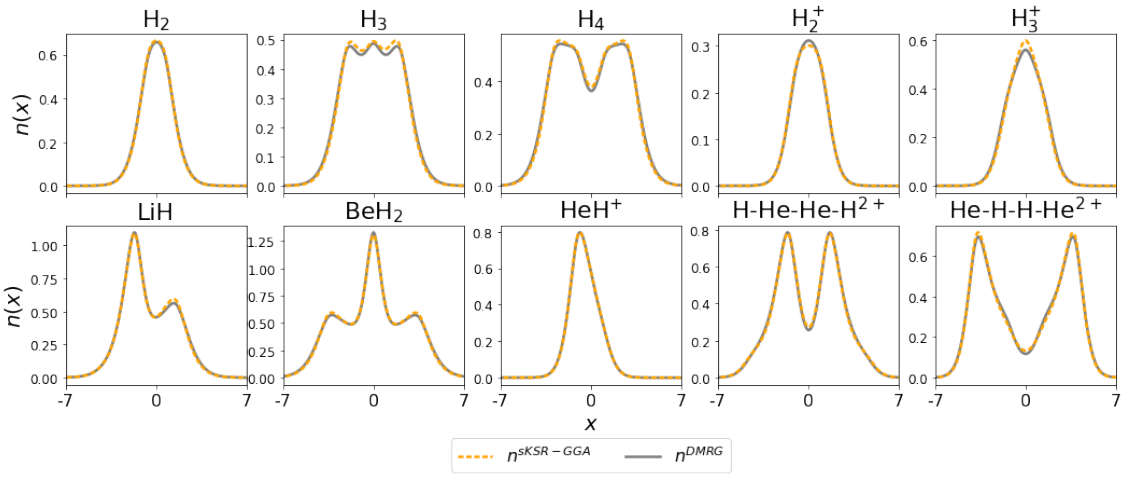


(a)

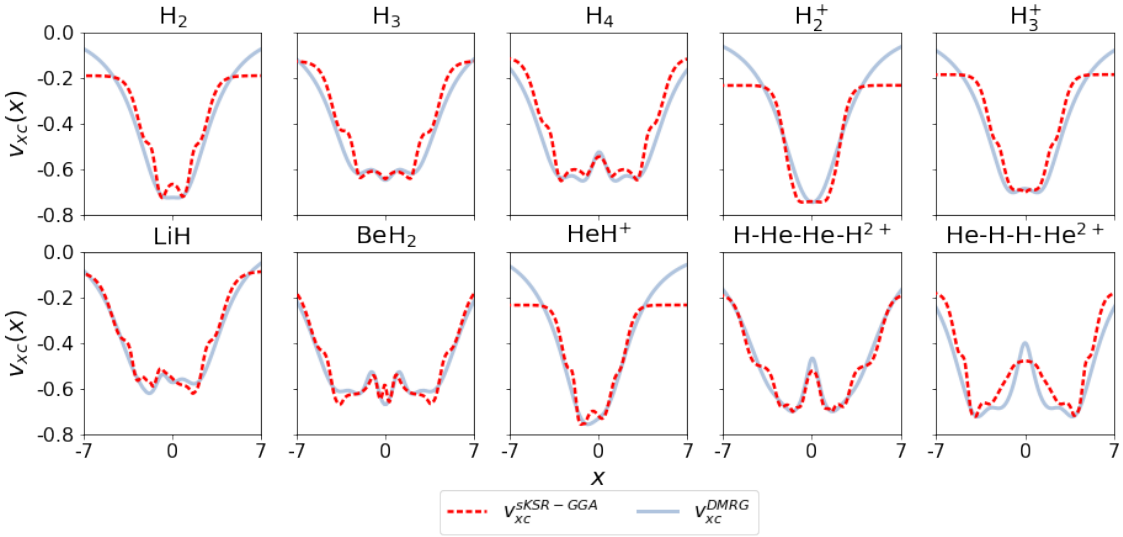


(b)

Figure A3: The exact ground-state density (gray) and the densities obtained using sKSR-LDA (orange dashes), (b) average XC potentials calculated from sKSR-LDA approximation (red dashes) and their exact counterparts (light blue) for the test molecules in Table. 4.1 at equilibrium separations.

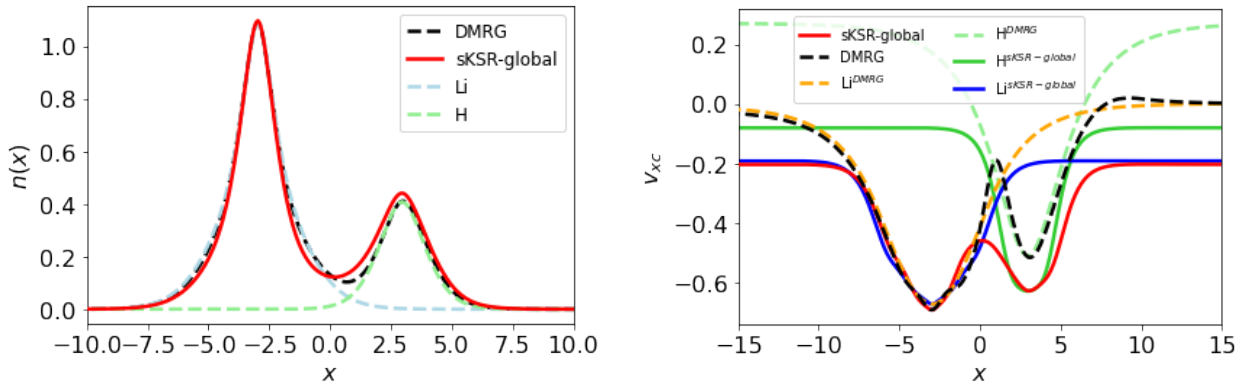


(a)



(b)

Figure A4: The exact ground-state density (gray) and the densities obtained using sKSR-GGA (orange dashes), (b) average XC potentials calculated from sKSR-GGA approximation (red dashes) and their exact counterparts (light blue) for the test molecules in Table. 4.1 at equilibrium separations.



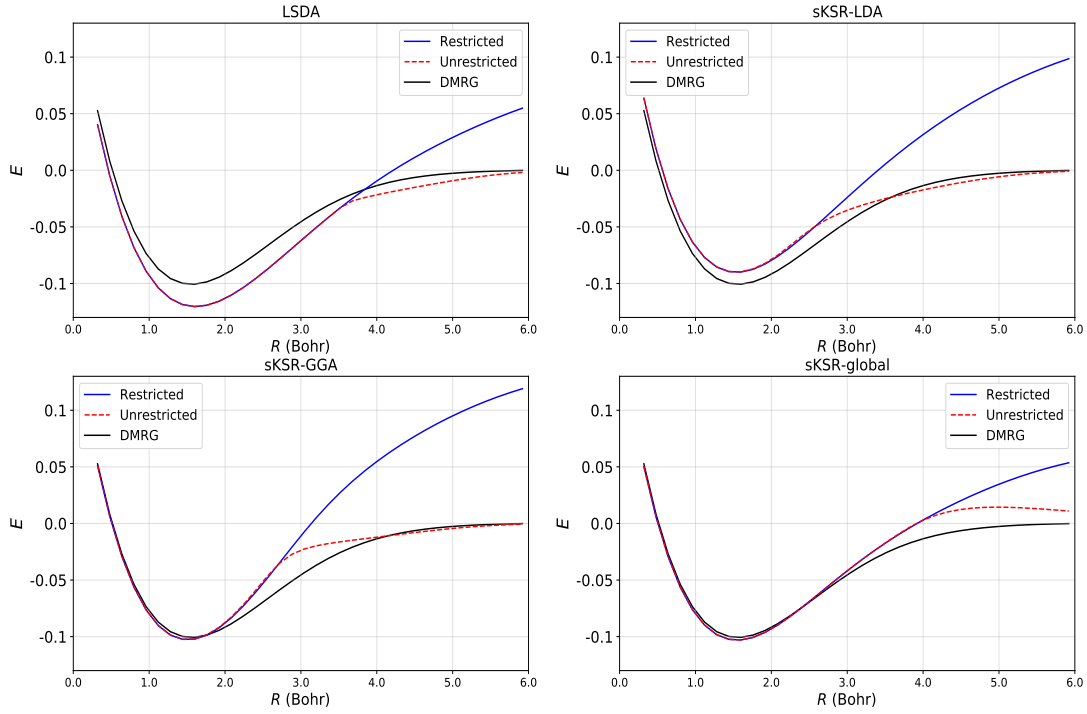
(a) (b)

Figure A5: (a) DMRG and the sKSR-global densities of stretched LiH (5.92 Bohr) and the atomic densities of Li (blue dashes) and hydrogen (green dashes). (b) The exact (black dashes) and the sKSR-global (red) average xc potentials of LiH at the same bond distance. The exact average xc potentials of Li (orange dashed) and H (green dashes) and the corresponding sKSR-global average XC potentials of Li (blue) and H (green) are included here for comparison.

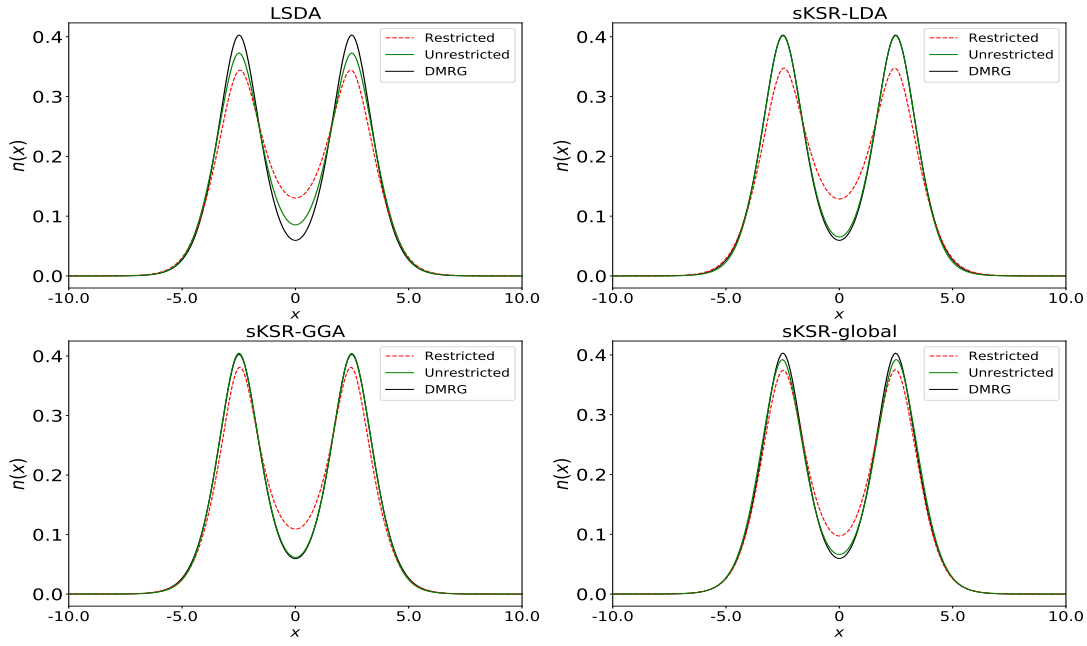
4.5.4 KSR-global results

The nonlocal approximation from Ref. [93] was considered without the self-interaction gate. The model was trained without adding energy trajectory loss. The errors in the total energy, atomization energy, and the density of the test molecules are given in Table. A2. The errors in the ionization potentials of the training and validation set are also included.

The training efficiencies of the KSR-global and the sKSR-global XC functional approximations are shown by plotting the training loss as a function of the number of training steps in Fig. A7. The total number of parameters to be learned are comparable for both KSR-global and sKSR-global. They are trained under identical conditions, with the same number of KS cycles in each training step. Using a 2.20 GHz Intel Xeon CPU (2 cores), the average time per training step is slightly higher for sKSR-global (19.13 seconds) compared to KSR-global (17.83 seconds). On a single NVIDIA Tesla K80 GPU, the average time per training step is 6.78 seconds for sKSR-global and 5.18 seconds for KSR-global, roughly a 3x speedup over the CPU setup.



(a)



(b)

Figure A6: (a) H_2 binding energy curves calculated using uniform gas LSDA, sKSR-LDA, sKSR-GGA and sKSR-global, using both restricted and unrestricted KS schemes and the corresponding DMRG results. (b) Density predictions at 4.96 Bohr using each of the three neural XC approximations and LSDA.

Table A2: Total energy errors (in mH), density losses (in 10^{-4} Bohr^{-1}), and errors in ionization potentials of atoms and atomization energies of molecules (in mH) calculated using KSR-global for the training, validation, and test sets in Table. 4.1.

Dataset	**Symbol**	**KSR-global**		
		ΔE	L_n	ΔIP
Training	H	1.27	0.26	-1.27
	He	-0.93	0.83	1.57
	Li	2.85	0.49	-2.55
	Be	-5.45	1.13	5.44
	Be^{++}	-0.02	0.26	0.26
	MAE	2.11	0.59	2.22
Validation	Be^{+}	-0.01	0.38	-0.01
				ΔAE
Test	H_2	-5.51	0.73	8.05
	H_3	16.1	0.64	-12.3
	H_4	-0.99	3.62	6.07
	H_2^+	-1.22	2.35	2.49
	H_3^+	-10.6	2.77	13.1
	LiH	-19.8	0.45	23.9
	BeH_2	33.6	2.75	-36.5
	HeH^+	-5.02	1.12	4.09
	H-He-He-H^{2+}	5.48	8.63	-7.34
	He-H-H-He^{2+}	1.89	3.20	-3.76
	MAE	10.02	2.63	11.8

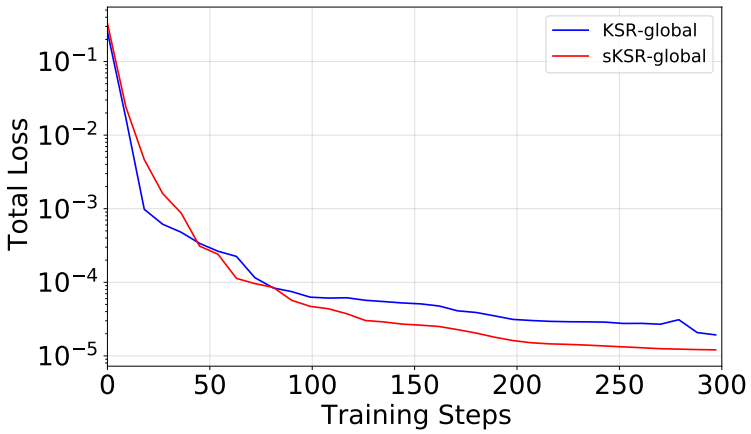


Figure A7: KSR-global (blue) and sKSR-global (red) training loss change with number of training steps when the two XC functional approximations are trained on the dataset given in Table. 4.1

4.5.5 Density-driven errors

For any approximate energy functional $\tilde{E}$ and approximate density $\tilde{n}$, the total error ΔE is,

$$\Delta E = \underbrace{\tilde{E}[\tilde{n}] - \tilde{E}[n^{exact}]}_{\Delta \mathrm{E_D}} + \underbrace{\tilde{E}[n^{exact}] - E^{exact}[n^{exact}]}_{\Delta \mathrm{E_F}}, \tag{A6}$$

where ΔE_D is the density-driven error, and ΔE_F is the functional error. The total error is the sum of the two, $\Delta E = \tilde{E}[\tilde{n}] - E^{exact}[n^{exact}]$. We calculated ΔE_D and ΔE_F for the atomization energies of equilibrium H_2 molecule with our three neural XC functional approximations and compared the values with uniform gas LSDA. These errors are reported in Table A3.

Table A3: The density driven errors and the functional driven errors in the atomization energy (in mH) for the H_2 molecule at equilibrium. For all KSR models, we use the same training set of 5 atomic systems: H, He, Li, Be, and Be^{++}, and validated on Be^{+}.

Method	ΔAE_D	ΔAE_F	ΔAE	$\%\lvert\Delta AE_D\rvert$
LSDA	-0.10	19.4	19.3	0.51
sKSR-LDA	-0.05	-10.5	-10.5	0.45
sKSR-GGA	-0.07	2.21	2.14	3.18
sKSR-global	-0.02	2.45	2.43	1.06

4.5.6 Dipole moments

Table A4: The molecular dipole moments (in atomic units) of the two one-dimensional test molecules calculated from DMRG, sKSR-global, sKSR-LDA and sKSR-GGA

Test molecule	**DMRG**	**sKSR-global**	**sKSR-LDA**	**sKSR-GGA**
LiH	-0.91	-0.84	-0.75	-0.87
HeH^{+}	0.363	0.357	0.362	0.346

The dipole moments for each molecule is calculated using the formula

$$D = \sum_i Z_i\,(x_i - x_c) - \int dx\, n(x)\,(x - x_c), \tag{A7}$$

where Z_i and x_i are the nuclear charge and positions of each atom in the molecule, and x_c is the

center of nuclear charge. Table A4 shows the dipole moments of the two polarized molecules in the test set calculated from the exact DMRG densities as well as the densities of the three sKSR approximations.

Part IV

Using Machine Learning to Categorize Density Functionals

Chapter 5

Unsupervised Learning

5.1 What is Unsupervised Learning

The primary goal of unsupervised learning is to find hidden patterns and insights in the data. So given a training dataset $\{x_i\}_{i=1,2,\ldots,N} \in X$, without any specific labels, $y_i \in Y$, an unsupervised learning algorithm will learn the structure of the training dataset by itself. However, unsupervised learning tasks are more challenging than supervised learning tasks since it is hard to assess the meaningfulness and accuracy of the predictions as there are no answer labels.

Unsupervised learning tasks help with many of the modern-day machine learning applications, such as decoding social media features [55], networking [170], sentiment analysis [21], product recommendations [148], image recognition [96], health and behavioral analysis [12, 74], and translation [110]. In the chemical and material sciences, (deep) unsupervised learning is extensively used to create meaningful feature representations for molecules and materials or to simplify the feature space [143, 52, 127]. These tasks are often combined with other supervised learning tasks, which helps minimize the uncertainties associated with unsupervised learning.

Based on types of applications, unsupervised learning tasks can be broadly classified into four categories,

- **Dimensionality reduction:** Dimensionality reduction techniques are often used for data visualization and data preparation before modeling. It projects high-dimensional data to a lower dimensional space while preserving the essence of the underlying structure in the data [108]. Having a large number of features implies that the volume of the feature space is enormous. Compared to the dimensionality of the feature space, the training set could be a tiny non-representative sample. Hence, machine learning algorithms trained on this small dataset cannot capture the behavior of the high-dimensional feature space and suffer from the *curse of dimensionality* [8].

- **Density estimation:** Unsupervised density estimation tasks determine the data distribution in space. It determines the underlying probability distribution function (PDF) of the dataset. Density estimation is a valuable task for the other two unsupervised learning techniques and generative models.

- **Clustering:** Clustering is a task of grouping different data points based on their similarity. A cluster is a collection of *similar* data points. Based on how *similarity* is defined, we have several clustering schemes and algorithms. These schemes will be discussed in detail in Section. 5.4.

- **Association:** Unlike clustering, which tries to find commonalities between data points, association tasks try to find interesting relationships between the variables in the dataset. Association has its most prominent applications in marketing analysis. For example, people who buy a specific item a may tend to purchase another item b if there is a more significant association between a and b.

The first three tasks are often combined to achieve different unsupervised learning goals. For example, density estimation is often used for data visualization, accompanied by dimensionality

reduction. It is also a part of the pipeline of several clustering algorithms. Often, clustering could be complementary to dimensionality reduction [52]. We will discuss these three tasks in detail in the following three sections. On the other hand, Association finds its applications in recommendation algorithms and filtering methods, basket data analysis, data mining, and bioinformatics. The details of the association rule and related unsupervised algorithms can be found elsewhere [49].

5.2 Dimensionality Reduction and Manifold Learning

Let us consider a data matrix X of size $N \times d$, where N is the number of data points, and d is the dimensionality. A dimensionality reduction technique will try to form a new data matrix Y, of size $N \times d'$, where $d' \ll d$ in a way that most of the information contained in the original dataset is preserved. With reduced dimensionality, training time and cost of machine learning algorithms also decrease, and data visualization becomes easy. Unfortunately, the most widely used dimensionality reduction method, the principal component analysis (PCA) [75] is a transformation into linear space and hence may often fail to capture nonlinear structure in the data.

A manifold, in simple words, is a surface of any shape. Manifold learning is equivalent to nonlinear dimensionality reduction. These techniques try to find a low-dimensional manifold for very high-dimensional data. Manifold learning algorithms are based on the idea that the dimensionality of many datasets is only artificially high, and it is possible to express the manifold of several features as a function of only a few underlying parameters.

This section will briefly discuss the theory of different linear and nonlinear dimensionality reduction algorithms. Examples of applications of these algorithms are discussed in the next chapter.

5.2.1 Linear dimensionality reduction methods

5.2.1.1 Principal component analysis

PCA [75] finds the set of orthogonal directions along which the data has maximum variance and then performs a change of basis of the data. It does so in four steps: i) it calculates a zero-mean data matrix by subtracting the average from each data point, ii) then it constructs a covariance matrix ($\frac{1}{N}X^TX$), iii) then it calculates the eigenvalues and eigenvectors of the covariance matrix, the largest eigenvalue identifies the first principal component, iv) then it projects the original datasets into the first d' principal components. The last step is accomplished by performing

$$X' = XV, \tag{5.1}$$

where V is the matrix of dimension $d \times d'$ containing the first d' eigenvectors of the covariance matrix. d' is chosen such that it captures a good portion of the total variance of the dataset.

5.2.1.2 Multidimensional scaling

Multidimensional scaling (MDS) [166] finds the d' dimensional space for a d-dimensional dataset that best preserves the pairwise distance between data points. Let us suppose d_{ij} is the pairwise distance between any two data points, i and j. We can use different distance measures to define d_{ij}. These distances then constitute a dissimilarity or distance matrix for the input data matrix X,

$$D = \begin{pmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,N} \\ d_{2,1} & d_{2,2} & & d_{2,N} \\ \vdots & & \ddots & \\ d_{N,1} & d_{N,2} & \dots & d_{N,N} \end{pmatrix}. \tag{5.2}$$

Now given D matrix, MDS tries to find N vectors y_1, ..., y_N such that the vector norm, $\|y_i - y_j\| = d_{i,j}$ for all i, j. Different forms of the loss function can be used to accomplish this task. One important loss function is the *stress*,

$$stress = \sqrt{\sum_{i \neq j = 1,\ldots,N} (d_{ij} - \|y_i - y_j\|)^2}. \tag{5.3}$$

This loss function is minimized using *stress majorization*. Numerical optimization techniques are used to find a solution. In the particular case when $d_{i,j}$ is expressed as vector norm, the solution can be found from eigendecomposition [44]. MDS is somewhat related to PCA, and both can be equivalent under certain conditions. In terms of computational efficiency, PCA is more efficient when $N \gg d$ and MDS is more efficient for $N \ll d$. However, MDS is also used when only the distance matrix, D, is given instead of the data matrix X. Hence its usage can be extended to nonlinear dimensionality reduction.

5.2.2 Nonlinear dimensionality reduction methods

5.2.2.1 Isometric feature mapping

Isometric feature mapping (Isomap) [165] is a nonlinear extension of the MDS method or kernel-PCA. It seeks a low-dimensional embedding that maintains geodesic distances between all points in the data manifold rather than Cartesian distances. It does so by initially constructing a graph where each data point is linked with its kth nearest neighbor with edges weighted by the pairwise Cartesian distances. An approximate geodesic distance is computed between all pairs of points as the shortest path. Then an MDS is performed on the geodesic distance matrix. A loss function similar to Eq. 5.3 can be used. Modified loss functions are also available for Isomap for specific applications [19], such as Sketch-Map. The hyperparameters involved in isomap, such as the number of k connected neighbors or limiting distance to consider neighbors, should be chosen carefully.

The usefulness of Isomap, compared to MDS, will depend on the degree of the nonlinearity of the data manifold.

5.2.2.2 Kernel PCA

Kernel PCA [144] is very similar to PCA in its essence. Only the linear transformation is replaced by a nonlinear transformation, $\phi(x_i)$, where ϕ can be a high-dimensional vector function. The transformation function is chosen such that the transformed dataset is approximately linear, and we can perform linear PCA or MDS. The transformed matrix is not explicitly computed, but obtained through a kernel function $K(x_i,x_j)$, using the *kernel trick*,

$$K(x_i,x_j) = \phi(x_i)^T \phi(x_j). \tag{5.4}$$

The choice of the kernel is important. The most widely used kernel is the Gaussian kernel, $K(x_i,x_j) = e^{-\|x_i-x_j\|^2/2\sigma^2}$. σ is a hyperparameter that decides the width of the Gaussian. Kernel PCA can help to deal with the problem of choosing the largest pairwise distance between a pair of data points that plagues MDS and Isomap by choosing an appropriate kernel.

Several modifications of Kernel PCA, such as diffusion map [25], are formulated for specific applications, and we will not review them further.

5.2.2.3 t-Distributed stochastic neighbor embedding

t-Distributed stochastic neighbor embedding (t-SNE) [30] is a probabilistic dimensionality reduction method. The affinities in the original data space are expressed by Gaussian joint probabilities,

$$P_{ij} = \frac{K(x_i,x_j)}{\sum_{k\neq i} K(x_i,x_k)} \tag{5.5}$$

and the affinities in the embedded space are expressed using Student's t-distribution,

$$Q_{ij} = \frac{(1 + \|y_i - y_i\|^2)^{-1}}{\sum_{k \neq i}(1 + \|y_i - y_i\|^2)^{-1}} \tag{5.6}$$

The loss function is defined with the Kullback-Leibler (KL) divergence [87],

$$KL(\mathbf{P}\|\mathbf{Q}) = \sum_{j \neq i} P_{ij} log \frac{P_{ij}}{Q_{ij}}. \tag{5.7}$$

This loss function is not convex and is optimized iteratively. t-SNE focuses on grouping data points based on the local structure of the data and therefore is useful for visualizing datasets that comprise several manifolds. However, due to stochasticity, the embeddings and the global structure of the transformed dataset are not preserved.

5.2.2.4 Other nonlinear methods

There are several other, comparatively less used, nonlinear dimensionality reduction methods. Similar to MDS, locally linear embedding looks for a lower-dimensional projection of the data that preserves distances within local neighborhoods. We can think of it as a series of local PCA compared globally to find the best embedding. The spectral embedding method attempts dimensionality reduction through spectral decomposition of the graph Laplacian. The graph itself approximates the low-dimensional manifold, and a cost function is minimized based on the graph.

Deep learning architectures such as autoencoders, variational autoencoders (VAE), and generative adversarial networks (GAN) can efficiently accomplish dimensionality reduction tasks. However, for the scale of our problem discussed in the next chapter, these architectures are irrelevant and hence not discussed here.

5.3 Density Estimation

Density estimation is integral to several dimensionality reduction techniques and clustering algorithms. Here we briefly go over parametric and nonparametric density estimation methods.

5.3.1 Parametric density estimation

In parametric density estimation, we assume that the training data belongs to a population of the probability distribution characterized by a fixed set of parameters. For example, if we consider a normal distribution of data points, the related probabilities are characterized by the mean and the standard deviation. Since the underlying distribution of the datasets might not be a perfect normal distribution, we can make this approach slightly flexible by modeling the probability distribution function as a mixture of k distributions. In Gaussian mixture models (GMM) [32] a mixture of Gaussian distribution function is used. The weights of each Gaussian in the mixture can be determined by maximum likelihood estimation. The number of distributions used in the mixture is a hyperparameter that requires careful evaluation.

5.3.2 Nonparametric density estimation

Nonparametric learning algorithms do not require the model to make any assumptions regarding the underlying distribution of the data points. Instead, these algorithms form clusters describing the data's categories and classes. These methods are preferable for small datasets. Most parametric methods require choosing hyperparameters. An optimal balance of bias and variance is required to make the best choices. Some popular nonparametric density estimation methods are discussed below.

- **Histograms:** In this popular method, the dataset is divided into several bins, and the PDF

is estimated based on the number of data points in each bin. Histograms are suitable for low-dimensional problems, $d \leq 3$, as it suffers from the curse of dimensionality.

- **Kernel density estimation (KDE):** KDE instead estimates the probability distribution function as a sum of kernel functions centered at each data point. Different kernel functions can be used. Although, Gaussian seems to be the most popular choice. Kernel density estimation is also differentiable. The choice of the hyperparameter σ in the Gaussian kernel requires careful evaluation for high-dimensional problems.

- ***k*-Nearest neighbor estimator:** This is a particular form of KDE where the kernel is expressed in terms of the hyperparameter k-nearest neighbor (k-NN). This method, in principle, can work in any dimension. However, it suffers from the curse of dimensionality as the distance between the closest neighbors becomes more and more similar to the distance between the furthest neighbors in higher dimensions.

5.4 Clustering Methods

Most unsupervised learning tasks primarily refer to clustering. Dimensionality reduction and density estimation are often used as sub-task for clustering. While clustering finds patterns in data, dimensionality reduction and density estimation help in visualization and analysis. We will discuss several clustering algorithms in this section. It is important to note that the evaluation and appropriateness of a clustering algorithm for a particular dataset are debatable. A suitable measure hardly exists to confirm the output from a particular algorithm. Hence, the algorithm should be chosen carefully based on the dataset and the expected results. Also, the choice of hyperparameters for each algorithm can have a sizable impact on the results.

5.4.1 Clustering tendency evaluation

Before evaluating the clustering performance, it is crucial to ensure that the data set has a clustering tendency and does not contain uniformly distributed points. If the data does not have a clustering tendency, then clusters identified by state-of-the-art clustering algorithms may be irrelevant. We can evaluate clustering tendency by calculating the Hopkins statistic [68]. The Hopkins statistic, H, is a statistical hypothesis test where the null hypothesis assumes that the data is generated from a Poisson point process and hence is uniformly randomly distributed.

Null Hypothesis: Data points are generated by uniform distribution (there are no meaningful clusters).

Alternate Hypothesis: Data points are generated by random data points (clusters are present).

If we consider a random sample (without replacement) of $m < N$ and generate a set Y of m uniformly randomly distributed data points, we can calculate the Hopkins statistic as,

$$H = \frac{\sum_{i=1}^{m} u_i}{\sum_{i=1}^{m} u_i + \sum_{i=1}^{m} w_i}. \tag{5.8}$$

Here, u_i is the distance of $y_i \in Y$ from its nearest neighbor in the original dataset, and w_i is the distance of m number of randomly chosen data points in the original dataset from its nearest neighbor in the same dataset. The Euclidean distance measure is the most common choice for calculating the nearest neighbor distance. If $H > 0.5$, the null hypothesis can be rejected, and the data likely contains clusters. If H is closer to 0, we can conclude that the data set has no clustering tendencies.

Since the Hopkins statistic can vary based on the set of randomly chosen points, it is a common practice to take the average over several random distributions of points.

5.4.2 Distance measures

Before we discuss different clustering algorithms, let us highlight the properties of a dissimilarity matrix or the distance matrix. For a distance matrix D, there are four main axioms for d_{ij}, the distances of the objects (A and B) in the feature space,

- $d(A,B) \geq 0$, i.e. distances are non-negative
- $d(A,B) = 0$ if $A = B$
- $d(A,B) = d(B,A)$, i.e. D is symmetric
- $d(A,C) \leq d(A,B) + d(B,C)$, i.e., the triangular inequality holds.

Non-metric dissimilarities: We can have dissimilarities that do not obey the triangle inequality or are not symmetric. Few clustering algorithms support non-symmetric dissimilarities.

The dissimilarity measure is also known as the kernel function. Distances are dissimilarities to the properties discussed above. Cosine and correlation (Pearson's correlation) functions are generally considered a measure of similarity.

For any type of clustering, the results depend on the distance measures used to build the distance matrix, D. Here are some standard dissimilarity measures:

Euclidean distance (L2-norm): If x_i and y_i are the coordinates of the data points in the feature space, then the distance between the points,

$$D(x,y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}, \tag{5.9}$$

Where N is the number of data points, x and y are the vectors representing the two points in the feature space. However, Euclidean distance often suffers from the curse of dimensionality for datasets with many features [50]. In such cases, other distance measures are preferable.

Manhattan distance (L1-norm): Manhattan distance has the form:

$$D(x,y) = \sum_{i=1}^{N} |x_i - y_i|, \tag{5.10}$$

It was suggested in Ref. [50] that for Manhattan distance, the difference between the minimum and maximum distances between points diverges with increasing number of dimensions. For the L2-norm, this value approaches a constant. Hence, for high dimensional data, L1-norm is preferable to L2-norm.

Cosine similarity: This is a similarity measure. It is a measure of orientation, and magnitude is not of importance. The similarity is calculated for the vectors x and y as,

$$D(x,y) = cos(\theta) = \frac{x.y}{||x||_2||y||_2}. \tag{5.11}$$

Cosine distance is the related dissimilarity measure. It is equivalent to $1-$cosine similarity. For normalized data, cosine distance is virtually equivalent to Euclidean distance.

Correlation distance: This is again a similarity measure. Highly similar data points will have a correlation distance close to 1.

$$D(x,y) = 1 - \frac{(x-\bar{x})(y-\bar{y})}{||x-\bar{x}||_2||y-\bar{y}||_2} \tag{5.12}$$

The distance measures discussed here are also shown in Fig. 5.1. There are several other distance measures, such as Hamming distance, Jaccard index, and Chebyshev distance. A suitable distance measure can be found by trial and error, depending on the problem.

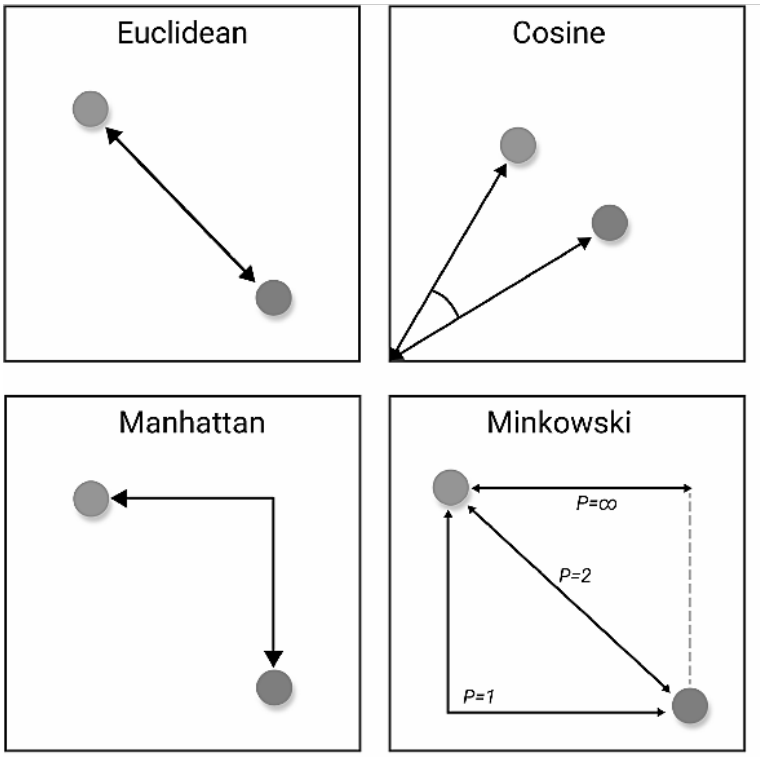


Figure 5.1: Cartoons depicting the Euclidean, Manhattan, Cosine, and Minkowski distance measures.

5.4.3 Classification of clustering algorithms

Well-known clustering algorithms can be broadly classified into two categories: 1) partition-based and 2) density-based algorithms.

1. **Partition-based clustering methods:** In these approaches, the datasets are roughly partitioned based on their similarities. In other words, they do not find clusters but minimize the intra-partition distance compared to inter-partition distance. Partitioning algorithms can again be centroid-based, graph-based, or hierarchical.

    - **Centroid based partition methods:** In these methods, a centroid for each cluster is defined first. Then the data points are assigned to the cluster of the nearby centroid. The number of desired clusters has to be specified by the user. These methods yield convex, globular clusters. Most popular clustering algorithms *k*-means [100] as well as

k-medoids [79] are centroid-based algorithms.

- **Graph based partitioning methods:** These algorithms convert the feature space to the vertices and edges of a graph and perform clustering on the transformed space. There are various ways to achieve space transformation. Algorithms like minimum spanning trees shared the nearest neighbor, and between centrality-based algorithms, cluster graphs. These are widely popular in community detection problems. Spectral clustering [149], on the other hand, finds the structural properties of the graph using spectral decomposition. Graph-based methods can produce clusters of non-convex shapes.
- **Hierarchical partitioning methods:** Hierarchical clustering seeks to build a hierarchy of clusters. This hierarchy is a tree structure called a dendrogram, representing an ensemble of clustering models with every possible number of clusters. It can either be agglomerative(bottom-up) or divisive (top-down). In agglomerative hierarchical clustering [79], each data point constitutes an individual cluster initially, and then pairs of clusters are merged as one moves up the hierarchy. In divisive clustering, we start with a single cluster of all data points and split the cluster recursively while moving down the hierarchy. Divisive clustering is a bit more complex, and the approach to the problem can be widely different. Girvan-Newman algorithm [51] is a sort of (graph-based) divisive clustering approach, which we will not discuss in this chapter.

2. **Density-based clustering:** Algorithms such as DBSCAN [39] clusters data based on density. Data points that densely populate a region are grouped to form a cluster. Data in the low-density regions are considered noise. The user can define the cut-off for the density. Most density-based algorithms initially transform the feature space to another space defined by the probability density. Consequently, clusters formed can have a non-convex shape in the feature space.

Recently, several new clustering algorithms have been developed that use elements from more than

one class of clustering methods. For example, HDBSCAN [17] is a density-based but hierarchical clustering algorithm. The mean shift method [26] is somewhat old but is a density-based method that yields convex clusters. Below we discuss a few of the popular clustering algorithms. Besides these methods, we can also use probabilistic methods based on the expectation-maximization algorithm and neural network-based techniques for complex clustering tasks. These techniques are outside the scope of this chapter.

5.4.4 k-Means clustering

The goal of k-means [100] clustering is to minimize the sum of the square of the distances of data points assigned to a cluster with respect to the respective cluster centers. This is an NP-hard problem, and an approximate solution is found with an iterative procedure. In this procedure, k-number data points (prespecified as the desired number of clusters) are randomly chosen as centers. Then, after assigning data points to the cluster of the closest center, new centroids are created by taking the mean value of all the data points assigned in the previous step. These two steps are repeated until the centroids stop changing.

The outcome of the k-means algorithm is highly dependent on the initialization step. Modified methods with improved initializations, such as mini-batch k-means, are also proposed. In fuzzy $c-$means algorithms, each data point shares a variable degree of membership to all possible clusters. k-means clustering is highly efficient, but its stability will depend on the dataset.

5.4.5 k-Medoids clustering

k-Medoids [79] partitions data into clusters by minimizing the sum of distances between each data point and the medoid of the cluster. The medoid is the data point with the least total distance to the other members in the partition. Since data points are defined as cluster centers, any form of

distance metric can be used to form the clusters. It is slightly costlier than k-means and hence more suitable for smaller datasets.

5.4.6 Affinity propagation

Affinity propagation [45] is a graph-based clustering method that utilizes the concept of "message passing" between data points. It is similar to k-medoids in one aspect that it finds *examplars* that already belong to the dataset as representative of the clusters to be formed. The messages sent between pairs of data points represent the suitability of one data point as the exemplar of the other. The process is repeated iteratively until the exemplars stop changing. Finally, data points are assigned to the cluster of the nearest examples. Unlike the k-means and k-medoids algorithms, we do not need to specify the number of clusters beforehand. However, the user must specify the preference for each data point and damping factor to avoid numerical oscillations. It supports non-metric dissimilarities and could be costlier for large datasets.

5.4.7 Spectral clustering

Spectral clustering [149] is a graph-based clustering method that performs manifold learning to transform the original high-dimensional data space into a meaningful low-dimensional data space and clusters data on that space. It initially constructs a dataset graph based on distances between points, then calculates the eigenvectors of the Laplacian to find a good embedding of the graph in a low-dimensional Euclidean space. Any clustering algorithms such as k-means can be used to perform clustering on the transformed space. However, if we use k-means, we have to specify the number of clusters.

Variations of spectral clustering utilize different schemes for constructing the initial graph or in the normalization of the Laplacian. In many cases, spectral clustering can be considered equivalent to

kernel k-means.

5.4.8 Agglomerative hierarchical clustering

Agglomerative hierarchical clustering depends on two factors, the choice of an appropriate distance measure (similarity function) and the linkage criteria that specify the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

We can choose the distance metric as specified in section 5.4.2.

Once the initial pairwise clusters are formed based on one of the distance measures (metric or non-metric), the merging of two such clusters is performed based on the chosen linkage criterion. Some commonly used linkage criteria are:

1. Single linkage clustering: The distance between two clusters is the minimum distance between the data points in each cluster. If A, B and C, D form two clusters, and $d(A,C) < d(B,C), d(A,D), d(B,D)$, then the distance between the two clusters is d(A,C).

$$d = \min d(a,b) : a \in A, b \in B \tag{5.13}$$

2. Complete linkage clustering: The distance between two clusters is the maximum distance between the data points in each cluster. If A, B and C, D form two clusters, and $d(A,C), d(B,C)$, $d(A,D) < d(B,D)$, then the distance between the two clusters is $d(B,D)$.

$$d = \max d(a,b) : a \in A, b \in B \tag{5.14}$$

3. Average linkage clustering: Also known as unweighted pair group method with arithmetic mean(UPGMA). The distance between two clusters is the average of all distances between

pairs of data points belonging to each cluster.

$$d = \frac{1}{|A|.|B|} \sum_{a \in A} \sum_{b \in B} d(a,b) \tag{5.15}$$

4. Ward linkage clustering: Uses Ward's variance minimization algorithm. The initial clusters are determined based on squared Euclidean distances.

$$d(u,v) = \sqrt{\frac{|v|+|s|}{T} d(v,s)^2 + \frac{|v|+|t|}{T} d(v,t)^2 - \frac{|v|}{T} d(s,t)^2}, \tag{5.16}$$

where u is the newly joined cluster consisting of clusters s and t, v is a remaining cluster, and $T = |v| + |s| + |t|$. $|*|$ signifies the number of elements in a cluster.

One must define a threshold distance to determine the number of clusters from hierarchical clustering. A horizontal line is drawn at this cut distance, and the number of tree branches it cuts through gives the number of clusters. The algorithm is pretty fast, and formed clusters are not globular. It is especially suitable for smaller datasets. The linkage criterion and distance threshold variations in agglomerative clustering can yield significantly different clusters.

5.4.9 Mean-shift clustering

Mean shift [26] is a centroid-based algorithm that produces clusters instead of partitions. It also does not require specifying the number of clusters. Instead, it assumes that the dataset is drawn from some underlying probability density function and then tries to place the centers of the clusters at the peak of that function. It uses the kernel density function to figure out the PDF and the bandwidth of the kernel is the only hyperparameter that needs optimization. However, since the mean shift is a centroid-based clustering method, it still aims for globular clusters and may not be suitable for all datasets.

5.4.10 DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) [39] defines a threshold density and produces clusters from connected regions with density above the threshold. It is not a partitioning algorithm; it can form clusters with any shape and does not require input for the number of expected clusters. In the first step, DBSCAN transforms the dataspace based on density. Dense regions are left alone and sparse regions are made sparser. Next, the algorithm defines *core samples* as those samples in the dataset that have a specific number of neighbors (defined by the user) within a cut-off distance. This process can be repeated multiple times to find all the members of a cluster. Data points that are not neighbors to any core samples are considered noise.

DBSCAN is in a spirit similar to performing a single-linkage clustering after a density-based transformation. However, the cut-off distance for the neighbors and the number of neighbors that controls the density are both hyperparameters that can significantly affect the output clusters.

5.4.11 HDBSCAN

Hierarchical DBSCAN (HDBSCAN) [17] is an improved form of DBSCAN. It can form clusters for varying densities. It forms a dendrogram after performing a single linkage on the density-transformed space. The tree can be cut at different heights, similar to hierarchical clustering, to pick clusters of different densities. However, the cut distance is determined by the algorithm based on another parameter, the minimum cluster size, which determines the size of the smallest cluster. This parameter is often more intuitive than the cut-off distance for DBSCAN or hierarchical clustering. HDBSCAN is very efficient and offers the benefits of both density-based clustering and hierarchical partitioning methods.

5.4.12 BIRCH

Balanced iterative reducing and clustering using hierarchies (BIRCH) [185] is a hierarchical clustering method suitable for large datasets. First, the user has to specify the desired number of clusters. In its initial step, BIRCH builds a clustering feature tree (CFT) for the dataset, similar to a dendrogram for hierarchical clustering. The initial CFT is then divided into smaller CFTs, and outliers are removed. Then one can use an agglomerative clustering algorithm to form the subclusters from the smaller CFT.

BIRCH is a memory-efficient algorithm as it only stores necessary information for the clustering, not the whole input dataset. It can be considered a data reduction method since it reduces the dataset into subclusters. Clustering is performed on those subclusters. It may not scale well with high-dimensional data, and alternatives such as minibatch k-means might be more desirable for datasets with a large number of features.

5.5 Performance Evaluation of Clustering Algorithms

We have discussed several clustering algorithms in this chapter. The choice of the algorithm will depend on the specific dataset under consideration. Since we do not have any labels to compare the accuracy of these clustering algorithms, evaluating their performance is a nontrivial task. An *evaluation metric* for clustering is generally defined as a similarity metric that compares the similarity among data points belonging to the same cluster to the similarity of members belonging to different clusters. There are mainly two types of measures to assess the clustering performance:

1. Intrinsic measures that do not require ground truth labels. Examples of intrinsic measures include Silhouette coefficient [134], Calinski-Harabasz index [16], Davies-Bouldin Index [31], etc.

2. Extrinsic measures that require ground truth labels. Examples include adjusted rand index, Fowlkes-Mallows scores, mutual information-based scores, homogeneity, completeness, and V-measure.

One can use extrinsic measures to compare different clustering algorithms, where we consider the clusters formed by one algorithm as the ground truth. We can also use these measures to compare clusters from different related datasets. Below we will discuss some of these evaluation metrics.

1. **Silhouette coefficient:** The Silhouette coefficient (S) [134] is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each cluster,

$$S = \frac{(b-a)}{max(a,b)}$$

.

Then we take the mean to produce the overall Silhouette coefficients. It can vary between -1 and 1. When the Silhouette coefficient is close to 0, it indicates overlapping clusters. This measure often yields higher scores for convex, globular clusters.

2. **Calinski-Harabasz (CH) index:** The CH index [16] is defined as the ratio of the sum of intra-cluster dispersion and inter-cluster dispersion for all clusters. Dispersion refers to the sum of the squared distances. It is fast to compute but, again, generates higher scores for centroid-based methods.

3. **Davies Bouldin (DB) index:** The DB index [31] is the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters farther apart and less dispersed will result in a better score. The similarity measure for two clusters i and j that is used to calculate the DB

index is,

$$R_{ij} = \frac{l_i + l_j}{d_{ij}}, \tag{5.17}$$

where l_i is the average distance from the centroid to each data point in the cluster i and d_{ij} is the distance between cluster centroids. The DB index is then,

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} R_{ij}, \tag{5.18}$$

where k is the number of clusters. The minimum score is zero, with lower values indicating better clustering. Similar to the Silhouette coefficient, this measure can overestimate the performance of centroid-based algorithms. The choice of distance measure is also limited to Euclidean distance.

4. **Normalized Mutual Information (NMI):** It is a concept from probability theory and information theory. The mutual information (MI) [171] measures the similarity between two labels of the same data. If $|U_i|$ is the number of the data points in cluster U_i and $|V_j|$ is the number of the data points in clusterV_j, the MI between clusterings U and V is given by

$$MI(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} log \frac{N|U_i \cap V_j|}{|U_i||V_j|} \tag{5.19}$$

This metric does not depend on the absolute values of the labels. If one permutes the class or cluster labels, it will not change the score value. This metric is also symmetric. It can be useful to measure the agreement of two independent label assignments on the same dataset when the ground truth is unknown.

Normalized MI (NMI) score normalizes the MI scores to scale between 0 (bad) and 1 (good).

5. **Adjusted mutual information (AMI)** Adjusted mutual information (AMI) [182] is an adjustment of the MI score to account for the chance. It considers that the MI is generally

higher for two clustering algorithms with a larger number of clusters regardless of whether more information is shared. For two clustering algorithms U and V, the AMI is given by

$$AMI(U,V) = \frac{MI(U,V) - E[MI(U,V)]}{mean(H(U),H(V)) - E[MI(U,V)]}, \tag{5.20}$$

where $H(U)$ is the entropy of uncertainty for clustering U, and $E[MI(U,V)]$ refers to the expected value of MI. The AMI returns a value of 1 when the two partitions are identical. Random partitions (independent labelings) have an expected AMI of around 0 on average and can be negative.

6. **Adjusted rand index.** The Rand index (RI) [167] computes a similarity measure between two clusterings by first considering all pairs of samples and then counting pairs assigned in the same or different clusters in the predicted and true clusterings. The raw RI score is:

   RI = (number of agreeing pairs) / (number of pairs)

   . However, the RI score may not equal zero for random labeling. Hence, we can also calculate adjusted RI [158] by using the expected RI ($E[RI]$) value,

   $$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \tag{5.21}$$

   Similarity scores between -1.0 and 1.0. Random labelings will yield an ARI close to 0.0.

7. **Homogeneity and completeness scores** Both of these are extrinsic measures based on conditional entropy analysis. A clustering algorithm satisfies homogeneity if all of its clusters contain only data points that are members of a single class,

   $$h = 1 - \frac{H(C|K)}{H(C)}, \tag{5.22}$$

   where $H(C|K)$ is the conditional entropy of the classes, Cs given the clustering assignments

Ks, and $H(C)$ is the entropy of the defined class labels. Completeness, on the other hand, assumes that all members of a given class are assigned to the same cluster,

$$c = 1 - \frac{H(K|C)}{H(K)}. \tag{5.23}$$

Both these scores are between 0 and 1; the higher is better. We can define another score called the V-measure [133], using these two extrinsic measures,

$$\mathrm{v} = \frac{(1+\beta) \times \text{homogeneity} \times \text{completeness}}{(\beta \times \text{homogeneity} + \text{completeness})}, \tag{5.24}$$

where the default β value is 1. V-measure is the same as NMI. These scores do not make any assumption regarding the cluster structures and can produce meaningful scores for non-globular clustering algorithms.

Intrinsic measures can also determine the optimal hyperparameters for a clustering algorithm. A frequently employed method is the elbow method. For example, we can use the Silhouette coefficients or the sum of the intra-cluster variance of all the data points from the centroid. We get an elbow plot if we plot either of these two quantities as a function of the clustering hyperparameter. An elbow is the hyperparameter value where the error measure starts to flatten. Therefore, the elbow should be considered the optimal choice for the hyperparameter. Gap statistics is another method that we can utilize to calculate optimal hyperparameter.

5.6 Summary

Unsupervised learning is a powerful machine learning technique that is closer in spirit to actual artificial intelligence compared to supervised learning. Applications of unsupervised learning for dimensionality reduction, density estimation, or clustering tasks can help understand rules and pat-

terns in data, leading to its widespread usage in the data-driven world. However, most of the current unsupervised learning algorithms require domain knowledge about the dataset. Such knowledge can induce bias in the choice of the algorithm and hyperparameters. Furthermore, optimizing hyperparameters involved in the learning algorithms is a nontrivial task. Evaluation metrics are ill-defined and often inherently favor one algorithm over the other. Hence parameter-free clustering methods are desirable, and there is work in progress in this direction. However, we still need unbiased intrinsic measures to evaluate clustering quality better.

Chapter 6

Categorizing Density Functionals with Unsupervised Learning

written with Suhwan Song, Ryan J. McCarty, Stefan Vuckovic, Eunji Sim, John Kozlowski, and Kieron Burke. This is a draft in preparation.

Abstract: *Density functional calculations are routine in many fields, especially chemistry and materials science. Debates abound over which of the hundreds of exchange-correlation approximations are best or even on what basis to judge. We introduce a measure of distinction between approximate functionals, the density-driven fractional difference (DDF) based on density-corrected DFT, and evaluate this functional fingerprint for the MGAE109 dataset from the Minnesota database for 33 popular approximations. We construct a metric space using these fingerprints, yielding distances between approximations. To evaluate the similarities among the functionals, we use three different unsupervised clustering algorithms suitable for small datasets, including one of our parameter-free clustering algorithms. In the DDF descriptor space, these algorithms create functional categories that largely (but not entirely) mimic those based on their ingredients, analogous to the famous Jacob's ladder categorization. Finally, we illustrate these functional*

clusters using various popular dimensionality reduction tools. Overall, our approach combines density and energy differences in a meaningful way, is independent of the origins of the functional approximation (e.g., how many empirical parameters it may have), and uses a parameter-free deterministic algorithm to cluster the approximations. Our scheme shows, in agreement with earlier speculation, that specific construction techniques of approximate functionals yield approximations that differ wildly from more established XC functionals.

6.1 Introduction

Kohn-Sham density functional theory (KS-DFT) [86] is used in tens of thousands of scientific papers each year, primarily within chemistry and materials science [14]. There are hundreds of useful approximations to the unknown exchange-correlation (XC) energy, employing different ingredients, using various conditions to choose parameters, and fitting to an increasingly large plethora of databases [102]. The over-arching goal of DFT approximation is to construct a single functional that *works* for all systems and properties of interest. In practice, different approximations yield higher accuracy for different problems, leading to the eternal question, Which functional should I use?, especially if the problem falls between databases [132].

Almost all discussion of the *quality* of functionals has centered around their accuracy (usually energetics compared to benchmarks in databases) or their intellectual purity (keeping the number of empirical parameters to a minimum). Twenty years ago, these ideas were baked into Jacob's ladder metaphor [121], where each rung corresponded to an approximation using specific ingredients, with the number of ingredients increasing as you climb the ladder. The aim is to find a *best* approximation at each rung, thereby balancing cost with accuracy. While many functionals have been constructed in the intervening decades that would now be difficult to assign to these old rungs, one could imagine mildly generalizing the ladder (as has occasionally been done) without losing its essence or utility.

However, new tools have since been developed that allow us to update the metaphor, using different construction principles and more sophisticated techniques. Here, we eschew the use of accuracy as a criterion, as this is mainly subjective (whose database do I use?). Instead of posing a supervised learning problem to determine which functional would be better under which condition, we propose an unsupervised learning scheme to compare approximate functionals to one another to determine how to categorize them. Moreover, by using machine learning techniques, we are seeing if (a) evidence of functional categories can be extracted from numerical performance alone, without knowledge of the ingredients used, and (b) the choice of categories (rungs on Jacob's ladder) themselves can be extracted. The primary goal is not to introduce any human bias and let the machine construct its functional ladder, stairs, or whatever it thinks is right.

There are three essential components in an unsupervised learning task. The first is to define a meaningful feature space where we can find meaningful clusters. The next is the clustering task itself. There are numerous clustering methods. In the absence of any measured property label (in unsupervised learning), determining which clustering algorithm will be best is an almost impossible task. The third part is dimensionality reduction for adequate visualization of the clusters. Here too, infinite techniques exist, but none might show the separation among the clusters. All these three tasks are highly indeterminate. Hence, domain knowledge comes in handy, especially for defining features and analyzing the clusters. In the sections below, we describe how we overcome the hurdles for all three tasks and try to learn what the machine tries to teach us.

6.2 Unsupervised Learning Density Functionals

We make a series of commonsense choices to set up the methodology. These choices are far from unique, but one can hope that reasonable alternatives will lead to similar results. Importantly, all our choices obey several strict conditions that minimize human bias.

6.2.1 Functional fingerprints from density-corrected DFT

Our first step is to avoid using exact results (usually energies) in constructing our scheme. There are endless papers evaluating the *performance* of density functional approximations. Here, we wish to categorize functionals based on their *behavior* for typical systems on which DFT calculations are run. Thus we construct a measure that requires *only* approximate DFT calculations and no exact results.

A second key feature is to include the behavior of densities and to compare different densities for different approximations. A primary reason for this is that, for some given system, two different approximations can easily yield negligibly different energies but typically have different densities. Such examples should contribute noticeably to any measure of the difference between two approximations. Also, quite a few density functionals are parameterized on the energy error, not the density error. The diverging trend for the maximal deviation of the densities for functional approximations discussed in Ref. [106] does not necessarily imply a diverging trend for the energy errors. This was pointed out in Ref. [80]. Hence, the best representative feature space for clustering should reflect differences in densities and energies. However, we must then convert density differences to energy differences in order to quantify such density differences meaningfully [150]. For example, Dick et al. [146] attempted to draw correlations among functionals by considering an integral of the L1-norm for the density difference. This is not a good measure of the density difference. Similarly, considering just the absolute errors in a specific type of energy quantity with reference energy and trying to correlate them in that energy space will be highly data-dependent and devoid of any density information [37]. If the density difference has little effect on energies, such differences are irrelevant to chemical properties and should not count significantly in our measure.

The theory of density-corrected DFT was developed for precisely this purpose [174]. Moreover, it was recently generalized to analyze differences between approximate functionals instead of errors

relative to the exact functional [150]. Finally, it has also been used to show the ambiguity in choices of measures of density and which density differences are relevant to chemical energetics.

Our first step is, for any KS-DFT calculation of ground-state energy with approximate functional A, to define:

$$D^A[B] = E^A[n_B] - E^A[n_A], \tag{6.1}$$

where $E^A[n]$ is the KS energy functional with XC approximation A and $n_C(\mathbf{r})$ is the self-consistent density of functional C for this problem. Thus $D^A[B]$ is the energy cost of using the *wrong* self-consistent density in the problem. This deceptively simple expression contains many essential features. First, it is non-negative, thanks to the variational principle. Second, it creates an energetic measure of the difference between $n_B(\mathbf{r})$ and $n_A(\mathbf{r})$, which can be compared to other relevant energies. Third, suppose B and A are (in some sense) similar. In that case, we expect this difference to be much smaller than relevant energy differences (indeed, to leading order in the difference between two functionals, it vanishes). Finally, if somehow A were the exact functional, this is a measure of the error in the density produced by B. In practice, one cares only about energy differences, and differences in D's are not non-negative, so we use absolute values. Moreover, since $D^A[B] \neq D^B[A]$, we take the average of the two as a symmetric measure of the difference between the functionals.

Finally, we must measure these differences on the only scale that matters: The energy differences predicted by the functionals themselves [155]. As mentioned above, when the difference between two functionals is small, we expect density-driven differences to be smaller than energy differences. A natural dimensionless choice is their ratio, which we expect to be about a few percent for typical density-insensitive cases. However, as mentioned, the energies can accidentally coincide, producing huge (and meaningless) ratios. Moreover, the more cases considered, the more likely such an accident will occur. We, therefore, compare the average of these differences over a

database to the root-mean-square average of energy differences in the database. Thus, for any pair of approximations A and B and a database of chemical reaction energies, we calculate

$$\eta(A,B) = \frac{\Delta\bar{D}(A,B)}{(\Delta E(A,B))_{rms}}. \tag{6.2}$$

We call this the density-driven fractional difference (DDF) of the two approximations. It is a general measure of how much they differ on the energy scale on which such differences matter. If $\eta << 1$, two functionals *behave* very similarly. It does not mean that they have similar accuracy, and often they will not. However, their treatment of systems and the densities they produce are markedly similar, and one should prefer the more accurate of the two (unless the cost difference is also substantial). On the other hand, if $\eta \sim 1$, the two functionals are very different and use very different features in the density.

Finally, we calculate the DDF matrix for 30 different functionals listed in Table. 6.1, using the Main Group Atomization Energies (MGAE109) dataset [187, 122]. The DDF is plotted with a color parametric plot in Fig. 6.1. Each row is a fingerprint for that given functional.

6.2.2 A metric space of approximate functionals

While the fingerprints of functionals provide an excellent measure of functional similarity, the DDF does not provide a metric on the space of functionals. The triangular inequality is violated, as the sum of the distances between PBE and MN11 and MN11 and BLYP are less than the distance between PBE and BLYP.

However, there is a convenient way to create a metric for this problem. We simply consider each functional as defining a new orthogonal direction in a P-dimensional space and the η-values are coordinates in that space. By construction, this is a vector space, and we choose the Manhattan

Table 6.1: A listing of the functionals used in this study. The name is the acronym or name of the functional. The year is the publication year. Type refers to which construction scheme was used. GH refers to global hybrid, RSH refers to range-separated hybrids, and NGA is nonseparable gradient approximation

Name	Year	Type	Short Range Exact Exchange	Long Range Exact Exchange	Reference
Hartree	1928	Hartree			[59]
HF	1935	HF			[60]
LSDA	1965	Local			[86]
PW91	1992	GGA			[116, 117]
PBE	1996	GGA			[115]
B3P86	1986	GH-GGA	0.2	0.2	[114]
LSDA-X	1965	Local			[86]
mPW91	1998	GGA			[2]
TPSSh	2003	GH-mGGA			[157]
BLYP	1988	GGA			[6, 88]
TPSS	2003	mGGA			[164]
B3LYP	1994	GH-GGA	0.2	0.2	[159]
revTPSS	2009	mGGA			[118, 119]
HSE06	2006	RSH-GGA			[63, 64]
PBE0	1999	GH-GGA	0.25	0.25	[3]
ωB97	2008	RSH-GGA	0	1	[20]
ωB97X	2008	RSH-GGA	0.1577	1	[20]
CAM-B3LYP	2004	RSH-GGA			[181]
LC-ωPBE	2006	RSH-GGA			[175]
M05	2005	GH-mGGA	0.28	0.28	[186]
M06	2006	GH-mGGA	0.27	0.27	[192]
M05-2X	2005	GH-mGGA	0.56	0.56	[188]
MO8-HX	2008	GH-mGGA	0.5223	0.5223	[191]
M06L	2006	mGGA			[189]
MN12-L	2012	mNGA			[124]
M11	2011	RSH-mGGA	0.428	1	[123]
MN15	2016	GH-mNGA	0.44	0.44	[184]
MN12-SX	2012	RSH-mNGA	0.25	0	[126]
M11-L	2011	mGGA			[125]
MN15L	2015	mNGA			[184]

(L1) norm,

$$d(A,B) = \sum_{i=1}^{P} |\eta(A,i) - \eta(B,i)|. \tag{6.3}$$

Note that with P different functionals, there will be 2 entries ($i = A$ and $i = B$) equal to $\eta(A,B)$ and $P-2$ is the difference between the η values of each functional on the other functionals. By construction, the distance or dissimilarity matrix, d, is real, symmetric, positive, and satisfies the triangular inequality. Therefore, we can use d to categorize our functionals, finding those clustered

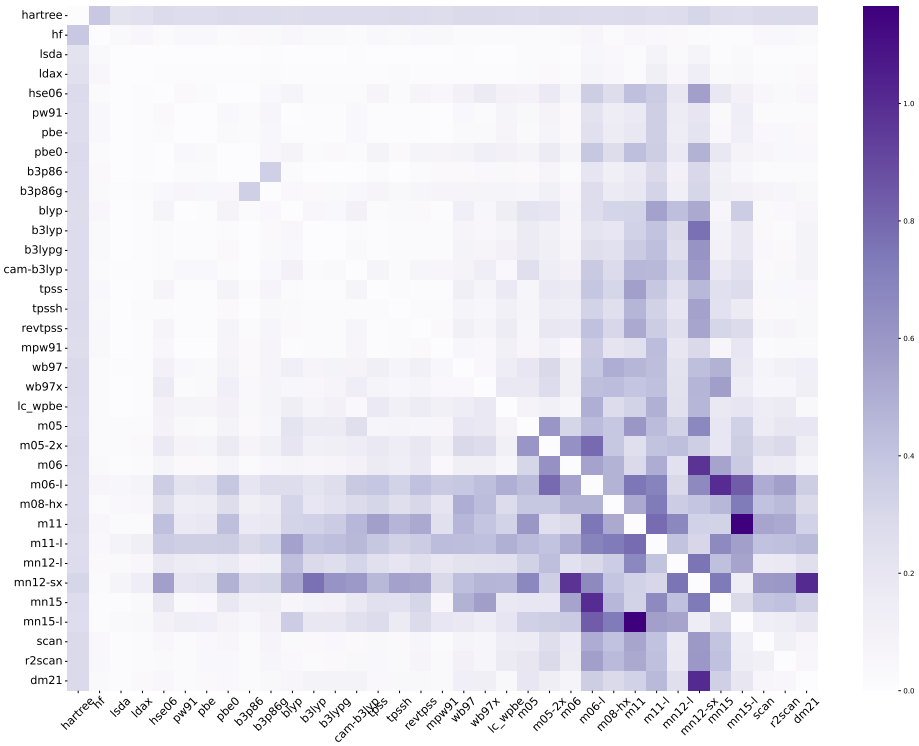


Figure 6.1: The DDF matrix of the MGAE109 dataset reaction energies for 30 functionals, plotted as a heatmap. Darker color corresponds to larger values.

in this P-dimensional space.

6.2.3 Clustering in d-space

Our second task is to choose an algorithm to perform unsupervised clustering of the approximate functionals in d-space. Again, many criteria were imposed on what might be an acceptable algorithm. We are looking for a clustering method suitable for small datasets with negligible dependence on user-defined parameters. First, all stochasticity should be eliminated. Most clustering algorithms (discussed in the previous chapter) require the specification of hyperparameters, including the possible number of clusters, cut-off radius, linkage criterion, or the number of neigh-

bors. The choice of these parameters can introduce a bias towards expectations based on domain knowledge. Since clustering is an unsupervised learning method and we do not have absolute labels, quantifying clustering error is another hurdle. Performance evaluation using Silhouette score analysis [134], Davies-Bouldin Index [31], or Calinski-Harabasz Index [16] tend to favor convex, centroid-based clusters. Since we only have very few functionals, and hence a small dataset to calculate the fingerprints, expecting the clusters to have convex globular structures (as most algorithms like k-means will produce) and evaluating clustering quality to optimize hyperparameters are equally impractical.

The agglomerative hierarchical partitioning method is often preferred for small datasets with non-convex clusters. We can easily construct a dendrogram for our problem, but deciding the cut-off distance for the dendrogram is a crucial parameter-optimization task. The linkage criterion decides how the clusters are combined once the initial clusters form. It is also a vital hyperparameter. However, there is a better way to look at hierarchical clusters. We can look at the dendrogram and quickly evaluate which functional is closer to PBE or M11.

HDBSCAN [17], a hierarchical density-based clustering method, is an attractive alternative to single-linkage agglomerative clustering, as it requires choosing a minimum cluster size that can be more intuitive. However, HDBSCAN declares data points as noise based on another less intuitive user-defined parameter. As a result, HDBSCAN can have difficulty deciding between noise and actual data for a tiny dataset, such as ours.

We developed a parameter-free clustering technique based on motivation from hierarchical clustering and HDBSCAN. For any given functional A, we find

$$d_{min}(A) = \min_{B \neq A} d(A,B), \quad B_{min}(A) = \arg\min_{B \neq A} d(A,B) \tag{6.4}$$

Next, we order all d_{min} in order of increasing value. For each value, consider the pair of functionals (A, B) it connects. If neither are already clustered, we begin a new cluster. We call such functionals

seeds. If either already appears in a cluster, the other is added to that cluster. In other words, we calculate an adjacency matrix based on the distance matrix. Functionals adjacent to each other are part of the same cluster. Thus, this is a chain-like clustering, depending only on connections with nearest neighbors. We call this one-step nearest-neighbor clustering (ONN).

Our algorithm is the first half of an algorithm created a few years ago, called the first integer neighbor clustering hierarchy (FINCH) [139]. Our algorithm is not hierarchical. While our algorithm and FINCH can use different distance measures to create the distance matrix from the DDF, we prefer using the L1 norm, and FINCH recommends cosine similarity. Cosine similarity or cosine distance (1-cosine similarity) is a trendy choice for making the distance matrix in data-intensive unsupervised learning tasks in natural language or image processing. For us, all the functional fingerprints are positive values and are located in the first quadrant of the $P-$dimenisonal space. The sense of direction is less important to us than magnitude; hence, we consider the L1 norm a better choice. Also, bot L2 norm or the Euclidean distance and cosine distance(or normalized Euclidean distance) suffer from the curse of dimensionality [50] to a more considerable extent compared to the L1-norm.

Our algorithm has many desirable features: It is speedy to run (although not very relevant for our problem), it decides how many clusters to make without any hyperparameter, and can easily be rerun when more data becomes available. It naturally uses the distance scale built into the data set and accounts for variations between different clusters. The clusters calculated from the ONN algorithm are shown in Fig. 6.2. We also introduce a second step by looking at the initially formed clusters. We define a distance cut-off to divide the clusters formed in the first step without reconstructing the distance matrix for each cluster. We define average cluster distance for cluster k as:

$$d_k = \frac{\sum_{i=1}^{N_k} d_i}{N_k}, \tag{6.5}$$

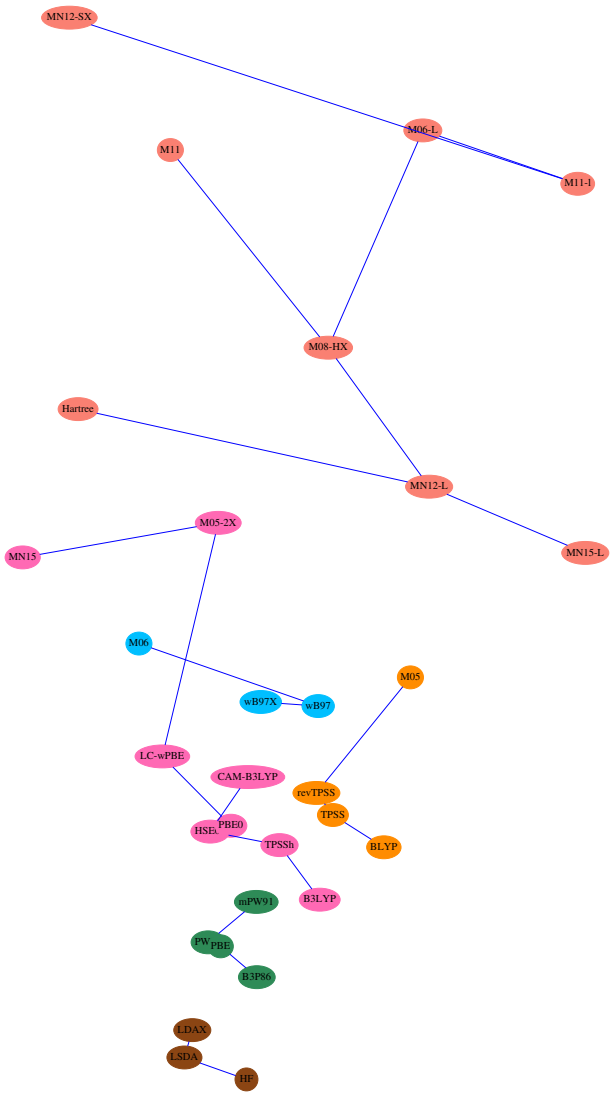


Figure 6.2: Visual representation of clusters found by unsupervised learning algorithm using functional fingerprints. Distances to neighbors are to scale.

where N_k is the total number of functionals in cluster k and d_i is the distance between neighboring functionals. The cut-off distance is defined as $d_{cut-off} = 2 \times d_k$. This measure is arbitrary. Any functional falling outside the cut-off radius can be considered noise (in the spirit of HDBSCAN). Hence we get clusters and not a simple partition of data. Since ONN yields chain-like clusters, this approximate extension is relevant. However, this is an additional step; if we do not want to disregard any functionals, we can avoid it. Including this step results in more meaningful clusters, as shown in Table. 6.2.

Fig. 6.2 reflects the actual distances between each pair of functionals. We cluster just 30 popular approximate functionals, using only DFT calculations on an atomization energy database and a clustering algorithm. The machine has found a cluster dominated by the local density approximation (saddle brown), a cluster of generalized gradient approximations (green), a cluster of meta-GGA's (yellow), a cluster of hybrids (pink), a cluster dominated by highly fitted functionals, such as ω-B97 and its variants (blue), and finally a cluster dominated by Minnesota functionals of later vintage (orange). B3P86 and BLYP, followed by a few Minnesota functionals, often deviate from the expected norms. The positions of these functionals in the $P-$dimensional spaces can be highly data-dependent. We may see them hopping to another cluster if we change the dataset.

Table 6.2: The clusters calculated using ONN with a distance cut-off, the identified nearest neighbor for each functional and the distance between them, the average cluster distance for distance cut-off, and the cluster numbers for the functionals calculated from agglomerative hierarchical clustering with complete linkage and HDBSCAN. L1-distance measure is used in all cases. The number of clusters = 6 for agglomerative clustering, minimum cluster size = 2, and the minimum number of samples = 1 for HDBSCAN. A cluster assignment -1 refers to outliers/noise

Functionals	Nearest Neighbor	Distance to Nearest Neighbor	Average Cluster Distance	Complete Linkage	HDBSCAN
PW91	PBE	0.1788	0.48	1	1
PBE	PW91	0.1788		1	1
B3P86	PBE	0.7271		1	1
mPW91	PW91	0.8379		1	1
HSE06	PBE0	0.3396	1.45	2	3
PBE0	HSE06	0.3396		2	3
TPSSh	HSE06	0.8499		2	3
CAM-B3LYP	HSE06	0.9603		2	-1
B3LYP	TPSSh	1.2137		2	-1
LC-ωPBE	PBE0	1.6039		2	-1
M05-2X	LC-ωPBE	3.081		3	-1
MN15	M05-2X	3.1823		3	-1
LSDA	LDAX	0.3893	0.55	1	2
LDAX	LSDA	0.3893		1	2
HF	LSDA	0.8867		1	2
TPSS	revTPSS	0.4038	1.19	2	4
revTPSS	TPSS	0.4038		2	4
BLYP	TPSS	1.591		2	-1
MO5	revTPSS	2.3844		2	-1
ωB97	ωB97X	0.9255	1.45	2	5
ωB97X	ωB97	0.9255		2	5
M06	ωB97	2.4975		2	-1
M08-HX	MN12-L	3.4659	4.1859	4	6
MN12-L	M08-HX	3.4659		4	6
M06-L	M08-HX	4.1795		5	-1
Hartree	MN12-L	4.1858		4	-1
MN15-L	MN12-L	4.3563		3	-1
M11-L	M06-L	4.3889		5	-1
M11	M08-HX	4.5122		6	-1
MN12-SX	M11-L	4.9329		6	-1

What do we learn from the ONN clusters? First, it somewhat generalizes Jacob's ladder (not necessarily in the given order), which categorizes functionals based on their ingredients rather than their results for realistic calculations. For example, a new functional might use ingredients from a higher rung of that ladder ineffectively However, it would be classified here on the lower-rung, as it would behave like its lower-rung analogs. Given some new functional approximation, any developer or user can find which category it belongs to and immediately draw insight from comparison with well-established members of that category, i.e., it should perform in some way better (or even best) against other members of its cluster, but need not be better than members of other clusters. Third, since ONN uses no information about where a functional comes from in the definition of the feature space, counting empirical parameters (or even exact conditions satisfied) is not used to determine its cluster (directly). (Of course, the judicious use of either parameters or conditions can still improve its accuracy). We also show the clusters calculated using hierarchical agglomerative clustering with complete linkage and HDBSCAN in Table. 6.2. The hyperparameters for these two algorithms are optimized to yield 6 clusters based on ONN clustering. We can quickly spot the similarities between ONN and HDBSCAN clusters. The only issue is that HDBSCAN identifies more than half of the functionals as outliers or noise. In a clustering space of 30 data points, constructing and separating high-density regions from low-density regions become meaningless.

Agglomerative hierarchical clustering with complete linkage, on the other hand, does produce linked clusters. It tends to combine clusters with the smallest distance (ONN cluster 1 with cluster 3, ONN cluster 2 with cluster 4, and cluster 5). However, these clusters are slightly misleading. Why does combining the clusters become more apparent if we look at the dendrogram? The heatmap of the distance matrix in Fig. 6.3 is representative of the clusters formed from the complete linkage. The orange and the green-colored functionals in the dendrogram represent two significant clusters. As we specified 6 clusters, all it does is cut the dendrogram at a distance that will dissect six vertical lines. The green-colored cluster of the Minnesota functionals will account for at least 3 of the 6 clusters. The dendrogram, however, reflects what will happen if we could cut it at

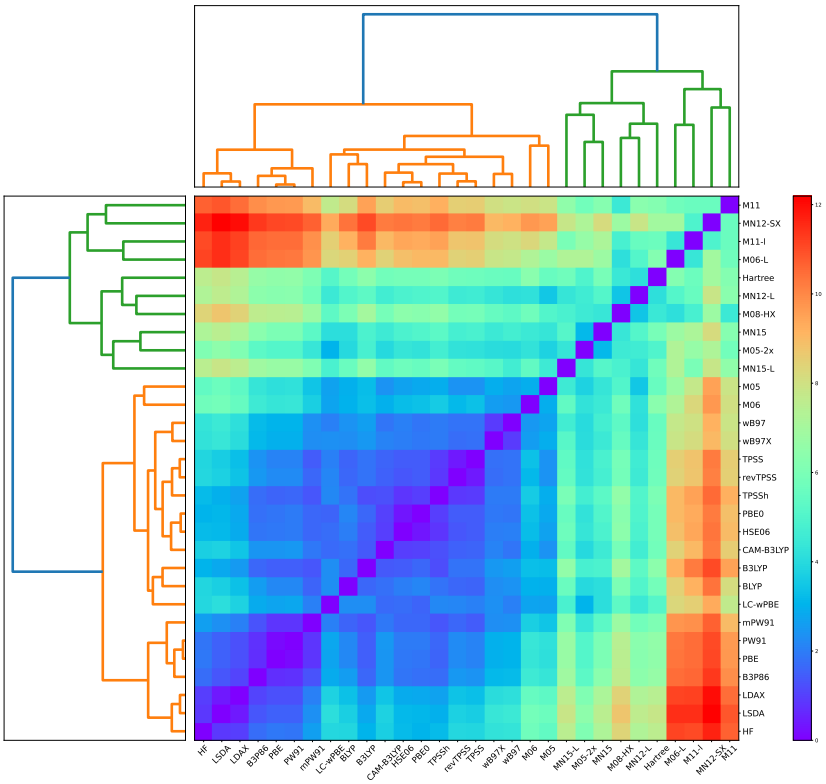


Figure 6.3: A comparison of the dendrogram for complete linkage hierarchical agglomerative clustering with optimal ordering and the colormap of the distance matrix constructed by calculating the Manhattan distance measures from the DDF matrix of the MGAE109 dataset

variable distances (like HDBSCAN), and we see a stair almost equivalent to Jacob's ladder for the functionals in orange. Most Minnesota functionals end up together, forming their own clusters. The situation is not so different if we perform single-linkage hierarchical clustering, as shown in Fig. 6.4. The only significant difference is that GGAs come before local functionals and HF. However, the relative ordering of the leaves in the dendrogram is not meaningful; only their height reflects the actual distances between them.

Now let us look at the connectivity of the pairs with the three clustering methods. We do not need to be experts to say that TPSS and revTPSS are similar to each other compared to other functionals. The same goes for ωB97 and ωB97X. Sometimes, it is helpful to look at the den-

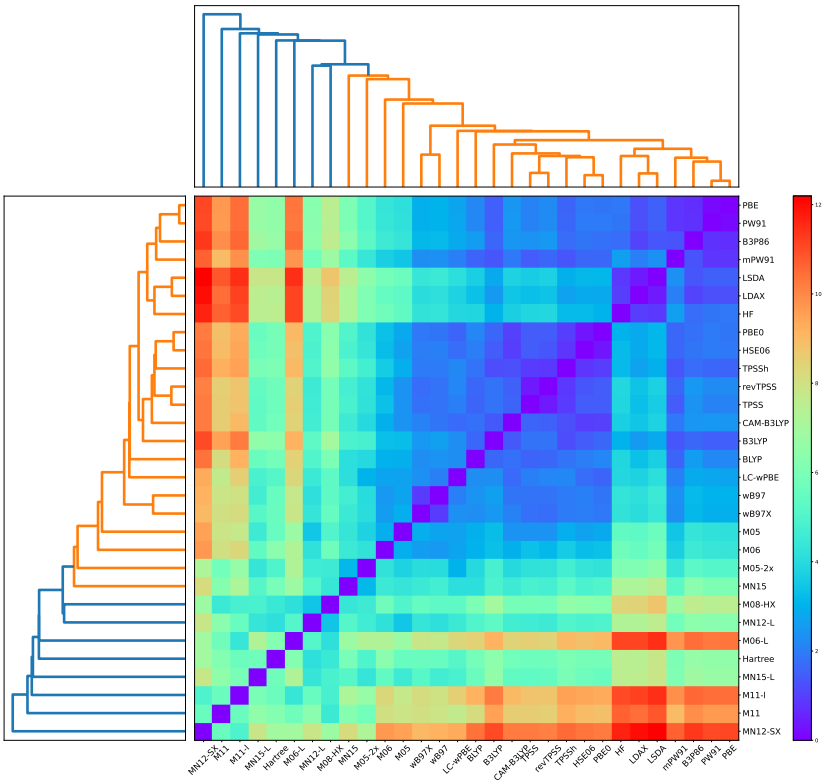


Figure 6.4: A comparison of the dendrogram for single linkage hierarchical agglomerative clustering with optimal ordering and the colormap of the distance matrix constructed by calculating the Manhattan distance measures from the functional fingerprints of the MGAE109 dataset.

drograms from agglomerative clustering as ONN, not being hierarchical, misses things like TPSS and revTPSS being closer to TPSSh by definition. While with ONN, B3LYP, CAM-B3LYP, and LC-ωPBE are happily in the hybrid cluster, these three, together with BLYP, form separate orders with complete and single linkage clustering (all are outliers with HDBSCAN). Why, with ONN, do BLYP and B3P86 end up in the wrong place? Why is a functional like B3LYP considered an outlier by HDBSCAN? Our list of functional approximations is highly dominated by nonempirical approximations that subscribe to Jacob's ladder. Our machine learning algorithm shows something similar. However, can we say that the BLYP and the post BLYP functionals abide by this ladder? Is there a connection with parameterization? We will undoubtedly need a larger functional space to provide a conclusive answer.

With all three clustering algorithms, we see something amiss with recent speculation of Ref. [106]. Minnesota functionals are confirmed by the appearance of the separate Minnesota clusters. In most cases, only M05 and M06 often cluster with the other functionals, and even they are often joined in last, just after ωB97 and ωB97X. The rest of the Minnesota functionals are more different from other functionals than any others, except for Hartree, which means setting XC to zero! These observations are consistent across all three clustering algorithms we discussed. It shows that their densities are more discernibly different from most other approximations suggested (but avoiding any other metric than the energy, in line with principles of DC-DFT [150, 151]). Both ONN and complete linkage dendrograms also provide some insights into the similarity and dissimilarity among different Minnesota functionals. One crucial insight from the Minnesota functional cluster is that the machine does not necessarily discern the effect of the number of parameters. We do not find any correlation between the number of parameters in each Minnesota functional and whom it chooses to be its neighbor. Hence, unsupervised learning inherently characterizes parameterization, and it is not just a function of the number of parameters; it also takes into consideration the datasets used, the relative importance of the parameters, and several other factors.

So what question does this clustering answer? Indeed, it does not directly tell you 'which functional to use.' Instead, it tells you which type of functional it is and how you can hope it will behave on problems of interest. For example, we have included about 30 functionals, possibly less than 5% of those in use. All others, and any new ones, can be run on the same database to see which cluster they fall within. In addition, a new functional using novel ingredients can be immediately compared with existing ones to see if those novel ingredients lead to new behavior.

We have also analyzed only a tiny database of standard chemical energies. One can also run on many others, such as non-covalent interactions, to see if the clustering changes. Based on our experiments, we can say that a dataset has to be density-sensitive. That is, the density difference should vary from functional to functional for our analysis to be valid. We automatically inherit this requirement from DC-DFT. In general, non-covalent interactions are less density-sensitive.

To demonstrate the usefulness of our approach, we discuss the clustering results with ONN for three state-of-the-art functionals, SCAN [162], r2SCAN [47, 48], and DM21 [85]. However, we propose a semi-supervised learning technique instead of remaking a DDF including these three functionals and then recalculating the distance matrix. We will now only determine the functional fingerprints for the three functionals and determine their distance from the rest of the thirty functionals. Then we will determine the nearest neighbor for the three functionals, and the cluster number associated with that nearest neighbor will also be the cluster number for the new functional. The results are presented in Table. 6.3. The results confer human expectations. Both SCAN and r2SCAN end up with meta-GGA functionals, and DM21, being a global hybrid, ends up with the hybrid functionals. DM21 is a particular member of the BLYP family, as it was trained based on B3LYP densities. Despite having different parameter-dependencies compared to B3LYP, and self-consistent evaluation of the functional (hence the density should be different from B3LYP), DM21 remains a close ally to B3LYP.

Table 6.3: Semi-supervised cluster assignments for SCAN, r2SCAN, and DM21 based on the ONN clustering.

Functional	Nearest Neighbor	Distance to Nearest Neighbor	Cluster assignments
SCAN	revTPSS	1.2589	4
r2SCAN	revTPSS	1.4081	4
DM21	CAM-B3LYP	1.6139	2

6.2.4 Dimensionality reduction

As we discussed, observations we made from the three clustering methods, Fig. 6.2, and the dendrograms in Fig. 6.3, and Fig. 6.4, raise questions that may not have any right answers. What could be another way to discern the complex layouts of these clusters? Of course, the best way would be if we could visualize the 30-dimensional space and see how the functional fingerprints are arranged in that space. However, unless we are superhumans, that is impossible. This is where dimensionality reduction, another unsupervised learning technique, comes in handy. The principal component

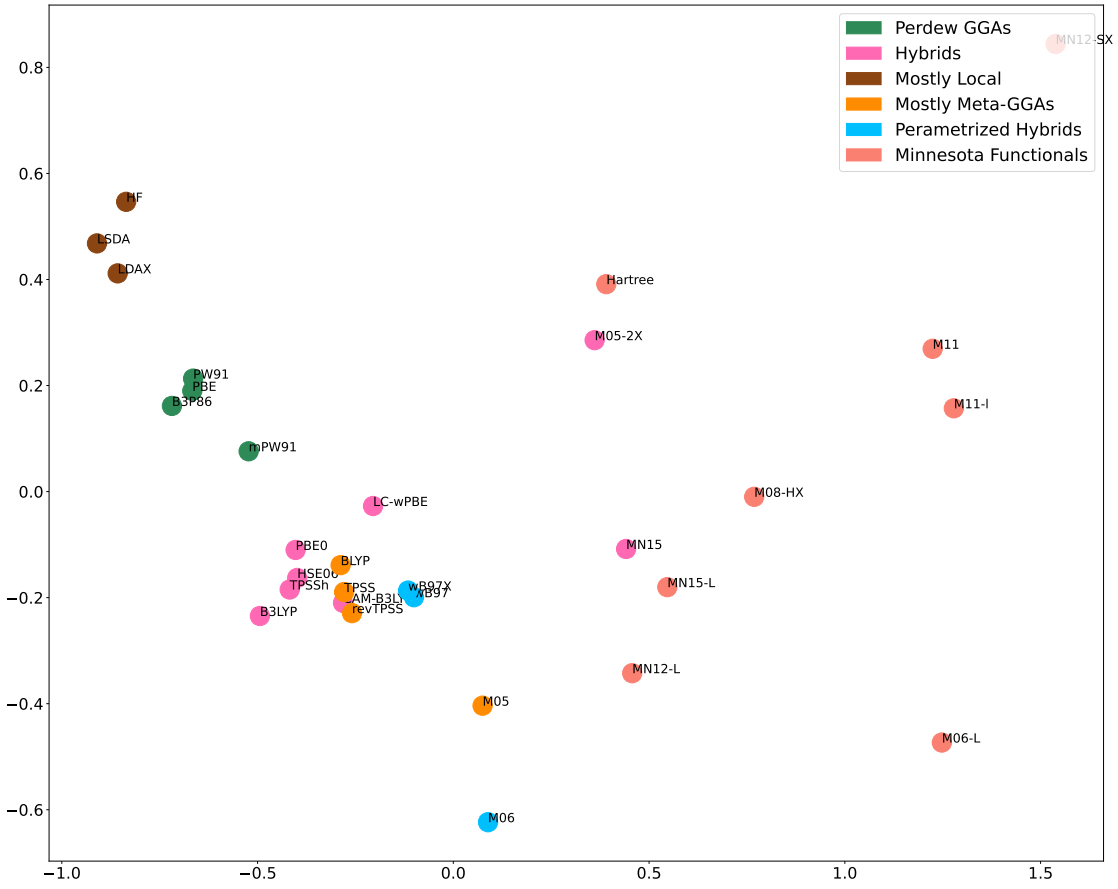


Figure 6.5: PCA plot of the DDF matrix. PCA was performed with five components based on the percentage variance associated with the eigenvectors (see Fig. B3 in Appendix). Clusters are marked based on the nearest-neighbor clustering results presented in Table. 6.2.

analysis is one of the most popular dimensionality reduction techniques. Fig. 6.5 show the relative ordering of the ONN clusters in the space of the first two principal components of the DDF matrix. While the first two clusters and the last cluster are somewhat apparent, we cannot exactly separate the three intermediate clusters. Other linear and nonlinear manifold learning methods, such as multidimensional scaling (MDS) [166] (Fig. 6.2 uses the same technique). The nonlinear extension of it, isometric feature mapping (Isomap) [165], can produce somewhat sensible plots (please

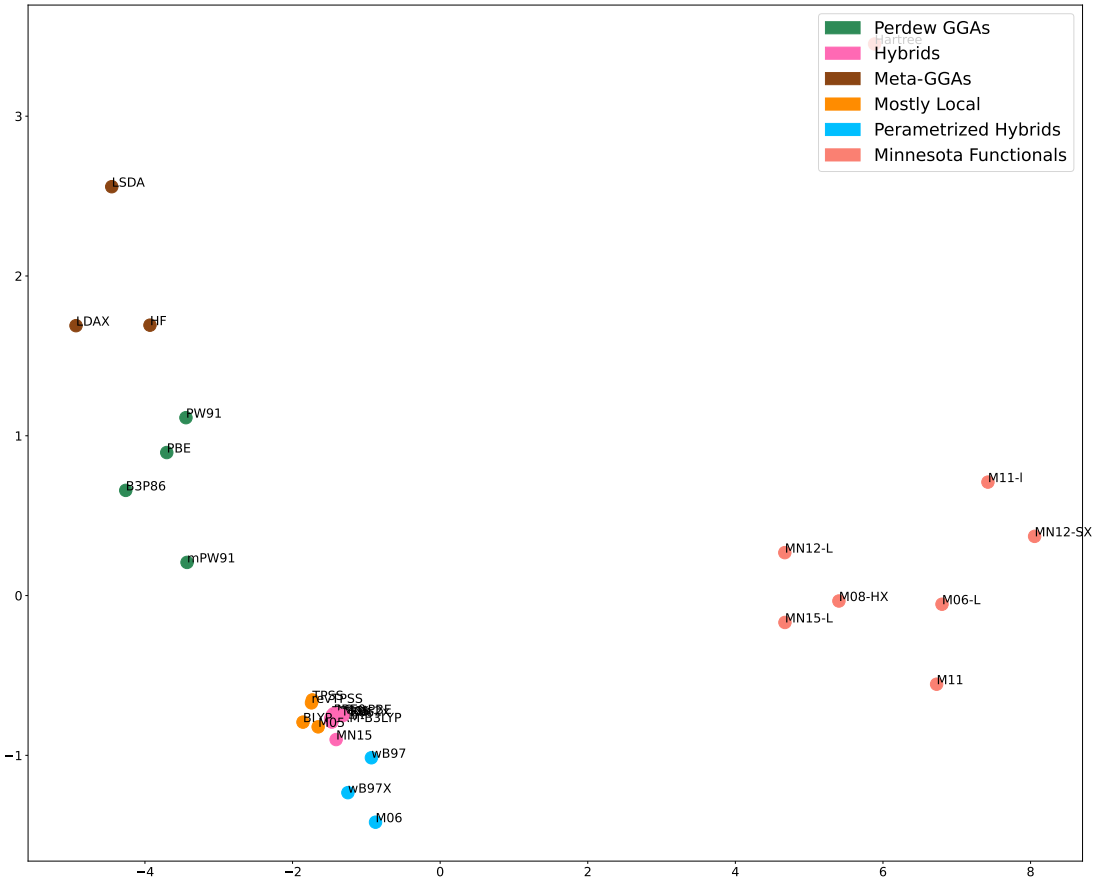


Figure 6.6: Vizualization of the functional fingerprints with h-NNE. Clusters are identified based on the nearest-neighbor clustering results presented in Table. 6.2. MDS and Isomap plots are included in the Appendix.

see Fig. B5 in Appendix). However, the choice of hyperparameter involved in these methods can influence the final results. On the other hand, probability distribution-based methods like t-SNE yield less than optimal representations of the clusters (see Fig. B4 in Appendix).

Following FINCH-related development, in Fig. 6.6, we plot the hierarchical nearest neighbor embedding or h-NNE [138]. h-NNE initially performs the FINCH clustering, then for each cluster, does a PCA to order the functionals within the cluster, and then defines a centroid for the cluster.

Then it performs another PCA with just the centroids to decide the relative ordering of the clusters. We modified our approach to accommodate the L1-norm and one-step clustering. It certainly separates the clusters, without one cluster overlapping with another cluster. However, clusters 3,4 and 5 are placed extremely close together, and almost all cluster 2 functionals reduce to a single point. Due to the small data size, the centroid-based approach may not work well for us. However, the x-axis or the ordering of the clusters is representative of Jacob's ladder, or more precisely, our machine's ladder. We need to define a different dimensionality reduction technique for ONN clustering to visualize the formed clusters better.

6.3 Conclusion

We have shown a new scheme for comparing density and energies from density functionals. We define a meaningful feature with information for the density-difference and energy difference of a functional with respect to other functionals. Then, we perform clustering on the feature space for 30 functionals using three different clustering methods, including one of our parameter-free methods. Finally, we analyzed the functional groups that the machine produced and tried to visualize them through dimensionality reduction. The relationship of functionals provides an effective measure with deep insight into functional density-energy performance even without a reference for accuracy. We have demonstrated its use on standard atomization energy datasets, highlighting how this method can identify unique components and give new insight into density functionals. Furthermore, our approach is easily extendable to larger datasets and other forms of energy.

Can we say some functionals stray from the path? While the initial work raising this question used chemically irrelevant densities and arbitrary choices of density measures, a careful search using DC-DFT (and therefore using the energy as the chemically meaningful measure of relevant density differences) found no examples where poor densities (even if rather odd looking) produced poor energies. However, our analysis focuses on similarities between approximations rather than

energetic accuracy, produces a clear picture of certain empirical functionals have very different densities than almost all others, presumably related to overfitting, thus putting the misgivings implied in Ref. [106] on a more systematic footing.

6.4 Appendix

6.4.1 Density-corrected DFT

For a given external potential v and an exchange-correlation functional $E^A_{\mathrm{XC}}[n]$, the total energy functional within the Kohn-Sham DFT (KS-DFT) framework reads as:

$$E^A_v[n] = T_{\mathrm{S}}[n] + U[n] + E^A_{\mathrm{XC}}[n] + \int d^3r\, n(\mathbf{r}) v(\mathbf{r}), \tag{B1}$$

where $T_{\mathrm{S}}[n]$ is the KS non-interacting kinetic energy and $U[n]$ is the classical Hartree energy. The corresponding ground-state energy is obtained from the following minimization:

$$E^A_v = \min_n E^A_v[n], \tag{B2}$$

where we refer to the minimizing density, n^A_v, as the *native* density of functional A. We now consider $E^B_{\mathrm{XC}}[n]$, a second exchange-correlation functional, which yields its native ground state density n^B_v and the ground state energy E^B_v. We define the difference between ground-state energies obtained from A and B,

$$\Delta E^{A/B}_v = E^A_v - E^B_v, \tag{B3}$$

and we will use the recently developed generalization of DC-DFT, *density functional analysis* [174], to decompose $\Delta E^{A/B}_v$ into functional- and density-driven terms. For any A, an energetic measure for a distance between its native n^A_v and any n^B_v that is isoelectronic with n^A_v, is given by:

$$d^{A/B}_v = E^A_v[n^B_v] - E^A_v[n^A_v] \geq 0. \tag{B4}$$

We can use this measure to partition $\Delta E_v^{A/B}$ as:

$$\Delta E_v^{A/B} = \Delta E_{\mathrm{XC}}^{A/B}[n_v^B] - d_v^{A/B}, \tag{B5}$$

where,

$$\Delta E_{\mathrm{XC}}^{A/B}[n] = E_{\mathrm{XC}}^{A}[n] - E_{\mathrm{XC}}^{B}[n]. \tag{B6}$$

Reversing the order of A and B, we can also write $\Delta E_v^{A/B}$ as:

$$\Delta E_v^{A/B} = \Delta E_{\mathrm{XC}}^{A/B}[n_v^A] + d_v^{B/A}. \tag{B7}$$

The first term on the right-hand side of Eqs. B5 and B7 is the difference between the two functionals evaluated on each of the two native densities and thus are functional-driven terms. The second term on the r.h.s. of Eqs. B5 and B7 is the density-driven term as it is given by the difference between the same energy functional evaluated on different densities.

Equations B5 and B7 are given for total energies for a given external potential v. For energy differences (e.g. atomization energies, electron affinities, etc.), Eqs. B5 and B7 take the same form and are formally derived in Section 5 of Ref. [174]. For example, the energy difference between total energies from two external potentials: $\Delta E^A = E_1^A[n_1^A] - E_2^A[n_2^A]$, the underlying $\Delta d^{A/B}$ reads as:

$$\Delta d^{A/B} = \underbrace{E_1^A[n_1^B] - E_1^A[n_1^A]}_{d_1^{A/B}} + \underbrace{E_2^A[n_2^A] - E_2^A[n_2^B]}_{-d_2^{A/B}}. \tag{B8}$$

More generally, for energy differences involving many external potentials indexed by i:

$$\Delta E_i^A = \sum_{p=1}^{P} E_{i,p}^A[n_{i,p}^A] - \sum_{q=1}^{Q} E_{i,q}^A[n_{i,q}^A], \tag{B9}$$

$\Delta d_i^{A/B}$ is given by:

$$\Delta d_i^{A/B} = \sum_{p=1}^{P} d_{i,p}^{A/B} - \sum_{q=1}^{Q} d_{i,q}^{A/B}. \tag{B10}$$

While $d_v^{A/B}$ is always non-negative (Eq. B4), the sign of $\Delta d_i^{A/B}$ is not definite. For this reason, we define:

$$\Delta D_i^{A/B} = \left|\Delta d_i^{A/B}\right|, \tag{B11}$$

which, in general, is not equal to $\Delta D_i^{B/A}$. We can write $\Delta D_i^{A/B}$ as:

$$\Delta D_i^{A/B} = \Delta D_i^{A+B} + \Delta D_i^{A-B}, \tag{B12}$$

where ΔD_i^{A+B} is symmetric, and ΔD_i^{A-B} is an anti-symmetric contribution to $\Delta D_i^{A/B}$, which are defined as:

$$\Delta D_i^{A\pm B} = \frac{1}{2}\left(\Delta D_i^{A/B} \pm \Delta D_i^{B/A}.\right) \tag{B13}$$

6.4.2 Functional fingerprints

We have now established key quantities that will be used to construct a fingerprint for a pair of functionals A and B, which we define here as their density-driven difference (DDF) on a scale of a full difference in their energies. Thus, for a given i energy of interest, one would naturally think of the following DDF indices: $D_i^{A/B} / \left|\Delta E_i^{A-B}\right|$, or symmetric $D_i^{A+B} / \left|\Delta E_i^{A-B}\right|$, where:

$$\Delta E_i^{A-B} = \Delta E_i^{A} - \Delta E_i^{B}. \tag{B14}$$

However, the two indices would be problematic when $\left|\Delta E_i^{A-B}\right|$ is small, as they would diverge when $\left|\Delta E_i^{A-B}\right| \to 0$. To fix this problem, we introduce the *functional fingerprint scale*. For a chosen property of interest (e.g., atomization energies of an organic molecule), we calculate the functional fingerprint scale by using a dataset with similar systems/properties (e.g., a dataset containing atomization energies of a few dozens of organic molecules). We set this scale as the root-mean-square of J data points (i.e., ΔE_i^{A-B} energies) that form the dataset:

$$K^{A+B} = \sqrt{\frac{1}{J}\sum_{i=1}^{J}\left(\Delta E_i^{A-B}\right)^2}, \tag{B15}$$

which is symmetric and non-negative by definition. We can now use K^{A+B} in the denominator of our functional fingerprint without worrying about its divergence. Thus, for a given i energy of interest, we define the functional fingerprint of A and B as:

$$\eta_i^{A/B} = \frac{\Delta D_i^{A/B}}{K^{A+B}} \geq 0. \tag{B16}$$

Plugging Eq. B12 into Eq. B16, we can partition $\eta_i^{A/B}$ into symmetric and anti-symmetric contributions:

$$\eta_i^{A/B} = \eta_i^{A+B} + \eta_i^{A-B}, \tag{B17}$$

where,

$$\eta_i^{A\pm B} = \frac{\Delta D_i^{A\pm B}}{K^{A+B}} = \frac{1}{2}\left(\eta_i^{A/B} \pm \eta_i^{B/A}\right). \tag{B18}$$

While $\eta_i^{A+B} \geq 0$, η_i^{A-B} can also be negative. We can now obtain $\eta^{A/B}$ averaged over the same dataset used to calculate K^{A+B} (Eq. B15):

$$\eta^{A/B} = \frac{1}{J}\sum_{i=1}^{J}\eta_i^{A/B}, \tag{B19}$$

Plugging Eq. B17 into Eq. B19, we obtain:

$$\eta^{A/B} = \eta^{A+B} + \eta^{A-B}. \tag{B20}$$

where,

$$\eta^{A\pm B} = \frac{1}{J}\sum_{i=1}^{J} \eta_i^{A\pm B} = \frac{1}{2}\left(\eta_i^{A/B} \pm \eta_i^{B/A}\right). \tag{B21}$$

Our η index or its symmetrized part allows the comparison of dozens of approximate exchange-correlation functionals without a need to have access to either exact energies or exact densities. Using the methodology developed in this section, we can take a specific dataset and construct the following matrix containing similarity indices for a selection of functionals:

$$M = \begin{pmatrix} \eta^{A/A} & \eta^{A/B} & \dots & \eta^{A/Z} \\ \eta^{B/A} & \eta^{B/B} & & \eta^{B/Z} \\ \vdots & & \ddots & \\ \eta^{Z/A} & \eta^{Z/B} & & \eta^{Z/Z} \end{pmatrix}, \tag{B22}$$

where A is the first and Z is the final tested functional.

6.4.3 Calculation details

All self-consistent and non-self-consistent calculations are performed with PYSCF 2.0 [163]. The AUG-cc-pVQZ basis set has been used for all the functionals. The energy convergence thresholds were set to 1e-8. Numerical quadrature grids of size seven are used for SCAN. For all other functionals, grid size was reduced to 4. An unrestricted scheme is used for all open-shell calculations.

Construction of the DDF matrix, clustering, and dimensionality reduction tasks are performed using different Python libraries. Both SciPy [172] and scikit-learn [113] libraries are used for the

machine-learning tasks. The HDBSCAN library [105] is used for clustering with HDBSCAN.

The NetworkX package [56] was used to generate the network graph for Fig. 6.2, and PyGraphviz 1.7 was used for visualization of the graph (with the *Neato* layout). The hierarchical nearest neighbor embedding plot in Fig. 6.6 was generated with the h-NNE [138] package.

6.4.4 Clustering tendency

We can ensure that the data set has a clustering tendency and does not contain uniformly distributed points by calculating the Hopkins statistic (H) [68]. The Hopkins statistic, H, is a statistical hypothesis test where the null hypothesis assumes that the data is generated from a Poisson point process and hence is uniformly randomly distributed.

The Hopkins statistic for the functional fingerprints constructed for the MGAE109 dataset is $H = 0.72$. Furthermore, it was averaged over 100 random iterations. Hence, the functional fingerprints have high clustering tendencies.

6.4.5 Clustering quality

The quality of the formed clusters (in the absence of absolute data labels) can be evaluated using several internal measures. Here we calculate the Silhouette scores [134] and the DB indices [31] for all four clustering algorithms reported in Table. 6.2. the calculated scores are reported in Table. B1. Silhouette scores and the DB index tend to favor convex globular clusters based on their definitions. Hence, these scores do not reflect the quality of the functional clusters for all methods covered in this manuscript. Also, due to the small size of the dataset, the Silhouette coefficients do not reach a value beyond 0.45 (for 2 clusters) even with k-means clustering. For 6 clusters, k-means clusters for the functional fingerprints of the MGAE109 dataset is close to 0.30.

Table B1: The Silhouette scores and the DB indices for the three clustering algorithms in Table. 6.2. Manhattan distance measure is used in all three cases.

Internal Measures	**Nearest-neighbor based clustering**	**Complete Linkage** (6 clusters)	**HDBSCAN** (6 clusters)
Silhouette scores	0.06	0.28	0.12
DB index	1.38	1.20	1.17

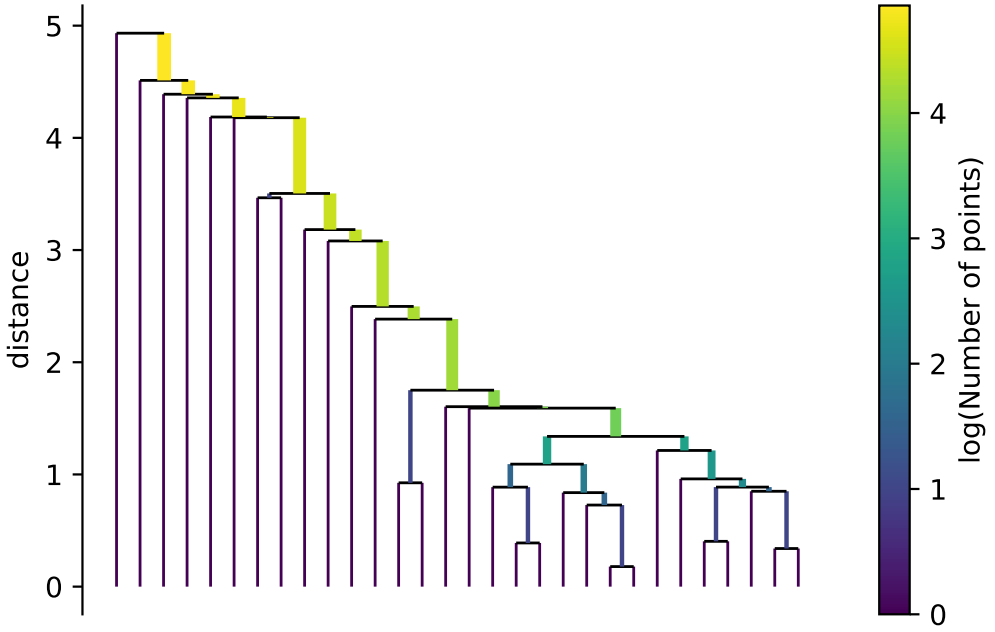
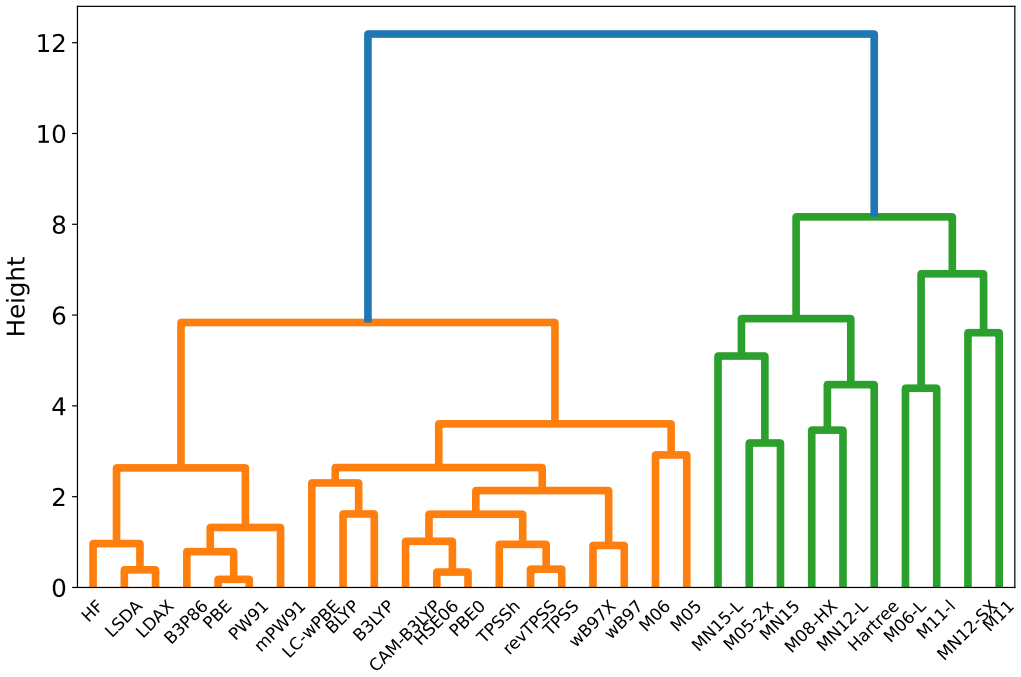
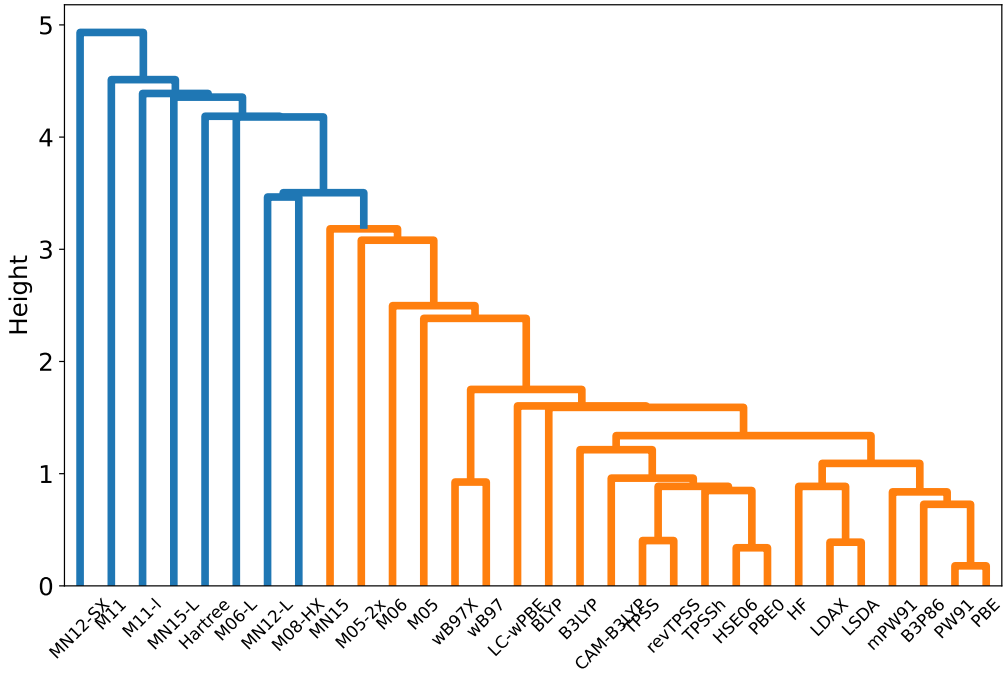


Figure B1: The HDBSCAN dendrogram for the MGAE109 dataset functional fingerprint is calculated with Manhattan distance measure. minimum cluster size = 2, the minimum number of samples = 1.



(a)



(b)

Figure B2: Hierarchical dendrogram for (a) complete linkage and (b) single linkage clustering with optimal ordering for the functional fingerprints of the MGAE109 dataset

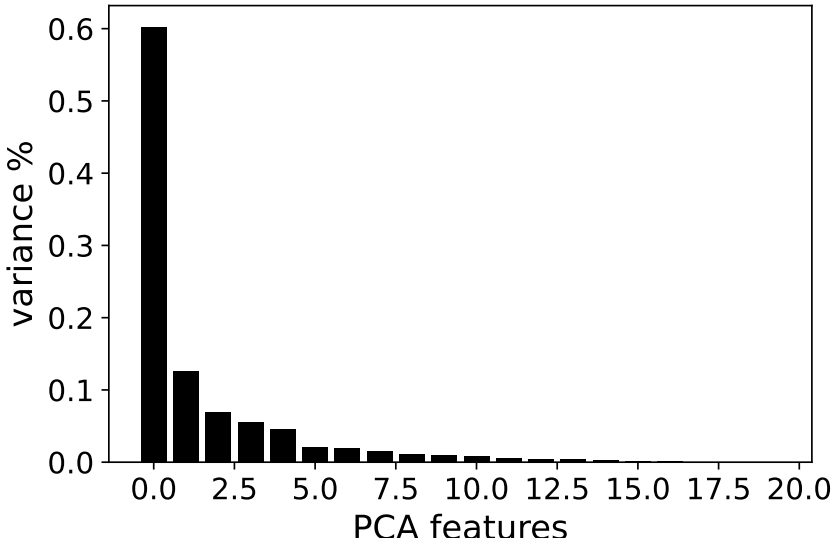


Figure B3: Percentage variance with respect to the first twenty PCA components of the DDF matrix of the MGAE109 dataset.

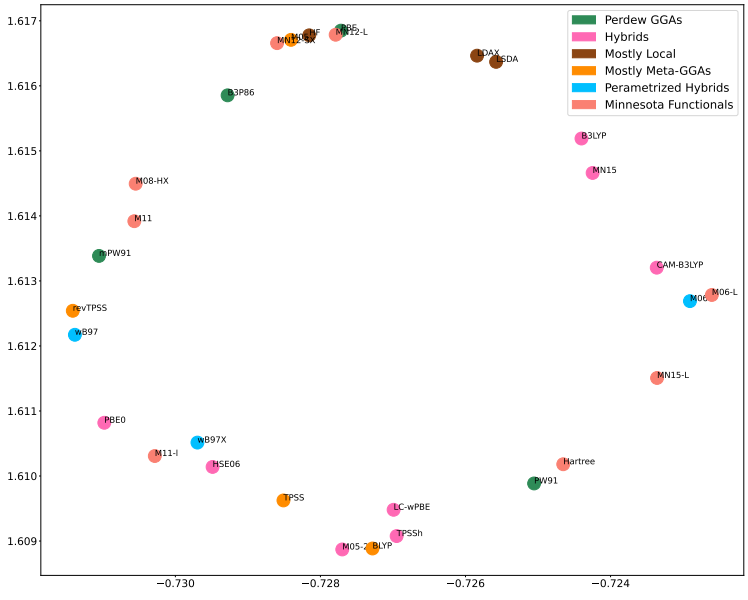
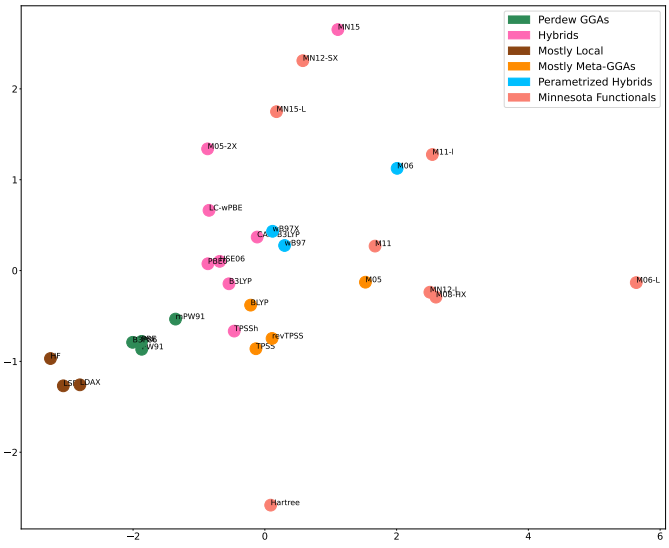
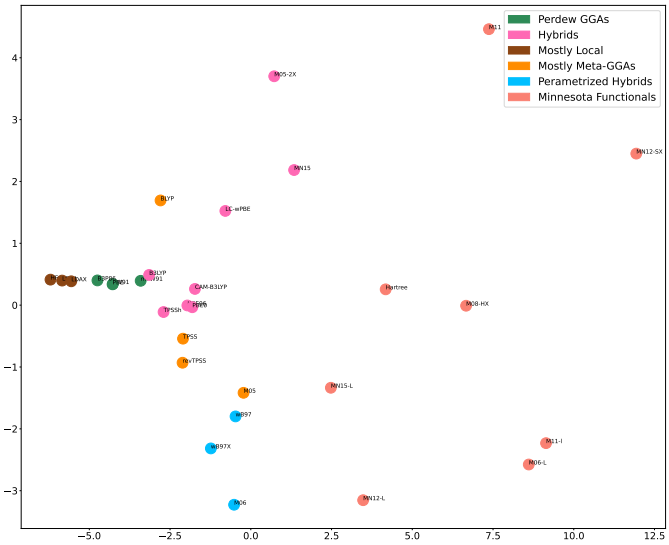


Figure B4: Dimensionality reduction of the MGAE109 dataset functional fingerprints with t-SNE



(a)



(b)

Figure B5: The 2D representations of the DDF matrix for the MGAE109 dataset. Low-dimensional projections are generated by (a) multidimensional scaling (MDS) and (b) isometric feature mapping (Isomap) manifold-learning methods, respectively.

Part V

Conclusions

Chapter 7

Summary and Future Work

In the past few years, the world has undergone significant changes in almost every field with the abundance of computational resources and the power of big data. It has experienced an entirely new dimension of data-oriented thinking and problem-solving that has helped with several exciting discoveries. One such avenue is physics-informed machine learning, which provides a compromise to combine human knowledge and the power of data, one complementing the other. This way, we are not entirely data-dependent; we can design and understand most of the machine learning components and solve a problem that would otherwise take decades.

There are several ways machine learning can be helpful in DFT. Chapter 2 briefly reviewed several machine learning DFT approaches that laid the foundations for most recent developments. Machine learning can help construct kinetic energy functional for orbital-free DFT that matches the accuracy of standard Kohn-Sham DFT calculations. Chapter 3 discussed a simple proof of concept for learning the kinetic energy functional for the Hubbard dimer and simple 1D real potentials using the Levy-Lieb constrained search approach [90, 97]. A careful extension of this concept to real molecules can help make generalizable orbital-free machine-learned density functionals.

Using machine learning to approximate the exchange-correlation functional and improve the ac-

curacy of Kohn-Sham DFT is another frequently explored research direction. In Chapter 4, I presented a spin-adapted modification of the fully-differentiable Kohn-Sham regularizer [93] in 1D with neural network nonlocal exchange-correlation approximations. As the machine-learned approximation was trained during a self-consistent solution of the Kohn-Sham equations, the neural network could learn the relation between the energy and the density at every step. Due to this automatic data augmentation and the regularizing effects, the training set of five atoms and ions suffice to accurately predict energies and densities for several weakly correlated molecules.

Other than the extension of the nonlocal approximation and the Kohn-Sham regularizer for real systems, future work can be directed towards incorporating physical intuition in the loss function and the training set. The original Kohn-Sham regularizer was tested for strongly-correlated molecules, but its generalizability was limited. Although it is still debatable, based on the observations from DM21 [85] which we briefly discussed in Chapter 1, one can incorporate fractional charge and fractional spin systems in a training set to improve generalizability and predictability for strongly-correlated molecules. On the other hand, the loss function can account for density-driven and functional-driven errors separately, leading to a more accurate exchange-correlation approximation.

Chapters 5 and 6 discuss the categorization of several established exchange-correlation functionals with unsupervised learning. In Chapter 5, I have detailed the concepts and methods frequently used in unsupervised learning tasks. The methods described here for dimensionality reduction, clustering, and clustering quality evaluation, are later used in Chapter 6 for 33 exchange-correlation approximations to understand their similarities and differences without human-induced bias. The feature space is uniquely defined to account for functional-driven and density-driven differences among functionals. The aim of this chapter was not to comment on which functional is best but to group them based on a simple connectivity graph constructed from a distance matrix made from functional fingerprints. Chapter 6 covers results for the MGAE109 dataset [187, 122]. Work is currently in progress for several other datasets and improving the clustering quality.

Machine learning DFT has come far in the last decade. Newly proposed, physics-informed machine-learned functionals, such as DM21, have provided insights into the drawbacks of DFT and suggestions for how one can cure them. While the applicability of machine-learned DFT functionals has not reached their full potential yet, we can draw motivation from machine-learned interatomic potentials for molecular dynamics calculations. Machine learned force fields have come a long way in terms of improving the accuracy of classical force fields and may soon dominate all molecular dynamics calculations. In DFT, I believe combining human intuition with the power of data can lead us to universally applicable functional approximations that yield usefully accurate results without incurring additional costs. With the recent developments, we have made some progress. Figuring out the rest of the missing pieces will help us realize a shorter path towards the exact functional.

Bibliography

[1] `https://github.com/pedersor/DFT_1d`, 2021.

[2] C. Adamo and V. Barone. Exchange functionals with improved long-range behavior and adiabatic connection methods without adjustable parameters: The mPW and mPW1PW models. *The Journal of Chemical Physics*, 108(2):664–675, jan 1998.

[3] C. Adamo and V. Barone. Toward reliable density functional methods without adjustable parameters: The PBE0 model. *The Journal of Chemical Physics*, 110(13):6158–6170, apr 1999.

[4] T. E. Baker, E. M. Stoudenmire, L. O. Wagner, K. Burke, and S. R. White. One-dimensional mimicking of electronic structure: The case for exponentials. *Phys. Rev. B*, 91:235141, Jun 2015.

[5] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.*, 18(1):5595–5637, Jan. 2017.

[6] A. D. Becke. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A*, 38:3098–3100, Sep 1988.

[7] A. D. Becke. Density-functional thermochemistry. iii. the role of exact exchange. *The Journal of Chemical Physics*, 98(7):5648–5652, 1993.

[8] R. E. Bellman. *Dynamic Programming*. Dover, 2015.

[9] M. Bogojeski, L. Vogt-Maranto, M. E. Tuckerman, K.-R. Müller, and K. Burke. Quantum chemical accuracy from density functional approximations via machine learning. *Nature Communications*, 11(1):5223, 2020.

[10] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[11] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller. Bypassing the Kohn-Sham equations with machine learning. *Nature Communications*, 8(1):872, 2017.

[12] R. Brydges, A. Dubrowski, and G. Regehr. A New Concept of Unsupervised Learning: Directed Self-Guided Learning in the Health Professions. *Academic Medicine*, 85(10), 2010.

[13] K. Burke. Perspective on density functional theory. *The Journal of Chemical Physics*, 136(15):150901, 2012.

[14] K. Burke. Perspective on density functional theory. *J. Chem. Phys.*, 136:150901, 2012.

[15] K. Burke and L. O. Wagner. Dft in a nutshell. *International Journal of Quantum Chemistry*, 113(2):96–101, 2013.

[16] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[17] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[18] D. J. Carrascal, J. Ferrer, J. C. Smith, and K. Burke. The hubbard dimer: a density functional case study of a many-body problem. *Journal of Physics: Condensed Matter*, 27(39):393001, 2015.

[19] M. Ceriotti, G. A. Tribello, and M. Parrinello. Simplifying the representation of complex free-energy landscapes using sketch-map. *Proceedings of the National Academy of Sciences*, 108(32):13023–13028, 2011.

[20] J.-D. Chai and M. Head-Gordon. Systematic optimization of long-range corrected hybrid density functionals. *The Journal of Chemical Physics*, 128(8):084106, feb 2008.

[21] G. Chandrasekaran, T. N. Nguyen, and J. Hemanth D. Multimodal sentimental analysis for social media applications: A comprehensive review. *WIREs Data Mining and Knowledge Discovery*, 11(5):e1415, 2021.

[22] Y. Chen, L. Zhang, H. Wang, and W. E. DeePKS: A Comprehensive Data-Driven Approach toward Chemically Accurate Density Functional Theory. *Journal of Chemical Theory and Computation*, 17(1):170–181, jan 2021.

[23] A. J. Cohen, P. Mori-Sánchez, and W. Yang. Insights into current limitations of density functional theory. *Science*, 321(5890):792–794, 2008.

[24] A. J. Cohen, P. Mori-Sánchez, and W. Yang. Challenges for density functional theory. *Chemical Reviews*, 112(1):289–320, 2012.

[25] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006. Special Issue: Diffusion Maps and Wavelets.

[26] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[27] F. G. Cruz, K.-C. Lam, and K. Burke. Exchange- correlation energy density from virial theorem. *The Journal of Physical Chemistry A*, 102(25):4911–4917, 1998.

[28] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople. Gaussian-2 theory for molecular energies of first- and second-row compounds. *The Journal of Chemical Physics*, 94(11):7221–7230, 1991.

[29] C. A. Custódio, É. R. Filletti, and V. V. Franca. Artificial neural networks for density-functional optimizations in fermionic systems. *Scientific Reports*, 9(1):1886, 2019.

[30] M. L. V. D. Visualizing datausing t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[31] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[33] S. Dick and M. Fernandez-Serra. Learning from the density to correct total energy and forces in first principle simulations. *The Journal of Chemical Physics*, 151(14):144102, 2019.

[34] S. Dick and M. Fernandez-Serra. Machine learning accurate exchange and correlation functionals of the electronic density. *Nature Communications*, 11(1):3509, 2020.

[35] P. A. M. Dirac. Note on exchange phenomena in the thomas atom. *Mathematical Proceedings of the Cambridge Philosophical Society*, 26(3):376–385, 1930.

[36] R. M. Dreizler and E. K. U. Gross. *Density Functional Theory: An Approach to the Quantum Many-Body Problem*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1990.

[37] C. Duan, S. Chen, M. G. Taylor, F. Liu, and H. J. Kulik. Machine learning to tame divergent density functional approximations: a new path to consensus materials design principles. *Chem. Sci.*, 12:13021–13036, 2021.

[38] S. Elfwing, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. Special issue on deep reinforcement learning.

[39] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[40] B. A. et al. Nersc-10 workload analysis (data from 2018). `https://portal.nersc.gov/project/m888/nersc10/workload/N10_Workload_Analysis.latest.pdf`, 2020.

[41] K. Finzel, P. W. Ayers, and P. Bultinck. A simple algorithm for the kohn-sham inversion problem applicable to general target densities. *THEORETICAL CHEMISTRY ACCOUNTS*, 137(3):6, 2018.

[42] M. Fishman, S. R. White, and E. M. Stoudenmire. The ITensor Software Library for Tensor Network Calculations, 2020.

[43] T. L. Fletcher, S. J. Davie, and P. L. A. Popelier. Prediction of intramolecular polarization of aromatic amino acids using kriging machine learning. *Journal of Chemical Theory and Computation*, 10(9):3708–3719, 2014. PMID: 26588516.

[44] S. L. France and J. D. Carroll. Two-way multidimensional scaling: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(5):644–661, 2011.

[45] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[46] M. Fritz, M. Fernández-Serra, and J. M. Soler. Optimization of an exchange-correlation density functional for water. *The Journal of Chemical Physics*, 144(22):224101, 2016.

[47] J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun. Accurate and Numerically Efficient r2SCAN Meta-Generalized Gradient Approximation. *The Journal of Physical Chemistry Letters*, 11(19):8208–8215, oct 2020.

[48] J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun. Correction to "Accurate and Numerically Efficient r2SCAN Meta-Generalized Gradient Approximation". *The Journal of Physical Chemistry Letters*, 11(21):9248, nov 2020.

[49] J. Fürnkranz and T. Kliegr. A brief overview of rule learning. In N. Bassiliades, G. Gottlob, F. Sadri, A. Paschke, and D. Roman, editors, *Rule Technologies: Foundations, Tools, and Applications*, pages 54–69, Cham, 2015. Springer International Publishing.

[50] C. C. GAggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In J. Van den Bussche and V. Vianu, editors, *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[51] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[52] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé, and A. Laio. Unsupervised Learning Methods for Molecular Simulation Data. *Chemical Reviews*, 121(16):9722–9758, aug 2021.

[53] L. Goerigk, A. Hansen, C. Bauer, S. Ehrlich, A. Najibi, and S. Grimme. A look at the density functional theory zoo with the advanced gmtkn55 database for general main group thermochemistry, kinetics and noncovalent interactions. *Phys. Chem. Chem. Phys.*, 19:32184–32215, 2017.

[54] P. Golub and S. Manzhos. Kinetic energy densities based on the fourth order gradient expansion: performance in different classes of materials and improvement via machine learning. *Phys. Chem. Chem. Phys.*, 21:378–395, 2019.

[55] P. Gundecha and H. Liu. *Mining Social Media: A Brief Introduction*, chapter Chapter 1, pages 1–17. Informs, 2014.

[56] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[57] K. A. Hallberg. New trends in density matrix renormalization. *Advances in Physics*, 55(5-6):477–526, 2006.

[58] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[59] D. R. Hartree. The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part II. Some Results and Discussion. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(1):111–132, jan 1928.

[60] D. R. Hartree and W. Hartree. Self-consistent field, with exchange, for beryllium. *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, 150(869):9–33, may 1935.

[61] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, New York, NY, 2009.

[62] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[63] J. Heyd, G. E. Scuseria, and M. Ernzerhof. Hybrid functionals based on a screened Coulomb potential. *The Journal of Chemical Physics*, 118(18):8207–8215, may 2003.

[64] J. Heyd, G. E. Scuseria, and M. Ernzerhof. Erratum: "Hybrid functionals based on a screened Coulomb potential" [J. Chem. Phys. 118, 8207 (2003)]. *The Journal of Chemical Physics*, 124(21):219906, jun 2006.

[65] F. L. Hirshfeld. Bonded-atom fragments for describing molecular charge densities. *Theoretica chimica acta*, 44(2):129–138, Jun 1977.

[66] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.

[67] J. Hollingsworth, L. Li, T. E. Baker, and K. Burke. Can exact conditions improve machine-learned density functionals? *The Journal of Chemical Physics*, 148(24):241743, 2018.

[68] B. Hopkins and J. G. Skellam. A new method for determining the type of distribution of plant individuals. *Annals of Botany*, 18:213–227, 1954.

[69] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991.

[70] C. Huntingford, E. S. Jeffers, M. B. Bonsall, H. M. Christensen, T. Lees, and H. Yang. Machine learning and artificial intelligence to aid climate change research and preparedness. *Environmental Research Letters*, 14(12), 2019.

[71] M. Innes, A. Edelman, K. Fischer, C. Rackauckas, E. Saba, V. B. Shah, and W. Tebbutt. A differentiable programming system to bridge machine learning and scientific computing, 2019.

[72] H. Ji and Y. Jung. A local environment descriptor for machine-learned density functional theory at the generalized gradient approximation level. *The Journal of Chemical Physics*, 148(24):241742, 2018.

[73] R. Jinnouchi, J. Lahnsteiner, F. Karsai, G. Kresse, and M. Bokdam. Phase transitions of hybrid perovskites simulated by machine-learning force fields trained on the fly with bayesian inference. *Phys. Rev. Lett.*, 122:225701, Jun 2019.

[74] K. W. Johnson, J. T. Soto, B. S. Glicksberg, K. Shameer, R. Miotto, M. Ali, E. Ashley, and J. T. Dudley. Artificial intelligence in cardiology. *Journal of the American College of Cardiology*, 71(23):2668–2679, 2018.

[75] I. T. Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[76] B. Kalita, L. Li, R. J. McCarty, and K. Burke. Learning to Approximate Density Functionals. *Accounts of Chemical Research*, 54(4):818–826, feb 2021.

[77] V. Karasiev and S. Trickey. Issues and challenges in orbital-free density functional calculations. *Computer Physics Communications*, 183(12):2519 – 2527, 2012.

[78] M. F. Kasim and S. M. Vinko. Learning the exchange-correlation functional from nature with fully differentiable density functional theory. *Phys. Rev. Lett.*, 127:126403, Sep 2021.

[79] L. Kaufman and P. J. Rousseeuw. Divisive analysis (program DIANA). In *Finding Groups in Data*, pages 253–279. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2008.

[80] K. P. Kepp. Comment on “density functional theory is straying from the path toward the exact functional”. *Science*, 356(6337):496–496, 2017.

[81] T. Kikutsuji, K. Kim, and N. Matubayasi. How do hydrogen bonds break in supercooled water?: Detecting pathways not going through saddle point of two-dimensional potential of mean force. *The Journal of Chemical Physics*, 148(24):244501, 2018.

[82] M.-C. Kim, E. Sim, and K. Burke. Understanding and reducing errors in density functional calculations. *Phys. Rev. Lett.*, 111:073003, Aug 2013.

[83] M.-C. Kim, E. Sim, and K. Burke. Ions in solution: Density corrected density functional theory (dc-dft). *The Journal of Chemical Physics*, 140(18):18A528, 2014.

[84] Y. Kim, S. Song, E. Sim, and K. Burke. Halogen and chalcogen binding dominated by density-driven errors. *The Journal of Physical Chemistry Letters*, 10(2):295–301, December 2019.

[85] J. Kirkpatrick, B. McMorrow, D. H. P. Turban, A. L. Gaunt, J. S. Spencer, A. G. D. G. Matthews, A. Obika, L. Thiry, M. Fortunato, D. Pfau, L. R. Castellanos, S. Petersen, A. W. R. Nelson, P. Kohli, P. Mori-Sánchez, D. Hassabis, and A. J. Cohen. Pushing the frontiers of density functionals by solving the fractional electron problem. *Science*, 374(6573):1385–1389, 2021.

[86] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965.

[87] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[88] C. Lee, W. Yang, and R. G. Parr. Development of the colle-salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B*, 37:785–789, Jan 1988.

[89] X. Lei and A. J. Medford. Design and analysis of machine learning exchange-correlation functionals via rotationally invariant convolutional descriptors. *Phys. Rev. Materials*, 3:063801, Jun 2019.

[90] M. Levy. Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the v-representability problem. *Proceedings of the National Academy of Sciences*, 76(12):6062–6065, 1979.

[91] L. Li, T. E. Baker, S. R. White, and K. Burke. Pure density functional for strong correlation and the thermodynamic limit from machine learning. *Phys. Rev. B*, 94:245129, Dec 2016.

[92] L. Li, S. Hoyer, R. Pederson, R. Sun, E. D. Cubuk, P. Riley, and K. Burke. Kohn-sham equations as regularizer: Building prior knowledge into machine-learned physics, 2020.

[93] L. Li, S. Hoyer, R. Pederson, R. Sun, E. D. Cubuk, P. Riley, and K. Burke. Kohn-sham equations as regularizer: Building prior knowledge into machine-learned physics. *Phys. Rev. Lett.*, 126:036401, Jan 2021.

[94] L. Li, J. C. Snyder, I. M. Pelaschier, J. Huang, U.-N. Niranjan, P. Duncan, M. Rupp, K.-R. Müller, and K. Burke. Understanding machine-learned density functionals. *International Journal of Quantum Chemistry*, 116(11):819–833, 2016.

[95] L. Li, J. C. Snyder, I. M. Pelaschier, J. Huang, U.-N. Niranjan, P. Duncan, M. Rupp, K.-R. Müller, and K. Burke. Understanding machine-learned density functionals. *International Journal of Quantum Chemistry*, 116(11):819–833, 2016.

[96] Y. Li. Research and application of deep learning in image recognition. In *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, pages 994–999, 2022.

[97] E. H. Lieb. Density functionals for coulomb systems. *Int. J. Quantum Chem.*, 24(3):243–277, 1983.

[98] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[99] Q. Liu, J. Wang, P. Du, L. Hu, X. Zheng, and G. Chen. Improving the Performance of Long-Range-Corrected Exchange-Correlation Functional with an Embedded Neural Network. *The Journal of Physical Chemistry A*, 121(38):7273–7281, sep 2017.

[100] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[101] S. Manzhos. Machine learning for the solution of the Schrödinger equation. *Machine Learning: Science and Technology*, 1(1):013002, 2020.

[102] N. Mardirossian and M. Head-Gordon. Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 200 density functionals. *Molecular Physics*, 115(19):2315–2372, 2017.

[103] T. E. Markland and M. Ceriotti. Nuclear quantum effects enter the mainstream. *Nature Reviews Chemistry*, 2(3):109, 2018.

[104] R. T. McGibbon and V. S. Pande. Learning kinetic distance metrics for markov state models of protein conformational dynamics. *Journal of Chemical Theory and Computation*, 9(7):2900–2906, 2013. PMID: 26583974.

[105] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[106] M. G. Medvedev, I. S. Bushmarinov, J. Sun, J. P. Perdew, and K. A. Lyssenko. Density functional theory is straying from the path toward the exact functional. *Science*, 355(6320):49–52, 2017.

[107] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen. Deep learning for cellular image analysis. *Nature Methods*, 16(12):1233–1246, 2019.

[108] K. P. Murphy. *Machine learning: A probabilistic perspective*. MIT Press, 2013.

[109] R. Nagai, R. Akashi, and O. Sugino. Completing density functional theory by machine learning hidden messages from molecules. *npj Computational Materials*, 6(1):43, 2020.

[110] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

[111] J. Nelson, R. Tiwari, and S. Sanvito. Machine learning density functional theory for the hubbard model. *Phys. Rev. B*, 99:075132, Feb 2019.

[112] T. T. Nguyen, E. Székely, G. Imbalzano, J. Behler, G. Csányi, M. Ceriotti, A. W. Götz, and F. Paesani. Comparison of permutationally invariant polynomials, neural networks, and gaussian approximation potentials in representing water interactions through many-body expansions. *The Journal of Chemical Physics*, 148(24):241725, 2018.

[113] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[114] J. P. Perdew. Density-functional approximation for the correlation energy of the inhomogeneous electron gas. *Physical Review B*, 33(12):8822–8824, jun 1986.

[115] J. P. Perdew, K. Burke, and M. Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996.

[116] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais. Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Phys. Rev. B*, 46:6671–6687, Sep 1992.

[117] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais. Erratum: Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Phys. Rev. B*, 48:4978–4978, Aug 1993.

[118] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, L. A. Constantin, and J. Sun. Workhorse semilocal density functional for condensed matter physics and quantum chemistry. *Phys. Rev. Lett.*, 103:026403, Jul 2009.

[119] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, L. A. Constantin, and J. Sun. Erratum: Workhorse semilocal density functional for condensed matter physics and quantum chemistry [phys. rev. lett. 103, 026403 (2009)]. *Phys. Rev. Lett.*, 106:179902, Apr 2011.

[120] J. P. Perdew, A. Ruzsinszky, J. Tao, V. N. Staroverov, G. E. Scuseria, and G. I. Csonka. Prescription for the design and selection of density functional approximations: More constraint satisfaction with fewer fits. *The Journal of Chemical Physics*, 123(6):062201, 2005.

[121] J. P. Perdew and K. Schmidt. Jacob's ladder of density functional approximations for the exchange-correlation energy. *AIP Conference Proceedings*, 577(1):1–20, 2001.

[122] R. Peverati and D. G. Truhlar. Communication: A global hybrid generalized gradient approximation to the exchange-correlation functional that satisfies the second-order density-gradient constraint and has broad applicability in chemistry. *The Journal of Chemical Physics*, 135(19):191102, 2011.

[123] R. Peverati and D. G. Truhlar. Improving the Accuracy of Hybrid Meta-GGA Density Functionals by Range Separation. *The Journal of Physical Chemistry Letters*, 2(21):2810–2817, nov 2011.

[124] R. Peverati and D. G. Truhlar. An improved and broadly accurate local approximation to the exchange–correlation density functional: The MN12-L functional for electronic structure calculations in chemistry and physics. *Physical Chemistry Chemical Physics*, 14(38):13171, 2012.

[125] R. Peverati and D. G. Truhlar. M11-L: A Local Density Functional That Provides Improved Accuracy for Electronic Structure Calculations in Chemistry and Physics. *The Journal of Physical Chemistry Letters*, 3(1):117–124, jan 2012.

[126] R. Peverati and D. G. Truhlar. Screened-exchange density functionals with broad accuracy for chemistry and solid-state physics. *Physical Chemistry Chemical Physics*, 14(47):16187, 2012.

[127] J. Polanski. Unsupervised learning in drug design from self-organization to deep chemistry. *International Journal of Molecular Sciences*, 23(5), 2022.

[128] S. N. Pozdnyakov, M. J. Willatt, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti. Incompleteness of atomic structure representations. *Physical Review Letters*, 125(16), Oct 2020.

[129] Z. D. Pozun, K. Hansen, D. Sheppard, M. Rupp, K.-R. Müller, and G. Henkelman. Optimizing transition states via kernel-based machine learning. *The Journal of Chemical Physics*, 136(17):174101, 2012.

[130] K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon. A fifth-order perturbation comparison of electron correlation theories. *Chemical Physics Letters*, 157(6):479 – 483, 1989.

[131] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions, 2018.

[132] D. Rappoport, N. R. M. Crawford, F. Furche, and K. Burke. *Approximate Density Functionals: Which should I choose?* Wiley, Chichester. Hoboken: Wiley, John & Sons, Inc., 2009.

[133] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[134] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[135] M. Rupp, O. A. von Lilienfeld, and K. Burke. Guest editorial: Special topic on data-enabled theoretical chemistry. *The Journal of Chemical Physics*, 148(24):241401, 2018.

[136] A. Ryabov, I. Akhatov, and P. Zhilyaev. Neural network interpolation of exchange-correlation functional. *Scientific Reports*, 10(1):8000, 2020.

[137] K. Ryczko, D. A. Strubbe, and I. Tamblyn. Deep learning and density-functional theory. *Phys. Rev. A*, 100:022512, Aug 2019.

[138] M. S. Sarfraz, M. Koulakis, C. Seibold, and R. Stiefelhagen. Hierarchical nearest neighbor graph embedding for efficient dimensionality reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[139] M. S. Sarfraz, V. Sharma, and R. Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2019.

[140] J. Schmidt, C. L. Benavides-Riveros, and M. A. L. Marques. Machine Learning the Physical Nonlocal Exchange–Correlation Functional of Density-Functional Theory. *The Journal of Physical Chemistry Letters*, 10(20):6425–6431, oct 2019.

[141] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[142] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nature Communications*, 10(1):5024, 2019.

[143] P. Schwaller, B. Hoover, J.-L. Reymond, H. Strobelt, and T. Laino. Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Science Advances*, 7(15):eabe4166, 2021.

[144] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 07 1998.

[145] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. Schnet – a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

[146] Sebastian and M. Fernandez-Serra. Highly accurate and constrained density functional obtained with differentiable programming. *Phys. Rev. B*, 104:L161109, Oct 2021.

[147] J. Seino, R. Kageyama, M. Fujinami, Y. Ikabata, and H. Nakai. Semi-local machine-learned kinetic energy density functional with third-order gradients of electron density. *The Journal of Chemical Physics*, 148(24):241705, 2018.

[148] K. Shah, A. Salunke, S. Dongare, and K. Antala. Recommender systems: An overview of different approaches to recommendations. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–4, 2017.

[149] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[150] E. Sim, S. Song, and K. Burke. Quantifying density errors in dft. *The Journal of Physical Chemistry Letters*, 9(22):6385–6392, 2018.

[151] E. Sim, S. Song, S. Vuckovic, and K. Burke. Improving Results by Improving Densities: Density-Corrected Density Functional Theory. *Journal of the American Chemical Society*, 144(15):6625–6639, apr 2022.

[152] J. C. Snyder, M. Rupp, K. Hansen, L. Blooston, K.-R. Müller, and K. Burke. Orbital-free bond breaking via machine learning. *The Journal of Chemical Physics*, 139(22):224104, 2013.

[153] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke. Finding density functionals with machine learning. *Phys. Rev. Lett.*, 108:253002, Jun 2012.

[154] J. C. Snyder, M. Rupp, K.-R. Müller, and K. Burke. Nonlinear gradient denoising: Finding accurate extrema from inaccurate functional derivatives. *International Journal of Quantum Chemistry*, 115(16):1102–1114, 2015.

[155] S. Song, S. Vuckovic, E. Sim, and K. Burke. Density Sensitivity of Empirical Functionals. *The Journal of Physical Chemistry Letters*, 12(2):800–807, jan 2021.

[156] B. K. Spears, J. Brase, P.-T. Bremer, B. Chen, J. Field, J. Gaffney, M. Kruse, S. Langer, K. Lewis, R. Nora, J. L. Peterson, J. Jayaraman Thiagarajan, B. Van Essen, and K. Humbird. Deep learning: A guide for practitioners in the physical sciences. *Physics of Plasmas*, 25(8):080901, 2018.

[157] V. N. Staroverov, G. E. Scuseria, J. Tao, and J. P. Perdew. Comparative assessment of a new nonempirical density functional: Molecules and hydrogen-bonded complexes. *The Journal of Chemical Physics*, 119(23):12129–12137, 2003.

[158] D. Steinley. Properties of the Hubert-Arable Adjusted Rand Index., 2004.

[159] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *The Journal of Physical Chemistry*, 98(45):11623–11627, nov 1994.

[160] E. M. Stoudenmire, L. O. Wagner, S. R. White, and K. Burke. One-dimensional continuum electronic structure with the density-matrix renormalization group and its implications for density-functional theory. *Phys. Rev. Lett.*, 109:056402, Aug 2012.

[161] J. Sun, A. Ruzsinszky, and J. P. Perdew. Strongly constrained and appropriately normed semilocal density functional. *Phys. Rev. Lett.*, 115:036402, Jul 2015.

[162] J. Sun, A. Ruzsinszky, and J. P. Perdew. Strongly constrained and appropriately normed semilocal density functional. *Phys. Rev. Lett.*, 115:036402, Jul 2015.

[163] Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, J. J. Eriksen, Y. Gao, S. Guo, J. Hermann, M. R. Hermes, K. Koh, P. Koval, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J. D. McClain, M. Motta, B. Mussard, H. Q.

Pham, A. Pulkin, W. Purwanto, P. J. Robinson, E. Ronca, E. R. Sayfutyarova, M. Scheurer, H. F. Schurkus, J. E. T. Smith, C. Sun, S.-N. Sun, S. Upadhyay, L. K. Wagner, X. Wang, A. White, J. D. Whitfield, M. J. Williamson, S. Wouters, J. Yang, J. M. Yu, T. Zhu, T. C. Berkelbach, S. Sharma, A. Y. Sokolov, and G. K.-L. Chan. Recent developments in the pyscf program package. *The Journal of Chemical Physics*, 153(2):024109, 2020.

[164] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria. Climbing the Density Functional Ladder: Nonempirical Meta–Generalized Gradient Approximation Designed for Molecules and Solids. *Physical Review Letters*, 91(14):146401, sep 2003.

[165] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[166] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.

[167] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.

[168] D. J. Tozer, V. E. Ingamells, and N. C. Handy. Exchange-correlation potentials. *The Journal of Chemical Physics*, 105(20):9200–9213, 1996.

[169] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller. Machine learning force fields, 2020.

[170] M. Usama, J. Qadir, A. Raza, H. Arif, K.-l. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha. Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE Access*, 7:65579–65615, 2019.

[171] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1073–1080, New York, NY, USA, 2009. Association for Computing Machinery.

[172] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[173] K. Vu, J. C. Snyder, L. Li, M. Rupp, B. F. Chen, T. Khelif, K.-R. Müller, and K. Burke. Understanding kernel ridge regression: Common behaviors from simple functions to density functionals. *International Journal of Quantum Chemistry*, 115(16):1115–1128, 2015.

[174] S. Vuckovic, S. Song, J. Kozlowski, E. Sim, and K. Burke. Density functional analysis: The theory of density-corrected dft. *Journal of Chemical Theory and Computation*, 15(12):6636–6646, November 2019. PMID: 31682433.

[175] O. A. Vydrov and G. E. Scuseria. Assessment of a long-range corrected hybrid functional. *The Journal of Chemical Physics*, 125(23):234109, dec 2006.

[176] L. O. Wagner, E. Stoudenmire, K. Burke, and S. R. White. Reference electronic structure calculations in one dimension. *Physical Chemistry Chemical Physics*, 14(24):8581–8590, 2012.

[177] J. Wellendorff, K. T. Lundgaard, A. Møgelhøj, V. Petzold, D. D. Landis, J. K. Nørskov, T. Bligaard, and K. W. Jacobsen. Density functionals for surface science: Exchange-correlation model development with bayesian error estimation. *Phys. Rev. B*, 85:235149, Jun 2012.

[178] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992.

[179] D.-O. Won, K.-R. Müller, and S.-W. Lee. An adaptive deep reinforcement learning framework enables curling robots with human-like performance in real-world conditions. *Science Robotics*, 5(46), 2020.

[180] S. Wouters and D. Van Neck. The density matrix renormalization group for ab initio quantum chemistry. *The European Physical Journal D*, 68(9):272, 2014.

[181] T. Yanai, D. P. Tew, and N. C. Handy. A new hybrid exchange–correlation functional using the Coulomb-attenuating method (CAM-B3LYP). *Chemical Physics Letters*, 393(1-3):51–57, jul 2004.

[182] Z. Yang, R. Algesheimer, and C. J. Tessone. A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports*, 6(1):30750, 2016.

[183] K. Yao and J. Parkhill. Kinetic energy of hydrocarbons as a function of electron density and convolutional neural networks. *Journal of Chemical Theory and Computation*, 12(3):1139–1147, 2016. PMID: 26812530.

[184] H. S. Yu, X. He, S. L. Li, and D. G. Truhlar. MN15: A Kohn–Sham global-hybrid exchange–correlation density functional with broad accuracy for multi-reference and single-reference systems and noncovalent interactions. *Chemical Science*, 7(8):5032–5051, 2016.

[185] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, page 103–114, New York, NY, USA, 1996. Association for Computing Machinery.

[186] Y. Zhao, N. E. Schultz, and D. G. Truhlar. Exchange-correlation functional with broad accuracy for metallic and nonmetallic compounds, kinetics, and noncovalent interactions. *The Journal of Chemical Physics*, 123(16):161103, oct 2005.

[187] Y. Zhao, N. E. Schultz, and D. G. Truhlar. Design of Density Functionals by Combining the Method of Constraint Satisfaction with Parametrization for Thermochemistry, Thermochemical Kinetics, and Noncovalent Interactions. *Journal of Chemical Theory and Computation*, 2(2):364–382, mar 2006.

[188] Y. Zhao, N. E. Schultz, and D. G. Truhlar. Design of Density Functionals by Combining the Method of Constraint Satisfaction with Parametrization for Thermochemistry, Thermochemical Kinetics, and Noncovalent Interactions. *Journal of Chemical Theory and Computation*, 2(2):364–382, mar 2006.

[189] Y. Zhao and D. G. Truhlar. A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions. *The Journal of Chemical Physics*, 125(19):194101, nov 2006.

[190] Y. Zhao and D. G. Truhlar. A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions. *The Journal of Chemical Physics*, 125(19):194101, 2006.

[191] Y. Zhao and D. G. Truhlar. Exploring the Limit of Accuracy of the Global Hybrid Meta Density Functional for Main-Group Thermochemistry, Kinetics, and Noncovalent Interactions. *Journal of Chemical Theory and Computation*, 4(11):1849–1868, nov 2008.

[192] Y. Zhao and D. G. Truhlar. The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other function. *Theoretical Chemistry Accounts*, 120(1-3):215–241, may 2008.