# ABSTRACT


Title of Dissertation: SPEEDING UP DENSITY FUNCTIONAL THEORY CALCULATIONS WITH MACHINE LEARNING: A DENSITY LEARNING APPROACH

Phillip E. Pope
Doctor of Philosophy, 2025

Dissertation Directed by: Professor David W. Jacobs
Department of Computer Science

Professor Hong-Zhou Ye
Department of Chemistry



The electronic structure of molecules and materials determines chemical reactivity. If we could only compute it accurately and efficiently, we could accelerate molecular research and help solve some of society's biggest problems. One prominent approach to electronic structure is *Density Functional Theory* (DFT), at the heart of which are the *Kohn-Sham* (KS) equations. These equations are a nonlinear eigenvalue problem of the form $H[\rho]\Psi = E\Psi$, where $H$ is a real symmetric matrix called the *Hamiltonian*, $\Psi$ is an eigenvector called the *wave function*, $E$ is an eigenvalue called the *energy*, and $\rho(\mathbf{r}) = \sum_i |\Psi_i(\mathbf{r})|^2$ is a real-valued field called the *charge density*, which is unknown a priori. In this thesis, we investigate the use of machine-learning models for reducing the amount of computation in solving the KS equations. Our strategy is to develop models to predict the *charge density* using equivariant graph-neural-networks. We show on materials and molecules that our method may obtain highly-accurate results leading to computational savings, sometimes obtaining *chemical accuracy*, commonly defined to be 1 kcal/mol, using a single step of KS-DFT. Our results show that density learning is a reliable means of speeding up DFT computations.

# SPEEDING UP DENSITY FUNCTIONAL THEORY CALCULATIONS WITH MACHINE LEARNING: A DENSITY LEARNING APPROACH

by

Phillip E. Pope

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:
Professor David W. Jacobs, Chair/Advisor
Professor Hong-Zhou Ye, Co-Chair/Advisor
Professor Maria K. Cameron, Dean's Representative
Professor Thomas Goldstein
Professor Soheil Feizi

# Acknowledgements

First I would like to thank my advisor David for allowing me to work on this thesis topic. It was very exciting to work in this area and I grew a lot growth from it. It would not have been possible without his support. I am grateful to him for letting me take this risk and encouraging me to work on impactful problems.

I would like to thank my co-advisor Hong-Zhou for mentoring me during the last year of my PhD. I benefited a lot from working with a real quantum chemist.

Thanks to my friend Sam for kindling my interest in scientific computing. This unexpectedly turned out to have a large influence on my research direction.

I am grateful to Josh and the Maryland grappling community for all the training together. Sunday open mats were a constant fixture and a great source of vitality.

Thanks to Zach for our friendship. Hiking, kayaking, and camping together was much needed during this PhD, especially the pandemic years.

I would like to express my gratitude to my partner Judith for supporting me during this thesis. Our travels together are a source of joy and inspiration.

Thanks to my parents, sisters, and family for their love and support, although they don't understand what I do.

This thesis is dedicated to my undergraduate advisor Patrick McDonald, who taught me mathematics and grappling, and inspired me to do research.

# Table of contents

# Chapter 1

# Introduction

## 1.1 Motivation

Electrons are the glue that hold everything together, and if we could only calculate their behavior accurately and efficiently, we could design new molecules and materials for societally important applications like for energy, industry, agriculture, and medicine. Since the discovery of quantum mechanics over a century ago, methods for the computation of electronic structure have been developed. These methods can be highly accurate for a range of atomic systems, however their reach is limited due to their large computational cost. Modeling of realistic systems may require hundreds or thousands of atoms, and is often out of reach for many methods.

*Density Functional Theory* (DFT) is an prominent approach to electronic structure which reformulates the $N$-particle quantum many-body problem in terms of an effective one-particle density [LL19; Esc96]. At the heart of this theory are the Kohn-Sham Equations, a nonlinear eigenvalue problem of the form $H[\rho]\Psi = E\Psi$, where $H$ is a real symmetric matrix called the *Hamiltonian*, the eigenvector $\Psi$ the *wavefunction*, the eigenvalue $E$ the *energy*, and apriori-unknown scalar field $\rho(\mathbf{r}) = \sum_i |\Psi_i(\mathbf{r})|^2$ the *charge density*. The Kohn-Sham equations are nonlinear in the sense that the matrix $H$ depends on the charge density $\rho$, which in turn

depends on the eigenvectors $\Psi$ of $H$. These equations are solved by fixed-point iteration, or *Self-consistent field* (SCF) iteration, where an initial guess for $\rho$ is made and then a sequence of *linear* eigenvalue problems are solved until convergence.

Reducing the cost of DFT calculations is of significant practical interest. Machine Learning stands to contribute to this problem through the develop of fast and accurate surrogate models. In this thesis we design a novel predictive system to accelerate DFT calculations based on learning from prior data. By predicting a core quantity which arises in such calculations, the *charge density*, we show that the amount of computation may be reduced while still satisfying accuracy requirements.

## 1.2 Contributions

Our contributions are as follows:

- **Designing Chemically Relevant Evaluations**: we design our model evaluations to focus on metrics and contexts relevant to practitioners in quantum chemistry. In the first result Chapter, we measure iterations saved over a standard baselines. In the remaining Chapters, we target the conventional standard of *chemical accuracy*, commonly defined to be 1 kcal/mol. This is commonly interpreted as the level of accuracy comparable to that of which may be obtained from chemical experiment. Achieving this level of *numerical accuracy* in the context of a quantum-chemical simulation, means that the simulations can make *testable* predictions. We report metrics in chemically relevant units and quantities like the total and HOMO/LUMO energies. We believe the right way to evaluate models for quantum chemistry is to evaluate on molecules/structures that are in some sense more complicated than those in the training set, for some suitable measure of complexity. These evaluations are more informative for real-world use of such models, as practitioners will always want to use the model on a larger sets of molecules/structures than those in the training set. We

do this in two ways. First, in the case of molecules, we train on structures with fewer atoms and test on ones with more. Second, in the case of materials, we train on structures with fewer unique elements, and test on structures with more. In doing so we try to focus on evaluations on scenarios which are likely to arise in practice.

- **Achieving near chemical accuracy with density learning**: Using modern *geometric graph neural networks*, we design a *query node* approach to learn the charge density pointwise across the computational grids on which they lie. By modifying the geometric graphs which represent the molecule with this node, the model gains the capacity for predicting a scalar grid-based quantity. Using this approach, we demonstrate highly accurate results may be achieved. We do so in terms of accuracy of the density and the total energy, where the later is obtained from the learned densities after one KS-DFT step. Using structures from a common molecular benchmark dataset, we achieve on average near chemical accuracy on a test set of molecules larger than in the training set.

- **Predicting on Excited States**: We use the flexibility of our query node approach to extend density predictions to *excited states*, involving non-zero spin and charge states, with a *multi-query node* approach. By encoding the molecular state with unique *type* of query node, implemented by an unused atomic index, the model can learn to distinguish excited states. To showcase this capability, we estimate the *Singlet-Triplet* gap, the energy difference between the ground state and a particular spin state of a moleucule. We show an initial degree of accuracy may be achieved via this method, and discuss improvements.

Our results show that density prediction is a reliable means of obtaining accurate energy estimates while reducing the number of KS-DFT iterations to a single step. Additional improvements are still necessary to raise the accuracy of the estimation of more difficult quanities like the HOMO/LUMO and Singlet/Triplet gaps. The results also show that a de-

gree of generalization to more complex molecules/materials is obtainable by density learning, while simultaneously meeting high accuracy requirements. Together these achievements represent a milestone towards machine learning models which are practically usable by quantum chemists.

The structure of the remaining part of this thesis is as follows. In the next Chapter, we discuss relevant background on DFT, equivariant machine learning, and density learning. In the following three Chapters, we present our results: first on materials, second on molecules, and third on excited states. In the last Chapter, we discuss our results and make concluding results. In the Appendix we also provide supporting details.

# Chapter 2

# Background

## 2.1 Electronic Structure Theory

In this subsection we describe important background on electronic structure theory. We introduce the mathematical and physical ideas necessary for understanding the context in which our contributions in machine-learning occur. Our goal is to make this background more accessible to the machine-learning community.

We begin by introducing the Schrödinger equation, the core equation of non-relativistic quantum mechanics. We focus on the spin-independent and fixed particle number case.

We then introduce the Quantum Many-Body problem, the problem of solving the Schrödinger equation for $N$-particles and many-body interaction, and discuss reasons for its intractability. Lastly we give a concise introduction to Density Functional Theory, the most popular approach for addressing the Quantum Many-Body problem, and show how the Kohn-Sham equations arise from the derivative of a certain energy functional.

### 2.1.1 Notation

We denote vectors in real-space $\mathbb{R}^3$ in bold-faced font. We use lowercase $\mathbf{r}$ to denote the position vector of electrons, and uppercase to denote the position vector of nuclei $\mathbf{R}$. Indices

on these vectors follow the same lower/upper-case convention, e.g. $\{\mathbf{r}_i\}_{i=1}^N$ and $\{\mathbf{R}_I\}_{I=1}^M$.

We denote vectors in Hilbert-space with Greek letters. In particular we denote wavefunctions in $L^2(\mathbb{R}^3)$ as $\Psi$ or $\varphi$. We denote the Hermitean adjoint of $\Psi$ by $\Psi^*$. For Hilbert-space vectors we use Dirac notation [Dir39], where the vectors $\langle\Psi|$ and $|\varphi\rangle$ are called *bra* and *ket* respectively, and the inner product $\langle\varphi|\Psi\rangle := \int \varphi^*(x)\Psi(x)dx$ is called a *braket*. A linear operator $A$ (also called an *observable*) acting on a ket $|\Psi\rangle$ obtains another ket denoted $A|\Psi\rangle$. The quantity $\langle\Psi|A|\Psi\rangle := \int \Psi^*(x)A(x)\Psi(x)dx$ is called the *expectation* of the linear operator $A$.

The three-dimensional Laplace operator with respect to a coordinate $\mathbf{r}$ is denoted as $\Delta_{\mathbf{r}} := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$. In some cases where the intended coordinates are unambiguous, we may simply write $\Delta$. In the multi-particle case we may also write $\Delta = \sum_i \Delta_{\mathbf{r}_i}$. Unless otherwise stated, we will use atomic units ($m = e = 1/4\pi\epsilon_0 = \hbar = 1$).

### 2.1.2 The Schrödinger Equation

The time-independent Schrödinger Equation is an eigenvalue problem of the form

$$H\Psi = E\Psi \tag{2.1}$$

where $H$ is a self-adjoint operator called the *Hamiltonian*, $\Psi$ is a complex-valued eigenfunction called the *wave function*, and $E$ is a real-valued eigenvalue called the *energy*. The Hamiltonian $H$ is of the form:

$$H := T + V \tag{2.2}$$

where the $T := -\frac{\Delta}{2}$ is the kinetic energy operator, and $V$ is system-specific potential term. For example, one important class of potentials for atomic systems is that of radial potentials

$v(\mathbf{r}) = \frac{1}{r}$, where $r = \|\mathbf{r}\|$.

The Pauli Exclusion Principle states that two electrons cannot occupy the same quantum state. Consequently any wavefunction $\Psi$ is anti-symmetric, i.e. that the interchange of any two-arguments induces a sign-change:

$$\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_i, \ldots, \mathbf{r}_j, \ldots, \mathbf{r}_N) = -\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_j, \ldots, \mathbf{r}_i, \ldots, \mathbf{r}_N) \tag{2.3}$$

This anti-symmetry property puts important limitations on the function class of admissible wavefunctions. For $N$-particle systems this space is defined as

$$\mathcal{A}_N := \{\Psi | \Psi \in \bigwedge^N L^2(\mathbb{R}^3), \langle \Psi | \Psi \rangle = 1\} \tag{2.4}$$

where the wedge product $\bigwedge$ captures the alternating property required for anti-symmetry.

In the next section, we discuss the many-body case of the Schrödinger Equation, which involves an arbitrary number of *interacting* nuclei and electrons, each of which emit a radial potential. This case has special relevance in chemistry applications.

### 2.1.3 The Quantum Many-Body Problem

The Quantum Many-Body Problem is the problem of computing the interactions between an arbitrary number of particles according to the laws of quantum theory.

In chemical applications only two types of particles are relevant: nuclei and electrons. We depict a generic many-body system in Figure 2.1.

#### The Ground State

Often in chemical applications, the ground state is most important. This is because for systems such as non-metals, the difference between the ground state and the first excited

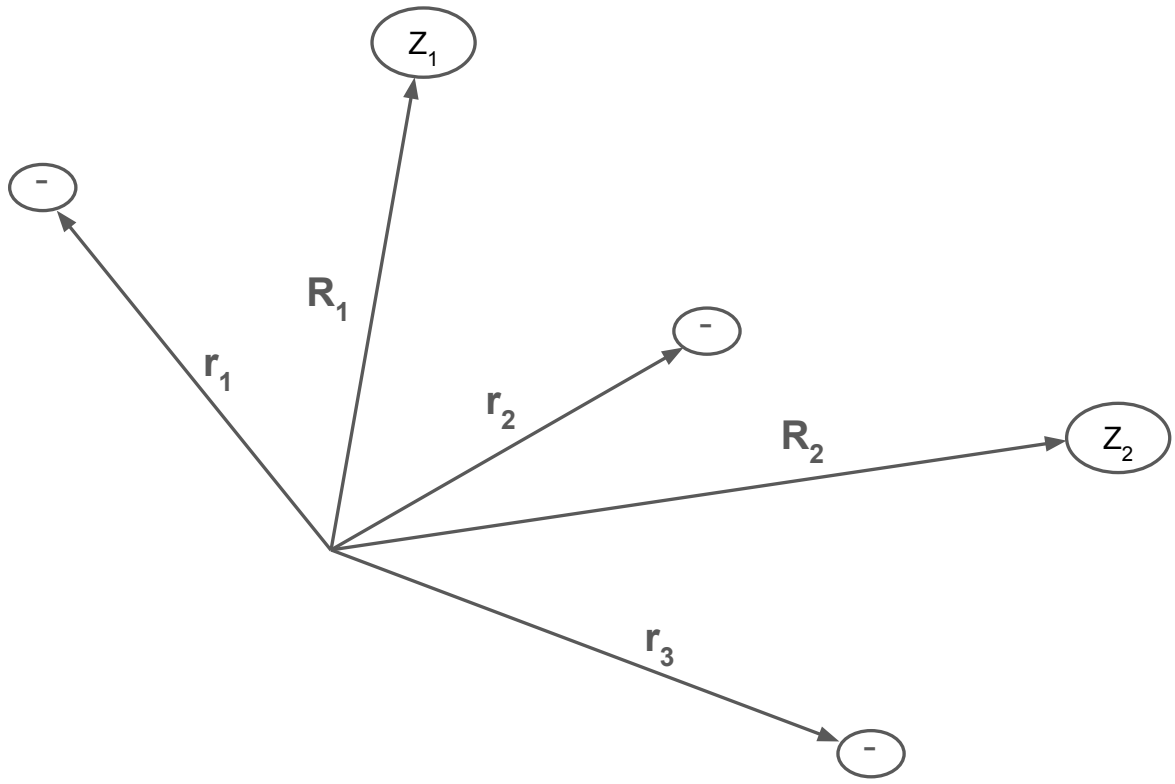


Figure 2.1: Illustration of a many-body problem with $N = 3$ electrons at positions $\{\mathbf{r}_i\}$ and $M = 2$ nuclei at positions $\{\mathbf{R}_j\}$ with atomic charges $Z_1$ and $Z_2$.

state $E_1 - E_0$ may be very large, on the order of electron volts (eV) [LL19].

Using the Courant-Fischer Min/Max Theorem [LL19], the eigenvalue problem 2.1 can be written as:

$$E = \inf_{\substack{\Psi \in \mathcal{A}_N \\ \langle \Psi | \Psi \rangle = 1}} \langle \Psi | H | \Psi \rangle \tag{2.5}$$

The ground state energy satisfies $E \leq E[\Psi]$ for all admissible $\Psi$. This is the *variational principle* in Quantum Mechanics. Such principles have often been a source of insight in physics [Esc96].

### The Charge Density

The charge density has played an important role since the early days of quantum theory. In the 1930's Thomas and Fermi independently proposed density theories which ignored correlation effects and neglected exchange effects, which although naive, still have conceptual importance [Esc96].

Intuitively speaking, the charge density is a real-valued scalar field which quantifies how likely an electron is to be found at a given location in space. More precisely, given an $N$-electron wave function $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$, the charge density is the integral of the squared wave-function amplitude over all coordinates except one:

$$\rho(\mathbf{r}) = N \int |\Psi(\mathbf{r}, \mathbf{r}_2, \ldots, \mathbf{r}_N))|^2 d\mathbf{r}_2, \ldots, d\mathbf{r}_N \tag{2.6}$$

It is important to note that the charge density must integrate to the total number of electrons $N$, each of which has unit charge.

### The Many-Body Schrödinger Equation

The many-body Hamiltonian in ordinary units with $M$ nuclei (ions), $N$ electrons is given by

$$\begin{aligned} H = & \underbrace{-\sum_{I=1}^{M} \frac{\hbar^2 \Delta_{\mathbf{R}_I}}{2M_I}}_{\text{Kinetic energy nuclei}} + \underbrace{\sum_{I<J}^{M} \frac{Z_I Z_J e^2}{|\mathbf{R}_I - \mathbf{R}_J|}}_{\text{Ion-ion interaction}} \\ & \underbrace{-\sum_{i=1}^{N} \frac{\hbar^2 \Delta_{\mathbf{r}_i}}{2m}}_{\text{Kinetic energy electrons}} + \underbrace{\sum_{i<j}^{N} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|}}_{\text{Electron-electron interaction}} \\ & \underbrace{-\frac{1}{2} \sum_{I=1}^{M} \sum_{j=1}^{N} \frac{Z_I e^2}{|\mathbf{R}_I - \mathbf{r}_j|}}_{\text{Ion-electron interaction}} \end{aligned} \tag{2.7}$$

where $\mathbf{R}_i$ denotes nuclei coordinates, $Z_i$ denotes nuclei charges, $\mathbf{r}_i$ electron coordinates. This Hamiltonian acts on wave functions $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N, \mathbf{R}_1, \ldots, \mathbf{R}_M)$, which has dimensionality $3(N + M)$.

The terms in this equation are described as follows. The first and third terms are the kinetic energy operators for nuclei and electrons respectively. The second term corresponds to interactions between nuclei (ions), and the third term corresponds to interactions between electrons. Finally the last term corresponds to interactions of nuclei and electrons.

### The Born-Oppenheimer Approximation

The mass of a proton is much greater than that of an electron, around 2000 times so. Thus the nuclei will move much slower than the electrons. The Born-Oppenheimer approximation assumes that nuclei are fixed: only electrons are allowed to move. This is often a reasonable choice for describing chemical systems, where electrons are responsible for reactivity. Two reductions occur to the many-body Hamiltonian in this approximation. First the kinetic energy of nuclei vanishes due to the nuclei being assumed fixed (i.e. their velocities are zero). Second, fixing the nuclei positions determines the ion-ion interaction term, which we denote $E_{II}$. We have

$$H = \underbrace{-\sum_{i=1}^{N} \frac{\hbar^2 \Delta_{\mathbf{r}_i}}{2m}}_{\text{Kinetic energy electrons}} + \underbrace{\sum_{i<j}^{N} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|}}_{\text{Electron-electron interaction}} \underbrace{-\frac{1}{2} \sum_{I=1}^{M} \sum_{j=1}^{N} \frac{Z_I e^2}{|\mathbf{R}_I - \mathbf{r}_j|}}_{\text{Ion-electron interaction}} + E_{II} \tag{2.8}$$

where $E_{II} = \sum_{I<J}^{M} \frac{Z_I Z_J e^2}{|\mathbf{R}_I - \mathbf{R}_J|}$. This Hamiltonian acts on wave functions $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$, which has dimensionality $3N$.

Under the Born-Oppenheimer approximation, we can see that the number of particles has been reduced from $M+N$ to $N$. This is often a great help, but it may still be insufficient

when the number of electrons $N$ is large. In the next section, we will see one reason why.

#### Numerical difficulties

Numerical methods for PDE such as the finite-element method (FEM) can in principle be applied to the many-body Schrödinger's equation [LS85]. However, systems of practical importance may still contain a large number of electrons. This results in very high dimensional problem: each particle lives in 3-dimensions, resulting in a $3N$ total dimensions.

To see why this is a problem, consider a uniform grid with $d$ grid points in each coordinate. The total number of grid points is then $d^{3N}$, which is exponentially large. Although the FEM reduces the number of grid points versus simple uniform grids, it is known to not perform well in high-dimensions. Thus some new ideas are needed. One popular approach is so called Density Functional Theory (DFT), which we discuss in the next section.

### 2.1.4 Density Functional Theory

Density Functional Theory (DFT) is an approach to electronic structure theory which reformulates the $N$-particle quantum many-body problem in terms of an effective one-particle density [Mar20]. DFT is widely used in a number of physics and chemistry applications. Indeed some of the most cited papers in all of science are about DFT [VMN14]. DFT was developed by American chemist Walter Kohn and collaborators in the 1960's. He shared the Nobel prize in Chemistry in 1998 with John Pople for these contributions [Pop21]. Later developments were made by Levy and Lieb [Lev79; Lie83], who put the theory on rigorous basis using the perspective of constrained optimization.

In this work we state the minimal aspects of the theory which we need to formulate our contributions with machine-learning. We refer the reader to the references for more details [LL19; Esc96].

**Mapping Wave Functions to Charge Densities**

In a series of influential papers [HK64; KS65], Kohn and collaborators proved the existence of a unique mapping between *non-degenerate ground state* $N$-particle wavefunctions and the three-dimensional *ground-state* charge density.

$$\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N) \leftrightarrow \rho(\mathbf{r}) \tag{2.9}$$

Proof of this theorem, in particular the fact that every non-degenerate ground-state potential has a unique density, may be found in standard references [Esc96].

That such a map exists suggests that the many-body problem may be reformulated in terms of densities instead of wave-functions. Here we do not prove that these two approaches yield the same energy. Instead in the remaining subsections of this chapter we turn to the *Kohn-Sham energy functional* and the associated minimization problem to derive the Kohn-Sham equations.

Degeneracy of states is an important phenomenon which occurs in real-world physics. A more general density functional theory which eliminates the non-degeneracy assumption using the perspective of constrained-optimization was developed by Levy and Lieb [Lev79; Lie83]. We refer the reader to references for the full details [LL19; Esc96]

### 2.1.5 The Kohn-Sham Proposal

Kohn-Sham Density Functional Theory (KS-DFT) proposes the following energy functional [LL19]

$$\mathcal{E}^{KS}\left(\{\psi_i\}_{i=1}^N\right) = \frac{1}{2}\sum_{i=1}^{N}\int |\nabla_{\mathbf{r}}\psi_i(\mathbf{r})|^2 d\mathbf{r} + \int \rho(\mathbf{r})V_{ext}(\mathbf{r})d\mathbf{r} + \frac{1}{2}\int\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' + E_{XC}[\rho] \tag{2.10}$$

where $V_{ext}(\mathbf{r}) := V_{ext}(\mathbf{r}, \{\mathbf{R}_I\})$ is the external potential induced by the nuclei and $E_{XC}[\rho]$ is the *exchange-correlation* (XC) functional, which we discuss in Section 2.1.6, and the third-term is called the *Hartree* or *Coulomb* term. Note that the external potential $V_{ext}$ is independent of the charge-density $\rho$, and only depends on the input configuration of atomic nuclei.

Following the variational principle, the KS-DFT proposal for the ground-state energy is the following minimization problem [LL19]

$$E^{KS} = \inf_{\substack{\{\psi_i\}_{i=1}^N \\ \langle\psi_i,\psi_j\rangle = \delta_{ij}}} \mathcal{E}^{KS}(\{\psi_i\}) + E_{II} \tag{2.11}$$

where we denote the Kohn-Sham ground-state energy energy as $E^{KS}$ and $E_{II}$ is the nuclei interaction constant from Section 2.1.3.

### 2.1.6 The Kohn-Sham Equations

At the heart of DFT is a non-linear eigenvalue problem called the Kohn-Sham equations. The Kohn-Sham equations are derived by finding the stationary point of the functional in Equation 2.10. The functional derivative of $\mathcal{E}^{KS}$ is [LL19]

$$\frac{1}{2}\frac{\delta \mathcal{E}^{KS}(\{\varphi_i\})}{\delta \varphi_i^*(\mathbf{r})} = \left(-\frac{1}{2}\Delta_{\mathbf{r}} + V_{ext}(\mathbf{r}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}' + V_{xc}[\rho](\mathbf{r})\right)\varphi_i(\mathbf{r})$$
$$=: H^{KS}[\rho]\varphi_i(\mathbf{r}) \tag{2.12}$$

where $V_{xc}$ is the functional derivative of $E_{xc}$, $\rho(\mathbf{r}) = 2\sum_i^N |\varphi_i(\mathbf{r})|^2$, and the last line defines the Kohn-Sham operator.

After a suitable rotation of the orbitals [LL19], we obtain the *Kohn-Sham* equations:

$$H^{KS}[\rho]\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}) \tag{2.13}$$

These equations, a non-linear eigenvalue problem, are typically solved through fixed-point iteration. An initial guess for $\rho$ is made and then iterated until convergence. This iteration is an example of *self-consistent field* (SCF) iteration, a term used in the computational chemistry literature.

### The Exchange-Correlation Functional

The XC functional captures *exchange* and *correlation* effects. The term exchange refers to purely quantum effects like anti-symmetry required by the Pauli exclusion principle, e.g. if the position of two particles are *exchanged* then the wave-function must have a sign-change. The term correlation refers to many-body interactions, e.g. the dynamics of particles in electronic interactions are said to *correlate*.

The XC functional in some sense represents everything we don't know about the system, i.e. all contributions to the energy other than the kinetic, external, and Hartree terms. Proven to exist in theory and to be universal for all electronic configurations, the true form of the XC functional remains unknown. This makes DFT formal in nature. In practice

approximations are used, some of which are theoretically motivated, others of which are empirically fit to ground-truth cases. The design of such functionals is a major topic in computational chemistry.

## 2.2 Machine Learning

In this section we introduce relevant background on the machine learning used in this project. We begin a discussion of the data representation used in this project: geometric graphs, and to represent atomic structures as graphs.

### 2.2.1 Graph Representations of Atomic Structures

An atomic structure is completely characterized by a *set* of atoms and their positions in space. Formally, we can write this as follows: $N$ atoms at positions $\mathbf{R}_i \in \mathbb{R}^3$ with atomic numbers $Z_i$, define the structure $s = \{(\mathbf{R}_i, Z_i)\}_{i=1}^{I}$.

One can consider attempting to learn on sets directly [Zah+17; Zep+21]. However recent learning success [GGG20; Zit+22; Lia+24] has largely focused on using *geometric graphs* as the input representation for atomic systems. Geometric graphs are constructed by (1) treating atoms as nodes, (2) drawing edges between nearby nodes and (3) retaining two pieces of information as node features, the spatial location of each node and an integer *type*. The integer type of the node is important in atomic modeling as it tracks the atomic number of the atom.

The promixity of nodes can be defined in different ways [Duv+23]. Perhaps the simplest is using a hard radial cutoff, where two nodes $v_1, v_2$ at positions $\mathbf{R}_1, \mathbf{R}_2$ share an edge $e$ if $d(\mathbf{R}_1, \mathbf{R}_2) \leq c$ for some cutoff $c$ and distance $d$.

### 2.2.2 Graph Neural Networks

Graph neural networks (GNNs) are neural networks designed to learn on graph structure data [Sca+08; KW16; Gil+17]. The main paradigm for learning latent representations in such networks is *message passing* [Gil+17]. This involves there parts: (1) computation of messages (2) aggregation of messages and (3) updating of node latent representations [Duv+23]

Given nodes $i$ and $j$ at layer $t$ with latent representations $s_i^{(t)}, s_j^{(t)}$, we can write a basic expression for these operations as follows:

$$m_{ij}^{(t)} = \texttt{MESSAGE}_\theta \left( s_i^{(t)}, s_j^{(t)} \right) \tag{2.14}$$

$$s_i^{(t+1)} = \texttt{UPDATE}_\phi \left( s_i^{(t)}, \bigoplus_{j \in \mathcal{N}_i} m_{ij}^{(t)} \right) \tag{2.15}$$

where $\mathcal{N}_i$ is the set of neighbors around node $i$, subscripts $\theta, \phi$ indicate learnable parameters, and $\bigoplus$ represents a *permutation-invariant* aggregation function (e.g. sum or mean). Lastly, to obtain a scalar output, for example in the case of regression to predict continuous density values as we do here, a *graph pooling* operation is applied.

### 2.2.3 Equivariant Learning

A major goal in machine learning is the design of architectures suitable for a given problem, frequently referred to as an *inductive bias*. Historically perhaps the most prominent example of inductive bias is the so-called translation shift-invariance of convolutional neural networks. Given a pattern within the width or *receptive field* of a convolutional kernel, it does not matter where in the image the pattern lies, the kernel will activate none-the-less.

Invariance to a certain transformation may be formally stated as follows. A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be invariant with respect to the action of a group $G$ if

$$y := f(x) = f(g \cdot x) \quad \forall g \in G \tag{2.16}$$

where $y$ is a scalar. In other words the function value does not change when inputs are transformed.

In the context of shift-invariance for image classification, given a binary classifer on size $N \times N$ greyscale images $\mathcal{X} = [0,1]^{N \times N}$ such that $f : \mathcal{X} \to \{0,1\}$, the relevant group of transformations is the pixel translational group.

More generally, for functions with non-scalar outputs we have the concept of equivariance, where outputs are transformed in a way which matches the transformation of the inputs in some appropriate sense. To formally write this, we need the language of representation theory [Zee16].

A group representation is a map $D : G \to GL(n)$ from a group $G$ to the general linear group $GL(n)$ of order $n$, that is the space of $N \times N$ invertible matrices, with the additional property of $D(gh) = D(g)D(h)$, that is it is a *homeomorphism*.

We say that $f : \mathcal{X} \to \mathcal{Y}$ is *equivariant* with respect to the group $G$ if

$$D_{\mathcal{Y}}(g) \cdot f(x) = f(D_{\mathcal{X}}(g) \cdot x) \quad \forall g \in G \tag{2.17}$$

where $D_{\mathcal{X}}(g)$ and $D_{\mathcal{Y}}(g)$ are the actions of the group element $g$ on the vector spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively.

It is easily seen that the composition of two equivariant functions is again equivariant: if $f, h$ are $G$-equivariant functions, then for any $g \in G$ we have $h(f(g \cdot x)) = h(g \cdot f(x)) = g \cdot h(f(x))$. From this fact, we get the useful consequence that the task of building equivariant neural networks is reduced to the design of equivariant layers.

Equivariant functions with learnable parameters are functions $f : \Theta \times \mathcal{X} \to \mathcal{Y}$ with parameters $\theta \in \Theta$ such that $D_{\mathcal{Y}}(g) \cdot f(\theta, x) = f(\theta, D_{\mathcal{X}}(g) \cdot x)$ [GS22].

### 2.2.4 Invariant and Equivariant Graph Neural Networks

Invariant GNNs are graph neural networks which learn on geometric information from their input graphs. The level of geometric information used can be classified by the number of participating bodies. For example, the *distance* between involves two nodes, *angles* involve three, and *dihedral angles* involve four.

Some intuition can be gained with the following example. Suppose you define message passing to be a function of the distance between two nodes $m_{ij} = f_\theta(s_i, \|x_{ij}\|)$. Crucically distances are preserved under translations and rotations, and we can use this to define an *invariant* message functions. This is the idea from [Sch+18]. In this work, message passing is defined by the simple elementwise multiplication of node features with a radial basis function $\phi$ applied to the distance, $m_{ij} = s_j^{(t)} \odot \phi\left(\|x_{ij}\|\right)$. Then a simple sum of messages is taken to update the node representations.

A more complex method of message passing involves angles, which are also invariant to translations and rotations. In [GGG20], message passing is defined for triples of nodes $m_{ijk} = f_\theta(s_j, s_k, \angle ijk)$. Another involves the *dihedral angle* between a group of four nodes, which is essentially the relative orientation of two 3-atom sets sharing two nodes in common.This approach is used to define message passing in [GBG21].

The above are examples of *invariant* message passing layers. These layers used invariants such as distances and angles, which are fixed in the design of the network. More generally, it is desirable to define layers to *learn* the invariants that are appropriate for a problem, and are by design equivariant to the appropriate symmetry operations.

There are several ingredients needed to do so [Duv+23]. The first is that the *type* and *order* of each tensor is accounted for. Here the "type" of a tensor refers to the basis it is represented in: for example, in a *cartesian* or *spherical* basis. The "order" of a tensor refers to the number of indices needed to describe a tensor: it does not make sense (geometrically) to add scalars to vectors. Second, multiplication of tensors is done in such a way that respects the underlying types and symmetry transformations: if one argument of a product is rotated,

then the output is also suitably rotated. The *tensor product* $\bigotimes$ is an multiplication operation that does this, and can be used to build up higher order tensors from lower ones. The details of the product depend on the tensor type. Lastly, non-linear operations in the network need special treatment as they can break equivariance, and are typically only applied to scalar features.

## 2.3 Related Work

### 2.3.1 Molecular Machine Learning

Learning on molecules using the geometric graph paradigm has seen a number of works [XG18; Sch+18], modeling developments [Zit+22; GBG21; GGG20; Shu+21; Gas+22] and large-scale datasets [Cha+21; Tra+22]. See [Jos+23] for a summary of recent approaches based on the level of geometric information incorporated in the model.

### 2.3.2 Density Learning

Learning the Kohn-Sham charge density has been the focus of a number of works. One early work deserving of mention is [Bro+17][1], who give use a kernel-based density-learning approach to model the dynamics of small organic molecules.

Other notable works include (1) [Rac+22] who use Euclidean neural networks [GS22] to study generalization from smaller to larger clusters of water; (2) [Gri+19] who study transfer learning of density models from small molecules to larger ones such as hydrocarbons; (3) [JB20], who show small-scale results with a message-passing network using "two-step approach" to encode the structure graph and local environment separately, in contrast to the query node approach that the present work uses; (4) [Sch+19] who learn elements of the Hamiltonian matrix directly for single molecule cases using a modified Schnet model [Sch+18], and show savings in SCF iterations in select cases; (5) [Gon+19], who use a graph-

[1] From K. Burke and co-authors

neural-network based approach for density prediction which encodes the local environment with a related query node. Using the CGGCNN [XG18] model, a predecessor to more advanced equivariant and geometric graph models that we use here, they perform a small-scale study of model transferability to new kinds of bonds not seen at train time, e.g. train on linear C-C-C and orthogonal C-O-C and test on orthogonal C-C-C (See Figure 7 of [Gon+19]).

Another notable work on density prediction from is that of [Zep+21] [2], who represent an atomic structure as a *set* rather than a graph and define a novel network architecture with translation, rotation and permutation symmetries. They show up to four-digits of accuracy on non-catalyst test structures, including water, small organic molecules, and aluminum.

[2]From L. Lin and co-authors

# Chapter 3

# Computational Considerations for Density Learning

In this chapter we discuss important computational aspects of our density learning approach. We discuss the *linear scaling* of the charge density as a function of the number of atoms and implications for density learning approaches. We then compare density learning with energy learning approaches and with DFT for large systems.

## 3.1 Scaling of Density Functional Theory

The computational costs of DFT is *cubic* in the number of atoms [LL19]. For plane-wave basis approaches, this cost arises from the expensive diagonalization of the large and sparse Hamitonian matrix to compute the orbitals and their energies in every iteration of the SCF cycle. For Gaussian-basis approaches, this cost arises from the expensive computation of matrix elements.

The cubic cost of DFT has led to widespread adoption versus more expensive methods [YD23]. However for large systems, this cost may still be too large. This has motivated the development of *linearly scaling* electronic structure methods [Koh96; BM12; Goe99; LLY19], which seek to provide accurate electronic structure calculations at a an asymptotically linear

cost. Although linear scaling is clearly advantagous asymptotically, a large prefactor can still render such methods slower than cubically scaling counterparts. Developing practical linearly scaling methods remains an active area of interest [BM12; Moh+14; Yu+20; Nak+20].

## 3.2 Density Learning and Linear Scaling

The charge density is an example of a quantity in electronic structure theory that scales *linearly* in the number of atoms [Koh96; BM12; Goe99; LL19]. A physical argument for this comes from the *near-sightedness principle for electronic matter* [PK05; Koh96]. This principle states that the charge density and other electronic properties principally depend on the external potential only at *nearby* points, and that interaction effects decay rapidly with distance. Although this principle is not universally valid [Koh96], it is assumed to be widely applicable for molecules and materials without long-range interactions [Koh96]. It also lends support to the radial cutoff procedure used in constructing geometric graphs density learning.

Linear scaling has the following important consequences for learning approaches. First the amount of training data is linear in the total number of atoms in the training set. Second, the inference costs per test system are also linear in the number of atoms. Practically speaking, this means that given some reference system, 10x inference costs will be required for a system that has 10x the number of atoms in the reference.

To compute the energy with our approach, we also require one KS-DFT diagonalization step. As discussed in the previous Section, the cost of this is cubic, the same as DFT. Our results suggest that on average our method achieves a lower prefactor.

## 3.3 Comparison with Energy Learning

Many works attempt to learn the energy and other scalar values directly [Cha+21; Tra+22]. Such approaches target a fixed number of scalar values per input. This means that a *con-*

*stant* number of values, independent of molecular size, are used to describe each input. A consequence of this is that the same amount of computation is spent on smaller systems as larger systems. This may be problematic, as large systems may be more complex and require more computation to accurately approximate them.

Density learning incurs greater costs in training and inference versus energy learning. For applications involving a very large number of systems, energy approaches may be preferable. This may come with costs in accuracy however. In addition, if more information than just the energy is required, then density learning may be preferable.

More information than the energy may be obtained with density learning after one diagonalization step. In principle all one-electron quantities are computable from the ground-state density. In addition, the initial density estimates may be refined to greater accuracy by several steps. Energy based approaches do not allow for refinement in this way.

## 3.4 Comparison with DFT on Large Systems

In the next chapters we will show that our method achieves computational savings versus standard baselines in DFT. Importantly the measure we use to do so is *hardware independent*: the number of iterations saved when initializing from learned densities. Despite their cubic scaling, existing DFT codes are still faster in terms of *wall-clock* time on the small molecules we evaluated and the hardware we used. Inference at test time for our method may still take significant time, especially for large grids. The overall time depends on the amount of GPU hardware used. Grid predictions are independent and may be run in parallel, so in principle inference time can be made negligble. Nevertheless, we found GPU memory to be the limiting factor for overall runtime of our method. Due to the linear scaling inference costs, we expect our method to outperform the cubically scaling costs of DFT on large enough systems.

# Chapter 4

# Saving Iterations with Density Learning for Catalyst Materials

This chapter describes our work in [PJ23].

## 4.1 Introduction

Density Functional Theory is widely used in a number of quantum-chemical applications. One such application with important economic and environmental consequences and growing attention from the machine-learning community in recent years is the discovery of new catalysts [Zit+20]. Catalysts are chemical reagents which increase the rate of a reaction by lowering the activation energy required. Today catalysts lie at the heart of many economically important industrial processes. The discovery of new catalysts is envisioned to play an important role in the transition to a Green economy, for example in the production of Hydrogen fuel from renewable energy. The electronic structure of catalysts plays a fundamental role in their reactivity. However, as an instance of the quantum many-body problem, the determination of such electronic structure poses numerous difficulties. A number of computational methods have been developed, spanning a range of costs and accuracies. Density Functional Theory is the most widely used electronic structure method for catalysis.

In this Chapter, we investigate the application of machine-learning models for the electronic structure of candidate catalyst materials. Our strategy is to develop machine-learning models which provide cheap-and-fast approximations of numerical quantities in algorithms for electronic structure. Our main contribution is the use of equivariant graph-neural-networks to predict the Kohn-Sham charge-density. We show computational savings in terms of the number of requisite iterations needed to reach convergence. We develop our methods on data relevant to catalysis. Although our methods are general, we do so to connect our work to an application with potentially large environmental impact.

We carry out an empirical investigation of density based models in catalyst systems. The focus of our investigation is whether such models can generalize to new combinations of elements. Our contributions are as follows:

- We compute a new large-scale DFT dataset of $\mathcal{O}(1000)$ relaxed bulk catalyst structures *with charge densities* spanning $\mathcal{O}(100\text{M})$ unique points using the open-source DFT software Quantum Espresso and workflow software AiiDA [Piz+16; Gia+17; Hub+20; Hub+21; Uhr+21; Pra+18].
- We design an evaluation methodology to measure when learned densities improve convergence versus standard baselines for density initialization in DFT. To do so we define a threshold-independent metric and also report the number of iterations saved at convergence.
- We show that density models improve convergence on new structures not seen at train time. Specifically we find that learned densities outperform baselines in 83% and 86% of test cases for binary and ternary catalyst respectively. These savings amount to a reduction of 13% in the number of SCF iterations needed to reach convergence.

To the best of our knowledge, our results are the first density-based approach to show improvement over standard baselines for density initialization in DFT on new structures not seen at train time.

### 4.1.1 Combinatorial Size of Catalyst Space

Perhaps the greatest difficulty of catalyst discovery is the combinatorial size of the search space of structures. Even when reducing the candidate set of elements to 55, as done in the Open Catalyst Project [Cha+21], the number of possible element combinations grows very quickly: we have $\binom{55}{3} = 26,235$, $\binom{55}{4} = 341,055$, and so forth. Additional factors like choice of crystal lattice, surface orientations, binding sites, and adsorbates further complicate matters, but nevertheless the number of possible element combinations is a dominating factor. Given the immensity of this search space, *generalization to new combinations* is a key aspect for the success of property-predicting models in applications. Put differently, predictive models that cannot combinatorially generalize will fail to cover large parts of the search space. Moreover some authors argue that combinatorial generalization is an intrinsically important property for machine learning models in general [Bat+18].

## 4.2 Methods

### 4.2.1 Local Property Prediction with Geometric Graph Learning

Standard geometric graph learning typically predicts a global property of a structure/molecule, e.g. the energy or band gap. Here we modify this approach for local property prediction by adding a "virtual atom" representing the query point to the graph. This approach is similar to [Gon+19], except our case deals with geometric graphs.

Let $s = \{(\mathbf{R}_i, Z_i)\}_{i=1}^{I}$ denote a structure $s$ with $N$ atoms at positions $\mathbf{R}_i \in \mathbb{R}^3$ with atomic numbers $Z_i$. Let $\rho_s : \mathbb{R}^3 \to \mathbb{R}$ denote the density function associated to a structure $s$, evaluated on a grid of query points $\{\mathbf{r}_j\}_{j=1}^{J}$, with all $\mathbf{r}_j \in \mathbb{R}^3$.

In the usual formulation, radial graphs are typically created from each structure, i.e. draw an edge between two atoms if they lie within some radius of each other. To instead make the graph local to a point, we first adjoin the query point to the structure as a "virtual"

atom $(\mathbf{r}_j, Z_{I+1})$, where we interpret the atomic number $Z_{I+1}$ simply as just an unused index on atom types. Note we use the same index for all query points. Letting $\mathcal{G}(\cdot)$ denote the operation of taking the radial graph, the localized graph then takes the form $\mathcal{G}(s \cup (\mathbf{r}_j, Z_{I+1}))$. Denote all such pairs for a structure as $\mathcal{D}_s = \{\mathcal{G}(s \cup (\mathbf{r}_j, Z_{I+1})), \rho_s(\mathbf{r}_j)\}$. Lastly, we complete the dataset by taking a union over all structures $\mathcal{D} = \bigcup_s \mathcal{D}_s$.

Given this data, we proceed as usual to stochastically optimize parameters $\theta$ of a graph neural network GNN to minimize the expected loss $\mathcal{L}$ of samples $g, \rho$ drawn from a (uniform) probability measure $\mathbb{P}_{\mathcal{D}}$ on the dataset $\mathcal{D}$:

$$\min_{\theta} \mathop{\mathbb{E}}_{g,\rho\sim\mathbb{P}_{\mathcal{D}}} \Big[ \mathcal{L}\left(\mathrm{GNN}_\theta(g), \rho\right) \Big] \tag{4.1}$$

Since densities are continuous real-valued quantities, a regression loss is used.

### 4.2.2 Creating a Dataset of Catalyst Densities

Although there are emerging datasets of charge-densities [She+21], our evaluation requires that the settings of the data and solver be *exactly the same* to ensure numerical consistency. To that end, we generate our own dataset of relaxed bulk catalysts using the well-established and open-source DFT solver Quantum Espresso [Gia+09; Gia+17] and modern computational workflow management system AiiDA [Hub+20]. In particular we utilize the relaxation workflows of [Hub+21], which automate the selection of computational parameters critical for achieving convergence based on expert knowledge.

For initial candidates for bulk catalysts we use the list of bulk catalysts identified by [Cha+21] from the Materials Project [Jai+13], which contains $11,410$ structures sampled from 1, 2 or 3 element combinations of a set of 55 elements. The total number of element combinations represented in this set is 5099.

Grid sizes per structures can vary by up to four orders of magnitude, which can strongly skew the training data toward larger structures. To mitigate this we restrict the maximum

Table 4.1: Sample sizes of dataset

| | $N_{\text{structures}}$ | $N_{\text{samples}}$ |
|---|---|---|
| Train (unary) | 47 | 3.8M |
| Train (binary) | 392 | 69M |
| Val (binary) | 4 | 300k |
| Test (binary) | 360 | 72M |
| Test (ternary) | 1116 | 380M |

number of atoms to be 12 and the maximum volume of the (conventional) cells to 200 $\text{Å}^3$. We then remove candidates with oxidation states which only occur once in the dataset, since these are not well-represented enough to evaluate, using the oxidation state guesser in `pymatgen` [Ong+13].

We then partition the train/val/test splits as follows. First we assign all unary catalysts remaining to the train set. Then we randomly sample binaries with oxidation states not already represented in the unaries. Next, we remove oxidation states not present in the training set from the remaining binary and ternary candidates. This preprocessing step is performed to ensure all states in the val/test splits were represented in the training set, e.g. training on $H^+$ cannot be expected to generalize to $H^-$. Finally we assign a few binaries to the validation split and the remaining binaries and ternaries to the testing split. Finally we validate that train and test splits are disjoint by element combinations, i.e. no combination of elements in the training set is repeated in the test splits (although sub-combinations may).

We relax each structure in these splits using Quantum Espresso with parameters set by the AiiDA relaxation workflow [Hub+21]. We use the `fast` protocol with the PBE exchange-correlation functional [PBE96] for all relaxations. Initial structure positions were first perturbed with noise. Unconverged runs were dropped from the dataset. We report the number of structures and density samples in Table 6.1.

### 4.2.3 Evaluating Convergence Over DFT Baselines

An important baseline for density based models is the "atomic charge superposition" (ACS) initialization for charge densities. This initialization scheme is commonly used in DFT solvers. The ACS initialization is derived from the pre-computed collection of atomic *pseudopotentials* used with the DFT solver, an important technique that reduces the number of electrons in the computation. For further details see [SCS10; LLY19; Mar20; LL19].

More precisely, given a configuration of atoms $\{(\mathbf{R}_i, Z_i)\}_{i=1}^I$ and associated atomic charge densities $\rho_{Z_i}$, the ACS initialization is $\rho_0^{ACS}(\mathbf{r}) = \sum_i^I \rho_{Z_i}(\mathbf{r}-\mathbf{R}_i)$. In other words it is simply a sum of radial functions. Note this initialization corresponds to a scenario of non-interacting atoms: poor performance may be expected when the atoms interact, i.e. when there is chemistry.

In this work we consider learned densities that do not improve upon this baseline to *not* be of interest. From a practical point of view, clearly initializations which do not beat this baseline would not be relevant. Nevertheless, given the difficulty of obtaining high precision results with learning-based approaches, this baseline is highly non-trivial.

The degree of precision required depends on the application. One could set a threshold and check which density reached convergence in fewest iterations. However, we wish to perform the evaluations in a threshold independent way to remain agnostic to the requirements of the application, similar in spirit to "area-under-the-curve" (AUC) measures popular in classification.

To that end we propose a "signed area under curves" (s-AUC) measure, essentially the area between the curves of convergence for the baseline and learned densities. See Figure 4.5 for examples. Note the sign is necessary because the curves may cross. More formally, if $\{x_i\}_{i=1}^{n_x}$ and $\{y_j\}_{j=1}^{n_m}$ are the SCF accuracy values for the baseline and learned curves respectively, and $n = \min(n_x, n_y)$ we define

$$\text{s-AUC} = \frac{\sum_{i=1}^{n} \log(x_i) - \log(y_i)}{\sum_{i=1}^{n} |\max(\log(x_i), \log(y_i))|} \tag{4.2}$$

where we have used a log-scale to show rate of convergence and normalized by the point-wise max to facilitate comparison between difference curves.

We also report the percentage of iterations saved at convergence. Although this measure is threshold dependent, it has the benefit of being easily interpretable to the ML+DFT community.

### 4.2.4 Model Training

We train a reduced sized Spherical Channel Network (SCN) for density prediction. We reduce the number of interactions to 10 and hidden channels to 256, resulting in 35M parameters, which is smaller than state of the art models [Zit+22]. We use interaction cutoff to 6.0Å, and a maximum number of neighbors per node of 12. We list full hyperparameters in the Supplementary.

We train two replicates of SCNs for 420k steps with batch size 45 across 8 GPUs. Regarding compute, for training we used approximately 2000 GPU hours for both replicates across all GPUs. We use `NVIDIA A5000` cards with 24 GB of memory.

In initial experiments we also evaluated Dimenet++ [GGG20] and GemNet [Gas+22] models but generally found performance on these models to be less than SCN models.

## 4.3 Results

### 4.3.1 Model Performance

In training we obtain mean absolute errors (MAE) of $\mathcal{O}(10^{-4})$. Best validation MAE was 0.0011 for both replicates. One replicate was chosen to report the remainder of the results.

We show the distribution of MAE scores across each train/test split in Figure 4.1.

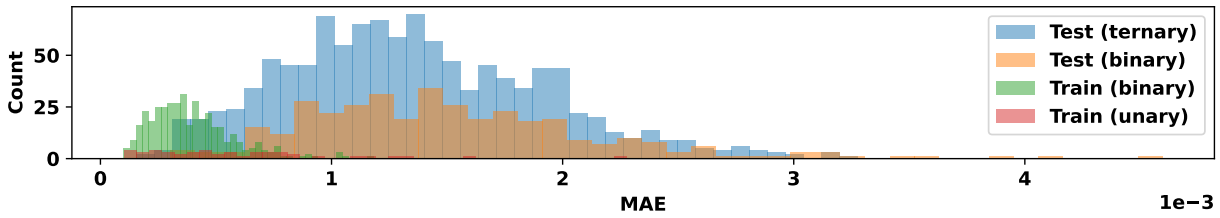


Figure 4.1: Distributions of MAE scores for train/test splits

### 4.3.2 Visualization of Learned Densities and Errors

One advantage of local property prediction is the ability to directly visualize errors as a function of input points. In Figure 4.2 we show the true density, predicted density, and (scaled) absolute error for the top and bottom test predictions by MAE.

These visualizations allow us to inspect of distribution of errors, which may facilitate e.g. model debugging, among other uses. Here we see a variety of patterns: ranging from nearly non-existent to quite strong errors near the core region of atoms. A closer analysis of these error patterns may inform future improvements to the dataset and model.

### 4.3.3 Evaluation of Learned Density Convergence

We evaluate whether initializing with learned densities leads to faster convergence in the SCF cycle. After predicting across the grid associated to each structure, we pass predictions to the DFT solver to initialize a new SCF cycle using *exactly the same runtime parameters* as the ground-truth run. Then we extract convergence information and compare them against the ground truth.

We show summary statistics of our results on SCF savings according to the s-AUC metric in Table 4.2. Notably we achieve positive savings in 83% of test binary cases and 86% of ternary cases, with a combined proportion of 86%. The mean and median savings in each case are positive.

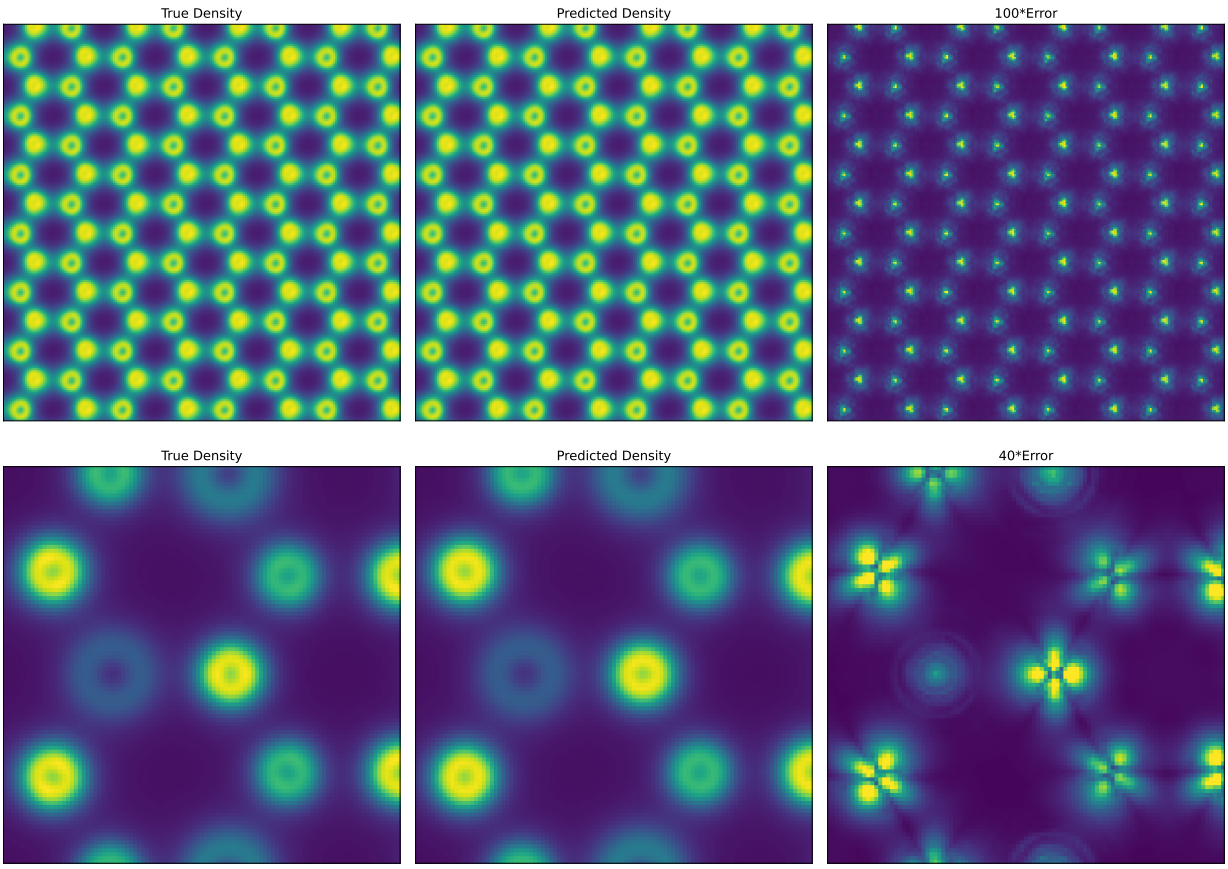


Figure 4.2: Test densities (2D projections): ground truth, predicted, and error. Top row is low error test structure $MoS_4W$(`mp-1023954`) and bottom row is higher error test structure $Fe_6Re_2$(`mp-865212`) with higher MAE error. Errors rescaled display more clearly. To visualize 3D densities are regridded to approximately a cube [She22] and summed along one axis.

Table 4.2: Summary statistics of SCF savings with the s-AUC metric.

| | N | $N_+$ | $N_+/N$ (%) | Mean | Median | Max | Min |
|---|---|---|---|---|---|---|---|
| Train (unary) | 47 | 45 | 96 | 0.6 | 0.3 | 7.3 | -0.1 |
| Train (binary) | 392 | 379 | 97 | 4.0 | 0.6 | $2.9 \times 10^2$ | -0.5 |
| Test (binary) | 360 | 298 | 83 | 7.8 | 0.2 | $1.9 \times 10^3$ | -3.7 |
| Test (ternary) | 1116 | 965 | 86 | 2.7 | 0.4 | $7.2 \times 10^2$ | $-2.8 \times 10^2$ |

Table 4.3: Summary statistics of relative iterations saved at convergence.

| | N | $N_+$ | $N_+/N$ (%) | Mean (%) | Median (%) | Max (%) | Min (%) |
|---|---|---|---|---|---|---|---|
| Train (unary) | 47 | 32 | 68 | 15 | 14 | 78 | -53 |
| Train (binary) | 392 | 361 | 92 | 22 | 21 | 63 | -50 |
| Test (binary) | 360 | 256 | 71 | 13 | 12 | 72 | -53 |
| Test (ternary) | 1116 | 841 | 75 | 13 | 12 | 65 | -90 |

In addition to the s-AUC metric, we also report savings in terms of the relative percentage of iterations saved at convergence in Table 4.3. The percentage of test structures with positive savings was 71% and 75% of binary and ternary test cases. The mean percentage of iterations saved was 13% for both binary and ternary splits.

We show the distribution of s-AUC scores and iteration savings across splits in Figures 4.3 and 4.4.

W plot the convergence behavior of top test cases by s-AUC in Figure 4.5. Positive savings can be seen when blue-dashed curves lie beneath the red-solid lines.

### 4.3.4 Validating Learned Densities

We perform two validations to check if learned densities converge to the same solution as the ground truth.

First we checked for any bias in converged energy values between ground-truth and learned densities. We find almost zero difference these energy values. On the test set we found 88% of structures had exactly zero difference in energy values. Of the cases with non-zero error, the relative error on average was $1.2 \times 10^{-11}$ and at maximum was $1.5 \times 10^{-9}$.

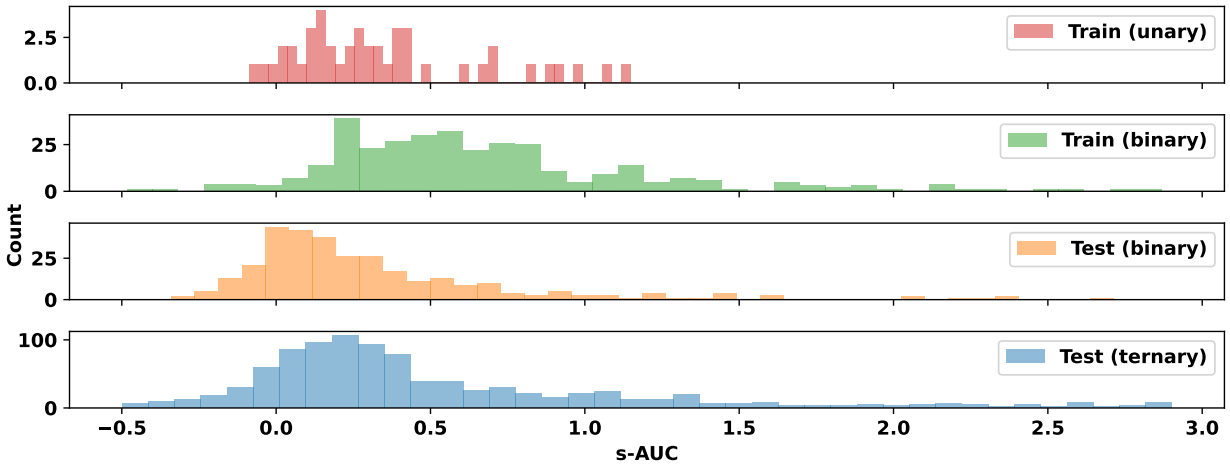


Figure 4.3: Distributions of `s-AUC` metrics. Outliers dropped for visualization.

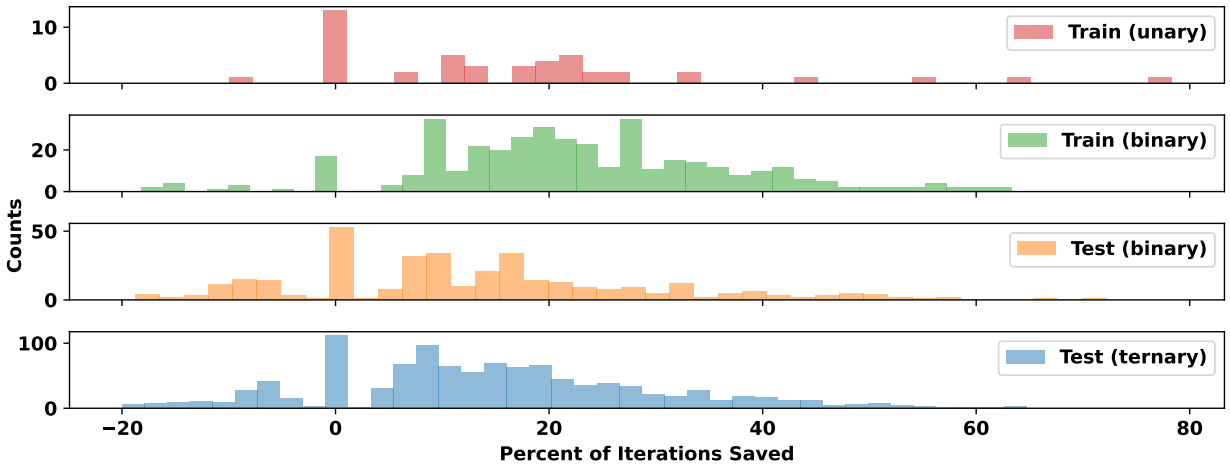


Figure 4.4: Distributions of relative iteration savings. Outliers dropped for visualization.

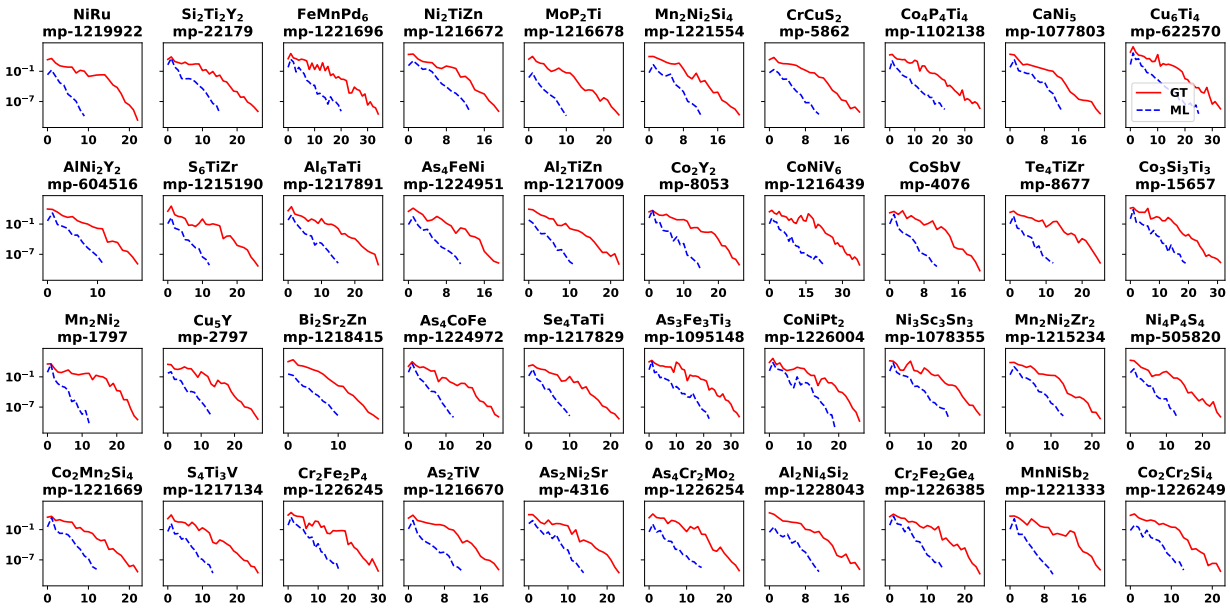


Figure 4.5: Top convergence results for test structures by s-AUC metric. Red solid lines are ground-truth DFT convergence, blue dashed lines are learned densities. Y-axis is SCF accuracy on a log-scale and x-axis is number of SCF iterations. Chemical formulas and material project ids given above each subplot. Note these structures were not seen at train time.

Second, we check if a downstream property computed from the learned densities matches their ground-truth counterparts. We compute and compare the band gaps computed from the ground-truth and ML densities using the built-in tool in ASE [Lar+17]. We find 95% of values to be exactly the same, with maximum relative error at $7.8 \times 10^{-5}$.

## 4.4 Discussion

Overall we find in over 80% of test cases learned densities achieve faster convergence versus the baseline. Importantly these test cases are out-of-distribution, involving structures with new combinations of elements not seen at train time. Our results show that training on simpler structures, i.e. unary and binary, can lead to generalization on more complex structures, i.e. ternary.

## 4.5 Limitations

In this work we only test generalization for combinations of elements, and ignore stoichiometric ratios. In addition, we only investigate bulk catalysts, not the full complexity of catalyst systems as modeled in [Cha+21] which includes adsorbates, surfaces, and non-equilibrium points. Extending our analysis to these more complex systems is left to future work.

Our energy values are not directly comparable with those of [Cha+21], because we generated our dataset with a different DFT solver and pseudo-potentials. Systematic numerical differences may exist between different solvers, which makes comparison of energy values difficult.

Another limitation is that we do not report timings of our method versus traditional CPU-bound methods. Because of the high GPU-bound inference costs of our method, we do not expect timing improvement versus traditional methods. Sufficiently many GPUs will help mitigate this, as well as recent hardware development trends towards larger GPU memory.

## 4.6 Summary

We generated a new dataset of bulk catalyst relaxations with charge densities using open-source software. We trained a density-model pointwise on unary and binary catalysts, and showed this model can generalize to new binary and ternary catalysts, a form of combinatorial generalization. We found learned densities lead to faster convergence than standard baselines used to initialize the SCF cycle of DFT in over 80% of cases, amounting to an average of 13% saving in iterations.

## 4.7 Broader Impact

One way this work may have positive broader impact is through reducing costs of DFT computations. For example, supposing that the USA National Labs spend $100M on DFT per year, and a negligible cost of inference, then a 13% reduction in SCF iterations would yield a savings of $13M [1]. Although there are a number of practical details to resolve, we believe this work is an important first step.

The potential impacts of new catalysts discoveries may have significant economic and environmental consequences. Positive impacts may include accelerating the transition towards renewable energy, an important goal for future societies. Negative impacts may include unintended consequences such as contributing to unsustainable population growth. In addition, as with any new technology with potentially large impacts in the economy, there is risk that control of the technology may concentrate in the hands of the few and limit the distribution of benefits.

In early stages of research is difficult to discern if the aims will come to fruition. The computational discovery of new catalysts does not necessarily imply that the results are translatable into real-world materials, e.g. not all materials are known how to be made easily or economically. If such technology can be discovered, we believe that the benefits can be more equitably distributed by making this research completely open-source and accessible to global researchers. Our use of open-source DFT solvers is one notable step towards this goal over previous approaches [Cha+21].

[1] We thank an anonymous reviewer for suggesting this way of interpreting our results

# Chapter 5

# Chemically Accurate Total Energy Estimation With One-Shot Density Learning

## 5.1 Introduction

Approximating the outputs of quantum chemical simulations such as Kohn-Sham Density Functional Theory (KS-DFT) is a major goal for machine learning and has the potential to accelerate scientific discovery for a range of societally important applications. Geometric graph learning with equivariant models is a prominent approach to this goal that has seen success on large-scale benchmarks, however the accuracy of such models remains insufficient for many applications. In this work, we show that highly accurate estimation of energy values may be obtained with a density learning approach, where the energy values are computed in "one-shot" from with a single KS-DFT step. Using state-of-the-art equivariant graph neural networks, we show on molecules that learned densities can be use to estimate the energy with accuracy sometimes exceeding one milli-Hartree per atom. To do so, we devise strategies to address the large amount of density, which arise from large computational grids

in DFT simulations. Along the way, we create a new dataset of charge densities using the open source solver `PySCF`. Our results demonstrate that high accuracy may be achieved with density learning.

## 5.2 Theory

We use Density Functional Theory at the level of the *Perdew-Burke-Ernzerhof* functional (PBE) with GTO basis sets. The Fock matrix is defined as

$$F_{\mu\nu} = h_{\mu\nu} + J_{\mu\nu}[\rho] + V^{xc}_{\mu\nu}[\rho] \tag{5.1}$$

where: $h$ is a one-electron energy, $J$ is a Coulomb term, and $V^{xc}$ is the exchange-correlation functional.

The eigenvalue problem for this matrix is typically solved iteratively starting from some initial guess for $\rho$, and then solved until convergence using a DIIS algorithm. Standard strategies for SCF initialization include the sum of atomic densities, and superposition of atomic potentials [Leh19].

Improving the initializations are of practical interest, as the closer the initialization is to the solution, the fewer iterations will be required to reach it. Given a candidate density (and not a density matrix) to initialize the SCF computation requires the construction of the Fock matrix and one KS-DFT diagonalization. We refer to this as a "one-shot" initialization. In addition we make two further design choices.

First, to make computation of the Coulomb term $J$ fast, we require SCF computations with periodic boundary conditions, so that a Ewald summation techniques may be used. Second, since PBE is a GGA type functional that depends on the gradient of the density $\nabla\rho$, we use a second-order finite difference scheme to approximate it. These design decisions in place, initialization of SCF cycles with new densities may be achieved.

## 5.3 Methods

### 5.3.1 Learning Approach

We use the geometric graph learning paradigm for pointwise charge density prediction. Following the approach in the previous Chapter , use a *query node* approach for local property prediction on such graphs.

Briefly, this approach involves adding a special node to the molecular graph which represents the grid point at which the density target lies. This node is assigned an unused atomic index to make it fully consistent with other nodes in the graph.

Given a dataset collection of molecular graph and density $g, \rho$ pairs, we stochastically optimize the expected loss $\mathcal{L}$ with respect to parameters $\theta$ of a graph neural network GNN.

$$\min_{\theta} \mathop{\mathbb{E}}_{g,\rho\sim\mathbb{P}_{\mathcal{D}}} \Big[ \mathcal{L}\left(\mathrm{GNN}_{\theta}(g), \rho\right) \Big] \tag{5.2}$$

As we discuss later in Section 5.3.3 , we use a *weighted* probability measure $\mathbb{P}_{\mathcal{D}}$ on the dataset $\mathcal{D}$, where the weight of each sample is determined by the magnitude of the density value.

### 5.3.2 Data

#### Candidate Molecules

We start with the QM9 dataset for our collection of molecular candidates [Ram+14]. QM9 contains $134k$ molecular structures, drawn from a larger algorithmic enumeration of small organic molecules with up to 17 heavy atoms $\{C, N, O, S, F\}$ (not counting Hydrogen) [Rud+12].

### Design of Train and Test Splits

In designing our train and test splits, we deviate from the conventional practice of random sampling and make some design choices with practical relevance in mind.

First we explicitly evaluate the generalization of training on smaller molecules and testing on larger molecules. This scenario is expected to be more difficult than simple random splitting, since the model must *combine* information learned from smaller molecules. We envision this to be more relevant to real-world application of such models: practitioners will likely want to use models on larger molecules outside the training set. Consequently the extent to which models can perform in this regime is of practical interest.

Second, we attempt to encourage more structural diversity in our train/test splits by conditioning our sampling on chemical formulas (the number of each elements present in a molecule). QM9 spans 621 unique chemical formulas. We sample at least one structure from each.

### DFT with PySCF

Next, given a collection of candidate structures, we compute the electronic structure using the open-source computational chemistry library `PySCF`. Following the discussion in Section 6.2, we use Density Functional Theory at the level of PBE under periodic conditions. We describe the full settings used for our DFT runs in the Appendix. We select molecules which converged at an energy difference threshold of $10^{-8}$ within a maximum number of iterations of 50.

### Data Reduction Strategies

A large number of grid-points may be necessary for highly accurate DFT computations. In principle all such data is available for training and evaluations, however even for a dataset containing hundreds of structures, the total number of grid points may range into the hundreds of millions. Common wisdom in machine learning is that more data improves gener-

alization. However a tension exists between data and computational cost: more time and resources are needed to process larger amounts of data.

Here we discuss strategies for reducing the computational burden while simultaneously attempting to achieve high accuracy. We first discuss a simple strategy that may be applied at both train and test. Then we discuss training/testing specific strategies.

First, we use the common practice in computational chemistry and materials science of reducing grid resolutions in DFT calculations with a plane-wave basis with the kinetic energy (KE) cutoff.

Next, we observe that a significant fraction of grid-points in our DFT computations have density value nearly zero. There is a physics based argument for this: grid points away from a molecule contributes little to the total energy.

That the data is mostly empty space is problematic for the speed of training, because under a uniform sampling scheme most samples will have near zero density and small error signal. To address this we drop density values with magnitude $\rho(\mathbf{r}) < \rho_{cut}$.

At test time the magnitude of density values are not known a priori. Instead we use a different strategy based on atomic distances to reduce inference costs. After calculating the *minimum* distance to any atom for all grid points, we set points outside of $r_{cut}$Å to be zero.

To justify choices of these parameters, we conduct an analysis on reference densities to estimate the loss of accuracy as a function of each cutoff. We discuss these details in the Appendix. Our analysis indicates that $E^{KE}_{cut} = 100$, $\rho_{cut} = 0.001$ and $r_{cut} = 5$Å are suitable choices for mHa accuracy.

### Dataset Statistics

We report sample sizes after applying this strategies in Table 6.1.

Table 5.1: Sample sizes of dataset

| | $N_{\text{structures}}$ | $N_{\text{atoms}}$ min/med/max | $N_{\text{grid}}^{KE=200}$ | $N_{\text{grid}}^{KE=100}$ | $N_{\text{grid}}^{KE=100+cut}$ |
|---|---|---|---|---|---|
| Train-small | 10 | 3/4/5 | 42M | 15M | 343k |
| Train-medium | 100 | 3/8/8 | 530M | 190M | 5.6M |
| Train-large | 1000 | 3/10/11 | 6.1B | 2.2B | 72M |
| Test | 1000 | 17/19/27 | 8.5B | 3.1B | 286M |

### 5.3.3 Training

#### Model

We use highly parameterized equivariant models for density prediction. In particular we use the EquiformerV2 model [Lia+24]. We use the same architecture for all experiments, with hyperparameters based on [Che24]. The model contains 8 layers with width 128 spherical channels, resulting in 28M parameters. We use interaction cutoff of 8Å, and a maximum number of neighbors per node of 25. We list full hyperparameters in the Appendix.

We train models for at least one epoch with batch size 32. The number of GPUs used depends on the amount of training data. Models with $N_{train} \in \{10, 100, 1000\}$ used $1, 4,$ and16 GPUs respectively. The total compute used for training was approximately 24 hours per GPU. We use `NVIDIA 2080Ti` cards with 12 GB of memory.

#### Training Strategies

To increase training speed, we use a *weighted random sampling* scheme, where samples are weighted by the magnitude of their density value. More formally, let $\rho_j = \rho(\mathbf{r}_j)$. Given a dataset of $N$ graph, density pairs, $\mathcal{D} = \{(g_i, \rho_i\}_{i=1}^N$, let $R = \sum_i \rho_i$. If $X$ is random variable on $\mathcal{D}$, the probability that $X$ selects sample $j$ is:

$$\mathbb{P}_{\mathcal{D}}\left(\mathrm{X}{=}(g_j, \rho_j)\right) = \frac{\rho_j}{R} \tag{5.3}$$

Normalization by $R$ is necessary for this distribution to sum to unity. On uniform grids the ratio $\frac{\rho_j}{R}$ represents the fraction of charge at the location $\mathbf{r}_j$ over the *total* charge in the dataset. In the course of our training experiments we found that using weighted random sampling improved the speed of convergence.

We discuss additional implementation details for training in the Appendix.

### 5.3.4 Evaluation

We evaluate the quality of predictions at two levels: the level of density and the level of energy.

To evaluate predictions at the level of the density, we use the normalized MAE per structure [Gri+23], the absolute error of the predictions versus the reference density, integrated over all grid points and normalized by the total number of electrons:

$$\text{MAE} = \frac{1}{N_e} \int |\hat{\rho}(\mathbf{r}) - \rho(\mathbf{r})| d\mathbf{r} \tag{5.4}$$

This metric measures density error as a fraction of the total charge. We report it as percentage $(0 - 100\%)$. We obtain a distribution of errors across a collection of structures, which may further be summarized with point-estimates like the mean and median.

To evaluate predictions at the level of the energy, we first compute the total energy from the learned density by means of a single KS-DFT diagonalization step, and then evaluate the absolute error against the *converged* reference energy. In addition to the total energy, we also evaluate the accuracy of HOMO/LUMO energy values. These interior eigenvalues of the Fock matrix pose a greater challenge for estimation.

As a reminder, we design our test splits to contain strictly larger molecules than in the training set. See Table 6.1 for statistics on molecular sizes per split.

### 5.3.5 DFT Baseline

We compare the performance of our method versus a standard baseline for SCF initialization. Specifically we construct the sum of atomic densities (SAD) generated by the `minao` initialization method [Van+06], which is the default in `PySCF` [Sun+18; Sun+20] and then compute the energy with one-step of KS-DFT. This method generates an initial guess for the *density matrix*, from which the total energy may be computed in linear time without diagonalization. We take the additional step of constructing the SAD to make an "apples-to-apples" comparison with initializing from learned densities.

## 5.4 Results

### 5.4.1 Test Set Results

In Figures 5.1, 5.2, 5.3 we show results on the test set for the density MAE, total energy, and HOMO/LUMO metrics respectively. In each figure we show results for each of the three models trained with different number of structures.

We observe all metrics to improve with more training data. Most density MAEs are below 5% at $N_{train} = 1000$. Figure 5.2 shows that the mean error in the total energy (purple dashed line) is near chemical accuracy per atom at $N_{train} = 1000$. The HOMO/LUMO errors are consistently higher and improve more slowly with more training data as shown in Figure 5.3.

The error distributions tend to be bimodal. Further analysis of the structures within each mode is likely to suggest improvements to the training data.

## 5.5 Conclusion

We trained density models on small molecules and showed that near chemical accuracy in the total energy may be achieved after one diagonalization step. We show highly-accurate

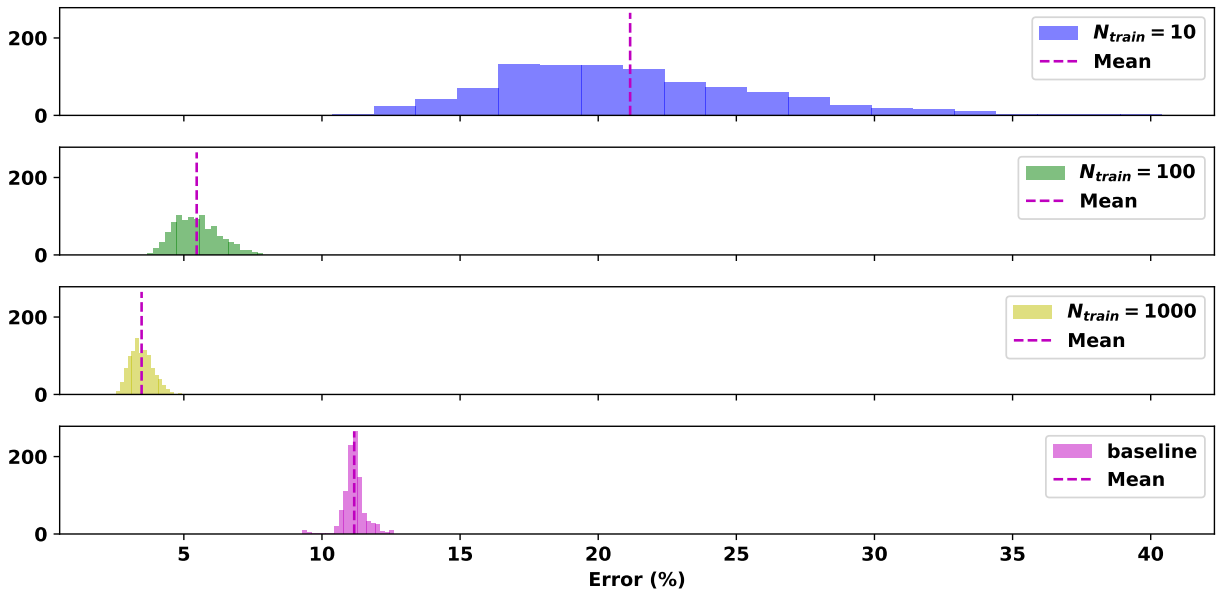


Figure 5.1: Distributions of test set density MAEs

generalization to larger molecules than in the training set. Our work suggests density learning is a reliable means of obtaining energy estimates for new structures.

One limitation of our approach is large inference costs at test time. We discussed data reduction strategies to address this. Further improvement can be expected in future work.

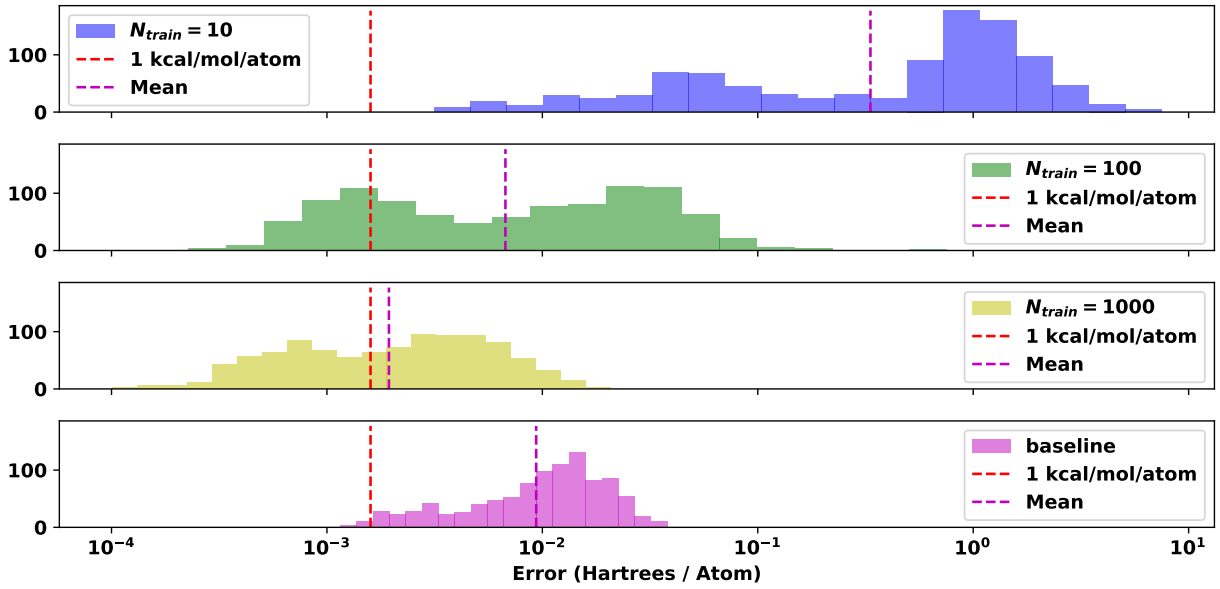


Figure 5.2: Distributions of test set errors in Total Energy

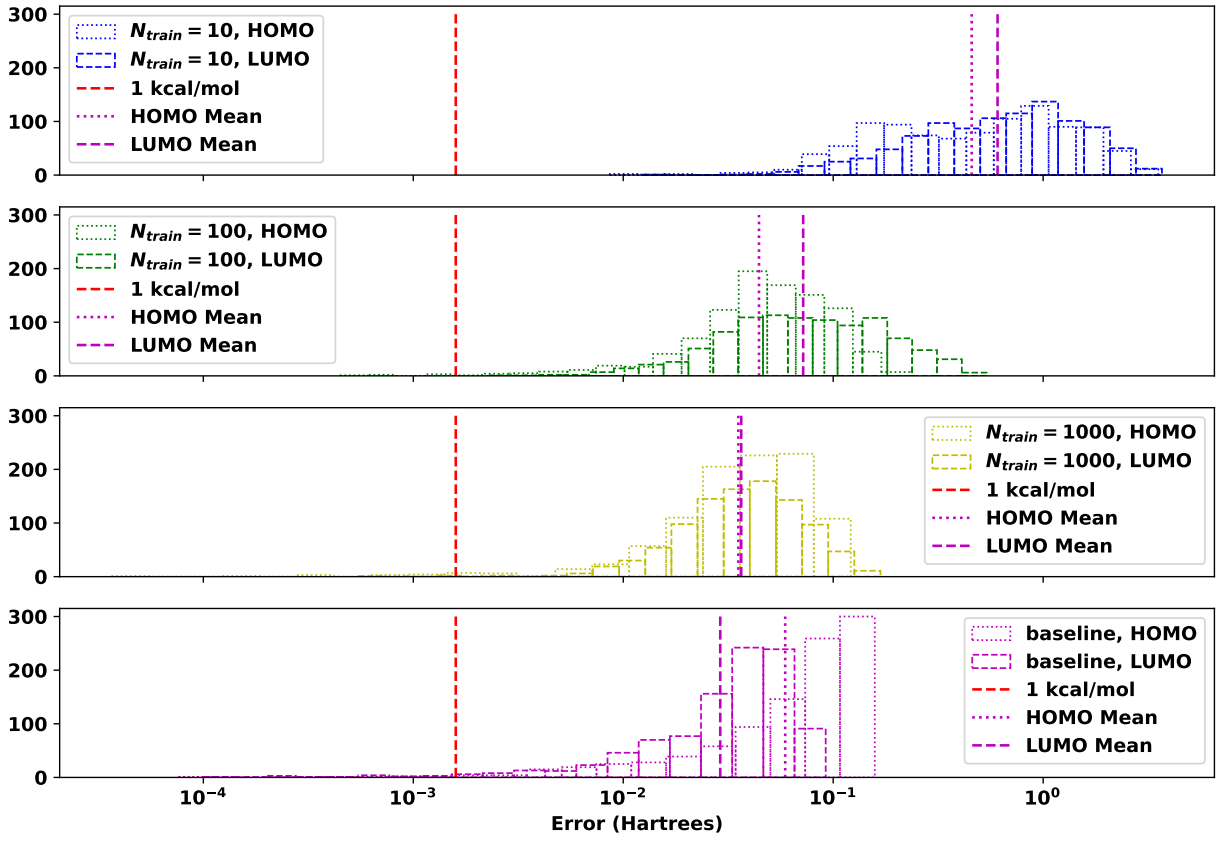


Figure 5.3: Distributions of test set errors in HOMO/LUMO values

# Chapter 6

# Towards Density Learning for Excited States

## 6.1 Introduction

Recent approaches to molecular learning have focused on learning target functions from purely geometric information from molecules [GGG20; Zit+22; Lia+24]. Although this approach has been at the forefront of progress in recent years, little attention has been paid to incorporating important *non-geometric* information in such models such as spin and charge. Spin and charge are *global* properties of a molecule, arising as a result of quantum theory. These properties are an essential feature of many chemical phenomenon such as $d$-orbital spin transfer in catalysis [CN23].

In this section we study density learning for the excited states of molecules. We extend the query-node approach to predict the spin density of the triplet state of molecules. To do so we use *multiple* query-node types to encode the spin/charge/channel of the system. By using a distinct node to distinguish between different molecular states, the model gains the capacity to predict the density from different states.

We evaluate the performance of this technique by training a model on triplet densities

and testing on larger molecules. We use this model and the one from the previous Chapter to calculate the *singlet/triplet gap*, a quantity of interest in quantum chemistry. We show that performance on average is less than for singlet density prediction. We attribute this observation to the asymmetrical nature of the spin-densities, and the inability of invariant pointwise density predictors to learn such densities. We discuss directions for future work.

## 6.2 Theory

We use spin-unrestricted Density Functional Theory at the level of PBE. Here the total density is the sum of the spin-resolved densities:

$$\rho(\mathbf{r}) = \rho^{\alpha}(\mathbf{r}) + \rho^{\beta}(\mathbf{r}) \tag{6.1}$$

where $\alpha$ and $\beta$ denote the *spin-up* and *spin-down* electrons and their corresponding densities.

The Fock matrix now has two channels

$$F^{\alpha}_{\mu\nu} = h_{\mu\nu} + J_{\mu\nu}[\rho] + V^{xc}_{\mu\nu}[\rho^{\alpha}] \tag{6.2}$$

$$F^{\beta}_{\mu\nu} = h_{\mu\nu} + J_{\mu\nu}[\rho] + V^{xc}_{\mu\nu}[\rho^{\beta}] \tag{6.3}$$

As before, this problem is solved iteratively starting from some initial guess for $\rho^{\alpha}$ and $\rho^{\beta}$. We make the same design decisions of (1) periodic boundary conditions and (2) restriction to GGA functionals such as PBE which require the density gradient.

## 6.3 Methods

### 6.3.1 Multi-type Query Node Approach

Here we discuss an extension of the query-node approach to excited states with spin. The essence of our approach is to assign a unique type of query node to every state and spin-channel pair.

Let a finite collection of molecular spin and charge states be denoted as $\{\mathcal{S}_m\}_{m=1}^{M}$. Denote the spin channels of state $m$ as $\mathcal{S}_m^{\alpha}, \mathcal{S}_m^{\beta}$ respectively. Assign an integer index to each of the $2M$ state and spin channel pair.

Let a structure $s$ with $I$ atoms at positions $\mathbf{R}_i \in \mathbb{R}^3$ with atomic numbers $Z_i$ be denoted as $s = \{(\mathbf{R}_i, Z_i)\}_{i=1}^{I}$. As before, let $\rho_s : \mathbb{R}^3 \to \mathbb{R}$ denote the density function associated to the structure $s$, evaluated on a grid of query points $\{\mathbf{r}_j\}_{j=1}^{J}$, with all $\mathbf{r}_j \in \mathbb{R}^3$, and draw an edge between two atoms if they lie within some radius of each other.

To define a query node for state $m$, add the "virtual" atom $(\mathbf{r}_j, Z_{I+2m})$ if the channel is $\alpha$ and $(\mathbf{r}_j, Z_{I+2m+1})$ if the channel is $\beta$, ensuring in all cases that the "virtual" atomic number has not been previously used. Then, we construct radial graphs and union over all structures and states to complete the dataset.

### 6.3.2 Triplet Density Data

We create a new dataset of the triplet densities of candidate molecules. We use the same split design/data reduction strategies for our dataset of triplet densities as in the previous Chapter.

DFT calculations for the triplet state of our candidate molecules were more difficult to converge than the singlet states. Only $N_{train} = 837$ and $N_{test} = 794$ of the structures in `train-1000` and `test-1000` splits of the singlet data in Chapter 2 converged.

Table 6.1: Grid sizes of triplet dataset. Number of density samples is twice grid size.

| | $N_{\text{structures}}$ | $N_{\text{atoms}}$ min/med/max | $N_{\text{grid}}^{KE=200}$ | $N_{\text{grid}}^{KE=100}$ | $N_{\text{grid}}^{KE=100+cut}$ |
|---|---|---|---|---|---|
| Train-small | 10 | 3/4/6 | 42M | 15M | 650k |
| Train-medium | 100 | 3/8/9 | 530M | 190M | 8.5M |
| Train-large | 837 | 3/10/11 | 5.1B | 1.8B | 95M |
| Test | 794 | 17/19/27 | 6.9B | 2.5B | - |

### 6.3.3 Further Quality Checks on Triplet Density Data

We furthermore apply two quality checks to select "well-converged" runs. First we drop candidates with small HOMO/LUMO gap $\leq 0.01$ Hartree. Second, using the default feature in `PySCF` of running an extra cycle after convergence without optimization data (DIIS), we drop candidates with energy deltas $|\Delta E| > 10^{-6}$. This checks if the convergence still holds after dropping information from previous iterates.

The number of structures dropped was 1495, of which 1043 were dropped because of small HOMO/LUMO gaps, and 944 of which were dropped because of high terminal energy delta [1].

To assess an appropriate KE cutoff and density cutoff, we conduct an analysis in the Appendix.

### 6.3.4 Triplet Dataset Statistics

We report sample sizes after applying these strategies in Table 6.1.

### 6.3.5 Training

#### Model

We use the same model and training strategies as in Chapter 2.

[1] Some structures did not meet either criterion.

### 6.3.6 Evaluation

We conduct the same evaluations as before, as well and an additional evaluation involving estimating the *singlet-triplet* gap

$$\Delta E_{st} = E_{\text{singlet}} - E_{\text{triplet}} \tag{6.4}$$

We use the $N = 1000$ singlet model of the previous Chapter to estimate singlet energies.

## 6.4 Results

### 6.4.1 Test Set Results: Energy and Density

We show results for triplet density predictions in Figures 6.1 and 6.2 for errors in total energy, and HOMO/LUMO energies respectively.

As in the previous chapter, we show results for each of the three models trained with different number of structures in each figure.

Overall the performance is less than for the singlet densities. As before we observe all metrics to improve with more training data, however at a slower rate than before. The errors for the Total Energy are shown in Figure 6.1. We see that the best model achieves slightly better than $10^{-2}$ Hartrees/Atom on average. Chemical accuracy (red-dashed line) is missed by about half an order of magnitude. In Figure 6.2 the HOMO/LUMO errors are also higher, and only in a few cases are highly accurate.

### 6.4.2 Test Set Results: Singlet-Triplet Gap

In Figure 6.3, we show results for the singlet-triplet gap estimation, using the singlet model from the previous Chapter and the triplet model from this section. For this metric we target a level of accuracy between $0.1 - 0.5eV$, which is comparable to the level of accuracy in experiments. We find that performance on average misses this target by about one order

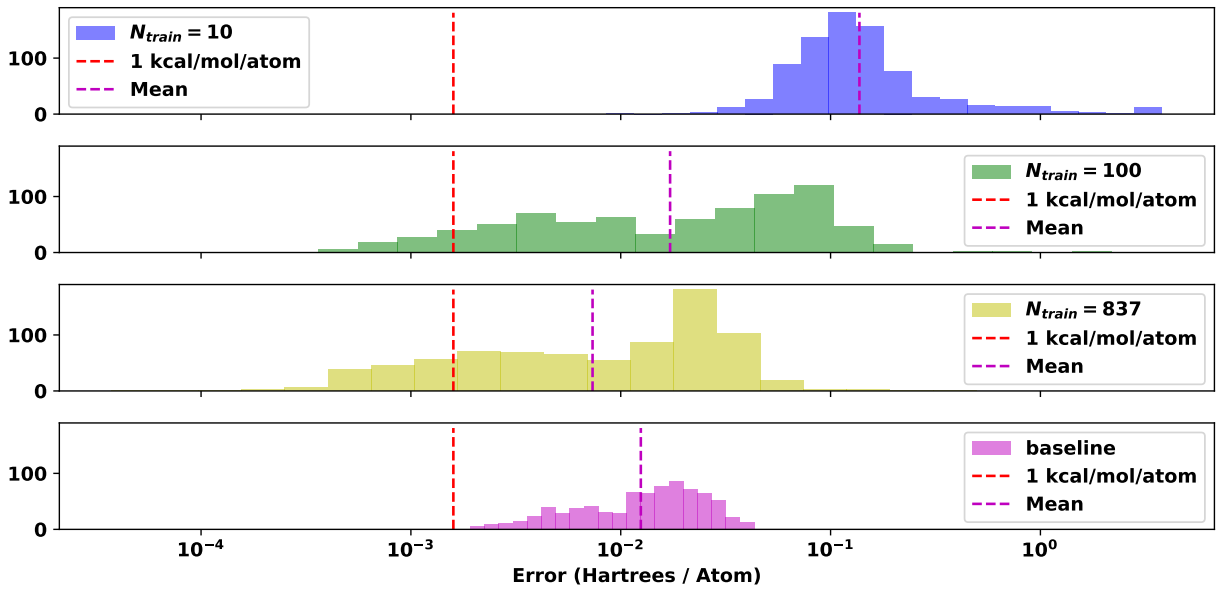


Figure 6.1: Distribution of test errors in total energy of triplet molecules

of magnitude. Note since that this gap is an *intensive* property, it is customary to not report metrics on a per atom basis. Thus errors are approximately one magnitude greater (for molecules with 10 atoms) than in Figure 6.1. We observe the errors tend to distribute in approximately two modes, where the right mode degrades average performance. Further inspection of the structures in this mode may reveal deficiencies in the training data and suggest further improvements.

### 6.4.3 Discussion

In this section we discuss reasons for poor performance on triplet data.

First we observe that a spin density may have lower symmetry than the underlying molecule to which it belongs. For example, in Figure 6.4 we show the $\beta$ densities for singlet and triplet methane. The $\beta$ density for the singlet state is simply half of the total density. The molecule has a tetrahedral symmetry, and so does the singlet density, but the triplet density does not.

We attribute one reason for degraded performance to the breaking of symmetry in the spin densities. Invariant density predictors cannot distinguish between geometrically identical

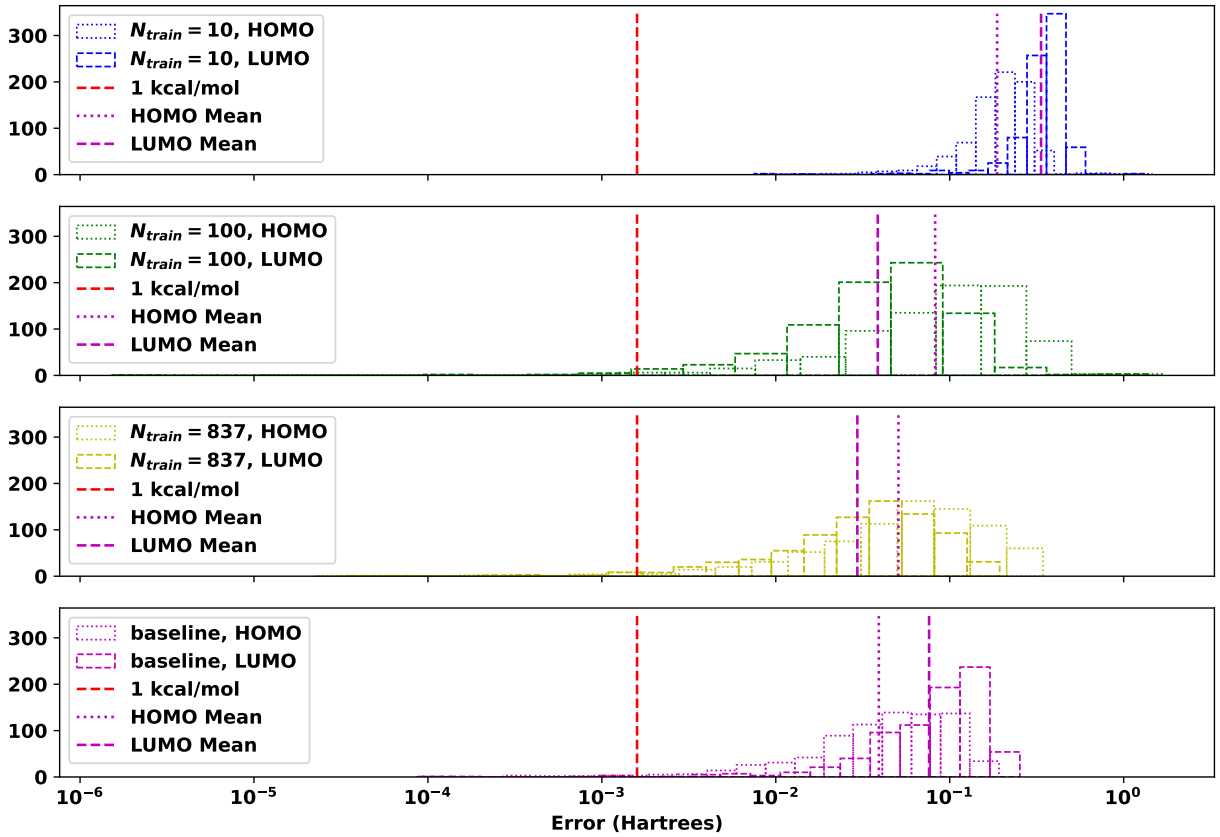


Figure 6.2: Distributions of test errors in HOMO/LUMO energies of triplet molecules

points on the molecule, so errors will be induced if the targets differ.

One possible resolution is to attempt to reduce the symmetry of the molecule. One method of doing so is randomly perturbing atomic positions, which breaks perfectly symmetrical arrangements to some degree of error. Another is to modify the input geometric graphs in some way, for instance by the addition of "virtual" node at some convenient location. Other symmetry-breaking strategies are discussed in [Xie24].

## 6.5 Conclusion

We trained *triplet* density models on small molecules. We used this model and the singlet density model from Chapter 2 to estimate the singlet-triplet gap. We found that performance did not achieve chemical accuracy, missing this target by at least of an order of magnitude of

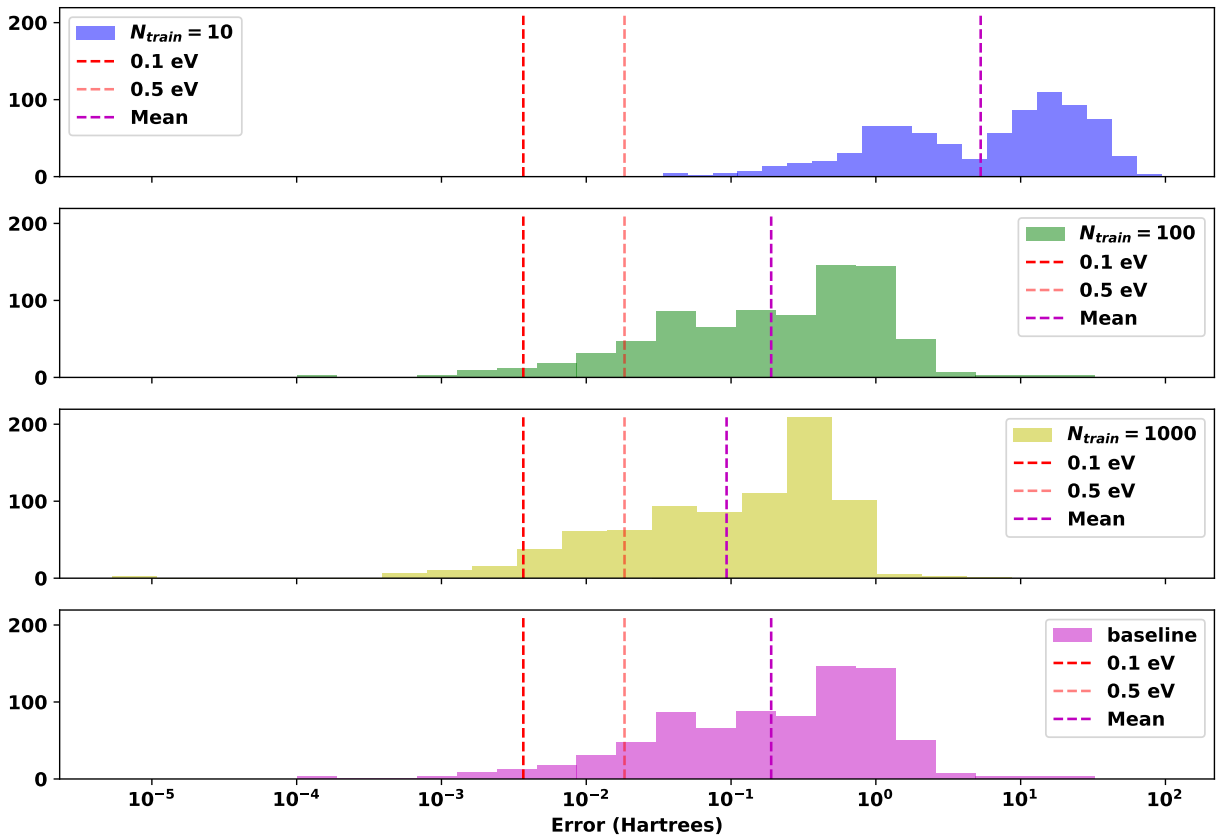


Figure 6.3: Distributions of test set errors in singlet/triplet gap estimations

Hartree. We tested on larger molecules than in the training set to reflect real-world practice. We discuss the problem of asymmetric targets for equivariant models. Our results show that equivariant models are not well-suited for triplet density learning, and that other approaches may be needed.

(a) Methane singlet ($\beta$ channel)

(b) Methane triplet ($\beta$ channel)

Figure 6.4: Reference density isosurfaces ($\rho = 0.8$) for methane singlet and triplet states

# Chapter 7

# Conclusion

In this thesis we developed a machine-learning method to reduce the amount of computation in an algorithm for electronic structure. Specifically we developed a density-based method to speed up Density Functional Theory, a prominent approach to the Quantum Many-Body Problem.

On materials and molecules we showed that a function may be learned to approximate the density to high accuracy, leading to a reduction in the number of iterations, sometimes giving highly-accurate results in a single iteration.

We show a novel capability of predicting the density for excited states of molecules. In particular we demonstrate prediction of the singlet-triplet gap, but at a level of accuracy not yet ready for practical use. We argue that this low accuracy is related to the limitations of equivariant models.

This work is a step forward towards practical use of machine-learning models in quantum chemistry, and moves us further towards the goal of accelerating scientific discovery to help address the problems of society in energy, industry, agriculture, and other important areas.

# Appendix A

# Additional Details

## A.1 Code and Data Availability

Our code and data may be found at: https://github.com/ppope/rho-learn.

## A.2 Numerical Assessment of Density Initialization Error

In this section we assess the error due to our density initialization method. Our implementation involves numerical methods which are accurate to a finite degree. It is important to assess this numerical error and ensure that it is less than the expected learning error. Otherwise, the assessment of learning performance may become unclear.

We perform a numerical study of initialization method (`rho_to_dm`) on reference molecules from QM7 across four spin/charge states: singlet, doublet cation, doublet anion, and triplet. Starting from converged densities, for each molecule we (1) construct the spatial density $\rho$ and then (2) re-construct the density matrix P from $\rho$, using the procedure discussed in the main text. We compute the errors induced by this process in the usual metrics of Total Energy, and HOMO/LUMO values. We show the results in Figure A.1.

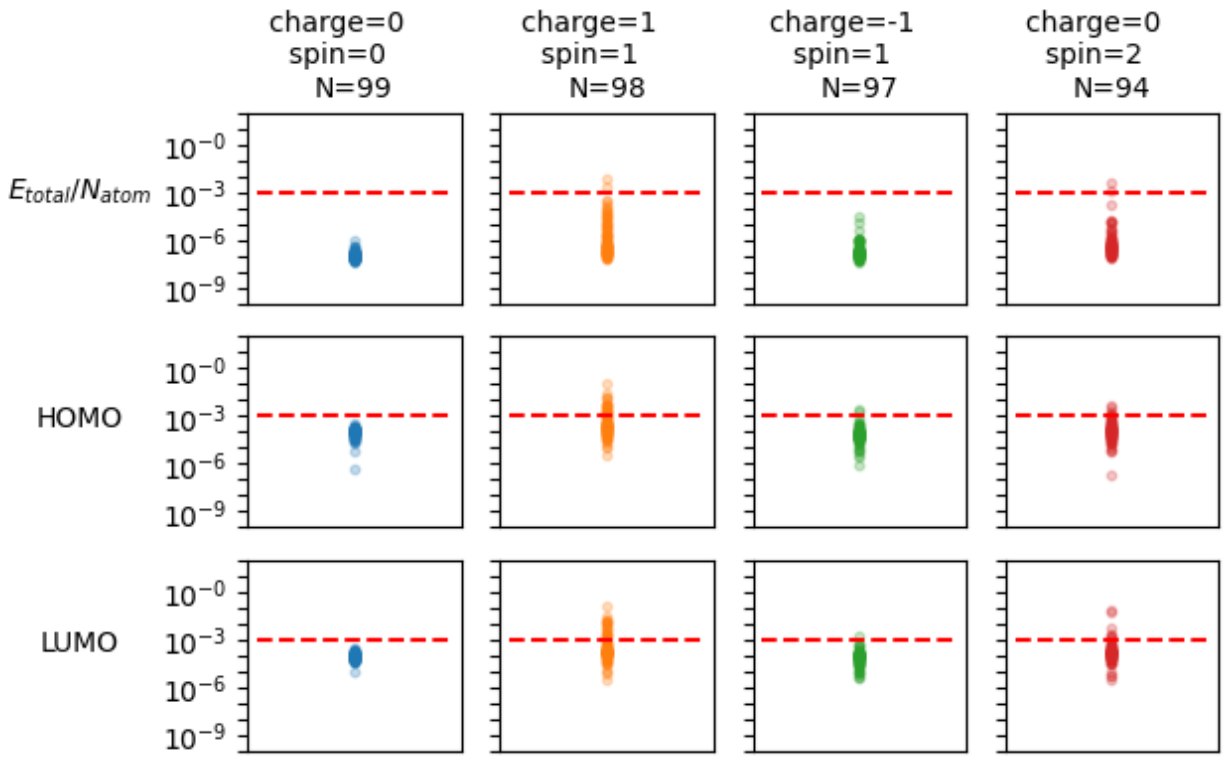


Figure A.1: Errors induced by our `rho_to_dm` initialization procedure on QM7 reference molecules and excited states.

We see that a number of triplet and doublet cations have error near or even above $10^{-3}$ Hartrees. This level of error is problematic because it leaves little room for generalization error, which is non-zero in practice.

## A.3 Exclusion Criteria for Molecules

Using the technique of the previous Section, we can assess values for the data quality checks we imposed: the HOMO/LUMO gap and the terminal energy delta obtained in an extra cycle of DFT to check convergence without previous iterates in the optimization. We exclude molecules with small HOMO/LUMO gap, which are known to be computationally unstable. We exclude molecules with a high terminal energy delta, which are questionable because the convergence depends on optimization data.

We target a minimum accuracy of $10^{-3}$ Hartrees in the Total Energy. Choosing a mini-

mum HOMO/LUMO gap of 0.01 and terminal energy delta of $10^{-6}$ meets the target accuracy. We show the errors from our procedure after filtering out molecules not meeting the criteria in Figure A.2.

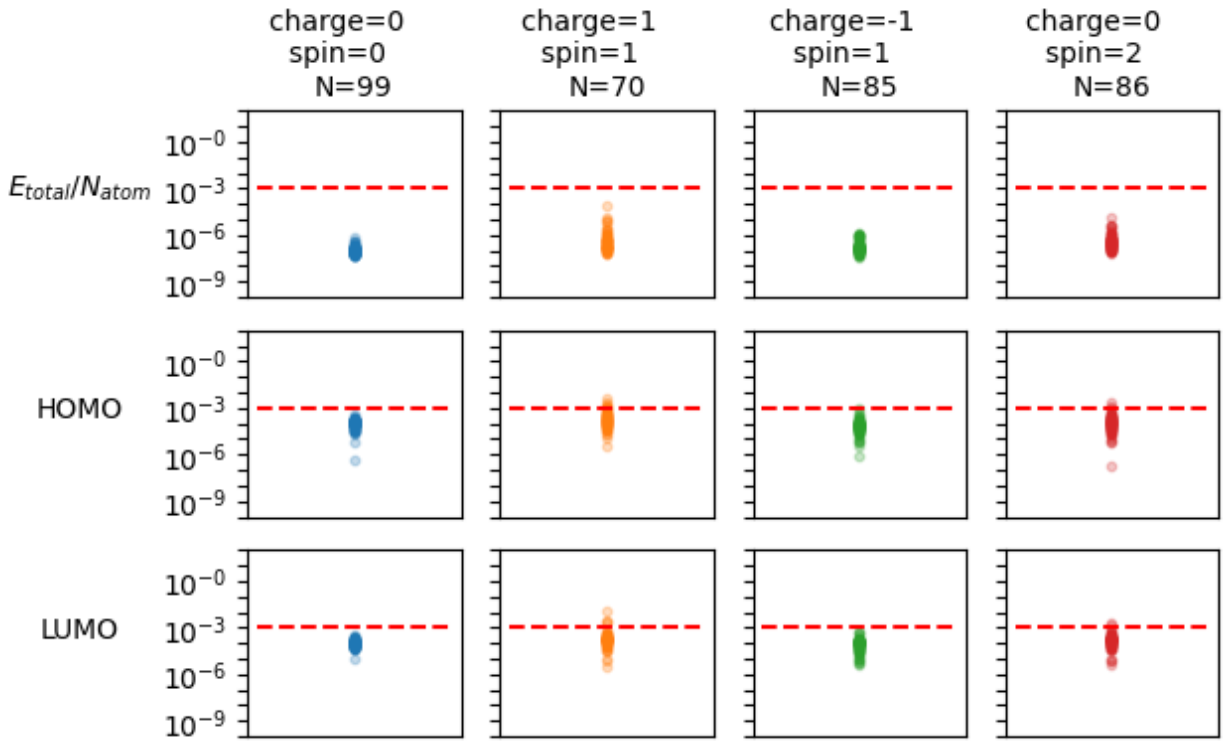


Figure A.2: Errors induced by our `rho_to_dm` procedure after applying the exclusion criteria.

We see that all Total Energy errors are below $10^{-3}$ for all spin/charge states. Singlet errors are well-below this threshold, and errors for doublet anion/cation and triplet states are somewhat higher, but still below the threshold.

For the HOMO/LUMO energies, we also note that the errors tend to be higher near $10^{-3}$. This suggests more stringent quality cutoffs may be necessary to achieve higher HOMO/LUMO accuracies, as the expense of excluding more candidate molecules, or higher accuracy methods implemented.

## A.4 Cutoff Strategies for Reducing Data

Several parameters were used for processing train/test data. We discuss these cutoffs and their justification in more detail here.

### A.4.1 Kinetic Energy Cutoff

One common data reduction strategy for periodic DFT calculations with plane-wave basis, is to reduce the resolution of the computational grid. The kinetic energy (KE) cutoff $E_{cut}$ is a natural parameter to do so, defined $|G|^2 <= 2E_{cut}$ where $G$ is a reciprocal space vector.

### A.4.2 Atomic Distance Cutoff

Given a grid point $\mathbf{r}$, and atomic coordinates $\mathcal{R} = \{\mathbf{R}_i\}_{i=1}^{I}$ define the cutoff distance as the minimum distance to any atom:

$$d(\mathbf{r}) = \min_{\mathbf{R}\in\mathcal{R}} \|\mathbf{r} - \mathbf{R}\| \tag{A.1}$$

Given a reference density $\rho$, the cutoff density at cutoff $c$ is the density

$$\rho_c(\mathbf{r}) = \begin{cases} \rho(\mathbf{r}) & \text{if } d(\mathbf{r}) \leq c, \\ 0 & \text{else} \end{cases} \tag{A.2}$$

We show an example map of this distance for in QM7 (molecule number 991) in Figure A.3. On the left is shown a slice of the charge density, where atoms are denoted with red stars. The right is a contour plot of atomic distances for this molecule in Bohrs, and illustrates the behavior of the cutoff density at different cutofss

One disadvantage of this distance is that it must be calculated for all points in grid. For

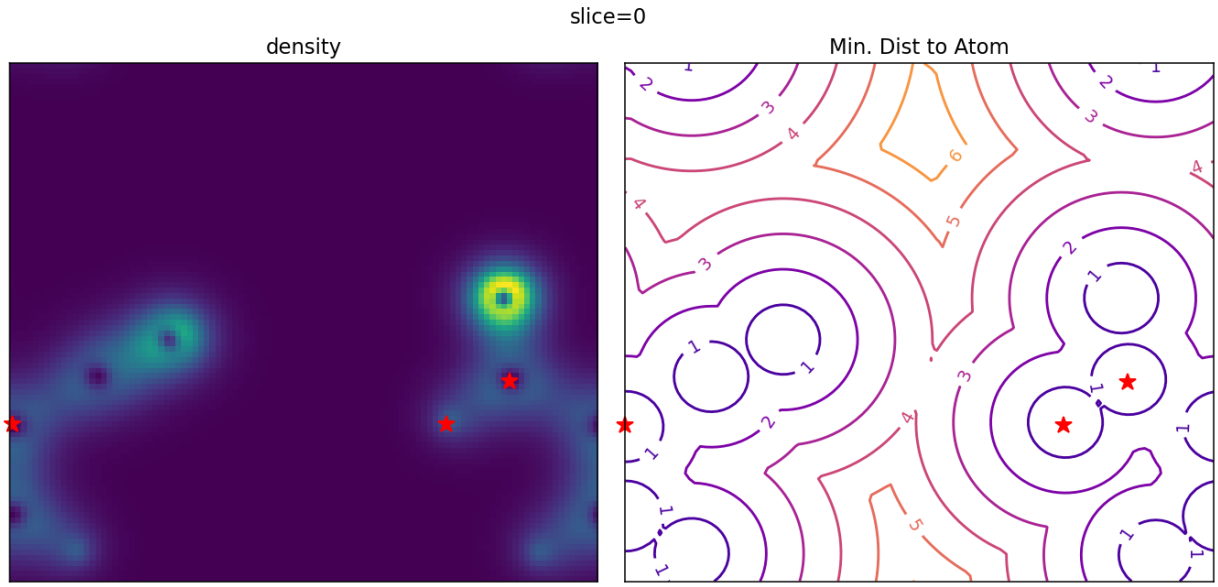


Figure A.3: Illustration of density atomic distance cutoff on an example molecule. Red stars indicate atomic centers.

large grids this may be slow. We discuss a faster alternative in the next section.

### A.4.3 Density Magnitude Cutoff

One simple alternative to the cutoff method of the previous section is to define the cutoff in terms of the magnitude of the density.

More formally, given a reference density $\rho$, the cutoff density at magnitude cutoff $c$ is the density

$$\rho_c(\mathbf{r}) = \begin{cases} \rho(\mathbf{r}) & \text{if } \rho(\mathbf{r}) \geq c, \\ 0 & \text{else} \end{cases} \tag{A.3}$$

This has the advantage of only requiring $N_{grid}$ comparisons, instead of the $N_{grid} \times N_{atoms}$ as above.

The disadvantange of this method is that it requires knowing the density magnitudes

prior to the cutoff. For reducing data requirements, this is only suitable at train time, where the reference densities are available. The previous method may be used at test time.

### A.4.4 Cutoff Study

We conduct a numerical experiment to assess what cutoff levels are appropriate to a desired level of accuracy.

On a collection of reference densities we apply the cutoffs at various levels and compute the accuracy in several metrics obtained after applying the cutoff to the density. Note we perform this analysis on *reference* densities to evaluate the best error obtainable by any density (predicted or not) with the given cutoff. We then use these cutoffs to reduce our training and inferences costs.

In Figures A.4 and A.5 we show the results for the KE cutoff and density distance cutoff respectively. The red dashed line indicates $10^{-3}$ Hartrees of error, which is our target accuracy level.

We observe that higher cutoffs are required to achieve that level of accuracy for excited states.

To allow for additional error from generalization, we choose conservatively: KE-cutoff = 100 and density-cutoff = 5 Bohrs.

## A.5 Training Implementation

One implementation detail is that we found very useful in training is to create geometric graphs "on-the-fly. Since our experiments often involving changing datasets, we find this helpful for avoiding large storage and preprocessing costs with creating graphs ahead-of-time, e.g. with LMDB files, at a negligible cost of increasing the time to create one batch.

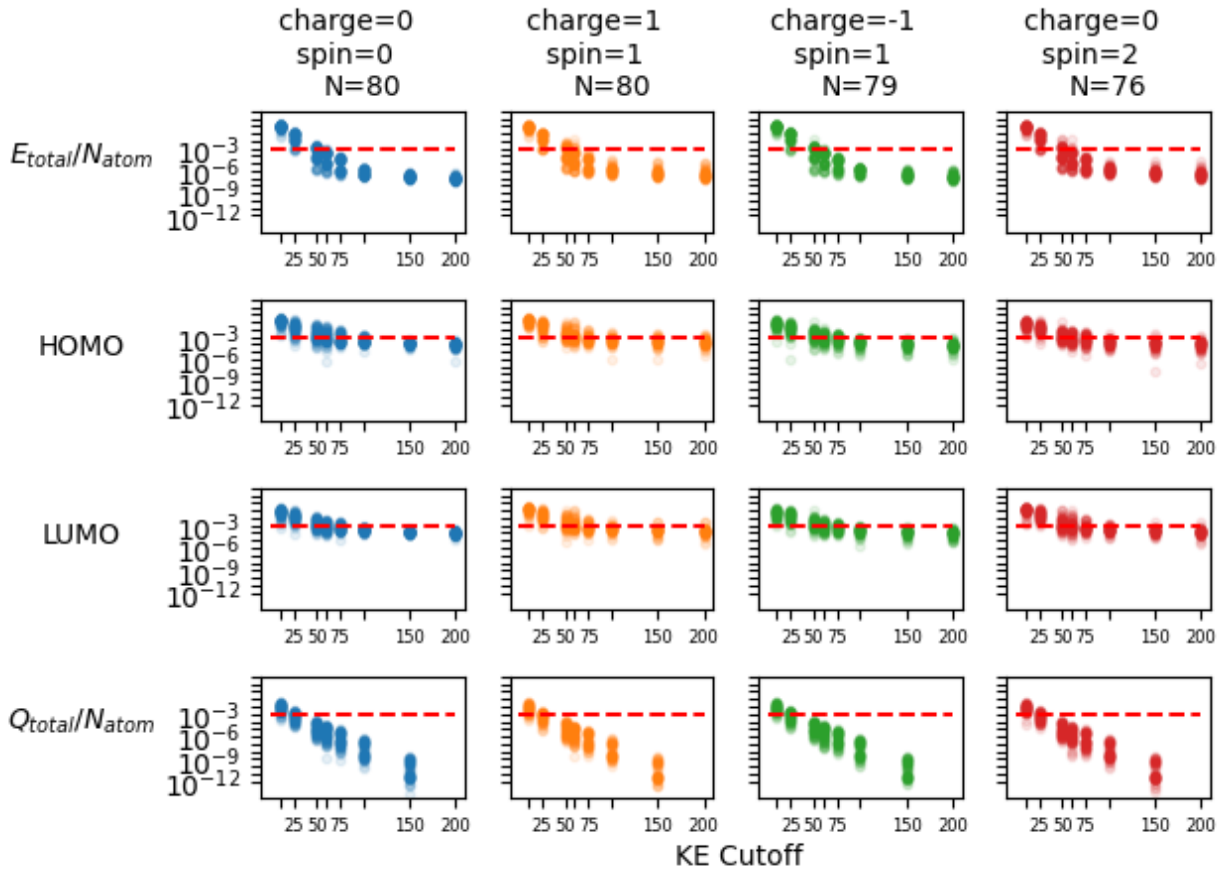


Figure A.4: Assessment of KE cutoff on reference molecules across different states and metrics. Y-axis is error, X-axis is cutoff.

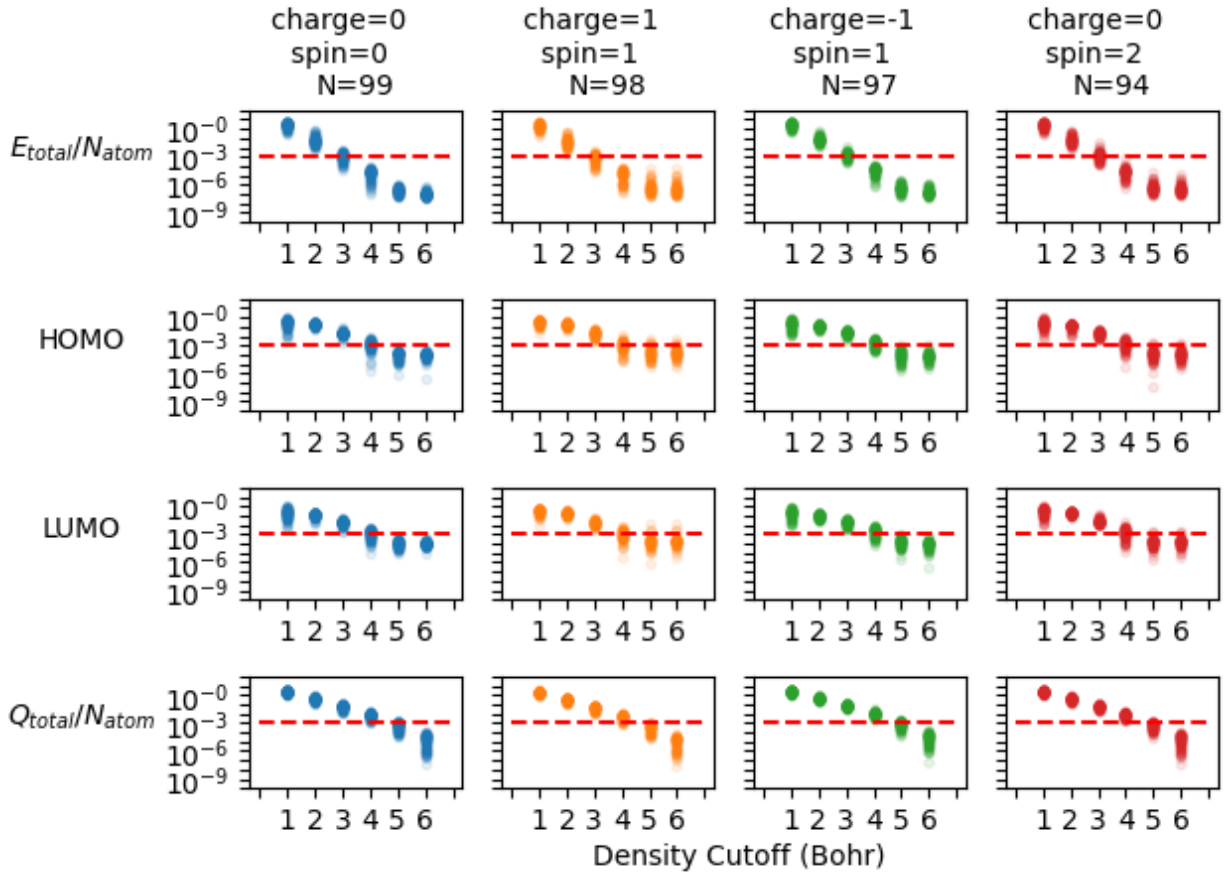


Figure A.5: Assessment of density distane cutoff on reference molecules across different states and metrics. Y-axis is error, X-axis is cutoff.

# Bibliography


[Bat+18] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[BM12] David R Bowler and Tsuyoshi Miyazaki. "Methods in electronic structure calculations". In: *Reports on Progress in Physics* 75.3 (2012), p. 036503.

[Bro+17] Felix Brockherde et al. "Bypassing the Kohn-Sham equations with machine learning". en. In: *Nature Communications* 8.1 (Oct. 2017), p. 872. ISSN: 2041-1723. DOI: 10.1038/s41467-017-00839-3. URL: https://www.nature.com/articles/s41467-017-00839-3 (visited on 08/22/2022).

[CN23] Ang Cao and Jens K. Nørskov. "Spin Effects in Chemisorption and Catalysis". In: *ACS Catalysis* 13.6 (Mar. 2023). Publisher: American Chemical Society, pp. 3456–3462. DOI: 10.1021/acscatal.2c06319. URL: https://doi.org/10.1021/acscatal.2c06319 (visited on 06/26/2025).

[Cha+21] Lowik Chanussot et al. "Open Catalyst 2020 (OC20) Dataset and Community Challenges". In: *ACS Catalysis* 11.10 (May 2021). Publisher: American Chemical Society, pp. 6059–6072. DOI: 10.1021/acscatal.0c04525. URL: https://doi.org/10.1021/acscatal.0c04525 (visited on 01/12/2022).

[Che24] FAIR Chemistry. *FAIR Chemistry - EquiformerV2 config*. 2024. URL: https://github.com/facebookresearch/fairchem/blob/fairchem_core-1.0.0/configs/s2ef/all/equiformer_v2/equiformer_v2_N%408_L%404_M%402_31M.yml (visited on 06/06/2025).

[Dir39] P. A. M. Dirac. "A new notation for quantum mechanics". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3 (1939), pp. 416–418. DOI: 10.1017/S0305004100021162.

[Duv+23] Alexandre Duval et al. "A hitchhiker's guide to geometric gnns for 3d atomic systems". In: *arXiv preprint arXiv:2312.07511* (2023).

[Esc96] Helmut Eschrig. *The fundamentals of density functional theory*. Vol. 32. Springer, 1996.

[GBG21] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. "Gemnet: Universal directional graph neural networks for molecules". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 6790–6802.

[GGG20] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. "Directional message passing for molecular graphs". In: *arXiv preprint arXiv:2003.03123* (2020).

[Gas+22] Johannes Gasteiger et al. *GemNet-OC: Developing Graph Neural Networks for Large and Diverse Molecular Simulation Datasets*. 2022. arXiv: 2204.02782 [cs.LG].

[GS22] Mario Geiger and Tess Smidt. *e3nn: Euclidean Neural Networks*. 2022. DOI: 10.48550/ARXIV.2207.09453. URL: https://arxiv.org/abs/2207.09453.

[Gia+17] P. Giannozzi et al. "Advanced capabilities for materials modelling with Quantum ESPRESSO". en. In: *Journal of Physics: Condensed Matter* 29.46 (Oct. 2017). Publisher: IOP Publishing, p. 465901. ISSN: 0953-8984. DOI: 10.1088/1361-648X/aa8f79. URL: https://doi.org/10.1088/1361-648x/aa8f79 (visited on 06/10/2022).

[Gia+09] Paolo Giannozzi et al. "QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials". en. In: *Journal of Physics: Condensed Matter* 21.39 (Sept. 2009), p. 395502. ISSN: 0953-8984, 1361-648X. DOI: 10.1088/0953-8984/21/39/395502. URL: https://iopscience.iop.org/article/10.1088/0953-8984/21/39/395502 (visited on 06/08/2022).

[Gil+17] Justin Gilmer et al. "Neural Message Passing for Quantum Chemistry". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1263–1272. URL: https://proceedings.mlr.press/v70/gilmer17a.html.

[Goe99] Stefan Goedecker. "Linear scaling electronic structure methods". In: *Reviews of Modern Physics* 71.4 (1999), p. 1085.

[Gon+19] Sheng Gong et al. "Predicting charge density distribution of materials using a local-environment-based graph convolutional network". In: *Physical Review B* 100.18 (Nov. 2019). Publisher: American Physical Society, p. 184103. DOI: 10.1103/PhysRevB.100.184103. URL: https://link.aps.org/doi/10.1103/PhysRevB.100.184103 (visited on 02/08/2022).

[Gri+19] Andrea Grisafi et al. "Transferable Machine-Learning Model of the Electron Density". In: *ACS Central Science* 5.1 (Jan. 2019). Publisher: American Chemical Society, pp. 57–64. ISSN: 2374-7943. DOI: 10.1021/acscentsci.8b00551. URL: https://doi.org/10.1021/acscentsci.8b00551 (visited on 05/17/2023).

[Gri+23] Andrea Grisafi et al. "Electronic-Structure Properties from Atom-Centered Predictions of the Electron Density". In: *Journal of Chemical Theory and Computation* 19.14 (July 2023). Publisher: American Chemical Society, pp. 4451–4460. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.2c00850. URL: https://doi.org/10.1021/acs.jctc.2c00850 (visited on 06/20/2025).

[HK64] P. Hohenberg and W. Kohn. "Inhomogeneous Electron Gas". In: *Phys. Rev.* 136 (3B Nov. 1964), B864–B871. DOI: 10.1103/PhysRev.136.B864. URL: https://link.aps.org/doi/10.1103/PhysRev.136.B864.

[Hub+20] Sebastiaan P Huber et al. "AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance". In: *Scientific data* 7.1 (2020), p. 300.

[Hub+21] Sebastiaan P Huber et al. "Common workflows for computing material properties using different quantum engines". In: *npj Computational Materials* 7.1 (2021), p. 136.

[Jai+13] Anubhav Jain et al. "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation". In: *APL materials* 1.1 (2013), p. 011002.

[JB20] Peter Bjørn Jørgensen and Arghya Bhowmik. *DeepDFT: Neural Message Passing Network for Accurate Charge Density Prediction*. 2020. arXiv: 2011.03346 [physics.comp-ph].

[Jos+23] Chaitanya K. Joshi et al. *On the Expressive Power of Geometric Graph Neural Networks*. 2023. arXiv: 2301.09308 [cs.LG].

[KW16] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[Koh96] W. Kohn. "Density Functional and Density Matrix Method Scaling Linearly with the Number of Atoms". In: *Phys. Rev. Lett.* 76 (17 Apr. 1996), pp. 3168–3171. DOI: 10.1103/PhysRevLett.76.3168. URL: https://link.aps.org/doi/10.1103/PhysRevLett.76.3168.

[KS65] W. Kohn and L. J. Sham. "Self-Consistent Equations Including Exchange and Correlation Effects". In: *Phys. Rev.* 140 (4A Nov. 1965), A1133–A1138. DOI: 10.1103/PhysRev.140.A1133. URL: https://link.aps.org/doi/10.1103/PhysRev.140.A1133.

[Lar+17] Ask Hjorth Larsen et al. "The atomic simulation environment—a Python library for working with atoms". In: *Journal of Physics: Condensed Matter* 29.27 (2017), p. 273002. URL: http://stacks.iop.org/0953-8984/29/i=27/a=273002.

[Leh19] Susi Lehtola. "Assessment of Initial Guesses for Self-Consistent Field Calculations. Superposition of Atomic Potentials: Simple yet Efficient". In: *Journal of Chemical Theory and Computation* 15.3 (Mar. 2019). Publisher: American Chemical Society, pp. 1593–1604. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.8b01089. URL: https://doi.org/10.1021/acs.jctc.8b01089 (visited on 06/19/2025).

[LS85] F. S. Levin and J. Shertzer. "Finite-element solution of the Schrödinger equation for the helium ground state". In: *Phys. Rev. A* 32 (6 Dec. 1985), pp. 3285–3290. DOI: 10.1103/PhysRevA.32.3285. URL: https://link.aps.org/doi/10.1103/PhysRevA.32.3285.

[Lev79] Mel Levy. "Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the v-representability problem". In: *Proceedings of the National Academy of Sciences* 76.12 (1979), pp. 6062–6065. DOI: 10.1073/pnas.76.12.6062. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.76.12.6062. URL: https://www.pnas.org/doi/abs/10.1073/pnas.76.12.6062.

[Lia+24] Yi-Lun Liao et al. *EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations*. 2024. arXiv: 2306.12059 [cs.LG]. URL: https://arxiv.org/abs/2306.12059.

[Lie83] Elliott H. Lieb. "Density functionals for coulomb systems". In: *International Journal of Quantum Chemistry* 24.3 (1983), pp. 243–277. DOI: https://doi.org/10.1002/qua.560240302. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.560240302. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qua.560240302.

[LL19] Lin Lin and Jianfeng Lu. *A mathematical introduction to electronic structure theory*. SIAM, 2019.

[LLY19] Lin Lin, Jianfeng Lu, and Lexing Ying. "Numerical methods for Kohn–Sham density functional theory". en. In: *Acta Numerica* 28 (May 2019). Publisher: Cambridge University Press, pp. 405–539. ISSN: 0962-4929, 1474-0508. DOI: 10.1017/S0962492919000047. (Visited on 03/03/2022).

[Mar20] Richard M Martin. *Electronic structure: basic theory and practical methods*. Cambridge university press, 2020.

[Moh+14] Stephan Mohr et al. "Daubechies wavelets for linear scaling density functional theory". In: *The Journal of chemical physics* 140.20 (2014).

[Nak+20] Ayako Nakata et al. "Large scale and linear scaling DFT with the CONQUEST code". In: *The Journal of chemical physics* 152.16 (2020).

[Ong+13] Shyue Ping Ong et al. "Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis". In: *Computational Materials Science* 68 (2013), pp. 314–319.

[PBE96] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. "Generalized Gradient Approximation Made Simple". In: *Phys. Rev. Lett.* 77 (18 Oct. 1996), pp. 3865–3868. DOI: 10.1103/PhysRevLett.77.3865. URL: https://link.aps.org/doi/10.1103/PhysRevLett.77.3865.

[Piz+16] Giovanni Pizzi et al. "AiiDA: automated interactive infrastructure and database for computational science". en. In: *Computational Materials Science* 111 (Jan. 2016), pp. 218–230. ISSN: 0927-0256. DOI: 10.1016/j.commatsci.2015.09.013. URL: https://www.sciencedirect.com/science/article/pii/S0927025615005820 (visited on 07/27/2022).

[PJ23] Phillip Pope and David Jacobs. "Towards Combinatorial Generalization for Catalysts: A Kohn-Sham Charge-Density Approach". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=JOHp5SmckS.

[Pop21] John Pople. "The nobel prize in chemistry 1998 explore press release". In: *Nobelprize. org* (2021).

[Pra+18] Gianluca Prandini et al. "Precision and efficiency in solid-state pseudopotential calculations". In: *npj Computational Materials* 4.1 (2018), p. 72.

[PK05] Emil Prodan and Walter Kohn. "Nearsightedness of electronic matter". In: *Proceedings of the National Academy of Sciences* 102.33 (2005), pp. 11635–11638.

[Rac+22] Joshua A. Rackers et al. *Cracking the Quantum Scaling Limit with Machine Learned Electron Densities*. 2022. arXiv: 2201.03726 [physics.chem-ph].

[Ram+14] Raghunathan Ramakrishnan et al. "Quantum chemistry structures and properties of 134 kilo molecules". In: *Scientific Data* 1 (2014).

[Rud+12] Lars Ruddigkeit et al. "Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17". In: *Journal of Chemical Information and Modeling* 52.11 (Nov. 2012). Publisher: American Chemical Society, pp. 2864–2875. ISSN: 1549-9596. DOI: 10.1021/ci300415d. URL: https://doi.org/10.1021/ci300415d (visited on 06/20/2025).

[SCS10] Yousef Saad, James R. Chelikowsky, and Suzanne M. Shontz. "Numerical Methods for Electronic Structure Calculations of Materials". en. In: *SIAM Review* 52.1 (Jan. 2010), pp. 3–54. ISSN: 0036-1445, 1095-7200. DOI: 10.1137/060651653. URL: http://epubs.siam.org/doi/10.1137/060651653 (visited on 04/09/2022).

[Sca+08] Franco Scarselli et al. "The graph neural network model". In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

[Sch+18] K. T. Schütt et al. "SchNet – A deep learning architecture for molecules and materials". In: *The Journal of Chemical Physics* 148.24 (June 2018). Publisher: American Institute of Physics, p. 241722. ISSN: 0021-9606. DOI: 10.1063/1.5019779. URL: https://aip.scitation.org/doi/10.1063/1.5019779 (visited on 02/09/2022).

[Sch+19] K. T. Schütt et al. *Unifying machine learning and quantum chemistry – a deep neural network for molecular wavefunctions*. 2019. arXiv: 1906.10033 [physics.chem-ph].

[She22] Jimmy-Xuan Shen. *mp-pyrho*. Version v0.3.0. Oct. 2022. DOI: 10.5281/zenodo.7227448. URL: https://doi.org/10.5281/zenodo.7227448.

[She+21] Jimmy-Xuan Shen et al. *A representation-independent electronic charge density database for crystalline materials*. 2021. eprint: 2107.03540.

[Shu+21] Muhammed Shuaibi et al. "Rotation Invariant Graph Neural Networks using Spin Convolutions". In: *arXiv:2106.09575 [cs]* (June 2021). URL: http://arxiv.org/abs/2106.09575 (visited on 01/06/2022).

[Sun+18] Qiming Sun et al. "PySCF: the Python-based simulations of chemistry framework". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 8.1 (2018), e1340.

[Sun+20] Qiming Sun et al. "Recent developments in the PySCF program package". In: *The Journal of chemical physics* 153.2 (2020).

[Tra+22] Richard Tran et al. "The Open Catalyst 2022 (OC22) Dataset and Challenges for Oxide Electrocatalysts". In: (2022). Publisher: arXiv Version Number: 3. DOI: 10.48550/ARXIV.2206.08917. URL: https://arxiv.org/abs/2206.08917 (visited on 02/01/2024).

[Uhr+21] Martin Uhrin et al. "Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows". In: *Computational Materials Science* 187 (2021), p. 110086.

[Van+06] JH Van Lenthe et al. "Starting SCF calculations by superposition of atomic densities". In: *Journal of computational chemistry* 27.8 (2006), pp. 926–932.

[VMN14] Richard Van Noorden, Brendan Maher, and Regina Nuzzo. "The top 100 papers". In: *Nature News* 514.7524 (2014), p. 550.

[XG18] Tian Xie and Jeffrey C. Grossman. "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties". In: *Phys. Rev. Lett.* 120 (14 Apr. 2018), p. 145301. DOI: 10.1103/PhysRevLett.120.145301. URL: https://link.aps.org/doi/10.1103/PhysRevLett.120.145301.

[Xie24] YuQing Xie. "Equivariant Symmetry Breaking Sets". English. PhD thesis. 2024, p. 59. ISBN: 9798346380528. URL: https://www.proquest.com/dissertations-theses/equivariant-symmetry-breaking-sets/docview/3132851126/se-2.

[YD23] Hong-Zhou Ye and Chenru Duan. *Quantum Chemistry*. 2023. URL: https://ai4science101.github.io/blogs/quantum_chemistry/ (visited on 01/31/2024).

[Yu+20] Victor Wen-zhe Yu et al. "ELSI—An open infrastructure for electronic structure solvers". In: *Computer Physics Communications* 256 (2020), p. 107459.

[Zah+17] Manzil Zaheer et al. "Deep Sets". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

[Zee16] Anthony Zee. *Group theory in a nutshell for physicists*. Princeton University Press, 2016.

[Zep+21] Leonardo Zepeda-Núñez et al. "Deep Density: Circumventing the Kohn-Sham equations via symmetry preserving neural networks". en. In: *Journal of Computational Physics* 443 (Oct. 2021), p. 110523. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2021.110523. URL: https://www.sciencedirect.com/science/article/pii/S0021999121004186 (visited on 01/13/2022).

[Zit+20] C. Lawrence Zitnick et al. "An Introduction to Electrocatalyst Design using Machine Learning for Renewable Energy Storage". In: *arXiv:2010.09435 [cond-mat]* (Oct. 2020). arXiv: 2010.09435. URL: http://arxiv.org/abs/2010.09435 (visited on 01/12/2022).

[Zit+22] C. Lawrence Zitnick et al. "Spherical Channels for Modeling Atomic Interactions". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022. URL: https://openreview.net/forum?id=5Z3GURcqwT.