



Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

ML-DFT: Powering Machine-Learning With Density Functional Theory

PhD Candidate: Eoin Monaghan

Supervisor: Prof. Stefano Sanvito

April 3, 2026

Declaration

I, Eoin Monaghan, hereby declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Eoin Monaghan

Eoin Monaghan

April 3, 2026

Abstract

This thesis investigates machine-learning force field development for atomic systems, progressing from molecular structures through periodic materials towards high-entropy alloy modelling. We develop equivariant graph neural network architectures leveraging density functional theory to predict atomic forces with near-quantum mechanical accuracy whilst achieving substantial computational efficiency.

We establish a rigorous set of data generation methodologies based on the MD17 dataset, and train models which yield force predictions with R^2 -scores exceeding 0.98 for a variety of molecules.

The central contribution establishes fundamental node alignment failures arising from interfacing equivariant distance-metric representations with neural networks. Through group-theoretic analysis, we demonstrate that molecular symmetries generate representational degeneracies due to the introduction of neural networks. We develop quantitative ambiguity metrics combining automorphism group cardinalities with structural complexity, revealing strong correlations with model performance. This transforms symmetry considerations into actionable pre-screening tools, whilst revealing that high-symmetry molecules counter-intuitively impose systematic training difficulties.

Extension to periodic systems reveals fundamental limitations for heterogeneous datasets. We integrate echo state network principles into Ni-Pd alloy systems, yielding exceptional energy predictions, whilst custom interpolation schemes produce force calculations rivalling backpropagation methods. This work establishes theoretical frameworks identifying symmetry-induced training difficulties whilst offering pathways toward improved efficiency and accuracy for high-entropy alloy research.

Keywords: *Machine-Learning, Graph Neural Networks, Message Passing, Density Functional Theory, Machine-Learning Interatomic Potentials*

Acknowledgements

I would like to firstly thank Prof Stefano Sanvito, for designing this project and for his excellent guidance, insight, and kindness. He has cultivated a fantastic group, a great environment, and was always there to help me through my problems. Furthermore, to Prof Iggy McGovern for his advice on delving into the world that is post-graduate academia, and the hot chocolate catchups that followed throughout. I would like to thank Research Ireland for funding this project (Project ID: GOIPG/2021/1332), TCHPC for their computational resources, and to Stefania and Aoife, for always being there to help when needed.

My journey throughout these four years has been anything but dull, and has been aided by a fantastic cast of characters that have helped along the way. I want to sincerely thank everyone that passed through CRANN 3.18 for all the fun times, debates, and stimulating conversations. You made each journey along the Dublin quays worth it every day. Please forgive the collective thanking, for it would be too long to mention everyone by name, and I'm doomed to unintentionally omit someone. There are however a couple of people, without whom, my time as a PhD researcher would be a far cry from the experience I'm glad to share today. In particular I want to thank Declan, for your wisdom, and sofas in Rome. To Paddy, for all the hiking getaways when I really needed it, and when my knees really didn't. To Michael, for all the bizarrely creative ways you approach any topic, and your sheer networking prowess. To Andrew, for always being a classy act, and for showing me new ways to code. To Lion, for all the music and procrastinating coffees. To Luke, for giving me the confidence to delve back into the arts, and for our mutual supports. To Matteo, for all our random gaming, book, and philosophical hot takes. To Laura, Michaelangelo and Stefania darling, for always bringing your fantastic individual flairs that our group so dearly misses. To Markus, Ümit, Annie, Eoghan, and Seán, who never fail to make a room brighter. To Hugo and Rutchapon, who helped in the dying minutes of this work. And lastly, to Urvesh, the oracle, who onboarded me into this realm and helped me every step of the way.

As an aside, I would be remiss to not include a word of thanks to four institutional pillars that helped sustain me throughout these last few years. Namely, Scéal, Honey Truffle, Bear Market, and Tang. The people that work in these buildings did more for my sanity than any online review could ever do justice.

Not to forget my roots, I need to thank my teenage pals Conor and Jamie, whose friendships I hope to hold on to for life. Thank you for all your shenanigans, musical wonder, and for the talents you share, please never stop. To Mart and Simon, for being my first true friends in this new city. To Josh and Caolan for cultivating our little flat into a home, and for all the late night gaming sessions. Thank you to Silvia, Jen, Izzy, Conor, Matthew, Tom, and the many already mentioned, for sustaining my bouldering habits, and continuing to let me feel like it counts as going to the gym. To Anton and Grete, whose chance introduction into my life catapulted me into unimaginable adventures.

You have all had a profound impact on my life, and if I were to start into more adjectives I would risk this section overtaking my thesis in length. However, there are some final people whose praises must be sung, and without whom this thesis would never have been written.

To my beautifully ridiculous boyfriend Rory. Thank you for never doubting my abilities in life. For always looking beyond the surface level, and giving me the confidence to be myself. Your attitude and outlook, no matter what life throws your way, is truly inspirational and drives me to be a better person every time I see you. I never dreamt that I could be so proudly happy with someone. Thank you for being the best goofball a boy could ever want.

I would never have gotten to the position to even contemplate undertaking a PhD if it were not for my family. To my sisters Áine and Maeve, thank you for putting up with the racket. To my mother Irene, and father Sean. You have both worked so incredibly hard to ensure we all had the opportunities to make good lives for ourselves. It could not have been easy, but I hope I can continue to do you proud and repay a lifetime of love and care.

Lastly, I would like to dedicate this thesis to my late grandmother Helen, who drove us to and from school every day, and who lovingly cared for us whilst mum and dad were working to provide for us, more as a third parent than grandparent. You selflessly spent your entire life caring for others, and your singing echoes in my mind anytime I come across home-made buns. They never come close. Thank you.

Contents

1	Introduction	1
1.1	Thesis Outline	3
2	Technical Background	5
2.1	The Quantum Many-Body Problem	5
2.2	Density Functional Theory	7
2.3	Kohn-Sham DFT	8
2.3.1	The Exchange-Correlation Functional	9
2.3.2	DFT as a Data Source for Machine-Learning Force Fields	11
2.4	Machine-Learning	12
2.4.1	Forward Propagation	13
2.4.2	Backpropagation	15
2.5	Powering Machine-Learning with Density Functional Theory	18
2.5.1	The Convergence of ML and DFT	18
2.5.2	The ML Approach to Property Prediction	19
2.5.3	Direct Force Prediction	20
2.5.4	Implications for Computational Physics	20
2.6	Molecules as Graphs	21
2.6.1	Mathematical Foundations of Graphs	22
2.6.2	Extended Definition for Molecular Systems	22
2.6.3	Graph Isomorphism and Molecular Identity	24
2.7	Periodic Systems as Graphs	26
2.7.1	From Molecular to Periodic Graph Theory	26
2.7.2	Unit Cell Construction	27
2.7.3	The Minimum Image Convention	28

2.7.4	Cutoffs, Connectivity and Neighbourhoods	29
2.7.5	Symmetry and Invariance	29
2.7.6	Computational Complexity	30
2.8	Graph Neural Networks and Message Passing	30
2.8.1	Message Computation	31
2.8.2	Aggregation	32
2.8.3	Node Update	32
2.9	Group Theory	33
2.9.1	Group Theory and Symmetry Operations	34
2.9.2	Invariance and Equivariance	34
2.10	Linear Representations and Group Actions	35
2.11	Equivariant Graph Neural Networks	36
2.11.1	Coordinate Equivariance	37
2.11.2	Training Space Reductions	37
2.11.3	Equivariant Message Passing Architecture	38
2.11.4	Aggregation and Update Operations	38

Part 1 - Molecules 41

3 GNN Implementation 42

3.1	Feature Engineering	42
3.1.1	Node Feature Construction	42
3.1.2	Edge Feature Construction via Adjacency Matrix	44
3.2	EMPL Architecture	45
3.2.1	Distance-Based Feature Extraction	45
3.2.2	Message Computation	46
3.2.3	Computational and Memory Complexity	47
3.2.4	Equivariant Update Mechanism	48
3.2.5	Output Layer Formulation	49
3.2.6	Mathematical Properties	50

3.3	Advanced EMPL Implementation Details	52
3.3.1	Numerical Stability Considerations	52
3.3.2	Gradient Clipping	54
4	Molecular Datasets	55
4.1	MD17	55
4.2	Custom Made Dataset	56
4.2.1	Configuration Space Sampling	57
4.2.2	Geometric Perturbation	58
4.2.3	DFT Calculation Methodology	59
5	Results - Molecules	62
5.1	Error and Distribution Analysis	64
5.2	Hyperparameter Testing	65
5.3	Molecular Parity Plots	67
6	Node Alignment Failures	71
6.1	The Isomorphism Problem	72
6.1.1	Group Theory Foundation	73
6.2	Network Incompatibility	73
6.2.1	Example	74
6.3	Molecular Graph Automorphism	76
6.3.1	The Automorphism Group as a Symmetry Quantifier	77
6.3.2	Molecular Automorphism Analysis	77
6.3.3	Network Training Implications	78
6.4	Quantifying Ambiguity	79
6.4.1	The Ambiguity Measure	79
6.4.2	The Ambiguity Score	80
6.5	Application to MD17 Dataset	81
6.6	Correlation Analysis and Theoretical Implications	84

Part 2 - Periodic Systems	87
7 Extension To Periodic Systems	87
7.1 Periodic Physics Foundations	88
7.2 Periodic Boundary Conditions	89
7.3 The Minimum Image Convention	90
7.3.1 Periodic Imaging Example	91
7.3.2 Implementation	91
7.4 Cutoffs and Interaction Truncations	93
7.5 Conclusion	94
8 Model Architecture - Silica	95
8.1 Padding and Masking	95
8.2 Network Integration	98
8.3 Silica Dataset	98
9 Results - Silica	100
9.1 Parity Plot Analysis	103
10 Node Alignment Failures (Continued)	106
10.1 Space Group Symmetries and Crystallographic Automorphisms	107
Part 3 - Towards High-Entropy Alloys and the Energy Surface Interpolation Scheme	111
11 Introduction to High-Entropy Alloys	111
12 Echo State Networks	113
12.1 Mathematical Formulation	116
12.2 Spectral Radius and the Echo State Property	118
12.3 Implementation as Message Passing in GNNs	118
12.4 Ni-Pd as a Binary Foundation	122

13 Force Calculations and Energy Surface Interpolations	124
13.1 Interpolation Methods	127
13.2 K-Nearest Neighbours for Local Sample Selection	129
13.3 Smooth Polynomial Interpolation Methods	131
13.3.1 Local Polynomial Fitting	131
13.3.2 Gaussian Weighted Regression	132
13.4 Finite Difference Calculations	133
14 Model Architecture - Ni-Pd	134
14.1 Ni-Pd Dataset	136
15 Results - Ni-Pd	137
15.1 X-Component Analysis	145
15.2 Y-Component Analysis	146
15.3 Z-Component Analysis	147
Conclusion	151
16 Conclusion	151
16.1 One Year Projection	153
16.2 Five Year Projection	154
A Publications	156

Chapter 1

Introduction

The use of computational methods for the prediction of molecular properties is one of the most challenging problems at the intersection of quantum mechanics, chemistry, and computational physics. Traditional approaches to quantum mechanical methods are entirely theoretically sound but scale poorly with system size, becoming computationally intractable for both large molecular and periodic systems with large cell sizes. This bottleneck is the main motivator in the development of machine-learning approaches, which aim to approximate quantum mechanical calculations at only a fraction of the overall computational cost.

At the heart of this challenge is the fundamental requirement that any predictive model must respect the inherent symmetries that are present within a given physical system. Molecules, for example, exist in three dimensional space and their properties remain either invariant or equivariant under the act of rotation, translation and permutation of identical atoms. The physical principles governing this behaviour can also act as a mathematical convenience as failure to incorporate these symmetries would lead to models requiring much larger training datasets and, at a surface level, could result in physically inconsistent predictions.

However, there exists an overlooked fundamental flaw with regard to the interfacing of equivariant representations in the form of Euclidean distances within discrete computational systems. Whilst the theoretical framework of equivariant Graph Neural Networks (GNNs) elegantly preserves symmetry properties through operations on relative distances and normalised vectors, the practical implementation and usage of these architectures will introduce subtle but critical inconsistencies that fundamentally undermine the guarantees they are intended to provide.

The central problem lies in what we term the node alignment failure. When molecular or crystalline systems are represented as graphs, atoms must be assigned to discrete node indices in a process that cannot be consistently applied in 3D space. The connectivity between these nodes must be determined through distance-based cutoff criteria and neighbourhood selection algorithms. This process creates a critical vulnerability, as if we have two configurations of the same molecule that differ only by a small rotation, under the principle of equivariance, these configurations should be treated as mathematically equivalent by the network. However, if this rotation causes atoms to be assigned to different node indices, the overall connectivity pattern with the adjacency matrix becomes internally inconsistent across equivalent configurations. The severity of this issue is amplified by the sensitivity of modern GNNs, with large models tending to be susceptible to even the most minor of changes in inputs.

Crucially, this inconsistency is not a rare edge case, as is with the two-body molecular graph isomorphism problem, but a systematic feature of metric based graph constructions, which will inherently affect a substantial portion of any reasonably sized dataset. Due to this, networks encounter multiple representations of what should be the same configuration, which will be processed differently through the network. The optimisation of such networks thus becomes an ill-defined objective as the network attempts to reconcile contradictory information about equivalent configurations.

This work addresses this fundamental inconsistency by first developing rigorous ambiguity metrics that quantify the extent of the node alignment problem within a given dataset. By characterising the degree of representational ambiguity present in a dataset, we establish quantitative measures that can serve as preprocessing tools to assess dataset quality, and identify problematic regions of configuration space before the training process commences.

The utility of these ambiguity metrics extends beyond mere diagnostic purposes. They provide actionable insights into which configurations are most likely to cause training instabilities, prediction inconsistencies, and in general, difficulties in overall model trainability. We move on to show that the metrics we develop bear a strong correlation with the overall R^2 -score on test datasets for our given model architectures, proving the stated use case of our metrics, as datasets can be filtered or augmented based on the ambiguity metric scores,

and training procedures can be adapted to account for high levels of ambiguity.

By quantifying the node alignment problem through rigorous ambiguity metrics, we transform an abstract theoretical concern into a measurable quantity that directly informs practical machine-learning usage for material science purposes.

1.1 Thesis Outline

This thesis is organised into three main parts, preceded by a technical background chapter and followed by concluding remarks.

Technical Background establishes the theoretical foundations required for this work. We begin with the quantum mechanical many-body problem and the development of density functional theory as a computationally tractable approach to electronic structure calculations. We then introduce machine-learning fundamentals, focusing on neural network architectures and the backpropagation algorithm. The representation of molecules as graphs is developed, along with the extension to periodic systems. We detail graph neural networks and message passing frameworks before introducing group theory concepts essential for understanding equivariance. The chapter concludes with a discussion of equivariant graph neural networks and their theoretical advantages for atomic property prediction.

Part 1: Molecules focuses on molecular force prediction. We present our equivariant message passing architecture, address limitations of the MD17 benchmark through custom dataset generation, and analyse model performance across ten organic molecules. This reveals our central contribution, the node alignment failure problem. We develop ambiguity metrics to quantify representational degeneracies from molecular symmetry and demonstrate their correlation with model performance.

Part 2: Periodic Systems extends the methodology to crystalline materials. We adapt the framework for periodic boundary conditions, investigate silica polymorphs across diverse cell sizes and space groups, and validate our ambiguity metrics in periodic structures. Results reveal fundamental limitations for heterogeneous datasets, motivating architectural modifications.

Part 3: Towards High-Entropy Alloys integrates echo state network principles into

our message passing scheme for Ni-Pd binary alloys, laying the foundation for further work with the goal of analysis of high-entropy alloys within our framework. We develop energy surface interpolation methods as alternatives to direct backpropagation for force calculations. The Echo State Network (ESN)-based architecture achieves good energy predictions whilst the interpolation schemes provide force calculations rivalling traditional approaches.

Conclusion summarises key findings, discusses implications for equivariant machine-learning in computational materials science, and outlines future research directions toward high-entropy alloy force field development.

Chapter 2

Technical Background

2.1 The Quantum Many-Body Problem

In order to describe a quantum many-body system, we are required to solve the time-independent Schrödinger equation for a system of N interacting electrons. Within the Born-Oppenheimer approximation, which separates nuclear and electronic motion due to their large mass difference, we can treat the nuclear positions as fixed parameters. This reduces our problem to solving the following equation,

$$\hat{H}\Psi(r_1, r_2, \dots, r_N) = E\Psi(r_1, r_2, \dots, r_N), \quad (2.1)$$

where $\hat{H}$ is the Hamiltonian operator, a matrix which contains all kinetic and potential energy contributions for the given system, and E is the energy of the system. For most systems of interest (realistically, anything above a simple hydrogen molecule) this equation cannot be solved analytically as the combinatorics of each interacting electron explode this problem in complexity, specifically, to an order $\mathcal{O}(N!)$ problem.

This complexity can be seen when studying the Benzene molecule. Containing 42 electrons, the wavefunction exists in a 126-dimensional space. Thus, a direct analytical attempt at a solution for the time-independent Schrödinger equation becomes computationally infeasible. This scaling is known as the curse of dimensionality, and it means that we must devise and utilise various approximations in order to encapsulate the important mechanics of a system whilst remaining within the realm of computational feasibility.

The electronic Hamiltonian from Eq. (2.1) can be written explicitly as,

$$\hat{H} = \frac{\hbar^2}{2m_e} \sum \nabla_i^2 + \frac{1}{2} \sum_{i \neq j} \frac{e^2}{|r_i - r_j|} + \sum_i V_{ext}(r_i). \quad (2.2)$$

The various terms represent the kinetic energy of the electrons and the external potential due to the nuclei and the many-body electron-electron interaction respectively. The electron-electron interaction term couples all of the electrons, leading to the $\mathcal{O}(n!)$ complexity problem for n electrons. It is this, which drives us to require more sophisticated approximation methods to simplify the problem further.

The first systematic approach to the many-body electronic problem came in the form of the Hartree-Fock method in the 1930s [1][2]. The approach is to treat the N -electron wavefunction as an anti-symmetrized product of single electron orbitals, which thus reduces the many-body problem to a set of coupled single-electron equations. Each electron moves in a mean-field potential created by averaging over the positions of all other electrons (the Hartree term), combined with an exchange term arising from wavefunction antisymmetry.

Two effects are present when considering the electrons in this case. The exchange interaction, which arises from the Pauli exclusion principle, states that electrons with parallel spins cannot occupy the same space, creating an 'exchange hole' around each electron. This is a purely quantum effect, which this method handles exactly through wavefunction anti-symmetry. The second effect at play is the correlation effect. While Hartree-Fock includes the mean-field electron-electron repulsion, it does not capture the instantaneous, correlated motion of electrons. This dynamic correlation is absent from the Hartree-Fock treatment. In summary, exchange arises from the Pauli exclusion principle and accounts for spin-dependent electron avoidance, whilst correlation captures the instantaneous charge repulsion beyond the mean-field approximation.

Though correlation typically only represents $< 2\%$ of the total energy of a system, the accuracy required for the desired use dictates that we hone in on this error.

A number of post Hartree-Fock methods exist, which can include the electron correlation term (Configuration Interaction (CI), Coupled Cluster Theory and Møller-Plesset perturbation theory are some examples of such methods). One of the regarded gold standards is

the coupled cluster single-double-triple, CCSD(T), method, which can achieve great accuracy but at the cost of $\mathcal{O}(n^7)$ computational complexity [3][4]. Naturally, this scaling makes high level correlation methods completely impractical for computational research once we reach low double-digit numbers of atoms. Thus, there is a significant gap between systems we can model accurately and systems of practical interest in materials science, nanotech and drug discovery. These limitations become particularly problematic when considering machine-learning applications, which require large datasets spanning a large range of chemical environments and system sizes. Clearly, alternative approaches with better scaling are needed.

2.2 Density Functional Theory

In 1964, Pierre Hohenberg and Walter Kohn provided a theoretical foundation for what would become Density Functional Theory (DFT) through two fundamental theorems [5]. The first states that the ground-state electron density given by,

$$n(r) = N \int |\Psi(r_1, r_2, \dots, r_n)|^2 dr_1 dr_2 \dots dr_n, \quad (2.3)$$

uniquely determines the external potential $V_{ext}(r)$, and thus all ground-state properties of the system. This implies that the ground-state energy can be written as a universal functional of the electron density,

$$E[n] = F[n] + \int V_{ext}(r)n(r)dr, \quad (2.4)$$

where $F[n]$ is a universal functional containing the kinetic energy and electron-electron interaction contributions.

The second theorem states that the ground-state energy functional obeys a variational principle. For any estimated density $\tilde{n}(r)$ which integrates to the correct number of electrons, $E[\tilde{n}] \geq E_0$, with equality when the estimated density is the true ground-state density. This provides a route to finding the ground-state density through a process of energy minimisation.

These theorems show that exact DFT exists in principle, providing that the ground-state energy and all other ground-state properties are unique functionals of the electron density.

However, explicit exact forms of the required functionals are not known.

2.3 Kohn-Sham DFT

Walter Kohn and Le Jeu Sham devised a solution to this problem. By introducing a fictitious system of non-interacting electrons that produces the same ground-state density as the real interacting system, we can calculate its kinetic energy and include all many-body effects to the exchange-correlation functional [6].

Their approach transforms the interacting N -electron problem into a set of N single-electron equations.

$$\left[\frac{-\hbar}{2m_e} \nabla^2 + V_{ext}(r) + H(r) + V_{xc}(r) \right] \Psi_i(r) = \epsilon_i \Psi_i(r) \quad (2.5)$$

where $\Psi_i(r)$ are the Kohn-Sham orbitals with eigenvalues ϵ_i , $H(r)$ is the Hartree potential and $V_{xc}(r)$ is the exchange-correlation potential. The electron density constructed by the Kohn-Sham orbitals is given as,

$$n(r) = \sum_i^N |\Psi_i(r)|^2. \quad (2.6)$$

The total energy in this formulation thus becomes,

$$E[n] = T_s[n] + \int V_{ext}(r)n(r)dr + \frac{1}{2} \int \int n(r)n(r')drdr' + E_{xc}[n], \quad (2.7)$$

where $T_s[n]$ is the kinetic energy of the non-interacting Kohn-Sham system. This formulation allows us to sidestep the unknown interacting kinetic energy functional by using the exact kinetic energy of the Kohn-Sham system.

This formulation scales with a computational complexity of $O(N^3)$, which is a great improvement compared to the much higher scaling of correlated wavefunction methods [7]. This scaling has made Kohn-Sham DFT the method of choice for large-scale quantum mechanical calculations and the primary source of training data for machine-learning force fields of various systems.

2.3.1 The Exchange-Correlation Functional

The exchange-correlation functional $E_{xc}[n]$ contains all of the many-body quantum effects beyond the mean-field Hartree interaction that distinguishes the real interacting system from the non-interacting Kohn-Sham reference system. It includes both the exchange energy arising from the Pauli exclusion principle, and the correlation energy arising from the electron-electron repulsion and the correction to the non-interacting kinetic energy. The exact form of this functional is unknown, decades of research have produced a hierarchy of approximations with varying levels of sophistication, computational cost and appropriate use cases for various purposes.

The simplest approximation is the Local Density Approximation (LDA), which treats the exchange-correlation energy density at each point in space as if it were that of a uniform electron gas with the same density [6][8]. Despite its simplicity, it performs well for systems with slowly varying electron densities, such as simple metals (Al, Ga, and others). However, there are known issues with how the LDA systematically overbinds molecules and solids.

The Generalised Gradient Approximation (GGA) represents a significant improvement over LDA [9][10]. This is done by incorporating not just the local density, but also the gradient of the local density. A number of GGA's have become popular due to, generally, correcting for the overbinding tendency of LDA and provide substantially improved geometries for molecules and solids, with respect to LDA. However, GGA's inherit the systematic values of semi-local approximations, specifically, the inability to describe dispersion interactions and the self-interaction error, which leads to the incorrect results in band gap and charge transfer calculations [11][12][13].

Meta-GGA'S extend the functional form by including further information pertaining to the density. For example they can include elements of the kinetic energy density or the Laplacian of the density. Some of these have garnered attention with their improvements in energy and geometry calculations simultaneously across diverse ranges of materials and systems.

Lastly, for our concern, hybrid functionals incorporate a fraction of the exact exchange from Hartree-Fock theory, mixing it with the DFT exchange and correlation [14][15]. A

popular choice of hybrid functional is the B3LYP, which combines elements of differing exchange and correlation functionals, and has balanced performance across many areas of molecular property calculations [16][17][18][19]. These functionals substantially reduce self-interaction error, but come at a considerably higher computational cost due to the evaluation of the non-local exact exchange. For our purposes, our DFT calculations will be conducted with use of this B3LYP functional and will be detailed in a later section, namely, Section 4.2.3.

Understanding the interplay between choice of exchange-interaction functional and overall DFT accuracy and limitations is crucial for proper calculations. Modern GGA and hybrid functionals achieve impressive accuracy for many molecular properties with bond lengths, bond angles, vibrational frequencies and atomic forces being well within acceptable ranges [19]. One of the most significant limitations is the complete failure to describe Van Der Waals dispersion interactions due to the semi-local nature of standard functionals, making them completely incapable of encapsulating the long-range correlation effects responsible for dispersion forces. This leads to systematic errors for weakly bound systems [20][21].

The selection of an appropriate exchange-correlation functional represents a critical decision in any DFT calculation, as it fundamentally determines the balance between accuracy, efficiency and the overall ability to describe the quantum phenomena of the underlying system being studied. No single functional is universally regarded as the best and we must therefore guide our choices by the specific chemical questions being asked, the nature of the system, and the resources that are available to us. While different functionals excel for different system types, we briefly highlight our choice of B3LYP for analysis of small to medium sized molecular systems. The most prominent reasoning for this is the computational cost. Whilst higher than pure GGA functionals, it remains entirely tractable for systems of this type and size. Furthermore, the inclusion of the exact exchange substantially reduces the self-interaction error, leading to a markedly improved description of molecular energies, orbital eigenvalues and charge distributions compared to semi-local approximations. The choice of this, or any functional, is not merely a technical detail but rather a fundamental aspect of the overall computational design of our data throughput processes that directly impacts the reliability and interpretability of the results, which in turn, ultimately decides whether or

not our calculations can provide meaningful chemical insights.

2.3.2 DFT as a Data Source for Machine-Learning Force Fields

Modern machine-learning for force field predictions relies heavily on DFT calculations as their primary source of training data. The $O(N^3)$ scaling makes it feasible to generate the thousands of electronic structure calculations needed to train accurate Machine-Learning (ML) models.

The process of generating DFT training data for ML force fields requires careful consideration of several factors.

- The choice of exchange-correlation functional determines the accuracy ceiling of the resulting model. For organic molecules, hybrid functionals such as B3LYP and PBE0 are most commonly used. When considering periodic systems beyond the realm of small to medium sized molecules, GGA's tend to be preferred.
- The diversity and unbiasedness of underlying datasets is crucial for model transferability. Transferability refers to how the model reacts and performs when shown data it has not seen before, be it data from the same dataset that was not used in training, or completely new samples, which may have drastically different properties but still fit the use case of the model.
- By including the atomic forces as part of our training data, this will drastically improve the ML force field quality. However, measures must be taken to ensure overfitting is not an issue.

The computational efficiency of DFT makes it uniquely suited for generating large-scale training datasets. Modern architectures can utilise parallelisation techniques in computational calculations, which makes it feasible to perform thousands of calculations in a relatively acceptable timeframe, with higher computational accuracy of leading to a cost in computational efficiency. While hybrid functionals provide superior accuracy for many properties, the higher computational cost may limit the size and diversity of the achievable training sets [7].

2.4 Machine-Learning

Machine-Learning (ML) is a powerful subset of Artificial Intelligence (AI). It encompasses many techniques, which allow us to approximate an unknown, many-dimensional function. In doing so, it can be used to generate predictions for behaviours pertaining to a specific set of data or environments similar to it. Methods vary from simple linear regression to complex neural network architectures. Neural networks are at the heart of most modern AI-related advancements within computational physics, and indeed, most AI related advancements overall.

The overall goal of a neural network is to approximate a high-dimensional, unknown, function through a feed-forward network of linked data transformations. This was originally devised as a method to mimic the workings of the human brain. In a similar vein to how a neuron works, by sending electronic signals of varying strengths amongst the connections between them, initial neural networks stacked together these layers of artificial neurons, which individually receive data through multiple input sources and transform it into an output signal. Mathematically, a neuron X in a densely connected layer l , X^l would have its output calculated by the weighted sum of all neuron outputs from the $(l - 1)^{th}$ layer, which is then shifted by a bias parameter,

$$X_i^l = b_i + \sum_{i=1}^{|X^{l-1}|} W_i^{l-1} X_i^{l-1} \quad (2.8)$$

where the vectors $\mathbf{W}$ and $\mathbf{b}$ are randomly initialised.

To visualise how the function of a single neuron translates into a full neural network, say we have a vector of input data X with a discrete number of points $|X|$, and we want to associate that data with a target Y in the form of a singular $y \in Y \in \mathbb{R}$. Each of these can be thought of as the first and last layers of a neural network, $X \equiv X^1$ and Y would be a single neuron output layer. A deep neural network (a network with more than one layer between the input and output layers) could thus be designed as shown in Fig. 2.1.

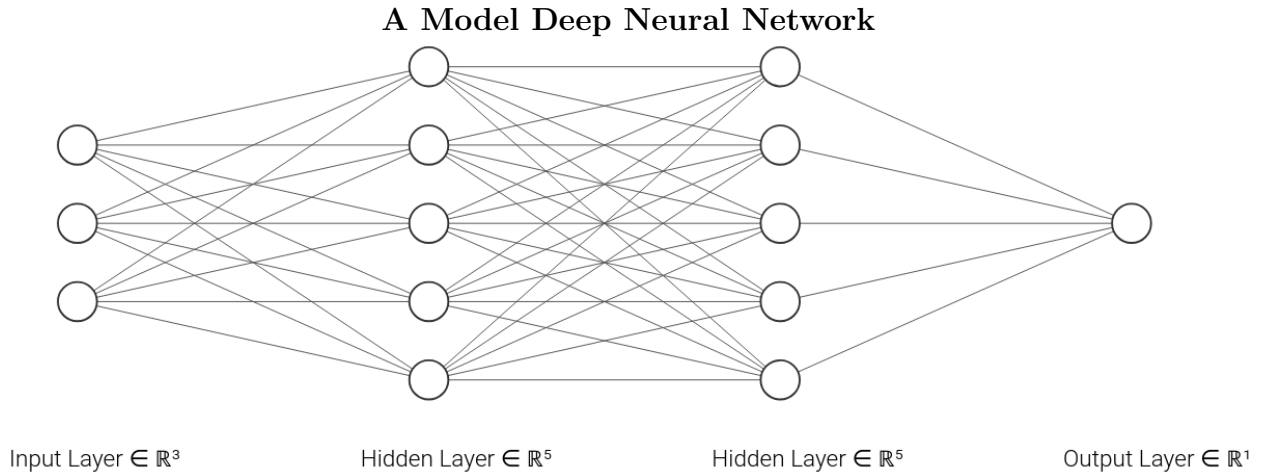


Figure 2.1: A schematic representation of a deep neural network architecture. The input layer X^0 contains a number of neurons which corresponds to the vector size of our data. Information flows forward through multiple hidden layers ($X^1, X^2, \dots, X^l$), where each neuron is densely connected to all the neurons in the previous layer via learnable weights and biases. The final output layer produces the target prediction Y . Each neuron's weight and bias is systematically updated through a procedure known as backpropagation.

The initial output of the network is, as expected, totally meaningless due to the randomised weights and biases from initialisation. We can, however, ‘train’ the network and fine-tune these components in order to get meaningful output from the network. This can then be used to generate predictions. The training process in our case relies on a dataset of input data, which is pre-labelled with corresponding output data that the network is expected to match. The training process relies on feeding our input data through and comparing the network output to the expected output from our dataset. This comparison is encapsulated within a loss function [specified in Eq. (2.14)]. This function exists purely to assign some sort of metric to how different the outputted value from the network is to the expected value correlated to the input. The derivatives of these can be used in a process known as backpropagation, which updates the weights and biases accordingly in a process known as gradient descent [22][23].

2.4.1 Forward Propagation

Forward propagation refers to the operation of data flowing through a network from the input layer to the output layer. Consider a neural network with L layers where each layer

$l \in L$ contains neurons with weight matrix W^l and bias vector b^l . The forward propagation computes the following layer outputs,

$$X^l = \sigma(Z^l), \quad (2.9)$$

where,

$$Z^l = \mathbf{W}^l X^{l-1} + \mathbf{b}^l, \quad (2.10)$$

represents the pre-activation values and σ denotes an activation function applied element-wise. Commonly used σ functions include the sigmoid, ReLu (rectified linear unit), and Tanh (hyperbolic tangent). These are the most commonly used throughout our work and are shown in Eq's (2.11), (2.12), and (2.13).

$$\sigma_{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (2.11)$$

$$\sigma_{ReLU}(x) = \max(0, x). \quad (2.12)$$

$$\sigma_{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (2.13)$$

For supervised learning tasks (machine-learning tasks where the input-output correlation is known beforehand, which covers all cases within this thesis), we employ the Mean Squared Error loss function (MSE) given in Eq. (2.14).

$$J = J(\mathbf{X}, \mathbf{Y}) = MSE(\mathbf{X}, \mathbf{Y}) = \frac{1}{2m} \sum_{i=1}^m ||Y_i - X_i^L||^2 \quad (2.14)$$

where m represents the training batch size. Batch sizing occurs as, in practice, the full training dataset is divided into smaller batches of size m , and the loss function is computed and minimised over each batch sequentially. This batching approach makes training computationally tractable for large datasets and can improve convergence properties. A complete pass through all batches in the training dataset constitutes one training epoch, and the optimisation procedure typically involves iterating over multiple epochs until convergence

criteria are met. Y_i denotes the target values for a given dataset entry, and X_i^L represents the model output for a given input i . We use J to denote a generic choice of loss function. The optimisation objective becomes minimizing this loss function with respect to all network parameters $\mathbf{W}^{L=\{l \in L\}}$ and $\mathbf{b}^{L=\{l \in L\}}$.

2.4.2 Backpropagation

The backpropagation algorithm leverages the chain rule of calculus to efficiently compute gradients $\frac{\partial J}{\partial \mathbf{W}}$ and $\frac{\partial J}{\partial \mathbf{b}}$ by propagating error signals backward through the network architecture. Beginning with the output layer, the error term for the final layer L is given by,

$$\delta^L = \frac{\partial J}{\partial Z^L} = \frac{1}{m}(Y - X^L) \cdot \sigma'(Z^L) \quad (2.15)$$

where $\frac{\partial J}{\partial Z^L}$ is the partial derivative of the loss function with respect to the pre-activation values in the output layer. This indicates the sensitivity of the loss to changes in the pre-activation values. The right hand side term comes from taking the derivative of our loss function in Eq. (2.14), which is multiplied element-wise with the derivative of the activation function. This is needed due to the chain rule, $X^L = \sigma(Z^L)$, as we need to multiply by $\sigma'(Z^L)$ to get the derivative with respect to Z^L instead of X^L . The complete term captures both how wrong our predictions are and how sensitive the activation function is at those particular pre-activation values. This becomes the error signal, which is propagated backwards through the network.

For the hidden layers $L = L - 1, L - 2, \dots, 1$ the error propagates recursively as follows,

$$\delta^L(l) = (\mathbf{W}^{l+1})^\top \cdot \sigma'(Z^l), \quad (2.16)$$

where $\delta(l)$ is the error signal for a specific layer l that we want to calculate, $(\mathbf{W}^{l+1})^\top$ is the transpose (due to the backwards propagation instead of forward) of the weight matrix connecting layers l and $l + 1$, δ^{l+1} is the error signal from the layer succeeding l that we have previously computed. By including this element, we are allowing ourselves to propagate the error information backwards. So overall, we are multiplying the error from the $(l + 1)^{th}$ layer

by the transposed weights. This tells us how each neuron in layer l contributed to the error in layer $l + 1$. $\sigma'(Z^l)$ applies the chain rule through element-wise multiplication with the derivative of the activation function at the currently layer l . This accounts for how sensitive the output of each neuron is to any changes in its pre-activation value.

The recursion works as each layer's error depends on the error from the layer succeeding it, creating a chain that propagates all the way back to the input. However, this recursive multiplication structure introduces potential instabilities in the gradient flow. When propagating errors through many layers, repeated multiplication by weight matrices and activation derivatives can cause gradients to either vanish or explode exponentially. Gradient explosion occurs when the product of weight magnitudes and activation derivatives exceeds unity across layers, causing gradients to grow unboundedly large and destabilising the training process. Conversely, gradient vanishing arises when this product is consistently less than unity, causing gradients to decay to near-zero values and preventing effective learning in earlier layers. These phenomena are particularly pronounced in deep networks and motivate the use of careful weight initialization schemes, gradient clipping, and architectural innovations such as residual connections.

This recursive formulation gives us great algorithmic efficiency as it only requires $O(|\mathbf{W}| + |\mathbf{b}|)$ operations. We can compare this to a finite differences approach, which would have a computational complexity of $O(|\mathbf{W}| + |\mathbf{b}|)^2$.

The parameter gradients of this process emerge directly from these error terms through local computations of,

$$\frac{\partial J}{\partial \mathbf{W}^l} = \frac{1}{m} \sum_{i=1}^m \delta_i^l (X_i^{l-1})^\top, \quad \frac{\partial J}{\partial \mathbf{b}^l} = \frac{1}{m} \sum_{i=1}^m \delta_i^l, \quad (2.17)$$

where the summation over the training examples i accounts for batch processing. These gradients allow us to update the model parameters through gradient descent via the following updating equations,

$$\mathbf{W}^l \leftarrow \mathbf{W}^l - \eta \frac{\partial J}{\partial \mathbf{W}^l}, \quad (2.18)$$

$$\mathbf{b}^l \leftarrow \mathbf{b}^l - \eta \frac{\partial J}{\partial \mathbf{b}^l}, \quad (2.19)$$

where η represents the learning rate hyperparameter controlling the magnitude of each update step.

Backpropagation allows us to decompose a global optimisation problem into one of local gradient computations, with each layers parameter updates depending solely on its local error signal and the outputs from the preceding layer.

The convergence properties of backpropagation critically depend on the network architecture, network initialisation schema, and loss function geometry. For the loss function the optimisation problem exhibits decently favourable properties in the regions near ideal solutions, though the non-convex nature of deep neural networks introduces one of the biggest pitfalls of the training routine, local minima. These networks can tend to get 'stuck' in the optimisation process within a local minima with no information as to whether or not it is the overall global minimum.

The gradient flow dynamics of this process can be analysed from the point of view of dynamical systems theory. Here, the parameter trajectory follows the negative gradient pathway $-\nabla J(\mathbf{W}, \mathbf{b})$, subject of course to the constraint that the gradients must propagate through the predetermined architecture of the neural network. This framework provides the theoretical groundwork for understanding both the empirical successes and fundamental limitations of gradient-based learning algorithms in deep neural network scenarios.

Despite the mathematical beauty of the overall training process for neural networks, training a neural network is not a trivial task. Many issues such as overfitting, gradient explosion and regularisation can greatly hamper our ability to obtain usable models after the training process.

It is clear to see how deep learning has been instrumental in the advancements of material property prediction and we move to detail how we can optimise the process of calculating properties through DFT with the help of machine-learning.

2.5 Powering Machine-Learning with Density Functional Theory

2.5.1 The Convergence of ML and DFT

The intersection of machine-learning and density functional theory represents a paradigmatic shift in computational materials science. The precision of quantum mechanical calculations combined with the scalability of modern artificial intelligence. This intersection raises an important question, namely, where can we draw the line between the fundamental trade-off between accuracy and computational cost?

Whilst DFT provides us with a powerful and robust framework for understanding the electronic structure and material properties of small to medium sized systems, it is very limited by the $O(N^3)$ scaling in its computational complexity. For example, while DFT calculations on large biomolecular systems like proteins are technically feasible, the computational cost becomes prohibitively expensive. This makes dynamical studies or investigations requiring many configurations effectively impossible. This scaling barrier is not only prohibitive for large systems, but also when dealing with long timescales requiring repeated calculations or, of increasing importance, high-throughput screening scenarios where thousands of configurations must be evaluated. Machine-learning offers a transformative solution by learning the complex mappings inherent in DFT calculations. Through this, we can achieve near quantum level accuracy at a fraction of the computational cost.

This synergy between ML and DFT operates on a number of different levels. At its core, DFT supplies high-quality training data that machine-learning models require to learn the intricate relationships between atomic configurations and the corresponding physical properties of interest. The electronic density, from which all ground-state properties can be calculated via the Kohn-Sham scheme, represents a rich source of information that can be utilised in the training of various models for the prediction of material properties of interest. This provides us with a positive feedback loop of DFT calculations serving machine-learning algorithms to conduct calculations at a fraction of the cost of a DFT calculation, allowing us to investigate more materials, in turn focusing our efforts of what to attempt in DFT

calculations, before repeating the cycle [24][25][26].

2.5.2 The ML Approach to Property Prediction

The application of ML to DFT-derived data has surpassed the simple use case of property prediction models to more intricate approaches that can capture the underlying quantum mechanical nature of various materials. Early efforts focused on learning scalar properties such as formation energies or band gaps, treating the atomic system as a field input to be mapped to a single output value. This initial testing ground showed us the feasibility of utilising machine-learning methods in this manner but were very much limited in scope and applicability [27].

The main challenge here lays in preserving the physical principles, which govern the nature of these materials whilst still achieving a computational efficiency allowing these efforts to be completed in a relatively quick time-frame without the need for an enormous amount of resources. Machine-learning models must be encoded to preserve and respect certain symmetries and the continuous properties of potential energy surfaces. This requirement has driven the development of specialised machine-learning schemes with various architectures and atomic representations. These approaches must encode atomic environments in such a way that preserves physical symmetries while remaining computationally tractable, though challenges remain as we later discuss.

The stochastic nature of the training process must be reconciled with the very much deterministic expectations of the given systems of interest. Models that seek to predict scalar energy values face inherent limitations in their ability to capture local variations in potential energy surfaces, of which our understanding is crucial for the use in calculations relating to atomic and molecular dynamics. These models are required to not only be highly accurate in their scalar predictions, but maintain stability across a range of chemical environments and structural spaces. These challenges are particularly acute for energy-based predictions. While forces can in principle be calculated by taking gradients of the predicted energy with respect to atomic positions, this approach suffers from error amplification and lacks explicit information about local force fields.

2.5.3 Direct Force Prediction

Direct force prediction, when compared to scalar energy value prediction, gives us several key advantages. Forces are inherently local quantities that depend primarily on the immediate atomic environment, making them more amenable to models which can capture local patterns and correlations. This locality means that models designed to predict atomic forces directly tend to be better at generalising across different system sizes and compositions, as the fundamental interactions that control atomic forces on a quantum mechanical level remain consistent regardless of global system context.

From a numerical perspective, direct atomic force prediction avoids compounding errors that arise from chained differentiation computations. When forces are computed as derivatives of the predicted energies, small errors in energy prediction can be amplified in the differentiation process, leading to much larger overall errors on the components of the atomic forces. This is quite problematic when we consider small systematic biases or noise that can be present even in the most accurate of energy prediction models, which will also propagate through and magnify error. By calculating our desired quantity directly, we can circumvent a lot of these issues.

Furthermore, the training for this scheme will inherently be more stable and efficient. Energy surfaces can be relatively flat in many regions within the configuration space. This makes it challenging for models to learn the subtle differences that could possibly correspond to large variations in forces. Being gradients, these highlight areas of the configuration space, where the most interesting physics occur, be it transition states, bond breaking events or other chemically relevant processes. The focus on dynamically important targets within machine-learning leads to more efficient use of training data and better performance in applications such as molecular dynamics simulations, an area where any advancements in computational accuracy and efficiency are highly sought after.

2.5.4 Implications for Computational Physics

All of this represents more than just a technical optimisation. It is a total recontextualisation of the direct energy solution. By focusing on the quantities that directly govern

quantum mechanical motion we automatically are aligning our models closer to the physical process of which they are designed to predict.

This alignment can be further seen by looking at the problems that now become computationally tractable. Long timescale molecular dynamics trajectories can require millions of force calculations, thus become much more feasible, if we could devise a method to get DFT level accuracy for a fraction of the computational cost. Likewise, large-scale materials screening can incorporate this level of accuracy, but without the throughput limit imposed by computational efficiency restrictions, which until recently have limited these approaches only to simplified models of small system sizes.

Thus, the direct force paradigm opens new avenues for integrating machine-learning into existing workflows, allowing for great optimisation and accuracy. Rather than replacing DFT calculations entirely, we can utilise ML as an accelerator within hybrid schemes, where occasional DFT calculations provide reference points for model calibration and uncertainty quantification. This allows for methods to gradually increase the role of machine-learning in computational materials science whilst maintaining the reliability and accuracy required for scientific discovery and technological publications.

2.6 Molecules as Graphs

Within mathematics, graphs are a form of data structure which can be used to represent pairwise relationships and interactions between unique and discrete entities through a collection of nodes and connecting edges. This structure provides a framework, which can naturally represent molecular systems, where the discrete nature of atomic connectivity largely overlaps with concepts within graph theory. Graphs provide a versatile representation, which makes them well-suited for problem solving within computational physics, where structural relationships fundamentally determine the properties of a system we want to analyse.

2.6.1 Mathematical Foundations of Graphs

A graph $\mathcal{G}$ is defined, in the most simplistic of cases, as a combination of two sets $\mathcal{V}$ and $\mathcal{E}$, denoted $\mathcal{G}(\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ represents a finite set of nodes (sometimes referred to as vertices). The set $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ contains n distinct elements with $|\mathcal{V}|$ denoting the cardinality of the set. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ consists of pairs of nodes, which can be either ordered or unordered. For our applications in the context of molecular analysis, we shall consider edges $e_i \in \mathcal{E}$ to be bidirectional as atomic bonds are assumed to exhibit symmetric interaction behaviour.

The degree of a vertex $v_i \in \mathcal{V}$, denoted $\deg(v_i)$, equals the number of incident edges: $\deg(v_i) = |\{e \in \mathcal{E} : v_i \in e\}|$.

Furthermore we have the adjacency matrix $\mathcal{A}$. This matrix provides representation of connectivity between nodes. For n nodes, $\mathcal{A}$ is an $n \times n$ matrix where $a_{ij} = 1$, if $(v_i, v_j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. These connections can also be weighted by attributing non-binary values to the elements of $\mathcal{A}$. In our case of undirected edges within graphs, $\mathcal{A}$ is a symmetric matrix as $a_{ij} = a_{ji}$.

2.6.2 Extended Definition for Molecular Systems

Molecular systems require a number of addenda to the typical simplistic graph structure in order to be useful in representing systems of interest for a machine-learning use case. We extend our definition of a graph to $\mathcal{G}(\mathcal{V}, \mathcal{A}, \mathcal{E}, \mathcal{V})$, where each matrix component serves a specific role in defining our representation.

The node matrix $\mathcal{V}$ is given as an $n \times m$ matrix $\mathcal{V} \in \mathbb{R}^{n \times m}$, where n corresponds to the atomic count $n = |\mathcal{V}|$ and m represents the dimensionality of the feature space. Each row $v_i \in \mathbb{R}^m$ constitutes a feature vector encoding individual atomic properties $v_i = [v_{i_1}, v_{i_2}, \dots, v_{i_m}]^T$.

Feature vectors can incorporate a wide range of atomic properties. These could include atomic number, mass, electronegativity, charge, hybridisation state and spatial coordinates to name a few potential use cases. In many cases where features are a form of categorisation, particularly in the case of atomic number, we encode these as a one-hot encoding rather than directly through integer value numbers. One-hot encoding allows us to represent categorical

properties through binary vectors where there is only one non-zero entry. For molecules containing K distinct elements, the atomic species encoding aspect of the feature vector $v_{ij}^{\text{species}} \in \mathcal{V}$ would be a sparse binary sub-vector of dimension K .

Cartesian spatial coordinates $\mathbf{r}_i = [x_i, y_i, z_i] \in \mathcal{R}$ are the simplest way to encode spatial information into our representation, and much like the one-hot encoded species information, forms a sub-vector $r_k = \{v_{i,j} : r_i \in \mathcal{R}\}$. This encoding will enable correspondence between various graph operations, whilst preserving the essence of the physical molecular structure.

The binary adjacency matrix $\mathcal{A} \in \{0, 1\}^{n \times n}$ keeps track of the existence of edges and which nodes they connect to. Edges in this case can simply correlate to covalent bonding, with $a_{ij} \in \mathcal{A} = 1$ indicating presence and $a_{ij} \in \mathcal{A} = 0$ absence. We are not, however, strictly limited by this, and for small systems, memory considerations permitting, we can use fully connected graphs to hopefully encode more information in our representation for use in our models, with the idea of giving the model a better chance to train well on a given dataset.

Furthermore, weighted extensions allow us to utilise the set of real numbers rather than binary inputs for each element of the adjacency matrix. Each $a_{ij} \in \mathcal{A} \in \mathbb{R}$ could be used to encode various bond properties such as bond orders or strengths, although binary representation often provides sufficient structural information, as most bond features would tend to be encoded in the edge matrix $\mathcal{E}$.

The edge feature matrix $\mathcal{E}$ can be used to encode bond-specific information beyond basic connectivity (which is covered by use of $\mathcal{A}$). Construction of this matrix is done by the enumeration of each non-zero $a_{ij} \in \mathcal{A}$. Here, each of these values are read in row-major order to ensure consistency in edge indexing across molecular graphs. For each non-zero a_{ij} we obtain the corresponding ordered feature vector $\mathbf{e}_i \in \mathcal{E}$ where $|\mathbf{e}| = \{|\mathcal{A}| : a_{ij} \neq 0 \forall (i, j) \in n \times n\}$ is the number of edges present in a graph $\mathcal{G}$ (and thus the number of feature vectors present within $\mathcal{G}$ and $|e_i|$ is the length of each individual edge feature vector $e_i \in \mathcal{E}$).

Edge feature encodings could include various properties and the descriptors used are down to developer preference. These can include bond order, length, aromatic character, stereochemistry or quantum mechanical parameters such as overlap integrals.

The global feature matrix $\mathcal{Y}$ can be utilised to encode any aspects of our system for which we want our machine-learning models to attempt to predict. These are typically used to encode molecular properties of interest, which characterise entire systems. Some examples of this could include electron density matrices, various different atom-wise energy components or other observables one might find of interest. There is no strict sizing or use case for this matrix, much like the choice of feature encoding within the node and edge matrices, this matrix can be comprised of anything one desires.

An illustration of a molecular graph is shown in Fig. 2.2.

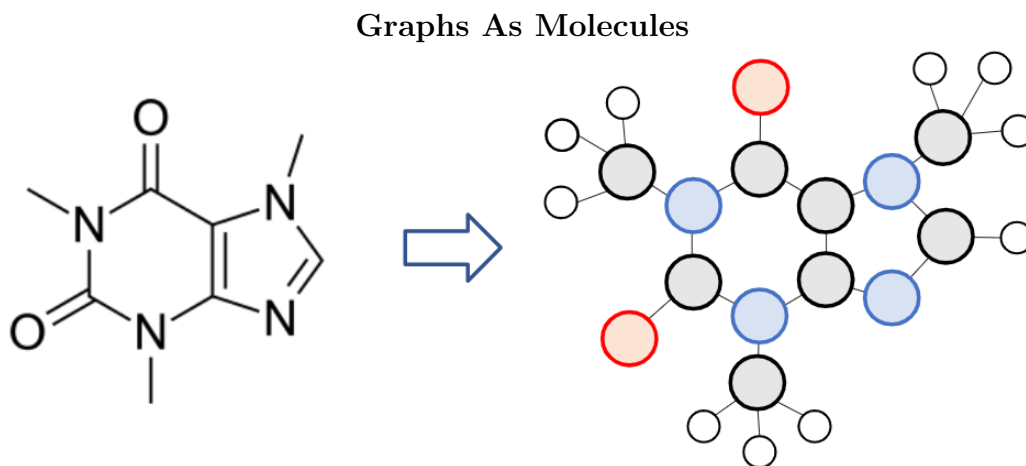


Figure 2.2: A simple diagram showing how a caffeine molecule can be envisioned as a graph. Each atom represents a graph’s node, with edges describing the bonds between atoms. In the graph the atomic attribute to each node follows the colour code: C = grey, O = red, N = blue, H = white.

For our purposes, we intend to target a matrix of atomic forces for a given system, thus $\mathcal{Y}$ would have a cardinality of $|\mathcal{Y}| = |\mathcal{V}| = n$ and be of size $\mathcal{Y} \in \mathbb{R}^{n \times 3}$, where 3 represents the x , y , and z components of each force vector $y_i \in \mathcal{Y}$.

2.6.3 Graph Isomorphism and Molecular Identity

A graph isomorphism is a graph-theory concept that provides a theoretical foundation for determining if two graph representations of data describe identical structures. Specifically, two graphs $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ are isomorphic if there exists a bijection $\phi : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ such that $(u, v) \in \mathcal{E}_1$ if and only if $(\phi(u), \phi(v)) \in \mathcal{E}_2$. A diagram illustrating this issue in a molecular context is shown in Fig. 2.3.

The Molecular Graph Isomorphism Problem

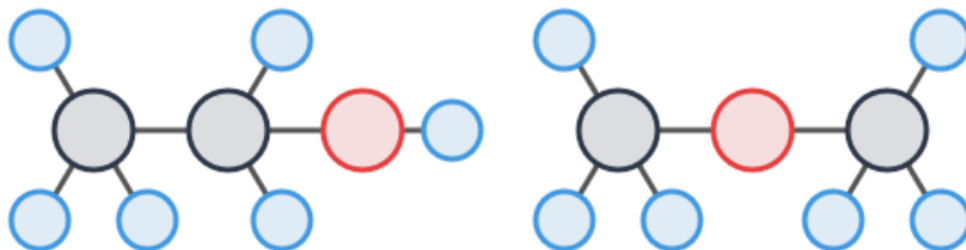


Figure 2.3: Graphical example of the graph isomorphism problem in molecular representations. The two molecules shown here have identical graph topology. They have the same number of vertices, edges and degree sequences. However, they possess completely different chemical properties. The left-hand side graph represents ethanol, while the right-hand side is dimethyl ether. The limitation of the graph isomorphism problem can, however, be easily overcome by encoding molecular properties with node and edge features. By representing atomic species as nodes features and bond connectivity within the edge features, models can easily distinguish these structural isomers. Furthermore, the use of an adjacency matrix reveals that ethanol has C-C-O connectivity, while dimethyl ether has C-O-C connectivity. Whilst the graph isomorphism problem is a theoretical problem, it is negligible in practice when proper encoding is implemented. The atomic attribute to each node follows the colour code: C = grey, O = red, H = blue.

The task of figuring out if graphs are isomorphic to each other is one of the fundamental problems within graph theory. They are generally not easy to solve efficiently and the most optimal algorithms have quasi-polynomial complexity of $N^{O(\log N)}$, where N is the number of vertices in the graph [28]. This complexity means that determining whether or not two molecular graphs represent the same molecule can be computationally challenging, especially for when we scale to larger systems. This would have large implications on how we design algorithms when it comes to degenerate representability [29].

It is known that two-body based molecular graph representations are fundamentally limited in regards to unique representability, being degenerate in two-body equivariant representations as discussed in Section 2.11. This limitation suggests an outright inability to tell two different structures apart if they happen to have the same graph encoding, namely, when two non-isomorphic graphs with identical local neighbourhoods are present in datasets and create ambiguities that compromise on learning performance.

However, this perspective overlooks the high level of mathematical richness that can

be had with a relatively simple graph-based representation of molecular systems. Molecular graphs capture essential structural information, which forms a great foundation for machine-learning approaches. Local edge connectivity, even if potentially ambiguous when looking at a system globally, encodes critical information on locality, bonding environments, functional groups and reactivity patterns, which strongly correlate to various molecular properties. The incompleteness of the two-body representation within a graph isomorphism realm does not negate nor strongly diminish a substantial predictive power than can be formed through machine-learning approaches. Rather, it provides a simplistic foundation, which suggests opportunities for enhancement through higher order structural information, encoding more intricate topological features.

2.7 Periodic Systems as Graphs

The extension of molecular graph representations to periodic systems introduces a number of unique mathematical challenges. These are problems that we must face as crystalline materials form a major sector of computational physics research of interest, having a much more rich and interesting feature set of applications, uses and key properties.

While molecular graphs possess a very well-defined set of boundaries and can easily construct node, edge and adjacency matrices with minimal ambiguity, periodic systems require more thought and care in the design of their representations. We are required to develop a specialised formalism which can deal with the infinitely repeating sequence of nodes and edges whilst retaining a reasonable level of computational efficiency.

2.7.1 From Molecular to Periodic Graph Theory

The representation of periodic systems as graphs extends upon our work in Section 2.6. This extension requires careful treatment of periodic boundary conditions whilst ensuring that the formalism can preserve both local atomic environments, as done previously with molecular, but still encapsulate the overall global crystalline properties and symmetries.

Consider a 3D periodic system defined by a Bravais lattice (an infinite array of discrete points with identical surroundings, generated by integer linear combinations of primitive

lattice vectors) with primitive vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$. Any lattice point can be uniquely defined with

$$\mathbf{R}_{n_1, n_2, n_3} = n_1 \mathbf{a}_1 + n_2 \mathbf{a}_2 + n_3 \mathbf{a}_3 \quad (2.20)$$

where $\mathbf{n} = (n_1, n_2, n_3) \in \mathbb{Z}$ are integers representing the unit cell indices, with each unit increment in n_i corresponding to a translation to an adjacent unit cell along the $\mathbf{a}_i$ direction. This periodic graph $\mathcal{G}_{\text{periodic}}(\mathcal{V}_{\text{periodic}}, \mathcal{E}_{\text{periodic}})$ must be able to capture the connectivity of this infinite system whilst still being within reasonable levels of computational complexity.

We can overcome this problem of infinitely repeating (n_1, n_2, n_3) by noting that we can decompose the graphs into a finite unit cell, which can then be coupled with our periodic boundary conditions to simplify the problem. We have,

$$\mathcal{G}_{\text{periodic}} = \mathcal{G}_{\text{cell}} \otimes \mathcal{P}, \quad (2.21)$$

where $\mathcal{G}_{\text{cell}} = \mathcal{G}(\mathcal{V}_{\text{cell}}, \mathcal{E}_{\text{cell}})$ is the graph comprising all information pertaining to the unit cell, $\mathcal{P}$ is the periodic boundary operator encapsulating the combined (n_1, n_2, n_3) , and $\otimes$ denotes a graph product, which extends connectivity over all unit cells.

2.7.2 Unit Cell Construction

The unit cell graph contains $N = |\mathcal{V}_{\text{cell}}|$ vertices. Each vertex $v_i \in \mathcal{V}_{\text{cell}}$ directly corresponds to a single atom within the unit cell. The position of this can then be described as,

$$\mathbf{r}_i = x_i \mathbf{a}_1 + y_i \mathbf{a}_2 + z_i \mathbf{a}_3, \quad (2.22)$$

where (x_i, y_i, z_i) are the *fractional* coordinates of the atom i at vertex z_i with $0 \leq x_i, y_i, z_i \leq 1$. Fractional coordinates translate the (x, y, z) position of an atom in a cell to a coordinate system, which is agnostic to overall global position. Instead it describes the position of an atom by its placement within a cell based on a fraction of the lattice vectors along the lattice points $\mathbf{R}$.

The construction of edges within this unit cell graph must now account for both atoms within the cell and atoms outside of the cell but still within the sphere of influence around

the atom in question. This requires the definition of a periodic metric function which can consider all periodic images of each atom. We define this distance between two atoms i and j to be

$$d(i, j) = \min_n |\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}| \quad (2.23)$$

where $\mathbf{n}$ encompasses all possible lattice translations as defined in Eq. (2.20).

2.7.3 The Minimum Image Convention

The Minimum Image Convention (MIC) gives us a method for tackling this problem. The process allows us to uniquely define distances and connectivity within periodic systems without large computational complexity.

For any pair of atoms (i, j) , there exists an infinite number of ways in which these two atoms could be connected under our unit cell representation due to the infinitely repeating nature of our periodic construction. The MIC selects which of these connections pertains to the shortest Euclidean distance between the atoms, regardless of the cell boundaries, and therefore is able to define a unique topological representation of the local environment.

The minimum image vector $\mathbf{n}_{ij}^{\min}$ between the pair (i, j) is given as,

$$\mathbf{n}_{ij}^{\min} = \operatorname{argmin}_{\mathbf{n}} |\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}|. \quad (2.24)$$

The minimum image distance is thus given by,

$$d_{ij}^{\min} = |\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}_{ij}^{\min}|. \quad (2.25)$$

Due to obvious computational limitations, the MIC is applied with a cap on the number of lattice translations away from the unit cell. We desire that,

$$\mathbf{n} \in \{n_1 \mathbf{a}_1 + n_2 \mathbf{a}_2 + n_3 \mathbf{a}_3 : |n_k| \leq N_k, k = 1, 2, 3, \dots\} \quad (2.26)$$

where N_k is specifically chosen so that any interactions we deem feasibly suitable for consideration are captured within our calculation.

2.7.4 Cutoffs, Connectivity and Neighbourhoods

Edge connectivity within periodic graphs is almost always determined to be within a predetermined cutoff radius r_{cut} . We say that an edge can only exist in our scheme if it supplies the shortest distance between two unique atoms, as give by the MIC, and if it falls within our predefined cutoff range.

$$e_{ij} \in E_{\text{cell}} \leftrightarrow d(i, j) \leq r_{\text{cut}}. \quad (2.27)$$

Thus we can define an atoms local neighbourhood as,

$$\mathcal{N}_i = \{j : d(i, j) \leq r_{\text{cut}}, j \neq i\}. \quad (2.28)$$

2.7.5 Symmetry and Invariance

Periodic representations are required to satisfy a number of criteria regarding symmetries in order to be physically meaningful. One of the most critical is translational invariance, given by,

$$\mathcal{G}(\mathbf{r}+\mathbf{R}) = \mathcal{G}(\mathbf{r}) \quad (2.29)$$

for any lattice vector $\mathbf{R}$. The MIC automatically adheres to this by construction, as the relative position of each atom is not changed under any form of lattice translation.

The preservation of point group symmetries within the repeating structure of interest is also considered. With any symmetry operation g within our systems point group, we should have that,

$$\mathcal{G}(g \cdot \mathbf{r}) = g \cdot \mathcal{G}(\mathbf{r}). \quad (2.30)$$

The same is true for rotational symmetries, with edge vectors transforming as,

$$\mathbf{R} \cdot \mathbf{e}_{ij} = \mathbf{R} \cdot (\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}_{ij}^{\text{min}}). \quad (2.31)$$

2.7.6 Computational Complexity

The complexity and scaling of construction of this formalism depends heavily on the choice of r_{cut} . For a given minimum lattice parameter a_{min} , we generate a number of replica unit cells. The number of these replicas generated scales according to,

$$N_{\text{replicas}} \sim \left\lceil \frac{r_{\text{cut}}}{a_{\text{min}}} \right\rceil^3 \quad (2.32)$$

The overall number of distance calculations required to formulate our representation and devise a neighbourhood construction for all atoms within the unit cell scales with,

$$O(|\mathcal{V}| \cdot N_{\text{replicas}}) \sim O(N^2) \quad (2.33)$$

per atom.

To ensure our implementation is optimised to account for this, along with the enhancement of computational efficiency, we implement various methods:

- *Spatial Partitioning*: Using cell lists to reduce the number of distance calculations
- *Early Stopping*: Screen interactions for applicability before computing Euclidean distances
- *Vectorisation*: Optimise distance calculations through vectorisation and parallelisation techniques

In practice, the generation of atomic neighbourhoods for periodic structures tends to scale in complexity with $O(|\mathcal{V}| \cdot N_{\text{replicas}})$, which poses a problem, but only for badly selected cutoff radii. Careful consideration of radius choice is needed in order for this to be computationally viable.

2.8 Graph Neural Networks and Message Passing

Graph Neural Networks (GNNs) are a subclass of neural networks, which are designed to operate with standard dense layer architecture, with graph structures as inputs. Traditional

networks are limited in a way due to them having to process fixed sized data structures. GNNs can handle variable topology built into graph data structures. The combination of these two are an obvious choice for our research into molecular and crystalline materials due to their varying atomic connectivity patterns across a range of different samples. However, care must be taken within the implementation to ensure regular forward flow through the network and consistent backpropagation calculations.

The main interface between graph matrices and their use as inputs in traditional neural network inputs layers is a series of aggregation and transformations of the local neighbourhood information of each node within the graph. This consistently transforms our data into a latent space of the correct sizing for input into our neural network. Each node in a graph contains a feature vector, $v_i \in \mathcal{V}$, which can be 'updated' through a series of *message passing* operations [30].

Message passing is a process which allows the network to infer complex data between different atomic environments whilst leaving the underlying topology of the system untouched. This framework can be decomposed into a sequence of computational layers. Each layer l performs one complete round of message passing, updating all node features based on their local neighbourhoods.

For a graph $\mathcal{G}(\mathcal{V}, \mathcal{A}, \mathcal{E}, \mathcal{Y})$ with node features $v_i^l \in \mathbb{R}$, where d is the dimension of the feature space and $l \in \mathcal{L}$ is any given in layer in the network, the message passing process contains three different steps, which are described here,

2.8.1 Message Computation

Each edge $e_{ij} \in \mathcal{E}$ generates an associated message vector m_{ij}^l through a *learnable function*. Learnable functions most often take the form of a set of densely connected and layered neurons. This essentially forms a 'sub-network' within the overall neural network. The parameters of these neurons are subject to the same forward pass and backpropagation algorithms as the main network, but pertain solely to this message passing function. The overall computation function is,

$$m_{ij}^l(h_i^l, h_j^l, e_{ij}) = \sigma(W_{msg}^l[h_i^l \otimes h_j^l \otimes e_{ij}] + b_{msg}^l) \quad (2.34)$$

where $\otimes$ denotes concatenation, σ is an activation function to be specified. The vectors W_{msg}^l and b_{msg}^l are sized to produce appropriate fixed-dimensional vectors.

2.8.2 Aggregation

Aggregation collects every message within a node neighbourhood, for each node within the system, through a permutation-invariant operator.

$$m_i^l = \bigoplus^l (\{m_{ij}^l\} : j \in \mathcal{N}_i) \quad (2.35)$$

where $\mathcal{N}_i$ denotes the neighbourhood of node i , and $\bigoplus$ is the aggregate function. This aggregation operator must be permutationally invariant so that the ordering of messages for each neighbour do not affect the output of this process. Commonly, the addition operator is used, so Eq. (2.35) takes the form,

$$\bigoplus^l (\{m_{ij}^l\} : j \in \mathcal{N}_i) = \sum_{j \in \mathcal{N}_i} W_{agg}^l m_{ij}^l + b_{agg}^l \quad (2.36)$$

where W_{agg}^l and b_{agg}^l are learnable parameters as before.

2.8.3 Node Update

The overall node update phase combines the aggregated message with the current node representation with an update layer within the overall network. The update function U is given as,

$$v_i^{l+1} = U^l(v_i^l, m_i^l). \quad (2.37)$$

For equivariance purposes, the update function here will be chosen to simply be the addition operator. This will be discussed further in Section 2.11.

Thus the complete message passing scheme can be summarised as the composition of Eq. (2.35) and Eq. (2.37) as follows,

$$v_i^{l+1} = U^l(v_i^l, (\bigoplus^l (\sigma^l(v_i^l, v_j^l, e_{ij})) : j \in \mathcal{N}(i))). \quad (2.38)$$

This formulation lets us propagate information across the graph prior to entering the core of the network whilst maintaining differentiability, which is essential for backpropagation. It is the baseline architecture for almost all GNNs and the expressivity of this network can be easily tweaked by altering our choice of functions. More complex variations of the architecture incorporate various techniques such as attention mechanisms or gated update mechanisms. These have gained a lot of popularity lately and concepts of each will be touched upon in our work.

The number of overall message passing layers that precede our densely connected neural network determines what is known as the *receptive field* of our network. After l layers of message passing, for any node v_i , it will have aggregated information from nodes within its l -hop neighbourhood. This allows us to gather information further across systems rather than be bounded by information dictated by our cutoff radius r_{cut} . This does, however, come at a price, as high-depth values can lead to network smoothing, a phenomenon wherein the network starts to predict the average target of a dataset to minimise a loss function rather than settle on a local minima.

The irregular, non-sequential and non-dense, connectivity of these layers introduce a heightened computational challenge in terms of efficiently batching data and allocating memory, especially in scenarios with long range connectivity. This makes it paramount that we strive to achieve maximum network predictive performance with as small of a parameter set as we can get away with. However, advancements in trainability of networks that message passing GNNs can provide undoubtedly make this a fruitful endeavour.

2.9 Group Theory

For reasons that will become clear when discussing the key findings of our work, providing an extensive understanding of how symmetry plays a role in the effectiveness of our representation will be needed to provide a solid baseline for the explanation of our results. To do so, we must look at the mathematical foundations of equivariance and understand how various functions and transformations preserve symmetry relationships under group actions [31]. These concepts form the foundation of representation theory within machine-learning

and are sometimes wrongly overlooked as a set-in-stone formulation, but must be understood in order to fully understand the actions of networks that are built upon them.

2.9.1 Group Theory and Symmetry Operations

A group G is a set which is paired with some form of binary operation $\cdot : G \times G \rightarrow G$ which satisfies the following fundamental axioms,

- *Closure* $\forall g, h \in G, g \cdot h \in G$.
- *Associativity* $\forall g, h, k \in G, (g \cdot h) \cdot k = g \cdot (h \cdot k)$.
- *Identity* $\exists e \in G$ s.t. $\forall g \in G, e \cdot g = g \cdot e = g$.
- *Invertibility* $\forall g \in G, \exists g^{-1}$ s.t. $g \cdot g^{-1} = g^{-1} \cdot g = e$.

This formalism will allow us to showcase how symmetry operations can transform various elements within a given space whilst preserving underlying structures.

2.9.2 Invariance and Equivariance

Any given mathematical function $f : X \rightarrow Y$ is said to be invariant under a group action G if the following holds,

$$f_{\text{inv}}(g \cdot x) = f_{\text{inv}}(x) \quad \forall g \in G, x \in X. \quad (2.39)$$

This property ensures that a functions output is not affected by any transformations under G .

Similarly, a function $f : X \rightarrow Y$ is said to be equivariant with respect to group actions $\alpha : G \times X \rightarrow X$ and $\beta : G \times Y \rightarrow Y$ if,

$$f(\alpha(g, x)) = \beta(g, f(x)) \quad \forall g \in G, x \in X. \quad (2.40)$$

This relationship establishes a commutative ordering of operations, wherein functions and transformations may be performed in any order and output the same result. This will become important later when considering training space reductions for identical molecules.

There is an important distinction to be made between these two transformation laws. Invariant functions will ignore the presence of symmetry operations when producing an output, whereas equivariant functions will transform their outputs in a way which is consistent with the transformation applied to their inputs.

This requirement of consistency leads to the imposition of strong constraints on the functional form and will lead to drastic simplifications with respect to the dataset size required to train our GNN. Indeed, any computational framework, which seeks physical consistency, requires that equivalent configurations under group transformations will produce appropriate and expected outputs.

2.10 Linear Representations and Group Actions

A linear representation of a given group G on a vector space V is defined as a homomorphism such that,

$$\tau : G \rightarrow GL(V), \quad (2.41)$$

where $GL(V)$ is known as a *general linear group*. The use of 'general' here implies that this covers all invertible linear transformations on the given vector space V . This mapping assigns, to each group element $g \in G$ a linear transformation as follows,

$$\tau(g) : V \rightarrow V \text{ s.t } \tau(gh) = \tau(g)\tau(h). \quad (2.42)$$

The homomorphism property is crucial to maintaining mathematical structural consistency. The implication here is a direct correspondence and relationship between the composition of group operations in an abstract space with the composition of transformations within a given representation space. When this is true, mathematical objects which are equivalent to each other map to equivalent representations, removing degeneracies and preserving the underlying symmetry structures.

The consistency of representations is an issue, which is especially prone to appearing when we attempt to interface abstract mathematical objects with computational systems, particularly when they require fixed-size inputs. Any representation $\tau : A \rightarrow \mathbb{R}^n$, where A is

a generic mathematical object, are bounded by the dimensionality requirement of n . Due to this, extra care is needed in order to ensure that equivalent (under the relevant symmetry group) objects transform predictably under corresponding matrix operations. Failure to do so will lead to scenarios where the same mathematical objects will appear as fundamentally different inputs to computational processes. Which such inputs are interfaced with systems that expect consistent input-output mappings, the breakdown of representational consistency will propagate throughout the entire calculation pipeline, leading to different processing of the same underlying information with no way to tell output is 'correct'.

Whilst continuous objects may possess perfect symmetries under an infinite number of groups, any finite-dimensional (thus, computationally relevant) representation will inevitably involve discretisation, which will induce labelling dependencies. This can easily break the homomorphism property and representations will then be required to rely on arbitrary labelling conventions, which not only ignores the intrinsic mathematical properties of the object of consideration, but may not even be possible to do consistently in some cases.

2.11 Equivariant Graph Neural Networks

By utilising the intersection of group theory and graph neural networks, we can utilise a powerful framework for complex systems of interest where physical laws strictly restraints on symmetry. Standard GNNs are capable of learning complex patterns, but they are agnostic to the underlying inherent symmetries present in atomic systems. Equivariant Graph Neural Networks (EGNNs) address this limitation by directly encoding symmetry constraints into the network architecture itself, ensuring that model predictions transform appropriately.

The motivation for the use of equivariance encoding in model architectures stems from the physical requirement that, in our use case, prediction targets must transform consistently with coordinate transformations. If we apply a rotation $\mathbf{R} \in SO(3)$, where $SO(3)$ is the special orthogonal group in 3D representing all possible ways to rotate a rigid object in 3D space, the outputs must undergo the same rotation $\mathbf{R}$. This reduces the complexity of the learning space for any neural network due to elimination of the need for the network to infer the constraints and symmetry relationships on its own.

2.11.1 Coordinate Equivariance

Given a molecular graph $\mathcal{G}(\mathcal{V}, \mathcal{A}, \mathcal{E}, \mathcal{Y})$ where each node $v_i \in \mathcal{V}$ contains both scalar features $h_i \in \mathcal{H}^{d_h}$ and coordinate vector features $x_i \in \mathbb{R}^3$, we require that for $E(3)$ -equivariance to hold then any Euclidean transformation $g(R, \mathbf{t}) \in E(3)$ must satisfy,

$$f(\mathbf{R}x_i + \mathbf{t}, h_i) = \mathbf{R} \cdot f(x_i, h_i) + \mathbf{t} \quad \forall \mathbf{R} \in SO(3), \quad \forall \mathbf{t} \in \mathbb{R}^3, \quad (2.43)$$

where $g \cdot x_i = \mathbf{R}x_i + \mathbf{t}$ is the group action on our coordinates and f represents an overarching abstract EGNN mapping.

Translation equivariance can be easily satisfied by shifting from an absolute coordinate position perspective to one of relative distances,

$$x_{ij} = |x_i - x_j|. \quad (2.44)$$

The implementation of rotational equivariance in EGNNs can be implemented through an *additive update mechanism*, which we detail in Section 3.2.

2.11.2 Training Space Reductions

The reduction of training space through the formulation of equivariant representations can be somewhat quantified. For a dataset with M unique entries, a standard network must learn across all symmetry transformations. The required training space scales as,

$$O(M \times |G|), \quad (2.45)$$

where $|G|$ represents the size of the symmetry group.

In contrast an equivariant network would require a training scale scaling in accordance with,

$$O_{\text{equi}}(M). \quad (2.46)$$

If we consider $E(3) = SO(3) \ltimes \mathbb{R}^3$, where $\ltimes$ is the semidirect product reflecting the non-commutative composition of rotations and translations. This hints at, at least theo-

retically, infinite potential savings. Empirically, it has been shown that models with and without equivariant representations achieve comparable predictive accuracies with around 2% of the number of training instances needed [32]. This is vitally important, recalling the computationally expensive DFT calculations needed to generate these datasets in the first place.

2.11.3 Equivariant Message Passing Architecture

We expand on the standard message passing scheme within GNNs by operating on scalar and vector features simultaneously. For a given layer in our network l , we have the following:

- *Scalar Node Features*: $h_i^l \in \mathbb{R}^{d_h}$.
- *Vector Node Features*: $\mathbf{x}_i^l \in \mathbb{R}^3$.
- *Edge Features*: $e_{ij} \in \mathbb{R}^{d_e}$.

The message computation for scalars is as previously described in Section 2.8. However, we introduce an additional vector message that will be responsible for updating the coordinates.

$$\mathbf{m}_{ij}^l = \mathbf{LSF}^l(h_i^l, h_j^l, e_{ij}) \cdot \frac{(\mathbf{x}_i - \mathbf{x}_j)}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|_2 + \epsilon} \quad (2.47)$$

where $\mathbf{LSF}^l : \mathbb{R}^{d_h} \times \mathbb{R}^{d_h} \times \mathbb{R}^{d_e} \rightarrow \mathbb{R}$ is a learnable scalar function, the subscript 2 indicates an $L2$ -norm, and ϵ is a small constant for numerical stability purposes.¹

The crux of this implementation is the fact that the direction of any vectorised message is determined purely by the normalised relative position vector $\frac{(\mathbf{x}_i - \mathbf{x}_j)}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|_2}$, whilst the magnitude of this vector is calculated by $\mathbf{LSF}^l$.

2.11.4 Aggregation and Update Operations

Aggregating the messages from all neighbouring nodes for both scalar and vector components is conducted as is described in Section 2.8.2, with only the vectorised messages needing

¹This is done to avoid accidentally dividing by zero, which is possible during the backpropagation process or, although exceedingly rare, when two atoms are listed to be a numerically identical positions due to float error.

to be updated.

$$\mathbf{m}_i^l = \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ij}^l. \quad (2.48)$$

We note that the summation operator indeed preserves translational and equivariant translations owing to its properties of linearity and permutation invariance.

The update operations can also be designed to maintain equivariance within the system.

$$h_i^{l+1} = h_i^l + \text{LSF}_h^l(h_i^l, m_i^l, \|\mathbf{m}_i^l\|_2), \quad (2.49)$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \mathbf{m}_i^l, \quad (2.50)$$

where LSF is a learnable function used in the updating of scalar latent features, messages and magnitude of message vector. Note that $\|\mathbf{m}_i^l\|_2$ is invariant, which ensures that further updates do not break equivariance within the representation.

Part 1 - Molecules

Having established the theoretical foundations of density functional theory, machine-learning, and equivariant graph neural networks, we now turn our attention to their practical application in molecular force prediction. Part 1 of this thesis focuses on molecular systems, which serve as an ideal proving ground for developing and validating our machine-learning force field methodology. Molecules present well-defined boundaries, finite dimensionality, and relatively tractable computational requirements compared to extended periodic systems, making them suitable for rigorous testing of our architectural choices and training procedures.

We begin by detailing our equivariant message passing graph neural network architecture designed specifically for direct atomic force prediction from molecular graphs. Following this, we present our approach to addressing the significant limitations of the widely-used MD17 dataset through the construction of a custom dataset generated with tight DFT convergence criteria. We then analyse the performance of our models across ten distinct organic molecules, revealing unexpected patterns in predictive accuracy that cannot be explained by system size or structural complexity alone. This leads us to our central finding, the identification of fundamental node alignment failures that arise from the interfacing of equivariant representations and neural network architectures. Through rigorous mathematical analysis and the development of quantitative ambiguity metrics, we demonstrate that molecular symmetry, rather than being purely advantageous from a model training perspective, introduces systematic representational degeneracies that directly impact model trainability and performance.

Chapter 3

GNN Implementation

Our implementation for an equivariant message passing graph neural network operates on molecular graphs of the form $\mathcal{G}(\mathcal{V}, \mathcal{A}, \mathcal{E})$ where nodes represent atoms, edges encode atomic interactions along known bonds with the goal of learning a mapping from the given molecular structure to the corresponding force vectors $\mathcal{Y} \in \mathbb{R}^{n \times 3}$

The core of our network architecture lies within our equivariant message passing layer, henceforth referred to as the EMPL. This layer maintains $E(3)$ -equivariance whilst processing both scalar and vectorial features from the graph matrices. This choice ensures that predicted forces transform consistently under any rigid body transformation. This is a fundamental requirement for any form of meaningful predictions of this nature.

3.1 Feature Engineering

3.1.1 Node Feature Construction

The node matrix $\mathcal{V} \in \mathbb{R}^{n \times (d_h + 3 + 3)}$ is constructed as a concatenation of three distinct sub-matrices,

$$\mathcal{V} = [\mathcal{H}, \mathcal{X}, \mathcal{F}] \tag{3.1}$$

where,

- $\mathcal{H} \in \mathbb{R}^{n \times d_h}$ is the one-hot encoded atomic species identifiers,
- $\mathcal{X} \in \mathbb{R}^{n \times 3}$ contains the Cartesian coordinates of each atom in Ångström,
- $\mathcal{F} \in \mathbb{R}^{n \times 3}$ is a latent vector initialised to zero.

The dimensionality of d_h is determined by the number of unique atomic species that are present in the dataset. The use of $\mathcal{F}$ allows us to update a latent representation directly whilst still having access to preserved invariant atomic identities $\mathcal{H}$ and the equivariant positional information in $\mathcal{X}$.

The concatenation structure in the node matrix has the ability to preserve permutation equivariance. This is a critical property within molecular graphs as they have no canonical node ordering schema. For any permutation matrix $\pi \in S_N$, corresponding to the reordering π of N atoms we have,

$$\mathcal{V}(\pi \cdot \mathcal{H}) = \pi \cdot \mathcal{V}(\mathcal{H}). \quad (3.2)$$

This follows component-wise, namely, one-hot encodings transform as $\pi \mathcal{H}_{ij} = \mathcal{H}_{\pi(i),j}$, coordinate matrices as $\pi \mathcal{X}_i = \mathcal{X}_{\pi(i)}$, and the latent features as $\pi \mathcal{F}_i = \mathcal{F}_{\pi(i)}$. The concatenation operation commutes with permutation appropriately,

$$[\sigma \mathcal{H}, \sigma \mathcal{X}, \sigma \mathcal{F}] = \sigma[\mathcal{H}, \mathcal{X}, \mathcal{F}]. \quad (3.3)$$

This would normally ensure that predictions are invariant to arbitrary relabelling of atoms. However, due to the nature of interfacing with neural networks, we show in Section 6 that this is not entirely the case.

The zero initialisation of the latent vector $\mathcal{F}$ ensures that the identity map is learnable, allowing the network to bypass layers if needed through the residual structures in later updates. This is inline with what is known as the residual learning principle: by initialisation to zero, the network starts this subsection in the form of the identity transformation and only learns the residual corrections needed for the task. Any non-zero initialisation would act as a bias for early layer throughput and gradients before any meaningful features may be extracted from the molecular structure. Formally, $\mathcal{F}^0 = 0$ ensures that $\frac{\partial J}{\partial \mathcal{F}^0}$, where J is an arbitrarily chosen loss function (see Eq. (2.14), reflects only the molecular structure encoded in $\mathcal{H}$ and $\mathcal{X}$, rather than being biased by random initialisation values.

The dimensionality d_h scales with the datasets atomic diversity and has a direct correlation to computational cost. For the MD17 dataset [33], $d_h = 4$ (H,C,N,O). Larger d_h increases the expressivity of the one-hot encoded representation, but scales the message

complexity linearly, as Ψ_{msg} processes inputs of dimension $(d_h + 2n)$.

The deliberate decomposition of our node feature matrix into separate components allows us to maintain physical interpretability, which we see as superior to learned embeddings. The atomic identities $\mathcal{H}$ remain human-readable throughout the entire training process. The coordinates $\mathcal{X}$ can be extracted at any point but crucially, $\mathcal{X}'$ can be expected to understand learned displacement fields within the model. Finally, the latent features vector $\mathcal{F}$ accumulate abstract structural information from $\mathcal{X}'$ without obfuscating the underlying physical quantities. This design choice allows for easy debugging as equivariance violations can be isolated to specific components, and the interpretable features give us a sanity check against known molecular behaviour at each layer of the network.

3.1.2 Edge Feature Construction via Adjacency Matrix

The adjacency matrix $\mathcal{A}$ and edge matrix $\mathcal{E}$ are constructed to encode Coulombic interactions between atomic pairs. The Coulomb matrix is formulated as,

$$C_{ij} = \begin{cases} \frac{Z_i^{2.4}}{2} & \text{for } i = j, \\ \frac{Z_i Z_j}{R_{ij}} & \text{for } i \neq j, \end{cases} \quad (3.4)$$

where Z_i represents the nuclear charge of atom i [34]. Here, the diagonal elements of this matrix represent a polynomial fit of atomic energies to nuclear charge. Due to the scalar nature of these features, and the absence of weighted edges in our use case, we can remove some computational redundancy by setting the values of the Coulomb matrix as the values in the adjacency matrix and reducing the amount of times we need to refer to differing matrices of the same values. If desired, we could still utilise the network with only the adjacency matrix, $\mathcal{A}_{ij} = \mathcal{E}_{ij} = C_{ij}$.

Despite already encoding atomic species and positions within the node matrix $\mathcal{V}$, we argue that these edge features provide a non-redundant role. The Coulomb matrix elements provide precomputed interaction potentials that can encode an interaction potential directly into the graph for a small computational cost, rather than requiring the network to derive these interactions through multiple message passing interactions.

We are essentially providing the network with a predetermined domain knowledge, as otherwise it would have to discover that,

- Electrostatic interactions follow a $\frac{1}{r}$ relationship.
- Interaction strengths depend on the product of nuclear charges $Z_i \times Z_j$.
- Self-energies scale with $Z^{2.4}$.

The integration of this domain knowledge allows us to accelerate the convergence of our model as without it, the network would require many more training examples and epochs to simply learn that two oxygen atoms interact more strongly than two hydrogen atoms and that this decreases with increasing distance. By being efficient with our limited computational resources, we can immediately leverage this information to learn the more subtle quantum mechanical effects that determine the target of interest, molecular forces.

3.2 EMPL Architecture

Having established how the node and edge feature matrices can be used to represent physical systems, we move to detail the architectural design of the equivariant message passing layer. This layer serves as the fundamental computational unit, transforming the graph through four stages, each designed to preserve $E(3)$ -equivariance whilst extracting structural information from the molecular geometry. The aforementioned stages are detailed in the following sections.

3.2.1 Distance-Based Feature Extraction

To maintain translation equivariance, all computations will utilise relative distance matrices rather than the absolute positions provided from the available datasets. Given the coordinate matrix $\mathcal{X}$, we extract the following distance matrices,

$$\mathcal{D}_{ij} = \|\mathcal{X}_i - \mathcal{X}_j\|_2, \quad (3.5)$$

$$\mathcal{D}_{ij}^2 = \|\mathcal{X}_i - \mathcal{X}_j\|_2^2. \quad (3.6)$$

Both of these will be utilised throughout the EMPL. They adhere to our constraints of rotational and translational invariance and form the foundation of our equivariant message construction.

The squared distance matrix $\mathcal{D}^2$ possesses differentiability everywhere in its domain, in contrast with the simple Euclidean distance matrix $\mathcal{D}$, which exhibits a gradient discontinuity at coincident points. The necessity for both $\mathcal{D}$ and $\mathcal{D}^2$ arises from their complementary roles. $\mathcal{D}$ provides geometric interpretability due to it's direct correspondence to the physical placement of atoms in a structure, whilst $\mathcal{D}^2$ ensures a smoother gradient flow during backpropagation.

Both distance matrices exhibit invariance under the full Euclidean group $E(3)$, encompassing the rotations of $SO(3)$ but also all reflections in 3D space, as for any orthogonal transformations $\mathbf{R} \in O(3)$ we have,

$$\mathcal{D}(\mathbf{R}\mathcal{X}) = \mathcal{D}(\mathcal{X}) \quad \forall \mathbf{R} \in O(3). \quad (3.7)$$

3.2.2 Message Computation

The computation of our messages within the network follows a two-stage transformation encapsulated in the following equation,

$$\mathbf{m} = \Phi_{\text{msg}}(\Psi_{\text{msg}}([\mathcal{H}, \mathcal{D}^2, \mathcal{A}])), \quad (3.8)$$

where,

- $\Psi_{\text{msg}} : \mathbb{R}^{n \times (d_h + n + n)} \rightarrow \mathbb{R}^{n \times 64}$,
- $\Phi_{\text{msg}} : \mathbb{R}^{n \times 64} \rightarrow \mathbb{R}^{64 \times 3}$.

The intermediate representation dimension of 64 arises from in-house testing showing that, through increasing dimensionality in powers of two, 128 and 256 dimensions yielded no noticeable improvement in trainability and predictive accuracy of the network, yet lowering the inner layer size to 32 resulted in slight underfitting of the data. We note that the difference was slight and it did not seem to steadily decrease, with some models with a

layer size of 4 still performing to a reasonable standard. We thus choose to implement the 64 neuron size layer as it sits in a reasonable middle ground between accuracy and computational cost, without the diminishing returns of increasing to a higher number of neurons within the layer.

Both of these layers further incorporate the swish activation function to provide smooth, non-monotonic behaviour to our data flow, which can help mitigate against gradient explosion whilst maintaining differentiability across the domain.

$$f_{swish}(x) = \frac{x}{1 + e^{-x}}. \quad (3.9)$$

We also note that the gradient of the swish activation function takes the following form,

$$f'(x) = f_{swish}(x) + \sigma(x)(1 - f_{swish}(x)), \quad (3.10)$$

where σ here denotes the sigmoid function. Critically, this prevents the emergence of a known issue within ReLU-based architectures with dead neurons [35]. The gradient magnitude of the swish activation function decays more gradually than ReLU in the negative domain. This maintains non-zero gradient flow, even for strongly negative activations, and therefore doing a large part in mitigating the vanishing gradient problem, which is inherent within deep neural networks.

3.2.3 Computational and Memory Complexity

The computational complexity of a single forward pass of the EMPL scales with $O(n^2 d_h + n^2 d_m + n d_m^2)$, where n is the number of atoms, d_h is the node feature dimension, and d_m is the message dimension. The dominant term arises from the pairwise distance computations $\mathcal{D}$ and $\mathcal{D}^2$ from Eq. (3.5) and Eq. (3.6). These require $O(n^2)$ operations per spatial dimension, resulting in $O(3n^2)$ for 3D Euclidean space. The message construction processes the concatenated features $[\mathcal{H}, \mathcal{D}^2, \mathcal{A}] \in \mathbb{R}^{n \times (d_h + 2n)}$, resulting in $O(n \times (d_h + 2n) \times d_m) \approx O(n^2)$ complexity. The subsequent message passing dense layer operations within Φ_{msg} require $O(n \times d_m^2)$ operations.

The memory requirements of this architecture scale as $O(n^2 + (n d_h + n d_m))$. This is

dominated by the storage capacity of the square adjacency matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$ and the square distance matrices $\mathcal{D} \in \mathbb{R}^{n \times n}$. The gradient computation during backpropagation maintains the same overall complexity, but requires additional $O(nd_m)$ allocation for the outputs of each layer to be stored whilst computing the gradients via the chain rule.

For a network depth L , the total complexity becomes $O(L \cdot n^2 \cdot \max\{d_h, d_m\})$, highlighting a quadratic bottleneck for large molecular systems. This motivates the imposed constraint that the total number of message passing layers will not surpass three, as this strikes the perfect balance between providing a sufficient receptive field coverage for learning the complex underlying patterns, whilst maintaining tractability for most organic molecules.

3.2.4 Equivariant Update Mechanism

The update procedure allows us to preserve the structure of the node matrix whilst being able to selectively modify the latent representation. The equations governing this mechanism are,

$$\mathcal{H}' = \Phi_{\mathcal{H}}(\Psi_{\mathcal{H}}(\mathcal{H}, \mathbf{m})), \quad (3.11)$$

$$\mathcal{X}' = \mathcal{X} + c \cdot \mathcal{D} \cdot \Psi_{\mathcal{X}}(\mathbf{m}), \quad (3.12)$$

$$\mathcal{F}' = \mathcal{F} + \mathcal{X}', \quad (3.13)$$

where $c = \frac{1}{N-1}$ is a normalisation constant, with N being the number of atoms in the system, and $\Phi_{\mathcal{H}}, \Psi_{\mathcal{H}}, \Psi_{\mathcal{X}}$ are learnable transformations of densely connected neural network layers.

The coordinate update $\mathcal{X}'$ allows us to maintain equivariance through the additive combination of the original coordinates and the displacement term extracted from our message passing scheme. The use of a scaling factor here is simply to scale our outputs in accordance with overall system size.

The normalisation constant $c = \frac{1}{n-1}$ is implemented in an attempt to prevent coordinate drift (unbounded growth of updates with system size) as the system size varies. We require the per-atom displacement magnitude, given as $\|(\Delta\mathcal{X})_i\|$, to scale with $O(1)$, completely independently of N . This would ensure that each individual atomic update remains bounded irrespective of the size of the molecule being analysed. The displacement for atom i is given

as,

$$(\Delta\mathcal{X})_i = c \sum_j \mathcal{D}_{ij}(\Psi_j(\mathbf{m})), \quad (3.14)$$

and aggregates contributions from $n - 1$ neighbours. By having $c = \frac{1}{n-1}$, we normalise the actual number of contributing neighbours and maintain per-atom stability.

Furthermore, the feature update $\mathcal{F}' = \mathcal{F} + \mathcal{X}'$ constitutes a residual learning architecture, analogous to models like ResNet [36]. This formulation enables effective training of deep architectures by circumventing the gradient degradation problem inherent in molecular GNNs. The gradient flow through residual connections can be analysed through the chain rule. For our formulation $\mathcal{F}^{(l+1)} = \mathcal{F}^{(l)} + \text{EMPL}^{(l)}(\mathcal{F}^{(l)})$, the gradient with respect to the layer l becomes,

$$\frac{\partial J}{\partial \mathcal{F}^{(l)}} = \frac{\partial J}{\partial \mathcal{F}^{(L)}} \prod_{i=l}^{L-1} \left(\mathbb{I} + \frac{\partial \text{EMPL}^{(i)}}{\partial \mathcal{F}^{(i)}} \right). \quad (3.15)$$

Expanding this product reveals an additive decomposition,

$$\frac{\partial J}{\partial \mathcal{F}^{(l)}} = \frac{\partial J}{\partial \mathcal{F}^{(L)}} + \text{correction terms}, \quad (3.16)$$

where the correction terms contain derivatives of EMPL layers but, crucially, the first term propagates unchanged throughout all residual connections. This direct path prevents the vanishing gradient problem inherent in deep feedforward networks, where gradients decay exponentially when activation derivatives are less than unity. The residual formulation thus enables a stable gradient flow across our EMPL architecture, without any multiplicative attenuation.

3.2.5 Output Layer Formulation

The output sub-layer of the EMPL involves the recompilation of our original graph structure with the updated embeddings included within the graphs matrices. As we want to propagate the original coordinates and the atomic connectivity throughout this process, we only update the latent features as follows,

$$\text{EMPL}(\mathcal{G}(\mathcal{V}, \mathcal{A}, \mathcal{E})) = \mathcal{G}'(\mathcal{V}', \mathcal{A}, \mathcal{E}), \quad (3.17)$$

where,

$$\mathcal{V}' = [\mathcal{H}, \mathcal{X}, \mathcal{F}']. \quad (3.18)$$

This design choice allows us to prioritise the accumulation of structural information within the latent space to be stored within $\mathcal{F}$, whilst maintaining physical meaning and interpretability of the atomic identities and positions. In particular, the decision to hold $\mathcal{A}$ and $\mathcal{E}$ fixed comes from the physical constraint that graph topology naturally encodes chemical bonding, which represents a molecular invariant and should not be subject to learning in our relatively simplistic analysis case, as bond breaking and formations are not assumed nor desired to be modelled for the architecture at this stage. These processes require crossing significant energy barriers, which are not captured by continuous, gradient-based updates. Thus the preservation of $\mathcal{A}$ and $\mathcal{E}$ enforces chemical accuracy throughout the initial and subsequent forward passes.

Restricting updates to $\mathcal{F}$ alone grants us further memory efficiency as only updating $\mathcal{F}$ requires $O(3n)$ storage for intermediate gradients, compared to the $O(n \times (d_h + 6))$ needed for full updates of $\mathcal{V}$. This is particularly important for computational setups with limited GPU resources as this constitutes an over 50% reduction in memory for this portion of the model, and will be critical in future expansions of the architecture.

3.2.6 Mathematical Properties

We can show that both translational and rotational equivariance are maintained throughout this process.

Consider the transformation $\mathcal{X} \mapsto \mathcal{X} + \mathbf{t}$ for any translation vector $\mathbf{t} \in \mathbb{R}^3$. All operations within this scheme are dependant only on the matrices $\mathcal{D}$ and $\mathcal{D}^2$. These matrices are invariant under translation as shown by,

$$\|\mathbf{x}_i + \mathbf{t} - (\mathbf{x}_j + \mathbf{t})\|_2 = \|\mathbf{x}_i - \mathbf{x}_j\|_2 \quad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}, \mathbf{t} \in \mathbb{R}^3. \quad (3.19)$$

Thus the message computation process leaves this unchanged, insuring that $\mathcal{F}'(\mathcal{X} + \mathbf{t}) = \mathcal{F}'(\mathcal{X})$ and translation equivariance is adhered to.

The proof for rotational equivariance follows from the above invariance of distance metrics

and the utilises the additive update structure. For any rotation matrix $\mathbf{R} \in SO(3)$ we have,

$$\mathcal{X}'(\mathbf{R}\mathcal{X}) = \mathbf{R}\mathcal{X} + c \cdot \mathcal{D} \cdot \Psi_{\mathcal{X}}(\mathbf{m}) = \mathbf{R}(\mathcal{X} + \mathbf{R}^\top \cdot c \cdot \mathcal{D} \cdot \Psi_{\mathcal{X}}(\mathbf{m})), \quad (3.20)$$

where the displacement term $c \cdot \mathcal{D} \cdot \Psi_{\mathcal{X}}(\mathbf{m})$ transforms appropriately under the action of $\mathbf{R}$. The distance matrix $\mathcal{D}$ remains invariant under rotation, since $\|\mathbf{R}(x_i - x_j)\|_2 = \|x_i - x_j\|_2 \forall \mathbf{R} \in SO(3)$. The learned transformation $\Psi_{\mathcal{X}}(\mathbf{m})$ produces a directional vectors in $\mathbb{R}^{n \times 3}$ that encode neighbourhood information and, importantly, the messages $\mathbf{m}$ themselves are constructed such that they are comprised of entirely invariant features. This invariance of the input features ensures that under rotation, the output of $\Psi_{\mathcal{X}}$ must transform equivariantly, as the mapping depends only on scalar and distance-based inputs, which themselves are invariant to rotation, giving,

$$\Psi_{\mathcal{X}}(\mathbf{m})(\mathbf{R}\mathcal{X}) = \mathbf{R} \cdot \Psi_{\mathcal{X}}(\mathbf{m})(\mathcal{X}). \quad (3.21)$$

The matrix product $\mathcal{D} \cdot \Psi_{\mathcal{X}}(\mathbf{m})$ inherits the property of equivariance, and the normalisation constant c is scalar, thus commuting with $\mathbf{R}$ and not affecting the property of equivariance. Substituting this into the factorisation yields $\mathcal{X}'(\mathbf{R}\mathcal{X}) = \mathbf{R}(\mathcal{X} + c \cdot \mathcal{D} \cdot \Psi_{\mathcal{X}}(\mathbf{m})) = \mathbf{R} \cdot \mathcal{X}'(\mathcal{X})$, satisfying the required equivariance condition for all $\mathbf{R} \in SO(3)$.

A diagram displaying the data flow of the overall layer is shown below in Fig. 3.1.

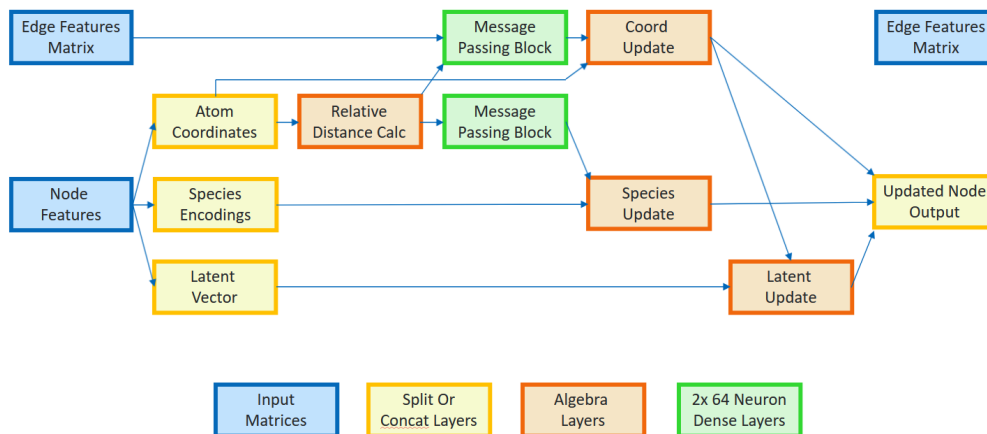


Figure 3.1: A diagram showing the data throughput for each instance of our equivariant message passing scheme. Blue boxes indicate input and output matrices, yellow indicates matrices derived from splitting or concatenating input or output matrices, orange pertains to mathematical operations, and green signifies out message passing blocks, outlined in Section 3.2.2.

3.3 Advanced EMPL Implementation Details

A number of commonly implemented measures pertaining to model trainability and stability cannot be blindly implemented in the off-the-shelf manner that many utilise, as doing so would violate many of our symmetry requirements previously stated. We detail the considerations and corrections made to these methods below.

3.3.1 Numerical Stability Considerations

The implementation of numerical stability measures such as batch normalisation, gradient clipping and residual connection interactions are commonplace within complex GNN architectures. However, there is a fundamental trade-off between training stability and the preservation of mathematical symmetries. Naive application of these techniques without second thought can unintentionally break the properties that guarantee equivariance in our representation.

Batch normalisation is a process which seeks to normalise a layers inputs by subtracting the mean value of a given batch of data before dividing by the standard deviation of the same subset of data [37][38]. This is then usually followed up with some affine transformation,

usually a linear activation function. It is argued that this can stabilise training as it ties each layers inputs to have consistent statistical characteristics, namely a mean of zero and variance of one. This supposedly reduces internal covariate shifting and can allow for higher learning rates, allowing us to reach desired loss function minima with fewer epochs of training time. However, due to our use of atomic coordinates as node features, this process will directly violate translation equivariance.

If we apply this procedure to our coordinate features,

$$\mathcal{X}_{\text{norm}} = \frac{\mathcal{X} - \mu_{\text{batch}}}{\sigma_{\text{batch}}}, \quad (3.22)$$

where $\mu_{\text{batch}} = \frac{1}{|B|} \sum_{k \in B} \mathcal{X}^{(k)}$ and $\sigma_{\text{batch}}^2 = \frac{1}{|B|} \sum_{k \in B} \|\mathcal{X}^{(k)} - \mu_{\text{batch}}\|^2$ with B being the batch for a given subset of data used for an epoch in training.

This introduces a bias in the form of a preferred reference frame, dependent on whatever specific molecular configurations are present in a given batch B . Under a translation $\mathcal{X} \mapsto \mathcal{X} + \mathbf{t}$, the normalised features become,

$$\mathcal{X}_{\text{norm}+\mathbf{t}} = ((\mathcal{X} + \mathbf{t}) - \frac{\mu_{\text{batch}} + \mathbf{t}}{\sigma_{\text{batch}}}) = \mathcal{X} - \frac{\mu_{\text{batch}}}{\sigma_{\text{batch}}} = \mathcal{X}_{\text{norm}}. \quad (3.23)$$

This appears to preserve translation equivariance. However, upon completion of the training process, when the model is used for inference, the values of μ and σ become fixed. In the training process these are calculated with every new batch. The overall fixed *running statistics* create a fixed reference frame that will break equivariance, as there is no guarantee that any batch of data entering will have the same mean and standard deviation of batch data compared to these running statistics.

$$\mathcal{X}_{\text{norm}}^{\text{inference } (\mathcal{X}+\mathbf{t})} = (\mathcal{X} + \mathbf{t}) - \frac{\mu_{\text{running}}}{\sigma_{\text{running}}} \neq \mathcal{X}_{\text{norm}}^{\text{inference } (\mathcal{X})} - \frac{\mu_{\text{running}}}{\sigma_{\text{running}}}. \quad (3.24)$$

This breaks the fundamental requirement that $f(g \cdot x) = g \cdot f(x) \forall g \in G$ where G is the translations group in $\mathbb{R}^3$.

3.3.2 Gradient Clipping

Gradient clipping is a method to prevent exploding gradients. It achieves this by scaling gradient vectors any time the norm of that vector exceeds a predetermined threshold limit. By bounding the magnitude of parameter updates, this should increase model stability in the training process. However, this clipping operation involves non-smooth transformations that will interfere with equivariant learning if activated.

The clipping operation $\nabla_{\text{clipped}} = \nabla \cdot \min(1, \frac{\tau}{\|\nabla\|_2})$ where τ is the maximum allowed gradient norm. This introduces a non differentiability at $\|\nabla\|_2 = \tau$. Furthermore, this act operates on the overall gradient norm.

$$\|\nabla_{\mathcal{X}} \cdot J\|_2 = \sqrt{\|\nabla_{\text{scalar}} \cdot J\|^2 + \|\nabla_{\text{vector}} \cdot J\|_F^2}, \quad (3.25)$$

where J is the Jacobian matrix of the loss function with respect to the model parameters, $\nabla_{\text{scalar}} \cdot J \in \mathbb{R}^n$ and $\nabla_{\text{vector}} \cdot J \in \mathbb{R}^{n \times 3}$ and the subscript F denotes the Frobenius norm of the given matrix.

Under any rotation $\mathbf{R} \in SO(3)$, we desire equivariant gradients to adhere to $\nabla_{\mathcal{X}} \cdot J(R \cdot \mathcal{X}) = R \cdot \nabla_{\mathcal{X}} \cdot J(\mathcal{X})$. Clearly, when we have that both $\|\nabla_{\mathcal{X}} \cdot J(\mathcal{X})\|_2 \leq \tau$ and $\|\nabla_{\mathcal{X}} \cdot J(R \cdot \mathcal{X})\|_2 \leq \tau$ (due to differing scalar/vector magnitudes), we create an inconsistency within scaling, breaking the desired rotational equivariance.

If we only apply normalisation to scalar invariant features, we can preserve the coordinate features. We can opt to mitigate these issues through selective feature normalisation and also through the implementation of an equivariant gradient scaling scheme.

Finally, we can replace the uniform clipping measure with a component-wise scaling as follows,

$$\nabla_{\text{eq-scaled}} = \min\left(1, \frac{\tau_{\text{scalar}}}{\|\nabla_{\text{scalar}} \cdot J\|}\right) \nabla_{\text{scalar}} \cdot J + \min\left(1, \frac{\tau_{\text{vector}}}{\|\nabla_{\text{vector}} \cdot J\|_F}\right) \nabla_{\text{vector}} \cdot J, \quad (3.26)$$

where τ_{scalar} and τ_{vector} are predetermined thresholds which preserve the scalar-vector decomposition.

Chapter 4

Molecular Datasets

4.1 MD17

The MD17 dataset instantly became very popular for the benchmarking of machine-learned force field, especially with GNN architectures [33]. It contains Molecular Dynamics (MD) trajectories for ten small organic molecules and provided a good test bed for rudimentary models due to the relatively small structures involved. Each calculation was performed using DFT, making use of the PBE+vdW-TS functional, implying a decent level of accuracy in both the provided energy and forces. It is comprised of tens of thousands of configurations per molecule and samples from room temperature MD trajectories computed with the FHI-aims code [39]. This extensive sampling intends to capture the full range of thermally accessible configurations and provides a comprehensive coverage of the potential energy surface of common organic molecules near equilibrium. However, this dataset was fraught with issues.

The dataset exhibits strong temporal correlations, which violate an important and crucial assumption in the devising of training and test sets, the Independent and Identically Distributed (IID) principle. The presence of this temporal correlation within the structure of the dataset causes *data leakage*, a phenomenon where data that should not be available to the model in the training set is present. For example, a molecule’s properties at a given time t are heavily influenced by the dataset entry for $t - \delta$ and will furthermore strongly influence the entry at $t + \delta$. This implies that the test data contains information highly correlated to the training set, which leads to an inflation of performance metrics and over-reaching benchmarks that do not at all reflect the true underlying predictability of any given model

trained on this dataset. The dataset exhibits strong temporal correlations, violating the independent and identically distributed assumptions of the dataset.

Configurations from MD trajectories are not guaranteed to be independent samples, reportedly evident from the autocorrelation function from the original MD17 dataset [40].

Furthermore, later studies showed severe limitations within the dataset’s configuration space coverage. Analysis by Leng et al. showed ”a significant sampling bias towards a narrow potential energy surface region close to the equilibrium structure.” This was later quantified to show an energy distribution limited to a maximum energy of ~ 30 *kcal/mol* [41][42][43].

The revised MD17 dataset was introduced by Christensen et al. [40], with the intention of addressing some of these issues. Whilst tackling issues relating to numerical noise and selectively decreasing the size of the dataset to attempt to avoid temporal correlation issues, these issues seem to persist, as identified by Peng et al. [44]. They note that the dataset exhibits the same sampling bias towards a narrow potential energy surface window near equilibrium along with the problematic energy distribution. Overall this revision marked an incremental improvement over the original, but still needed transformative improvement. The revised dataset reduced the configuration count to approximately 10,000 per molecule and recalculated forces with tighter convergence criteria, improving numerical consistency. However, the fundamental sampling limitations, temporal correlations, and equilibrium biases remain unaddressed.

These persistent limitations manifest directly in model performance. Preliminary training on MD17 configurations across multiple molecules (uracil, paracetamol, aspirin, and toluene) yielded poor R^2 -score performance, and solidified the requirement for fundamentally improved data with stricter DFT convergence. This inadequacy is demonstrated in Section 5.1, where direct comparison reveals improvements in predictive performance for identical model architectures.

4.2 Custom Made Dataset

Having established that MD17’s limitations impose fundamental performance ceilings, we developed a custom dataset generation methodology that addresses these issues sys-

tematically. Rather than discarding MD17’s geometric information entirely, we leverage its configurational diversity as a starting point, while implementing three critical improvements. Namely, elimination of temporal correlations, expanding the configurational space coverage, and conducting DFT calculations within PySCF with tight convergence criteria enforced [45].

Our approach combines strategic coordinate sampling from the original MD17 trajectories with controlled geometric perturbations. This methodology recognizes that the original MD17 coordinates represent physically reasonable configurations despite their sampling limitations. By applying K -means clustering and targeted perturbations, we effectively break temporal correlations whilst preserving the chemically realistic molecular geometries that emerge from room-temperature dynamics, thereby constructing a dataset that spans a more diverse region of configuration space without sacrificing physical validity.

4.2.1 Configuration Space Sampling

With this desired outcome in mind, we are challenged with adequately sampling the original configuration space whilst still maintaining full computation tractability. The minimisation of temporal correlation effects was achieved via a method of clustering based on structural similarity, with the intent to only select one data instance per cluster, thus reducing the overall level of temporal correlation within the overall original dataset. The similarity metric imposed was,

$$s_{ij} = \sqrt{\sum_{\text{bonds}} w_{\text{bond}}(r_i^{\text{bond}} - r_j^{\text{bond}})^2 + \sum_{\text{angles}} w_{\text{angle}}(\theta_i^{\text{angle}} - \theta_j^{\text{angle}})^2 + \dots} \quad (4.1)$$

$$\sqrt{\dots + \sum_{\text{dihedrals}} w_{\text{dihedral}}(\phi_i^{\text{dihedral}} - \phi_j^{\text{dihedral}})^2},$$

where $w_{\text{bond}} = 10\text{\AA}^{-2}$ is the bond length weighting parameter, $w_{\text{angle}} = 1(\text{rad}^{-2})$ is the angle bending weighting parameter, and $w_{\text{dihedral}} = 0.5(\text{rad}^{-2})$ is the dihedral torsion weighting parameter. These weights are chosen such that bond length deviations contribute most strongly to the similarity metric, followed by angular deviations, and finally torsional deviations, reflecting the relative importance of these structural features in determining molecular

configuration.

This weighted similarity metric operates in the internal coordinate space rather than the Cartesian space, a feature that provides invariance to global translations and rotations whilst emphasising chemically meaningful structural variations. The bond length weight is set to one order of magnitude higher than the angular terms in order to apply the notion that bond stretching represents the stiffest degree of freedom compared to angle bending and torsional rotation. The dihedral weight is further reduced to 0.5 because torsional degrees of freedom exhibit larger thermal fluctuations and should not dominate the clustering metric.

Clustering the results of this ensures that, by selecting from each cluster, the set of chosen outputs will represent a diverse range of molecular configurations whilst mitigating the temporal correlation issues present in the overall dataset. We employed K -means clustering with k chosen such that the average cluster population is roughly 150 configurations per cluster, resulting in 1000 cluster centers from the original overall MD17 dataset. The cluster representatives were selected as the configuration closest to each cluster centroid with respect to the similarity metric space, ensuring that the selected configurations for our overall custom dataset are genuinely representative of the underlying dataset in the least spatially correlated way possible.

4.2.2 Geometric Perturbation

We apply a direct Cartesian coordinate distortion to the atomic position vectors of all instances that were selected from the configuration space sampling. For each base configuration $x_i \in \mathcal{X}$, we generate a component-wise random displacement of the form,

$$x_i^p = x_i + \delta_i, \quad (4.2)$$

where the perturbation vector $\delta_i = [\delta_{x_i}, \delta_{y_i}, \delta_{z_i}]$ is a random placement sampled from a uniform distribution,

$$\delta_{x_i}, \delta_{y_i}, \delta_{z_i} \sim U(-\epsilon |x_i|, \epsilon |x_i|), \quad (4.3)$$

where ϵ is a displacement parameter.

We set this displacement parameter to be $\epsilon = 0.1$, allowing us to generate perturbations

within $\pm 10\%$ of the original configuration. This choice of perturbation parameter was chosen through root mean square displacement analysis of the original dataset. This process strikes a good balance between the expansion of the configuration space available to cover, whilst maintaining chemical reasonableness. The bond lengths remain within 0.2 Å of the equilibrium values, with angle shifts typically not exceeding $10 - 15^\circ$, with no bonds being spuriously broken or malformed.

Where multiple perturbations were needed in order to achieve ideal dataset sizes, we generated k realisations per base configuration. These k samples would be run through our configuration sampling scheme in order to choose the most diverse results, mitigating a potential flare up in temporal correlation.

The target of 1000 configurations per molecule represents a pragmatic compromise between the computational cost of the necessary DFT calculations, with a dataset size sufficient for neural network training, without the probable introduction of overfitting in the model training process.

4.2.3 DFT Calculation Methodology

All electronic structure calculations were conducted with the PySCF computational chemistry package. The B3LYP hybrid functional was chosen and we utilised the correlation-consistent cc-VTZ basis set.

The B3LYP functional incorporates 20% of the Hartree-Fock exchange functional along with 80% density functional exchange-correlation. This choice should provide a methodological improvement over the use of PBE+vdW-TS from the MD17 calculations. There are, however, some known systematic limitations when this functional is used in the analysis of large molecules and in instances involving dispersion-dominated interactions. These issues were highlighted by Grimme, who demonstrated that B3LYP underestimates dispersion interactions by up to 50% for prototypical systems such as stacked aromatic dimers, but should not pertain to the molecules within our chosen dataset [46][47]. It will still provide greatly improved accuracy compared to functionals similar to PBE for this use case. The molecules contained within MD17 are relatively small, all containing fewer than 25 atoms with limited opportunities for long range dispersion interactions. Thus, B3LYP’s known

accuracy for covalent bond energies, geometries and vibrational frequencies make it a solid choice for our work. Furthermore, the 20% exact exchange component will help to reduce the self-interaction error that is commonplace within pure DFT functionals, improving force predictions in regions of stretched or compressed bonds which may arise from our molecular perturbation scheme

The cc-pVTZ basis set represents Dunning’s correlation-consistent triple- ζ basis set. This includes polarisation functions, which are designed with the goal of achieving full systematic convergence towards the complete basis set limit [48][49]. In particular, for the molecular sizes involved with the MD17 dataset, this choice of basis-set provides near complete basis set convergence based on organic molecule benchmarks. Specifically, the cc-pVTZ employs contracted Gaussian functions with angular momentum of up to $l = 3$ for second-row elements, particularly C, N, and O, and $l = 2$ for hydrogen. This totals approximately 150 – 350 basis functions, depending on overall molecular size. The choice of this basis function thus provides confidence that our results are not artifacts of basis set incompleteness, whilst still remaining computationally tractable.

Through strict enforcement of $F_i = \frac{-\partial E}{\partial x_i}$, we can remove numerical inconsistencies that effect the original dataset. Where energy and force calculations were previously conducted with different convergence parameters, PySCF ensures this is rectified through this criteria. Each calculation cycle is set with small convergence criteria, namely 10^{-8} Hartree for energy convergence and a density convergence of 10^{-6} . These tight convergence thresholds ensure that the forces calculated via analytical gradients of the converged electronic density are suitably numerically consistent with the energy surface. This represents a substantial improvement over the MD17 dataset forces, as its forces were computed via finite differences, and introduced possibilities of noise depending on the local curvature of the potential energy surface within that scheme. The Direct Inversion of Iterative Subspace (DIIS) method is implemented to accelerate convergence through accuracy of initial density guesses. The DIIS algorithm, which is implemented in PySCF, stores the previous eight Fock matrices and error vectors by default and constructs an extrapolated solution that minimises the residual commutator $[F, \rho S]$ where F is the Fock matrix, ρ is the density matrix, and S is the overlap matrix. DIIS acceleration typically reduces the number of SCF iterations required for overall

convergence, although exact speedup statistics are dependent on the size of the molecular system and the quality of the initial guess for the system.

PySCF also implements level-shifting, a method of adding a constant energy shift to unoccupied orbitals during calculations in order to prevent oscillations. This is mainly utilised in systems with small HOMO-LUMO gaps and are of little concern here. The level shift parameter effectively increases the gap between the occupied and virtual orbitals during the SCF procedure. This is done in an attempt to stabilise the convergence for near-degenerate electronic states. For the closed-shell organic molecules within our dataset, the typical HOMO-LUMO gaps are sufficiently large that level shifting is rarely invoked, doing so in less than $\sim 5\%$ of calculations, specifically for some of the highly perturbed instances within the dataset, where bond compression or stretching may temporarily reduce the gap.

Configurations exhibiting issues in convergence are excluded from the final dataset. This approach gives us the knowledge that all instances present in this finalised custom database have met all specified accuracy criteria and, to the best of our ability, do not fall foul of the issues that were present within the MD17 dataset. The total process resulted in single-digit number of unconverged molecules, which were removed and resampled from pool detailed previously in Section 4.2.1. The process did not then need to be repeated. The resulting dataset gave us 1000 usable configurations each of Aspirin, Azobenzene, Benzene, Ethanol, Malonaldehyde, Naphthalene, Paracetamol, Salicylic Acid, Toluene and Uracil.

Chapter 5

Results - Molecules

Initially, benzene, uracil and paracetamol molecules were chosen to undergo many variations of model training with slightly varying hyperparameters in order to test which would be most suitable for testing all molecules on. These included varying the number of epochs per runs, number of message passing layers, L1/L2 regularisations, dropout layers and batch size variations. The parity plots corresponding to the best performing model for each molecule are given below with the respective hyperparameter settings for each described. Assume results were conducted on a 25,000-epoch run with a batch size of 4, with no additional hyperparameter features, henceforth referred to as the 'base case setup'. Metrics include Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 scores.

The results presented in Table 5.1 reveal a clear pattern in model performance that strongly correlates with molecular symmetry properties and compositional diversity. We can separate these results into three distinct performance tiers for analysis.

The highest performing molecules are indicated most prominently by their high R^2 -scores. These include uracil, naphthalene, salicylic acid, toluene and azobenzene. These all achieve near DFT-level accuracy, with MSE values below $0.06 \text{ (eV/\AA}^2\text{)}$. These molecules share the property of either high compositional diversity, with uracil containing four distinct atomic species for example, or spatial asymmetries that effectively break representational degeneracies despite geometric regularity.

The moderate performance tier includes paracetamol, aspirin, and benzene. These exhibit acceptable but noticeably degraded performance. The presence of benzene in this tier, despite its small size, provides compelling evidence for symmetry-induced training complications, which we indeed found and will explore in Section 6. There we show that Benzene's

Results for MD-Custom-Made PySCF Dataset

Molecule	MSE	MAE	R^2
Uracil	0.01867	0.09832	0.99024
Naphthalene	0.01701	0.09716	0.98926
Salicylic	0.01870	0.09963	0.98881
Toluene	0.04585	0.16088	0.96863
Azobenzene	0.06288	0.18410	0.96124
Paracetamol	0.11028	0.24540	0.93078
Aspirin	0.17688	0.30784	0.88963
Malonaldehyde	0.22379	0.35700	0.87453
Benzene	0.08738	0.22053	0.88755
Ethanol	0.24223	0.38288	0.81679

Table 5.1: MSE, MAE and R^2 results for the PySCF dataset, with corresponding parity plots shown in the following figures. MSE values are reported in units of (eV/Å²) for component-wise atomic forces. The coefficient of determination R^2 is computed as $R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$ where $\hat{y}_i$ are the predicted values and $\bar{y}$ is the mean of true values across all Cartesian components and all atoms. These results are presented in order of R^2 -score, and the table has been segmented to highlight the different groups of molecules discussed within the results.

D_{6h} point group symmetry, combined with compositional homogeneity, being comprised exclusively of carbon and hydrogen atoms, creates maximal node alignment ambiguity within our representational framework, and thus contributes to degraded model performance.

Lastly, we have the lowest performing molecules, malonaldehyde and ethanol. Models trained on these molecules demonstrate the poorest predictive capabilities. Unlike the high-performing molecules, these systems lack aromatic ring structures, which provide both geometric rigidity and distinctive π -electron distributions that facilitate feature learning. The geometric rigidity constrains these molecules to a narrower region of the configuration space, reducing the conformational variability the network must learn. Additionally, the delocalised π -electron systems create distinctive spatial patterns in the electronic density that the network can exploit as stable, learnable features. Without these structural anchors, these flexible molecules span multiple conformational basins through internal rotations about C-C and C-O bonds, creating substantially different geometric arrangements relative to the more rigid molecules in the dataset. The network thus encounters greater variability in distance matrix patterns for these systems, leading to a need for relatively more data instances in order to train a model to the same high standard exhibited by models trained on the other

molecules within this dataset.

5.1 Error and Distribution Analysis

The mean absolute error (MAE) provides complementary information to the mean squared error (MSE) by quantifying typical prediction deviations without the quadratic penalty that emphasises outliers. The MAE values span nearly a factor of four, from 0.097-0.383 eV/Å for Naphthalene to Ethanol respectively. This range significantly exceeds what would be expected from molecular size variations alone, reinforcing our theory that representation quality or ambiguity, rather than system complexity, is driving errors within our prediction accuracy across molecules.

Furthermore, we show that our custom PySCF generated dataset is superior to the original MD17 dataset by highlighting the large-scale errors between models trained on the two differing datasets with the same underlying model architecture, highlighted by the parity plots of model predictability between training on the two datasets in Fig .5.1. Note that the MD17 results are shown with forces scaled to the interval $[0, 1]$, as targetting this scaled set of values produced the best predictive model for that data. All other plots correctly display unscaled targets and predictions.

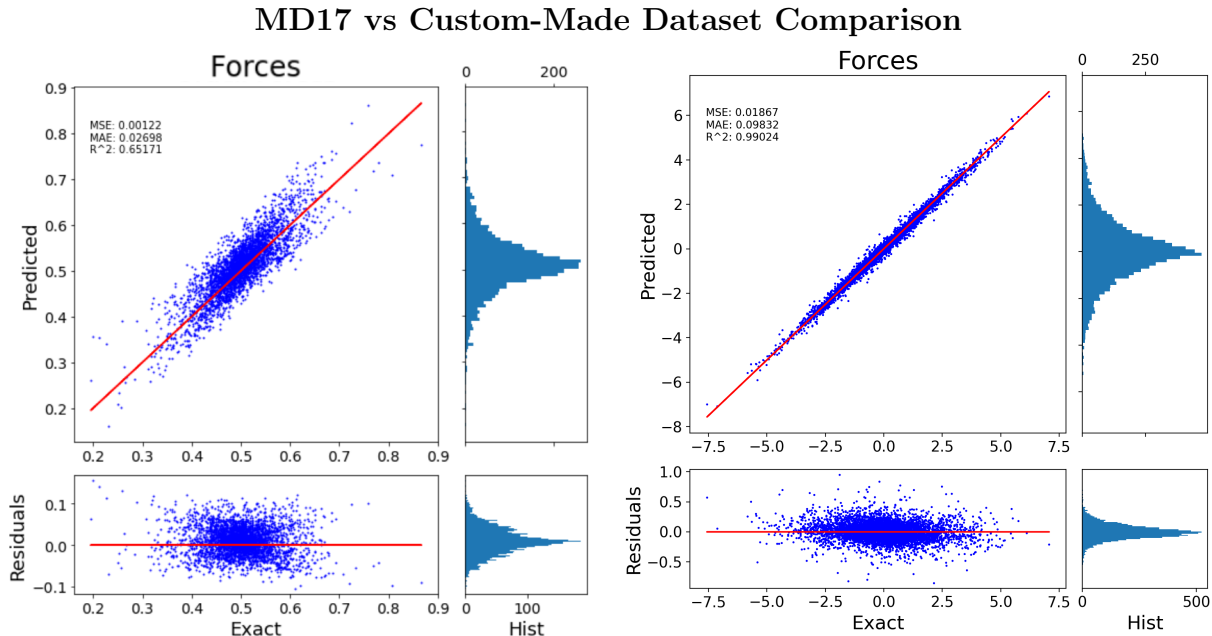


Figure 5.1: Parity plots for the atomic forces obtained with our graph-network architecture trained on the original MD17 data (upper panels) and our own DFT dataset (lower panels). The MD17 forces have been scaled to the interval $[0, 1]$ for convergence purposes detailed in/ Section 5.1. Results are presented for the uracil molecule. Clearly, a less noisy dataset as the one we produced by enforcing tight convergence criteria correlates to a better trained model under identical architecture. Together with the parity plots we present the residue and histograms of the various distributions. The mean squared error (MSE), mean absolute error (MAE) and R^2 coefficient for each model are reported in the legends.

The dramatic performance difference shown in Fig. 5.1, despite using identical architectures, definitively establishes that MD17’s limitations impose fundamental accuracy ceilings. Our custom-made dataset’s tight DFT convergence criteria and strategic sampling methodology can be seen to translate directly to superior model trainability.

5.2 Hyperparameter Testing

We conducted systematic hyperparameter searches across all molecules, varying training duration, learning rates, regularisation strategies, and architectural parameters. Whilst optimal configurations exhibited some molecule-specific variation, improvements over baseline configurations were generally marginal, suggesting our architecture is relatively robust to hyperparameter perturbations.

Hyperparameter Search Space		
Hyperparameter	Range Explored	Final Value
Training Epochs	10^3 to 10^6	25,000
Learning Rate η	10^{-3} to 10^{-7}	10^{-4}
Dropout probability	0.0 to 0.3	0.0
L1 Regularisation λ_1	0.0 to 10^{-4}	0.0
L2 Regularisation λ_2	0.0 to 10^{-2}	0.0
Message Layer Neurons	4, 8, 16, 32, 64	64
Number of Message Passing Layers	1, 2, 3, 4	3
Batch Size	1, 2, 4, 8, 16, 32	4

Table 5.2: Summary of hyperparameter search space explored for each molecule. The final column indicates baseline configurations, with molecule-specific deviations noted in the text.

Training duration exhibited diminishing returns beyond approximately 25,000 epochs for most molecules. Aspirin was an exception, wherein the extension to 100,000 epochs improved the R^2 -score by approximately 0.02, however this is within the realm of noise variation and alterations via random initialisation. Conversely, benzene showed minimal improvement beyond 25,000 epochs, possibly indicating that the performance ceiling is determined by representation limitations arising from high symmetry rather than insufficient training.

Learning rate variations revealed a broad optimal region around $\eta = 10^{-4}$, with rates exceeding 10^3 causing instabilities in training, and rates below 10^{-6} resulting in prohibitively slow convergence. Explicit $L1$ and $L2$ regularisation provided negligible benefits.

Dropout regularisation proved beneficial only for ethanol, where probability 0.15 enabled an improvement of approximately 0.04 in R^2 . This suggests that ethanol’s conformational flexibility creates effective noise in the training data that dropout can help mitigate.

The number of neurons within each layer of the learnable message passing functions was also analysed. Layer widths below 16 resulted in systematic underfitting, with 64 being the optimal number. The number of message passing layers stacked after each other was also investigated, with three layers benefitting all models for all molecules. In particular, uracil demonstrated a 0.02 rise in R^2 -score with three layers compared to one, two, or four, and

this was seen consistently throughout other molecules.

The baseline configuration proved robust across most molecules, with deviations limited to extended training for aspirin, naphthalene, paracetamol, and salicylic acid; dropout for ethanol; reduced batch sizing for toluene; and deeper architectures for uracil. The marginal nature of these improvements underscores that performance variations across molecules arise primarily from intrinsic symmetry-induced representational ambiguities rather than suboptimal hyperparameters.

5.3 Molecular Parity Plots

The parity plots below provide detailed visual assessment of model performance across all ten molecules in the custom PySCF dataset. Each plot displays predicted force components against DFT reference values, with accompanying residual distributions and histograms that reveal the quality and consistency of predictions. The red parity line represents perfect agreement between predicted and reference forces, with deviations from this line indicating prediction errors. The residual plots in the lower panels quantify systematic biases and the magnitude of prediction errors, whilst the histograms on the right panels show the statistical distribution of both predictions and errors. Together these visualisations enable comprehensive evaluation of model performance and identification of systematic patterns across different molecular systems.

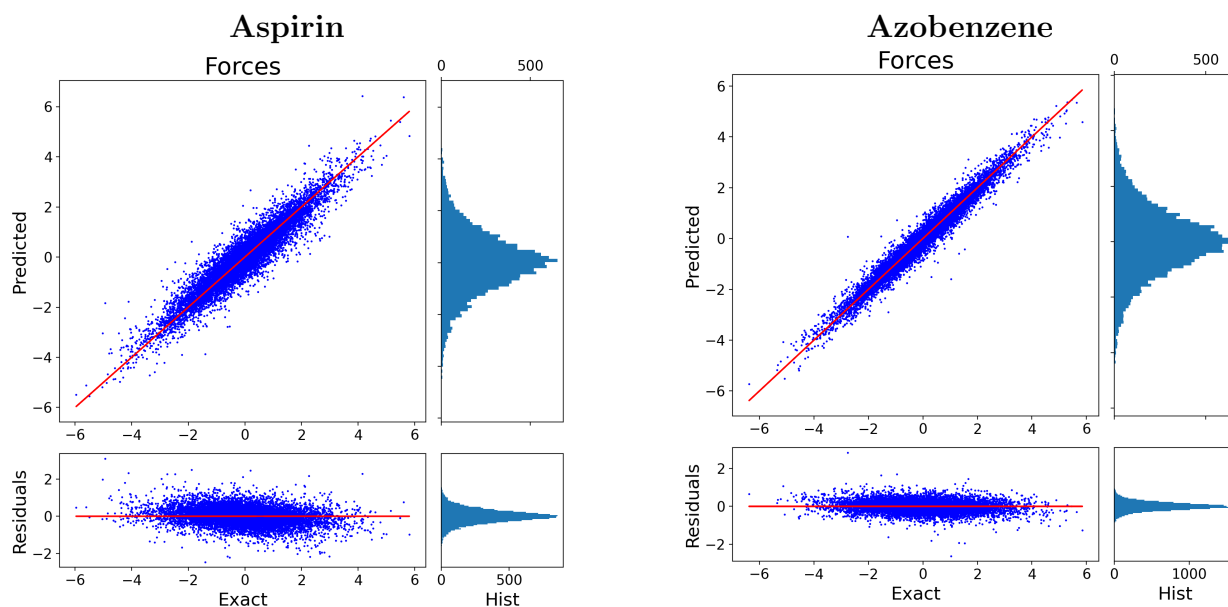


Figure 5.2: Parity plots for the Cartesian components of the atomic forces of aspirin and azobenzene. The red line in the main panel denotes the parity line. The bottom panels show the residuals of the main parity plot. The two panels on the right of each subplot show a histogram of data points corresponding to their respective panel on the left. The MSE/atom for aspirin is 8.42meV and the R^2 score is 0.8896. The MSE/atom for azobenzene is 2.62meV and the R^2 score is 0.96124.

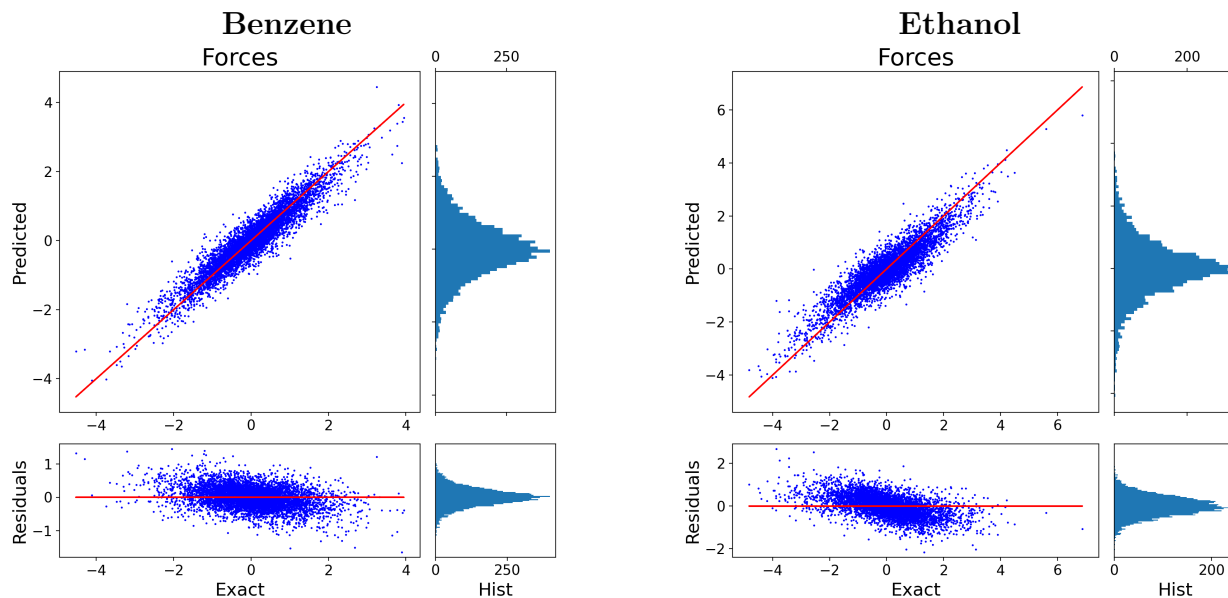


Figure 5.3: Parity plots for the Cartesian components of the atomic forces of benzene and ethanol. The red line in the main panel denotes the parity line. The bottom panels show the residuals of the main parity plot. The two right panels show a histogram of data points corresponding to their respective left panel. The MSE/atom for benzene is 7.28meV and the R^2 score is 0.8876. The MSE/atom for ethanol is 26.91meV and the R^2 score is 0.8168.

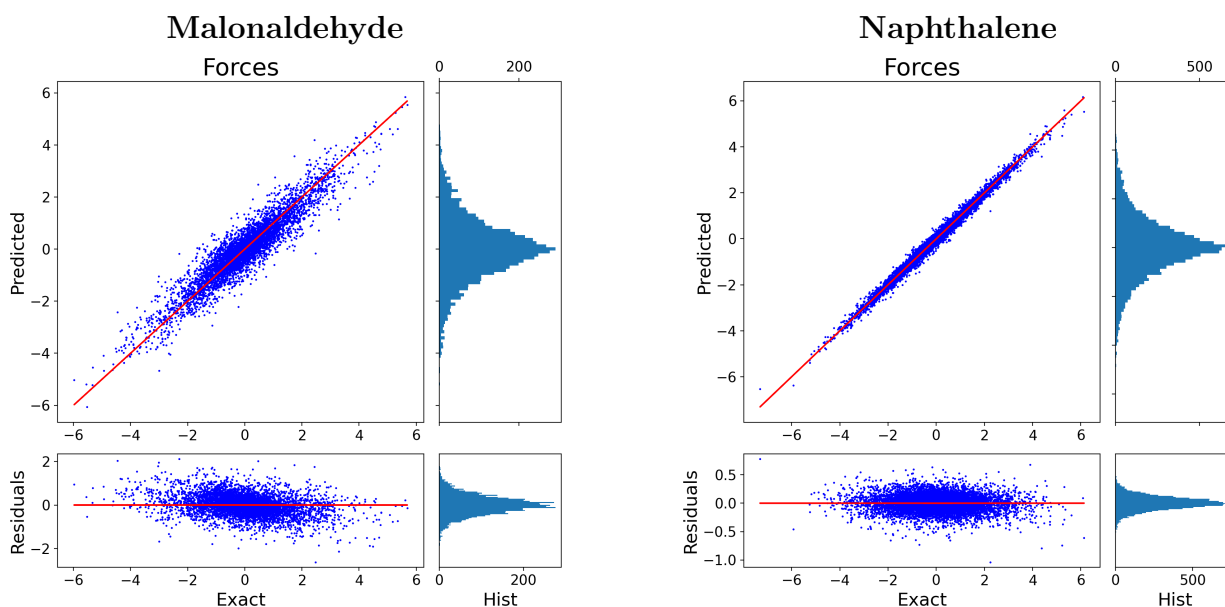


Figure 5.4: Parity plots for the Cartesian components of the atomic forces of malonaldehyde and naphthalene. The bottom panels show the residuals of the main parity plot. The two panels on the right of each subplot show a histogram of data points corresponding to their respective panel on the left. The MSE/atom for malonaldehyde is 24.87meV and the R^2 score is 0.8745. The MSE/atom for naphthalene is 0.95meV and the R^2 score is 0.9893.

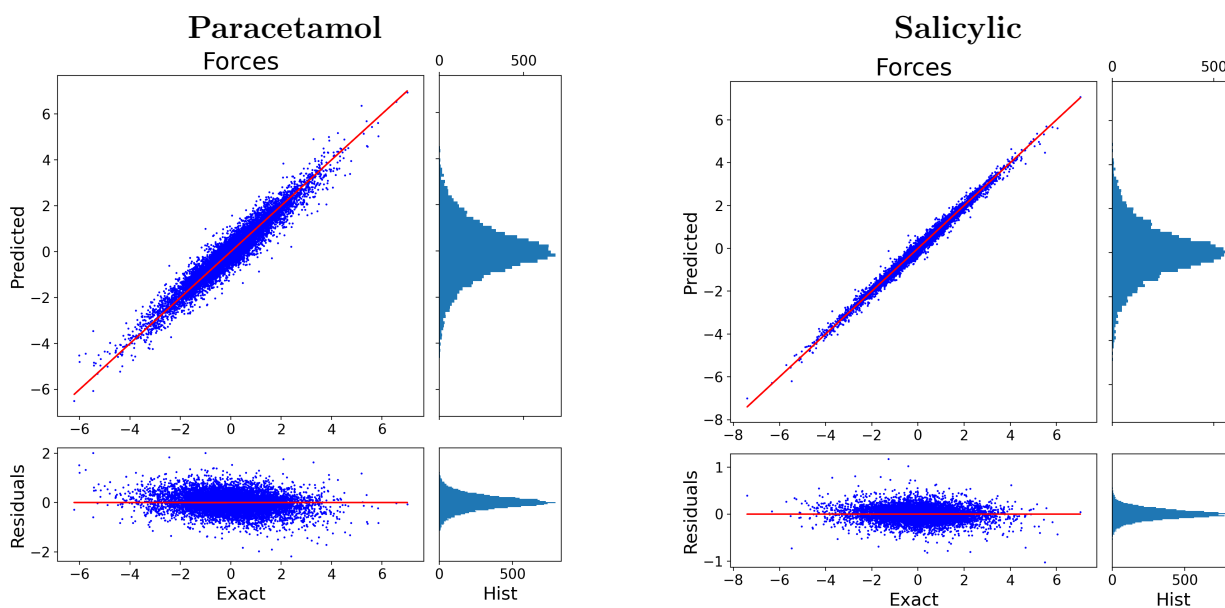


Figure 5.5: Parity plots for the Cartesian components of the atomic forces of paracetamol and salicylic. The red line in the main panel denotes the parity line. The bottom panels show the residuals of the main parity plot. The two right panels show a histogram of data points corresponding to their respective left panel. The MSE/atom for paracetamol is 5.51meV and the R^2 score is 0.9308. The MSE/atom for salicylic acid is 1.17meV and the R^2 score is 0.9889.

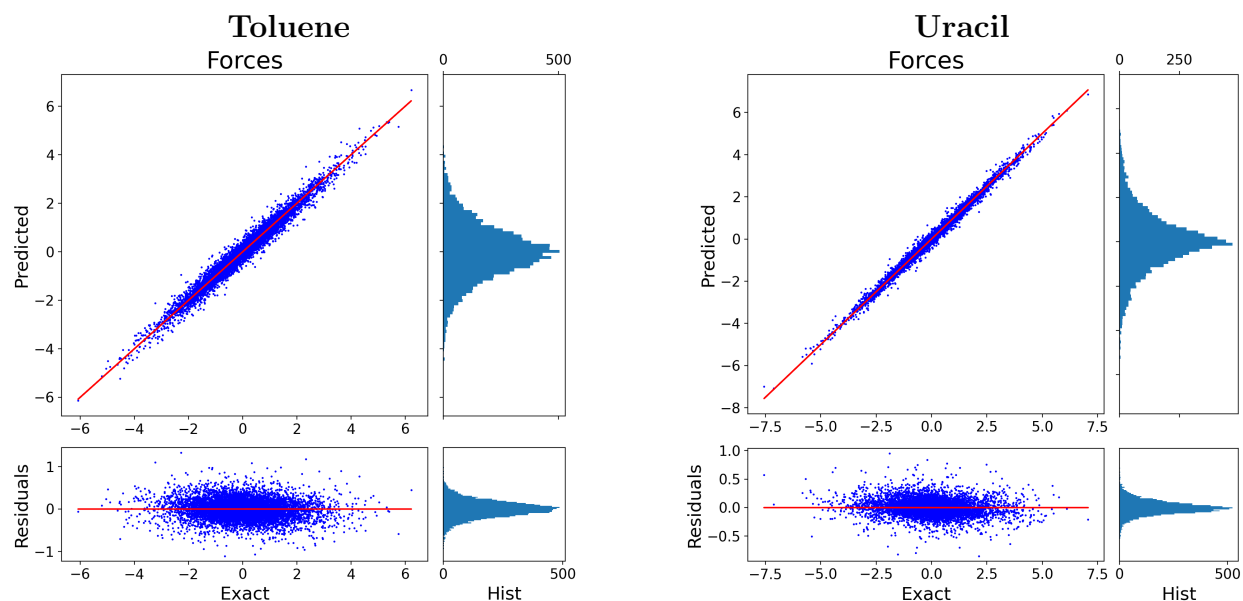


Figure 5.6: Parity plots for the Cartesian components of the atomic forces of toluene and uracil. The red line in the main panel denotes the parity line. The bottom panels show the residuals of the main parity plot. The two right panels show a histogram of data points corresponding to their respective left panel. The MSE/atom for toluene is 3.06meV and the R^2 score is 0.9686. The MSE/atom for uracil is 1.56meV and the R^2 score is 0.9902.

Chapter 6

Node Alignment Failures

There exists an overlooked and fundamental flaw in the interfacing of equivariant representations in the form of Euclidean distance matrices with any form of neural network. This issue is in particular exacerbated for molecules exhibiting high levels of representational symmetry. In this chapter we rigorously explain this issue, highlight scenarios in which it can arise, why it arises, and potential methods to combat it. We henceforth refer to this issue as *node alignment failures*, due to the issue only becoming apparent at the point in which graphs interface with the initial layer of a neural network, as will be described in detail within the following sections.

To give a brief overview of the problem, if we take a benzene molecule and task different observers to label the six carbon atoms on the internal ring with the numbers one through six, you are likely to receive different orderings of labels. Some may start ordering from where they see the uppermost atom and go clockwise, some anticlockwise, some may start with the rightmost etc. To us it is obvious that by choosing arbitrary labels does not change the underlying molecules. We still see them as identical in every way. However, to an equivariant neural network, this seemingly harmless act has potentially already completely obfuscated the network and made these identical molecules appear to be completely different inputs with differing property values.

When we represent molecules using distance matrices, different labellings as described above will produce different matrix arrangements. Even though the distances have not changed (the set of data within each matrix will be identical), the ordering in which they are stored in the matrix will be different.

GNNs can account for this on the basis that they are permutationally invariant. This is

an exceptionally important quality that in almost all use cases for GNNs, the row ordering of matrix inputs does not affect the network. However, in our representational choices for equivariant consistency, it is not only row permutation we have to be concerned about, but also columnar permutation. This is what we shall describe in detail over the following sections and highlight, with examples, how it can become such a major issue in the trainability of neural networks.

We show that this problem is directly tied to the levels of representational symmetry within a given molecule. Furthermore, we show that this particular issue scales exponentially as for n atoms in a given system, there are $n!$ possible ways to assign node labels. This can easily create an explosion in obfuscation beyond what any neural network can be trained to handle.

This problem stems from deep group-theoretic incompatibility between symmetric structures of physical representations and the asymmetric structure of neural networks. It provides us with a counter intuitive approach to the utilisation of representational symmetry for training set size reduction and highlights that there really never is a free lunch with machine-learning. Through spectral analysis, information theory and scaling law investigations, we hope to provide a thorough breakdown of an overlooked mathematical obstruction within computational physics that requires new theoretical approaches to tackle.

We shall first describe this problem for a system of n totally indistinguishable atoms and then move on to detail how this applies to our real systems in a later section.

6.1 The Isomorphism Problem

The core mathematical issue which comprises the node alignment failure issues emerges from a detailed variant of the isomorphism problem. By first establishing that the group of distance matrices D derived from any permutation of node relabellings is isomorphic to the symmetry group S_n , we can in turn show that this creates an exponential degeneracy that neural networks cannot resolve with any consistent operation scheme.

6.1.1 Group Theory Foundation

For a graph $\mathcal{G}$ with vertex set $\mathcal{V}$, $|V| = n$, the symmetric group S_n represents all possible ways in which we can arrange the n nodes, i.e, relabel the atoms of the system, within this matrix $\mathcal{V}$.

S_n acts on the relative distance matrix $\mathcal{X} \mapsto \mathcal{D}$ through simultaneous row-column permutations. For any permutation $\pi \in S_n$, we can obtain a transformed relative distance matrix D^π by applying the operation π to the row and column indices of $\mathcal{D}$, giving,

$$D^{\pi_{ij}} = D_{\pi(i)\pi(j)}. \quad (6.1)$$

We say that this procedure defines a *group homomorphism* $h : S_n \mapsto D$, wherein a group homomorphism is a function which maps between two groups such that the underlying group operation is preserved. Here, D is the group of all relative distance matrices that are derivable from the original distance matrix D under any of the permutation operations.

We have that $h(\pi) = D^\pi$ establishes a group homomorphism between D and S_n . We seek to show that this is a full isomorphism.

The homomorphism property $h(\pi_1 \circ \pi_2) = h(\pi_1) \circ h(\pi_2)$ can be seen to follow from the permutation composition by associativity. The bijectivity requirement is guaranteed due to the fact that distinct permutations produce inherently distinct matrix arrangements for abstract atomic configurations.

This isomorphism implies that every possible molecular structure containing n indistinguishable atoms is able to generate exactly $n!$ unique relative distance matrices under permutation operations. All of these correspond to identical configurations.

6.2 Network Incompatibility

A fundamental aspect of this problem is that neural networks are completely agnostic to the equivalences of permuted molecular configurations. This is true for our architecture, which we will use to describe the issue in more detail, but this can be applied abstractly to most scenarios involving the interfacing of neural networks and high symmetry representa-

tions.

The message computation function Ψ_{msg} takes $[\mathcal{H}, \mathcal{D}^\epsilon, \mathcal{A}]$ as input and processes this data through sets of densely connected neural network layers which operate on the individual rows of this concatenated matrix. However, the critical failure occurs due to permutations on $\mathcal{D}$ alter both row and column ordering of data within each row, obfuscating which data point enters the network at which neuron within a given layer.

Consider a 4-atom molecule where the relative distances matrix $\mathcal{D}$ is processed row-by-row by the function $\Psi_{msg} : \mathbb{R}^{d_h+4+4} \mapsto \mathbb{R}^{n \times 64}$. Each row contains the distances for a given node to every other node.

The neural network will process each of these rows initially with a weight matrix,

$$W \in \mathbb{R}^{d_h+4+4} \mapsto \mathbb{R}^{64}, \quad (6.2)$$

where the output specifically is given as $W^\top \cdot [\mathcal{H}_i, \mathcal{D}_i^2, \mathcal{A}_i]$. Each position within row i has a fixed semantic meaning. For example, one of the entries has a fixed position purely to describe the squared distance of the current atom to atom j within the system. By permuting the node labels we scramble both which atom occupies row i , which isn't an issue, but also which atoms the distances in that row refer to. This is where the catastrophic mismatch occurs, as shown in detail in the following example.

6.2.1 Example

We detail the example wherein we compare an original distance matrix with one which has been altered by the $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1)$ permutation explicitly. The information present in the original matrix gets shifted in this new matrix as follows,

$$\begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{pmatrix} \rightarrow \begin{pmatrix} d_{44} & d_{41} & d_{42} & d_{43} \\ d_{14} & d_{11} & d_{12} & d_{13} \\ d_{24} & d_{21} & d_{22} & d_{23} \\ d_{34} & d_{31} & d_{32} & d_{33} \end{pmatrix}. \quad (6.3)$$

Due to permutation invariance we can realign the rows so that the atoms involved have,

from the perspective of the original molecule, the same row ordering,

$$\begin{pmatrix} d_{44} & d_{41} & d_{42} & d_{43} \\ d_{14} & d_{11} & d_{12} & d_{13} \\ d_{24} & d_{21} & d_{22} & d_{23} \\ d_{34} & d_{31} & d_{32} & d_{33} \end{pmatrix} \rightarrow \begin{pmatrix} d_{14} & d_{11} & d_{12} & d_{13} \\ d_{24} & d_{21} & d_{22} & d_{23} \\ d_{34} & d_{31} & d_{32} & d_{33} \\ d_{44} & d_{41} & d_{42} & d_{43} \end{pmatrix}. \quad (6.4)$$

We can see that, compared to the original matrix, the resulting permutation has row elements containing distance information pertaining to completely different atom pairs. The network computes two differing outputs from it's initial layer,

$$W^\top \times [\mathcal{H}_1, d_{11}, d_{12}, d_{13}, d_{14}, \mathcal{A}_1] \neq W^\top \times [\mathcal{H}_1, d_{14}, d_{11}, d_{12}, d_{13}, \mathcal{A}_1]. \quad (6.5)$$

A diagram illustrating this effect is shown in Fig. 6.1.

This highlights the fundamental violation. The network processes the pairwise atomic distance $d_{\pi(i),\pi(j)}$ using weights that were intended for the distance d_{ij} . This creates a systematic misinterpretation of every pairwise relationship that can be subject to arbitrary node labelling.

This is due to the network weights being randomly initialised with fixed positional points within the architecture that assume consistent data structure flow within its inputs, throughout processing the forward pass and outputs. The probability of the randomly initialised weights- or indeed weights that are trained during backpropagation- accidentally create a scenario of consistent processing is practically zero for any non-trivial permutation n .

This mathematical certainty shows that any neural network with standard positional encoding of this form, and indeed any data structure reliant on similar metrics for representation, will systematically misinterpret distance relationships from the initial steps of training.

The Node Alignment Failure

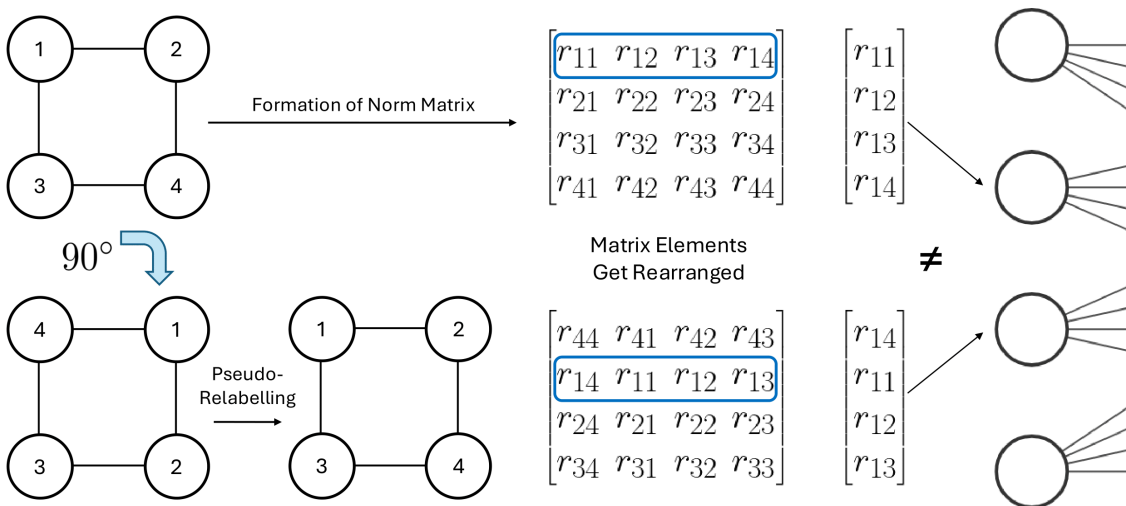


Figure 6.1: Illustration on how the inconsistent labelling of nodes may lead to irregularities with the interface of Euclidean distance matrices and neural networks. Here, the lower matrix contains the entries of the original upper matrix to highlight that data reordering can occur. We see that joint row-column swaps are present, with the swapping of columnar data causing identical data points to enter initial dense layers of the neural network at different points. With no guarantees that the weights and biases of any given node of neural network entry may share parameters, this causes training issues within the model that we call obfuscation of the initial dense layers.

6.3 Molecular Graph Automorphism

Now that we have established that the node alignment failures arise from a fundamental issue in the compatibility between molecular symmetry encodings and neural network architectures, we now endeavour to provide a mathematical framework, which will allow us to later quantify the extent of this problem. The automorphism group gives us a theoretic foundation for understanding why certain molecules are inherently more susceptible to these failures than others.

6.3.1 The Automorphism Group as a Symmetry Quantifier

For a molecular graph $\mathcal{G}(\mathcal{V}, \mathcal{A}, \mathcal{E})$, the automorphism group $Aut(\mathcal{G})$ consists of all bijections $\pi : \mathcal{V} \rightarrow \mathcal{V}$, which satisfy the following conditions.

- **Bond Preservation:** $(v_i, v_j) \in \mathcal{A} \iff (\pi(v_i), \pi(v_j)) \in \mathcal{A}$.
- **Distance Preservation:** $d_{ij} = d_{\pi(i)\pi(j)} \forall d_{ij} \in \mathcal{D}$.
- **Species Preservation:** $h_i = \pi(h_i) \forall h_i \subset v_i \in \mathcal{V}$.

This allows us to define the cardinality of the automorphism group described above, $|Aut(\mathcal{G})|$, as the number of structurally equivalent molecules which, when encoded in our graph scheme, preserve molecular identities and bonds yet generate distinct distance matrix arrangements $\mathcal{D}$.

The automorphism group directly constrains the isomorphism problem laid out in Section 6.1. We have that the symmetric group S_n represents all possible $n!$ permutations of node labels. However, the automorphism group $|Aut(\mathcal{G})|$ identifies the subset of permutations that preserve the molecular structure. Each automorphism corresponds to a valid relabelling that maintains chemical equivalence whilst producing a distinct distance matrix.

The minimum number of representationally distinct distance matrices that can be generated for a given molecular structure is $|Aut(\mathcal{G})|$. This quantifies the irreducible symmetry that no deterministic labelling strategy can eliminate within a pairwise representational scheme.

6.3.2 Molecular Automorphism Analysis

We analysed the molecules present in the initial MD17 dataset to reveal the varying levels of node alignment vulnerability across the structures. The results are shown in Table 6.1.

We can see that half the molecules involved in this dataset exhibit no alignment ambiguity under our representation. Some, namely Malonaldehyde, Toluene, Azobenzene and Naphthalene, show either single symmetry operations that involve a two-fold ambiguity or a slightly more moderate vulnerability. Benzene is an outlier in that it exhibits severe alignment failures.

Automorphism Group Sizes and Symmetry Analysis for MD17 Molecules

Molecule	Formula	$ Aut(\mathcal{G}) $	Symmetry Description
Aspirin	$C_9H_8O_4$	1	Asymmetric structure
Salicylic Acid	$C_7H_6O_3$	1	Asymmetric structure
Ethanol	C_2H_6O	1	Asymmetric structure
Uracil	$C_4H_4N_2O_2$	1	Asymmetric pyrimidine
Paracetamol	$C_8H_9NO_2$	1	Asymmetric para-distribution
Malonaldehyde	$C_3H_4O_2$	2	Mirror plane through central carbon
Toluene	C_7H_8	2	Mirror plane through methyl bond
Azobenzene	$C_{12}H_{10}N_2$	2	Mirror plane perpendicular to N=N
Naphthalene	$C_{10}H_8$	4	D_2 symmetry (2 rotations $\times$ 2 reflections)
Benzene	C_6H_6	12	D_6 symmetry (6 rotations + 6 reflections)

Table 6.1: A table displaying the symmetry details of all ten molecules present within the MD17 dataset. Included is the chemical formula for each molecule, and their respective cardinality number of their automorphism group.

6.3.3 Network Training Implications

The automorphism group size directly determines the multiplicative factor by which training data appears in an inconsistent manner to neural networks. For benzene, with $|Aut(\mathcal{G})| = 12$, each instance in the training set could appear as one of twelve distinct input patterns to the network, despite representing the same underlying chemical structure.

We thus conclude that the effective training set size for neural networks decreases proportionally to the cardinality of the automorphism group, with highly symmetric molecules under this representation contributing disproportionately to this degradation. For a molecular representation approaching absolute perfect symmetry $|Aut(\mathcal{G})| \rightarrow n!$.

This framework shows that node alignment failures are not merely an artifact of implementation, but are a fundamental consequence of the group-theoretic structure of molecular symmetries. The automorphism group provides us with a diagnostic tool for identifying problematic molecules within the training data and a theoretical benchmark for evaluating symmetry-handling methodologies.

6.4 Quantifying Ambiguity

Having established the fundamental isomorphism, which links possible distance matrix transformations under artificial node relabelling and abstract permutation groups, we now seek to develop a rigorous mathematical framework for quantifying the level of ambiguity that is introduced through the described node alignment failures. The quantification of such is essential for understanding barriers of computational complexity in training neural networks to desired degrees of accuracy prior to engaging in the training process.

6.4.1 The Ambiguity Measure

We seek to establish how much obfuscation can occur due to the reordering of terms within the distance matrix. Within our representation, only atomic species are encoded within the node features. This is a deliberate choice dictated to us by model simplicity. Each row of the distance matrix is inputted into a neural network with associated species information concatenated to the node’s distance information. If the species are one-hot encoded, then each row of the distances matrix will then be associated with a vector, which will distinguish the different species types from each other. Thus, within this representation obfuscation only occurs between nodes of the same species.

For each molecule, we assign a maximal obfuscation number $O_{\max}(\mathcal{G})$ as follows,

$$O_{\max}(\mathcal{G}) = \sum_{k=1}^s \frac{|e_k|!}{N!}, \quad (6.6)$$

where e_k is the set of atoms of species type k , $|e_k|$ denotes the number of atoms of that species within the molecule. s is the total number of distinct species present. N is the number of atoms within the molecule. The one-hot encoding vector h_i for any atom i takes one of s possible one-hot basis vectors. For our dataset containing four distinct species (H,C,N,O),

we have,

$$h_i \in \mathbf{e}_k = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\}. \quad (6.7)$$

This obfuscation measure quantifies the fraction of node permutations that preserve species identity whilst scrambling distance matrix column ordering. Consider a hypothetical four-atom molecules comprised of hydrogen, carbon, nitrogen and oxygen, each appearing once. We have $|e_k| = 1$ for all k , which gives us,

$$O_{\max}(\mathcal{G}) = \frac{1! + 1! + 1! + 1!}{4!} = \frac{1}{6}. \quad (6.8)$$

This fraction decreases rapidly with molecular size and species diversity. For a molecule with $N = 20$ atoms distributed equally among four species types ($|e_k| = 5$ for all k), we get,

$$O_{\max}(\mathcal{G}) = \frac{4 \times 5!}{20!} \approx 1.98 \times 10^{-15}. \quad (6.9)$$

This shows that species diversity dramatically suppresses alignment ambiguity. Conversely, for a maximally symmetric system comprised entirely of identical atoms, we have $|e_1| = N$ and $|e_k| = 0$ for $k > 1$, which gives $O_{\max}(\mathcal{G}) = \frac{N!}{N!} = 1$. This represents complete obfuscation where every permutation produces a valid yet differently ordered distance matrix.

6.4.2 The Ambiguity Score

The obfuscation measure $O_{\max}(\mathcal{G})$ captures symmetry-induced degeneracy, but neglects structural complexity. A small symmetric molecule and a large symmetric molecule may share similar $O_{\max}$ values yet present vastly different challenges to neural network training. We require an additional metric that encodes topological and compositional complexity.

For this purpose, we employ the Bertz complexity index T , as modified by Hendrickson, which quantifies the structural complexity of a given system through graph-theoretical anal-

ysis of connectivity patterns and heteroatom content [50][51]. The Bertz index increases with molecular size, free of graphical branching, ring content and heteroatom diversity. Values of T are computed using the CACTVS toolkit implementation and are retrievable from the PubChem database entries for each molecule [52][53].

We construct our ambiguity score by combining these two measures,

$$A(\mathcal{G}) = \frac{1 - T^2 \ln O_{\max}^2}{N}, \quad (6.10)$$

where the N normalisation accounts for system size. The squaring of both T and the logarithm of $O_{\max}$ serves to amplify sensitivity to variations in both complexity and symmetry, allowing the metric to distinguish between molecules that might otherwise appear similar under linear scaling.

The logarithmic transformation of $O_{\max}$ is particularly important as $O_{\max}$ spans many orders of magnitude, from near-unity for symmetric molecules to near-zero for diverse systems. The logarithm compresses this range whilst preserving ordering relationships.

The interplay between T^2 and $\ln O_{\max}^2$ captures competing effects. High structural complexity (T) generally benefits model training by providing rich representational information, whilst high obfuscation ($O_{\max}$) degrades training through alignment inconsistencies. The ambiguity score $A(\mathcal{G})$ thus reflects the net effect of these opposing factors.

6.5 Application to MD17 Dataset

We compute the components of the ambiguity scores for each molecule in our dataset. The results are presented in Table 6.2, where we report $\ln O_{\max}^2(\mathcal{G})$, the Bertz complexity T , the resulting $\ln |A|$, and the corresponding R^2 -score achieved by our neural network model.

The relationship between molecular symmetry and model performance becomes immediately apparent when comparing three representative cases, namely, benzene, toluene, and uracil, as shown in Fig. 6.2.

This progression directly validates our theoretical framework. Benzene’s D_{6h} symmetry generates twelve equivalent distance matrix arrangements through valid node relabellings,

Representational Molecular Complexity and Corresponding R^2 -Scores

Molecule	$\ln O_{\max}^2(\mathcal{G})$	T	$\ln A $	R^2
Uracil	-1.14975	161	7.81748	0.99024
Naphthalene	-34.91736	80.6	9.44161	0.98926
Salicylic	-20.27449	133	10.01748	0.98881
Toluene	-19.90072	42	7.75807	0.96863
Azobenzene	-65.64460	157	11.11869	0.96124
Paracetamol	-13.13481	139	9.44849	0.93078
Aspirin	-61.97655	212	11.79541	0.88963
Benzene	-13.20833	12	5.57794	0.88755
Malonaldehyde	-21.76834	34.2	7.94772	0.87453
Ethanol	-12.43684	2.8	2.39288	0.81679

Table 6.2: The components of the molecular complexity Eq. (6.10) compared to the R^2 predictability scores for each molecule in our dataset. We can see that scores $\ln |A| > 7$ are usually a good indicator for strong model performance (with the notable exception of malonaldehyde). Scores below 6 are associated to issues in the training process.

forcing the network to learn identical force fields through differing processing pathways. This manifests as increased scatter and wider residual distributions. Toluene’s methyl substituent breaks the six-fold rotational axis, dramatically reducing the number of overall symmetries and corresponds to a substantial performance improvement in model predictability. Uracil eliminates ambiguity entirely through compositional diversity, with four distinct atomic species ensuring each node labelling produces a unique distance matrix.

The systematic correlation between $|Aut(\mathcal{G})|$ and R^2 -score provides compelling empirical validation for the ambiguity metrics developed in Section 6.4, demonstrating that benzene would require much more training data than toluene to achieve comparable accuracy purely due to the higher levels of symmetry involved.

Examining Table 6.2 reveals a number of patterns. Uracil achieves the highest R^2 -score with $\ln |A|$, driven by high Bertz complexity ($T = 161$) from its four distinct atomic species and minimal obfuscation. The small obfuscation measure reflects the fact that uracil’s species diversity leaves few indistinguishable atoms, thereby suppressing alignment ambiguity.

Naphthalene and salicylic acid demonstrate similar behaviour to uracil. The R^2 -scores for both are exceptionally high and we see $\ln |A|$ values above 9.4. Despite naphthalene’s significant geometric symmetry (belonging to the D_{2h} point group), its moderate Bertz

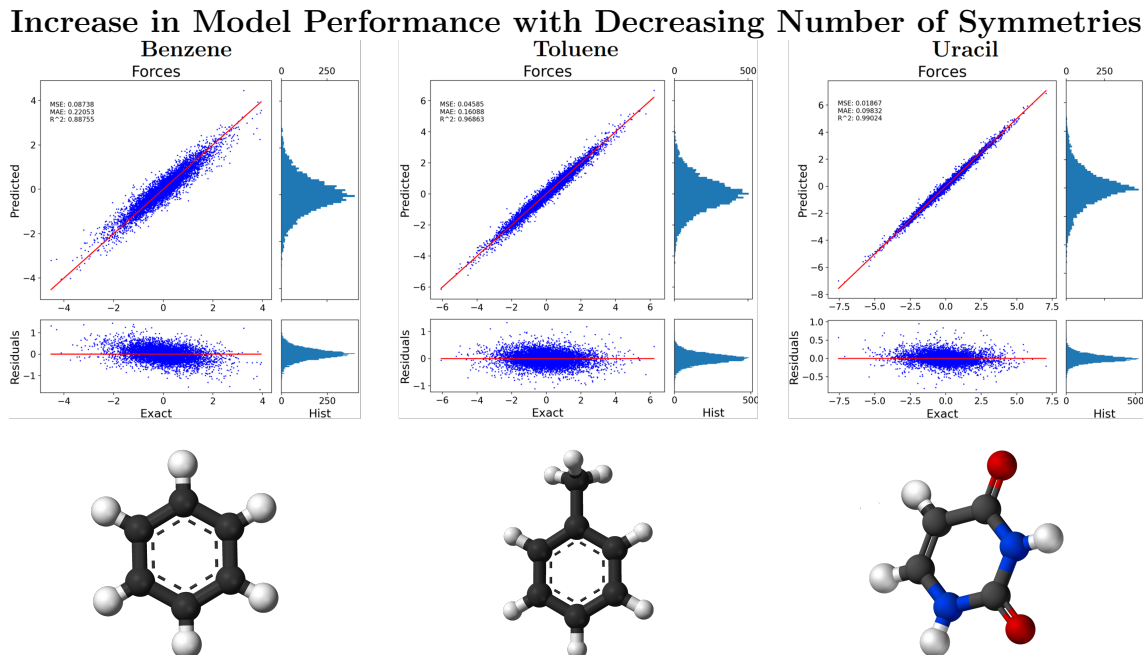


Figure 6.2: Figure showing that the improvement in model performance correlates with reducing the number of symmetries present within a given molecule. Note that, symmetries do not only apply to physical symmetries with respect to the atomic position, but also to species type.

complexity and reasonable obfuscation levels allow for effective training.

Benzene shows us the counter-intuitive case that has been discussed extensively in the previous sections. It has a Bertz complexity of $T = 12$, which is comparatively low compared to the rest of the molecules in our dataset. It also achieves one of the lowest ambiguity scores within the dataset. This low score correctly predicts its poor R^2 performance within the model, as its minimal structural complexity fails to compensate for the representational ambiguity that is induced by its D_{6h} symmetry.

Ethanol exhibits the lowest model predictive performance with an R^2 -score of 0.81679, and has a correspondingly minimal $\ln |A| = 2.39288$. Its exceptionally low Bertz complexity ($T = 2.8$) reflects its small size and lack of structural features. Furthermore, combined with moderate obfuscation from its inherent conformational flexibility, ethanol presents insufficient representational richness for the neural network to extract reliable patterns.

Malonaldehyde represents a slight anomaly in its ambiguity metrics. Despite its logged ambiguity value of $\ln |A| = 7.94772$ suggesting that it should have a reasonably well performing predictive model arise from training on this molecule, it only achieves an R^2 -score

of $R^2 = 0.87453$. We believe this discrepancy to arise from its conformational flexibility generating multiple equivalent spatial arrangements during molecular dynamics sampling. This can introduce effective obfuscation not fully captured by the static $O_{\max}$ measure derived from species counts alone.

6.6 Correlation Analysis and Theoretical Implications

To rigorously quantify the relationship between our ambiguity metric and the overall resulting model predictive performance, we compute the cosine similarity between $\ln |A|$ and R^2 across all molecules. The cosine similarity measures the angular relationship between two vectors in n -dimensional space through a normalised dot product,

$$\cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2} \sqrt{\sum_i b_i^2}}. \quad (6.11)$$

For our dataset, $\ln |A|$ and R^2 are effectively 10-dimensional vectors (one entry per molecule), which gives us,

$$S_C(\ln |A|, R^2) = 0.963... \quad (6.12)$$

This near-unity cosine similarity indicates strong directional alignment between the two quantities. The ambiguity metric operates independent of vector magnitude, eliminating bias from measuring different ranges whilst preserving proportional relationships.

The high correlation establishes that the ambiguity score $A(\mathcal{G})$ serves as a robust pre-screening tool for model trainability. Molecules with low values of $\ln |A|$ uniformly struggle and can be highlighted as a subsection of a dataset which need specific attention, potentially requiring dataset augmentation or alterations to the treatment of the data in the underlying model, without having to train a model and use computational resources before coming to this conclusion via post-processing analysis.

The strong correlations between ambiguity and performance provides compelling evidence that neural network success on molecular systems is not merely a function of architecture, hyperparameters and training data quantity. Rather, we show that fundamental mathematical properties of the underlying systems representation itself impose hard limits on achievable

accuracy.

The logarithmic relationship embedded in $A(\mathcal{G})$ suggests that even moderate increases in the overall structural complexity of a system, or any reductions in the obfuscation of a system through species diversification, can yield dramatic improvements in model performance. Furthermore, node features which extend beyond one-hot encoded species types that can further uniquely identify otherwise identical atoms will also increase the representational complexity of the underlying data and reduce the obfuscation, which occurs at the point of network interfacing. This insight supports the development of adaptive modelling frameworks where architectural choices, training data requirements and feature engineering strategies are guided by the ambiguity score, rather than sheer, costly, computational experimentation.

Furthermore, the metric highlights a counter-intuitive aspect of equivariant neural networks. Traditional thinking suggests that highly symmetric molecules should benefit the most from equivariant architectures, as these are explicitly designed to encode the underlying geometric symmetries present in a system in order to reduce training data requirements. Our analysis reveals a fundamental tension which that, whilst equivariance handles spatial transformations elegantly, it cannot resolve the combinatorial explosion of valid node labellings that symmetric molecules can generate.

The node alignment failures represent a fundamental incompatibility between the discrete, position-dependent structure of neural network weight matrices and the continuous, permutation-equivalent nature of molecular symmetries. The ambiguity metric thus transforms model development from an empirical exercise into a theoretically grounded process, providing clear guidance on when to invest in model complexity, and when we can indicate if a potential simpler approach will achieve near-optimal performance, without the need of large training sets, complex architectures and expensive computational resources.

Part 2 - Periodic Systems

Chapter 7

Extension To Periodic Systems

To extend our work of machine-learning force predictions from molecular systems to infinite periodic structures, we must tackle a number of significant barriers. Molecules have well-defined boundaries, unambiguous bonds and finite intersection networks. Periodic systems, however, are based on infinite lattices. While molecular force prediction in Part 1 achieved R^2 -scores exceeding 0.98 for high-performing molecules (uracil, toluene, naphthalene, salicylic acid, and azobenzene), the challenge of training high performing predictive models for periodic schemes faces challenges in obtaining this level of accuracy. In particular, periodic images introduce ambiguous distance calculations, which need representational attention in order to mitigate, increasing cell sizes will increase the computational cost of the overall model, and lastly, we will face amplified symmetry-induced node alignment failures from crystallographic space groups. This section aims to establish a foundation for the analysis of periodic systems within our previously devised architecture, and detail ways in which we can attempt to obtain the same high levels of predictive accuracy achieved in our molecular models despite the new challenges we face in periodic systems.

This is motivated by the fact that periodic systems exhibit interesting physical properties that emerge as a direct result of their infinite representation, which are of great value to researchers across all STEM domains. For example, the periodic potential creates electronic band structures utilised in determining the metallic or semi-conductive behaviour of crystals. The overall lattice can be used to determine mechanical stability. The study of phonons within these systems allow us to investigate transport, heat capacity and thermal expansion properties.

In order to conduct any of the wide variety of informationally rich calculations of this

nature, we need to ensure that our representation of the material at hand is relaxed towards its equilibrium geometries. It is thus vitally important that accurate calculations of atomic forces within a system can be conducted quickly so as to enable geometry optimisation, along with providing the data necessary for molecular dynamics calculations as before.

Furthermore, akin to how forces can be calculated with the use of gradients obtained from the backpropagation of a model mapping Cartesian coordinates to system energies, direct force prediction allows us to potentially calculate various quantities that rely on force derivatives. These could include elastic constants, piezoelectric constants or phonon-phonon coupling [54].

Thus, we endeavour to translate our model architecture for the purposes of periodic structure calculation and accurately predict atomic forces for a periodic system.

7.1 Periodic Physics Foundations

The study of periodic systems is fundamental to condensed matter physics, allowing us to understand macroscopic materials, potentially containing billions of atoms, through the analysis of a singular repeating unit cell. This reduction allows us to study complex systems of interest without having to attempt DFT calculations, which would be computationally prohibitive at its $O(N^3)$ complexity scaling [55][7]. This highlights an important feature, that macroscopic properties of these systems are determined by the arrangement of periodically repeating atoms.

The validity of the unit cell approach rests on Born-von Kármán periodic boundary conditions [56][57]. These conditions state that, for sufficiently large systems, boundary effects become negligible and bulk properties are determined entirely by the periodic arrangement within a single unit cell. The physical justification derives from the exponential decay of surface-to-volume ratio as $N^{-\frac{1}{3}}$, ensuring that in the thermodynamic limit $N \rightarrow \infty, V \rightarrow \infty, \frac{N}{V} \rightarrow \text{const}$, i.e, contributions from system boundaries vanish whilst bulk properties converge [58].

Periodic systems are characterised by infinitely repeating unit cells across three dimensions. This is representative of an underlying physical reality that crystalline materials, for

the most part, minimise their total energy through regular repeating atomic arrangements.

The lattice is constructed from three fundamental vectors, known as *lattice vectors* $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$. Equivalent positions in different unit cells can be reached via a series of translations, which take the form,

$$\mathbf{R}(\mathbf{n}) = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3 \quad (7.1)$$

where $\mathbf{n} = (n_1, n_2, n_3) \in \mathbb{Z}^3$.

The unit cell volume $V = |\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)|$ determines the density of the material.

We refer to the positions of atoms within the lattice via fractional coordinates. These natural units of the lattice are expressed as,

$$\mathbf{r} = u\mathbf{a}_1 + v\mathbf{a}_2 + w\mathbf{a}_3, \quad (7.2)$$

where (u, v, w) are the fractional coordinates. In this coordinate system, periodicity manifests by positions differing solely by integer values of $\mathbf{n}$. The transformation between the standard coordinate system and fractional is given as

$$\mathbf{r}_{\text{frac}} = \mathbf{L}^{-1}\mathbf{r}_{\text{cart}} \quad (7.3)$$

where $\mathbf{L}$ is the matrix comprising of lattice vectors $\mathbf{a}$. Note that this transformation preserves all physical observables. The choice of coordinate system affects the mathematical representation but not the physical content. In practice, our neural-network implementation operates in Cartesian coordinates to maintain compatibility with the equivariant message passing architecture established in Part 1. Fractional coordinates however, play a vital role during the preprocessing stage for consistent atom positioning across different cell geometries and are needed for efficient minimum image convention calculations, see Section 7.3.

7.2 Periodic Boundary Conditions

Periodic boundary conditions are the mathematical implementation of the assumption that our material of interest comprises of a unit cell, which repeats periodically in all direc-

tions with no discontinuities, surfaces or boundaries.

For any field $f(\mathbf{r})$, we require that,

$$f(\mathbf{r} + \mathbf{n}) = f(\mathbf{r}), \quad (7.4)$$

for all lattice vectors $\mathbf{n}$.

There are a number of implications for this implementation. Firstly, this approach eliminates the concept of material surfaces, which are a large area of interest in materials science. For this thesis, we restrict attention exclusively to bulk periodic systems. The extension to surface and defect calculations, which require slab geometries and vacuum regions, breaking translational symmetries within our representation, constitutes an important avenue of potential future work, but lies beyond the scope of this thesis. Thus, we restrict our attention purely to bulk behaviour. Although this limitation can be addressed with relatively simple alterations, we do not investigate it in our work. Furthermore, the validity of the boundary conditions relies on the thermodynamic limit. We assume that the system size becomes infinite, but with a constant density. Here, bulk properties emerge as intended, but surface contributions become negligible.

7.3 The Minimum Image Convention

A fundamental challenge with periodic systems represented by a singular unit cell is in devising a method for which the shortest distance between two pairs of atoms is considered as this may be a distance that crosses cell boundaries into other periodic images of the unit cell. The Minimum Image Convention (MIC) solves this problem and it is a critical component for graph neural networks, as our architectures rely heavily on distance calculations [59][60]. We will use this extensively for the construction of neighbourhood lists and the propagation of information through our message passing network.

7.3.1 Periodic Imaging Example

For any atom pair (i, j) in a periodic system, there exist infinitely many distance paths between them through periodic images. Without the MIC, a neural network would face ambiguous choices in measuring the direct intra-cell distance or considering shorter paths through periodic boundaries. This ambiguity creates inconsistent neighbourhood definitions and can cause many issues in equivariant representations, and further ambiguity as to which distance should be considered for use within our message passing scheme. Without a detailed scheme such as the minimum image convention, the following ambiguity arises.

Consider two atoms in a cubic cell with lattice parameter $a = 10\text{\AA}$. The atom A is at position $(1, 1, 1)$ and B is placed at $(9, 9, 9)$. With just the unit cell to look at, the calculated distance would be $\approx 13.86\text{\AA}$. However, we can easily observe that the image of B situated at the relative position of $(-1, -1, -1)$ is much closer, giving a distance of $\approx 3.46\text{\AA}$. Explicitly,

$$d_{\text{direct}} = \|(1, 1, 1) - (9, 9, 9)\| = \|(-8, -8, -8)\| = 8\sqrt{3} \approx 13.86\text{\AA}. \quad (7.5)$$

$$d_{\text{MIC}} = \|(1, 1, 1) - (-1, -1, -1)\| = \|(2, 2, 2)\| = 2\sqrt{3} \approx 3.46\text{\AA}. \quad (7.6)$$

A graph neural network that blindly utilises direct distances would classify these atoms as relatively distant and severely limit or exclude them from any neighbour calculations and data propagation through the network. This inconsistency, as shown previously with ambiguous data flow implications, would lead to obfuscated force predictions, difficult trainability and poor predictive performance.

7.3.2 Implementation

We select the most physically relevant distance between a given pair of atoms within the unit cell, which we define to be the shortest path between a central atom and the second atom within the unit cell, or any of the immediately surrounding periodic replications of it,

$$d_{\min}^{ij} = \min_{\mathbf{n} \in \mathbb{Z}^3} \{\|\mathbf{r}_i - \mathbf{r}_j - \mathbf{n}\|\}. \quad (7.7)$$

We can efficiently calculate this using fractional coordinates and not relying on explicitly checking every possible lattice translation. The following five-step process implements the MIC through modular arithmetic in fractional coordinates.

- Convert Cartesian coordinates to fractional coordinates.

$$\begin{aligned}\mathbf{u}_i &= L^{-1}\mathbf{x}_i, \\ \mathbf{u}_j &= L^{-1}\mathbf{x}_j.\end{aligned}\tag{7.8}$$

- Calculate the fractional separation between the two atoms.

$$\Delta\mathbf{u} = \mathbf{u}_i - \mathbf{u}_j.\tag{7.9}$$

- Wrap to the range $[-0.5, 0.5]$ instead of $[0, 1]$.

$$\Delta\mathbf{u}_{\min} = \Delta\mathbf{u} - \text{round}(\Delta\mathbf{u}).\tag{7.10}$$

- Convert back to Cartesian coordinates.

$$\Delta\mathbf{r}_{\min} = \mathbf{L} \times \Delta\mathbf{u}_{\min}.\tag{7.11}$$

- Calculate the minimum distance.

$$d_{\min} = ||\Delta\mathbf{r}_{\min}||.\tag{7.12}$$

In the third step we wrap to $[-0.5, 0.5]$ because this range captures the maximum distance at which an atom can be the nearest periodic image. Furthermore, the round function provides us with the ability to find which periodic image gives the shortest distance, effectively selecting the appropriate vector $\mathbf{n}$, minimising the distance we are interested in.

7.4 Cutoffs and Interaction Truncations

The assumed infinite connectivity of periodic systems requires us to truncate interactions to a finite set, which minimises the total information loss within our representation. We do this through the introduction of cutoff radii.

The use of finite cutoff radii relies on the fact that the atomic interaction strengths decay with distance and, beyond a certain distance, its effects are negligible. The decaying nature of interactions at a distance necessitates that we implement some form of smoothing to our choice of cutoff function so that, as we reach the predetermined distance threshold, our cutoff function returns values which tend towards zero. We do this through the use of the Behler-Parrinello cutoff function [24][61].

$$f_c(r) = \begin{cases} \frac{\cos(\frac{\pi r}{r_{\text{cut}}}) + 1}{2} & \text{for } r \leq r_{\text{cut}} \\ 0 & \text{for } r > r_{\text{cut}}, \end{cases} \quad (7.13)$$

where r_{cut} is the predetermined cutoff distance. This implementation of the cutoff function has the added benefit of being continuously smooth and differentiable at all points. The gradient is given as,

$$f'_c(r) = \begin{cases} -\frac{\pi}{2r_{\text{cut}}} \sin\left(\frac{\pi r}{r_{\text{cut}}}\right) & \text{for } r \leq r_{\text{cut}} \\ 0 & \text{for } r > r_{\text{cut}}. \end{cases} \quad (7.14)$$

This vanishes at $r = r_{\text{cut}}$, crucially ensuring continuity for force calculations.

We choose this function in particular as the gradual truncation maintains realistic profiles for consistency in force calculations with the added benefit of aiding in computational efficiency. This is due to the reduction in the neighbourhood space that is created for each atom within the unit cell, as any distance included in the minimum image convention calculation that exceeds the cutoff distance is completely ignored for calculation. This screening drastically reduces the computation time and the memory allocation needed for the message passing neighbourhood input building.

We note that different materials require different approaches in our choice of r_{cut} . Covalent compounds can utilise relatively shorter r_{cut} values due to the nature of localised

covalent bonding. Ionic materials require larger values in general, around twice as much as covalent, due to the electrostatic interactions. Metals are even more computationally expensive as their cutoff ranges need to accommodate metallic bonding. Overall we seek to minimise the value of r_{cut} to optimise computational efficiency without losing important bonding information.

7.5 Conclusion

Periodic systems introduce a number of challenges, which we need to overcome in order to produce predictive models of the same quality as our molecular GNNs. The infinite connectivity, many-body correlations and emergent collective phenomena require a careful mathematical approach that curtails computational complexity without costing us any loss in the authenticity of the physical systems we aim to represent.

The minimum image convention and cutoff functions provide good practical solutions to the most obvious issues we face in our architecture transformation, but they introduce limitations, which need to be deeply understood in order to comply with our goal of unambiguous, rich and accurate representations of underlying systems.

The remainder of this part will focus specifically on the analysis of silica, silicon, and various phases and combinations of silicon and oxygen atoms within a large range of unit cell structures and sizes.

We detail the alterations needed to the network architecture to accommodate for the periodic nature of these systems and detail how previous symmetry issues can cause further issues in this environment. The analysis of results obtained from network training will be used to assess the viability of our model architectures and procedures.

Having established the theoretical foundations of periodic representations, we move to describe the architectural modifications required to handle variable-sized unit cells and MIC-based distances in our own custom model architecture and model framework.

Chapter 8

Model Architecture - Silica

8.1 Padding and Masking

Given that our silica dataset contains many different types of unit cells of varying sizes and atomic numbers, we need to implement a dynamic padding protocol to provide our model with a procedure to deal with potential tensor sizing inconsistencies in periodic systems. Unlike the relatively uniform molecular dataset discussed in Part 1, silicon-based structures span a wider size range, with common instances of α -quartz cells at 9 atoms/cell, to amorphous cells of over 200 atoms [62][63]. This variation covers multiple orders of magnitude and creates large incompatibilities with any neural-network architecture due to the requirement of fixed size input dimensions.

The padding scheme works by identifying the largest cell in an overall training database, which is assumed to be the largest system of interest in both training and predicting scenarios. Each unit cell is then systematically expanded in representation to match the maximum dimension defined by the static largest unit cell. This is done through the addition of atomic null states,

$$V_i^{\text{pad}} = \begin{cases} V_i & \text{for } i \leq |\mathcal{V}| \\ 0_{\text{null}} & \text{for } i > |\mathcal{V}| \end{cases} \quad \text{for } i = 1, \dots, |\mathcal{V}|^{\text{max}}, \quad (8.1)$$

where 0_{null} represents the complete *null* atomic state, which is defined as having a set of atomic species ‘zero’ and pseudo-coordinates all situated at the origin. The same padding procedure is applied to all other matrices of the graph $\mathcal{G}(\mathcal{V}, \mathcal{A}, \mathcal{E}, \mathcal{Y})$.

This padded representation stretches the matrices of our graph so that all entries have

uniform tensor dimensions. Doing so not only allows us to interface with neural networks without tensor sizing issues, but furthermore enables vectorised batch processing. Uniform tensor dimensions allow multiple structures to be processed simultaneously through single matrix operations rather than sequential loops, significantly accelerating both training and inference. This parallelisation is particularly advantageous on GPU architectures, where thousands of cores can execute identical operations across batch elements concurrently [64].

Thus, the atomic position matrix becomes,

$$\mathcal{X}^{\text{pad}} = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \dots & \dots & \dots \\ x_{n-1} & y_{n-1} & z_{n-1} \\ x_n & y_n & z_n \\ 0 & 0 & 0 \\ \dots & \dots & \dots \\ 0 & 0 & 0 \end{pmatrix}_{|\mathcal{V}|^{\text{max}} \times 3}. \quad (8.2)$$

Similarly, the distance matrix becomes uniformly sized across all instances of the dataset.

$$\mathcal{D}^{\text{pad}} = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1|\mathcal{V}|} & 0 & \dots & 0 \\ d_{21} & d_{22} & \dots & d_{2|\mathcal{V}|} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ d_{|\mathcal{V}|1} & d_{|\mathcal{V}|2} & \dots & d_{|\mathcal{V}||\mathcal{V}|} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix}_{|\mathcal{V}|^{\text{max}} \times |\mathcal{V}|^{\text{max}}}. \quad (8.3)$$

This padding introduces computation overhead. For a dataset with $N_{\text{max}} = 200$ and $N_{\text{avg}} = 50$, for example, padding creates a $4\times$ memory increase. However, we can optimise the storage of such matrices with sparse matrix encoding and further use of optimised GPU operations.

This padding mechanism creates many pathways within the network, both forward in evaluation and backwards in the calculation of gradient updates, which we want to exclude. In backpropagation, particularly, the automatic differentiation process will attempt to compute gradients for all of these padded zero vectors along with the rest of the network as intended. To accommodate for this, we apply a technique known as masking, which ensures that these gradients are nullified, whilst minimising any numerical instabilities that may occur from the padding and masking process [65].

The masking process is implemented through the use of a *mask matrix*. This acts as a binary filter, which identifies the elements of the entire padded tensor corresponding to real data that we want to consider throughout the training process.

The mask matrix for this architecture needs to encode multiple different physical criteria simultaneously. We define our masking protocol as,

$$\mathbf{M}_{ij} = \begin{cases} 1 & \text{if } (v_i^{\text{pad}}, v_j^{\text{pad}}) \in \mathcal{V} \quad \text{and} \quad d_{ij}^{\text{min}} \leq r_{\text{cut}} \\ 0 & \text{if } (v_i^{\text{pad}}, v_j^{\text{pad}}) \notin \mathcal{V} \quad \text{or} \quad d_{ij}^{\text{min}} \geq r_{\text{cut}} \end{cases} \quad (8.4)$$

where $\mathcal{V}$ is the original, unpadded, node feature matrix and d_{ij}^{min} is the distance calculated in accordance with the minimum unit convention.

Although our dataset is utilised here for direct atomic force predictions, trained networks may subsequently be deployed in molecular dynamics scenarios. In such applications, atomic positions evolve at each timestep, potentially altering which atom pairs fall within the cutoff radius. The masking layer must therefore be updated dynamically to reflect changes in d_{ij}^{min} and maintain consistency with the predetermined cutoff. Although not utilised in this thesis, this functionality was encoded with the architecture and checks for updates at each step the masking layer is used. Furthermore, we highlight that the dynamic updating, and indeed all masking layers are required to be fully differentiable, as required for network stability, as it is highlighted in the padding implementations.

8.2 Network Integration

We now describe how two functions, crucial to network training, namely the loss function computation and the gradient updates within backpropagation, have to be altered to accommodate for padding and masking layers.

The loss function must account for the variable number of physical atoms across any given of unit cells,

$$L^M = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} M_i \cdot ||Y_i - \mathcal{Y}_i||, \quad (8.5)$$

where M is our mask matrix, $|\mathcal{V}|$ is the number of real atoms in our unit cell, Y_i is our predicted output from the network and $\mathcal{Y}_i$ is the true value target, which the network attempts to predict. The normalisation ensures that larger unit cells do not heavily bias the loss function.

The masking protocol is crucial when we consider the backpropagation process, and prevents gradients from passing through the padded elements. We thus alter the gradient computation as,

$$\frac{\partial L}{\partial \theta} = \sum_{ij} M_{ij} \cdot \frac{\partial L}{\partial \text{message}_{ij}} \cdot \frac{\partial \text{message}_{ij}}{\partial \theta}, \quad (8.6)$$

where L denotes the loss function and θ represents the overall network parameters. The gradient flow in accordance to this equation prevents the network from learning patterns that do not pertain to the underlying physics, which would lead to spurious results in atomic force predictions.

8.3 Silica Dataset

The silica dataset employed in this work was provided by [66][67]. Reference calculations were performed using density functional theory, as implemented in the Quantum ESPRESSO package [68], employing the generalised gradient approximation with GGA-PBE exchange-correlation functional [9], and Projector Augmented Wave (PAW) pseudopotentials. It features 1062 periodic silicon-based structures, spanning diverse crystallographic configurations, cell sizes, and thermodynamic states. These structures range from minimal single-atom cells,

to large 228-atom supercells, providing comprehensive coverage of silica-based periodic systems. This heterogeneity is essential for training a robust force-prediction model, which is capable of generalising across the varied landscape of silica polymorphs encountered in computational materials science. This, however, represents a substantially more challenging undertaking than our molecular work, as the model must simultaneously handle variable cell sizes spanning two orders of magnitude, diverse crystallographic symmetries, and the additional representational complexity introduced by periodic boundary conditions.

The cell size distribution exhibits significant heterogeneity with $N_{\text{cell}} \in [1, 288]$ atoms. Furthermore, the distribution is multi-modal with primary concentrations of cell sizes at $N_{\text{cell}} = 64$ (233 structures, 21.9%), $N_{\text{cell}} = 96$ (127 structures, 12.0%), $N_{\text{cell}} = 72$ (89 structures, 8.4%), $N_{\text{cell}} = 36$ (95 structures, 9.0%), and $N_{\text{cell}} = 12$ (78 structures, 7.3%).

The instances within this dataset span pure silicon entries, oxygen molecules, combinations of both, and silica dioxide. The atomic coordination and density patterns seen within it are consistent with various silica polymorphs, and ensures the network encounters differing coordination environments across the silica phase diagram.

The dataset also contains a strong cubic dominance, with 740 structures (69.7%) exhibiting cubic symmetry. The remaining dataset entries span a wide range of cell types, with triclinic, hexagonal, tetragonal, orthorhombic, rhombohedral and monoclinic cells present.

Force analysis also reveals a substantial amount of non-equilibrium content, which is of most interest to our model training and testing. Using an equilibrium threshold of 0.01 eV/Å, only 254 structures (23.9%) would qualify as equilibrium structures. This also comes with a strong cell-size dependence, with small primitive cells containing less than eight atoms having an equilibrium occurrence rate of 95.2%.

One consideration that needs to be taken into account is the effect of the extreme cell size variation will have with respect to memory overhead. The inclusion of padded matrices to maintain consistency throughout the network in terms of tensor sizing is the main contributor to this. Such issues can be partially mitigated through batch processing and GPU parallelisation, but these do not entirely solve the problem. Separate training and analysis will be conducted on subsets of the overall dataset to both help alleviate this issue, and investigate the change in model performance with a more focussed range of cell sizes.

Chapter 9

Results - Silica

We train our architecture extensively on the full silica dataset, and furthermore develop models for size-specific and cell specific subsets. The size-specific models use only unit cells containing 64, 72, and 96 atoms respectively, with the full dataset trained on all 1062 structures. This is done to investigate how the variety of physical phenomena present within the overall dataset would effect the trainability of our models, with the size-specific models being used as a trainability metric on comparison to this. Doing so explicitly allows us to isolate the effects of cell size variation on the model performance.

Table 9.1 shows the results from our model for the aforementioned dataset and subsets.

Results for Silicon Dataset Models				
Model	N _{structures}	MSE	MAE	R^2
64-atom cells	233	0.20479	0.35995	0.88921
72-atom cells	89	0.48023	0.55200	0.90076
96-atom cells	127	0.71757	0.67491	0.86021
Cubic cells	740	0.17303	0.33003	0.93033
Full dataset	1062	0.37541	0.48800	0.81376

Table 9.1: Performance metrics for models trained on size-specific subsets, cubic structures, and the complete silica dataset. MSE values in eV/Å², MAE in eV/Å. N_{structures} indicates training set size.

The parity plots for each of the models described in Table 9.1 are shown below. Note that values are truncated within the plots for visibility purposes, and forces greater than 10 eVÅ are omitted.

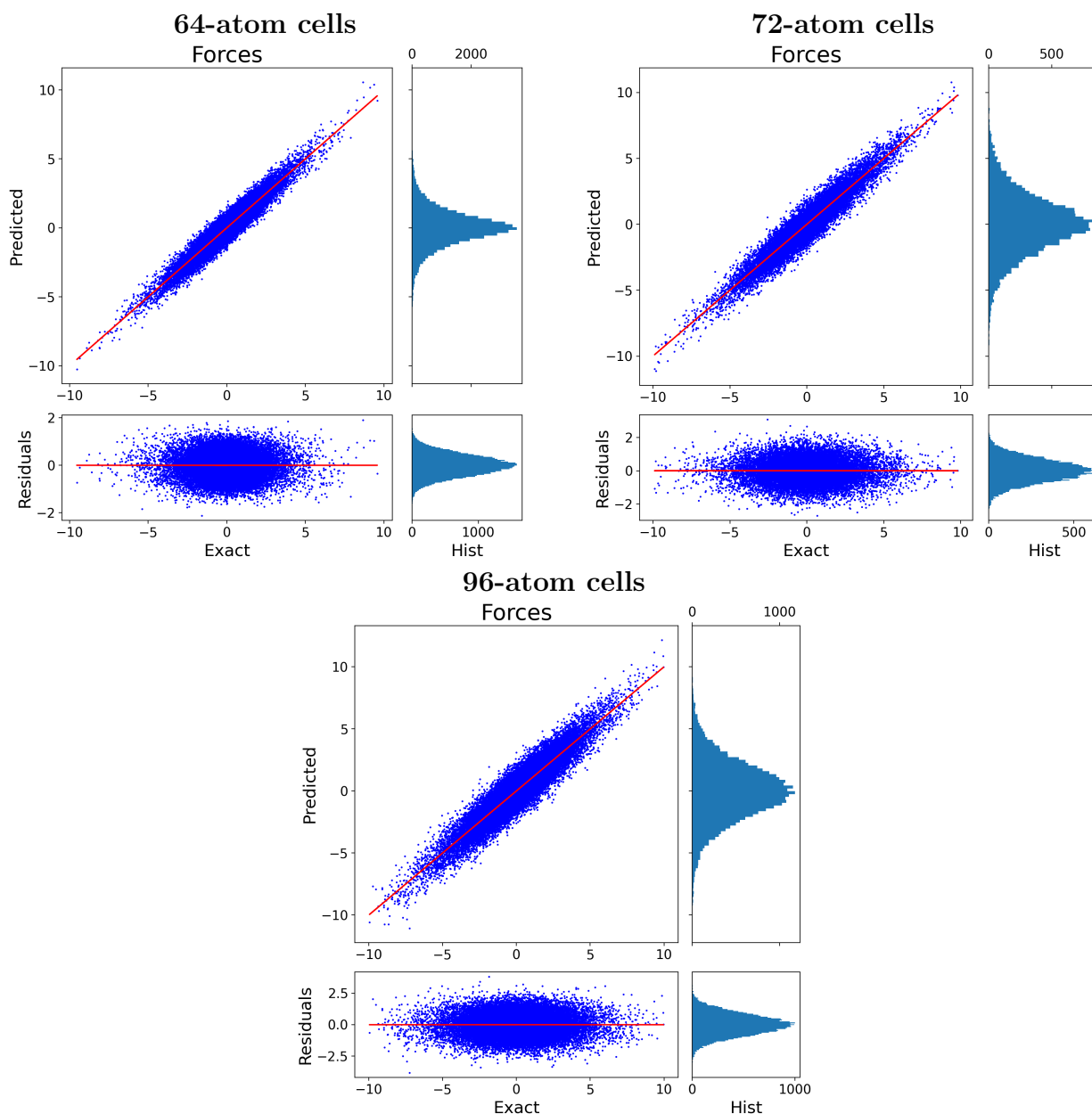


Figure 9.1: Parity plots for size-specific models trained on cells containing 64, 72, and 96 atoms respectively. The 64-atom model achieves $R^2 = 0.88921$ with 233 structures, demonstrating reasonable performance with adequate training data. The 72-atom model shows the highest $R^2 = 0.90076$ despite only 89 training structures, though this raises potential overfitting concerns given the limited dataset size relative to the others. The 96-atom model exhibits degraded performance ($R^2 = 0.86021$) with larger cell sizes and 127 structures, suggesting that message passing pathways struggle to propagate information accurately across the extended neighbourhoods presents in larger cells. These results isolate the effects of cell size variation on model performance and demonstrate that structural homogeneity within training subsets yields improved predictive accuracy compared to heterogeneous datasets.

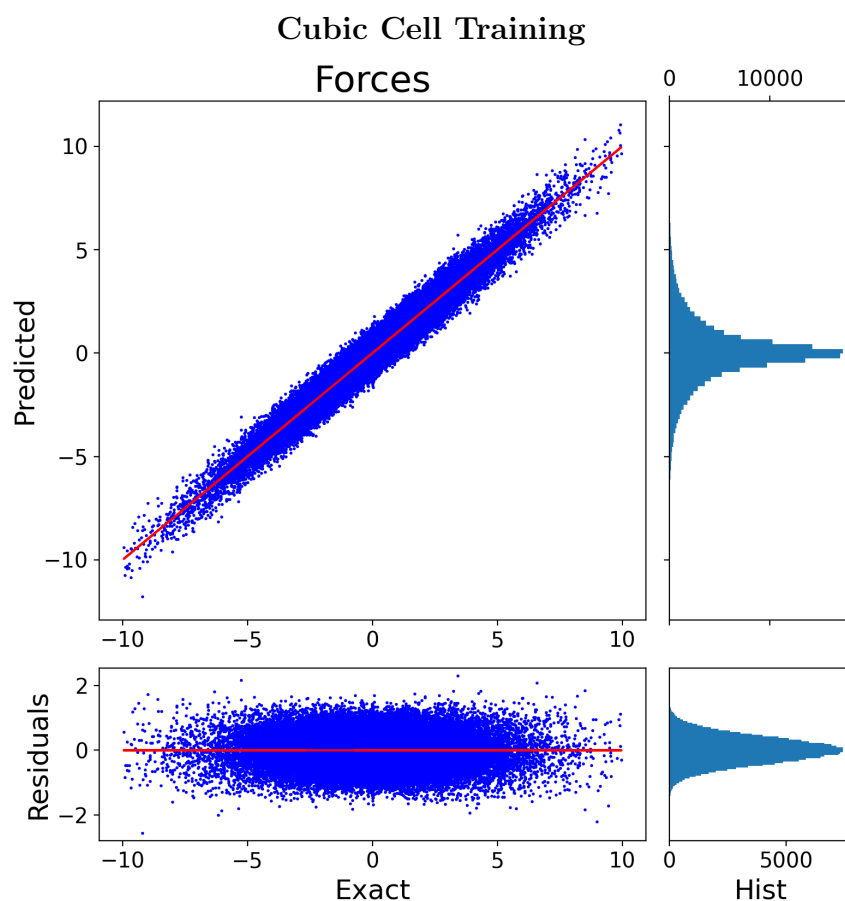


Figure 9.2: Parity plot for the cubic cell model. Training on the 740 cubic structures yields the best performance across all models ($R^2=0.93033$), demonstrating the advantage of symmetry-constrained datasets. The cubic symmetry reduces representational complexity by eliminating cell shape variations, allowing the network to focus on learning chemical and coordination environment effects without the influence of diverse crystallographic systems. This performance approaches the high-performing molecular systems from Part 1, suggesting that the architectural limitations may be partially overcome through strategic dataset curation.

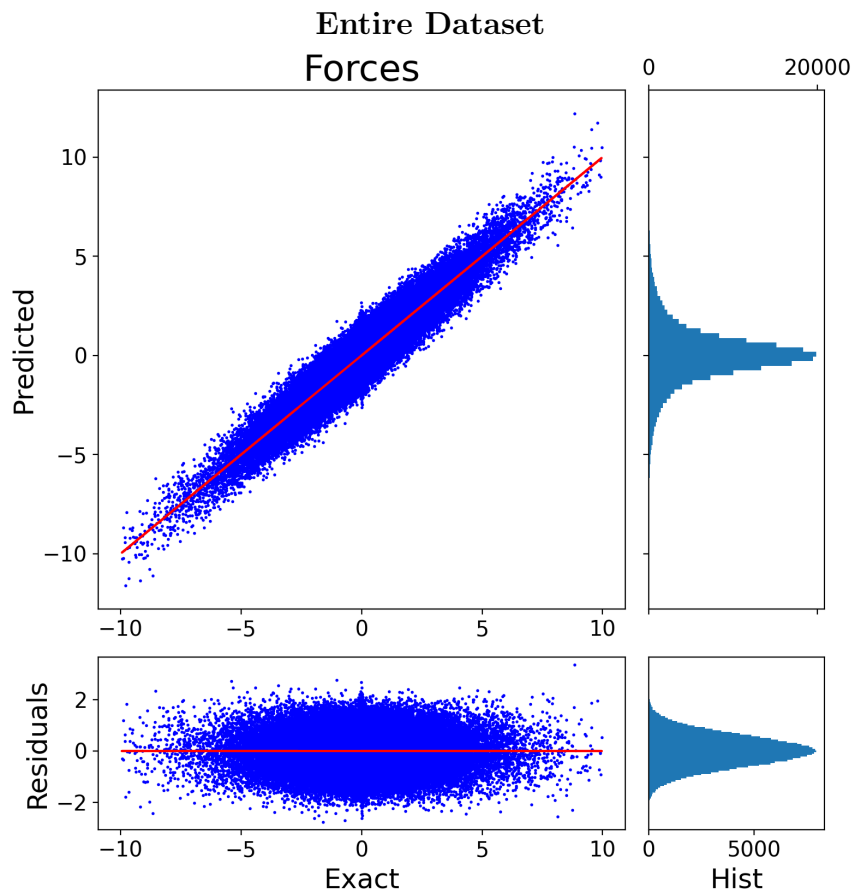


Figure 9.3: Parity plot for the full dataset model. The heterogeneous mixture of 1062 structures yields $R^2=0.81376$, reflecting the fundamental difficulty of generalising across diverse cell sizes and crystallographic systems. With cell sizes spanning two orders of magnitude and structures distributed across seven different space group types, the network encounters vastly different coordination environments, padding requirements, and distance calculations. The degraded performance of this model, indicated by the increased scattering along the parity line indicates that the network cannot reconcile the conflicting representational demands of simultaneously learning forces for small primitive cells and large supercells.

9.1 Parity Plot Analysis

The parity plots in Figures 9.1 through 9.3 exhibit similar visual characteristics. All of them show well-defined, linear correlations between predicted and exact forces with approximately symmetric residual distributions. This visual similarity, however, masks subtle differences in predictive performance, with the R^2 -scores span from 0.81367 to 0.93033. The consistency in plot appearance despite this varying performance metrics suggests that

the underlying dataset is internally well-structured. However, the quantitative performance differences reveal that visual inspection alone provides an insufficient assessment of model quality for periodic systems.

The results reveal a critical tension between dataset homogeneity and generalisation capability. The 72-atom model achieves a high R^2 -score, but with only 89 training structures, this should be interpreted with caution. The limited dataset size raises concerns about model overfitting. In particular, the elevated MSE scores compared to the 64-atom and cubic models may indicate that the model is able to capture overall trends, whilst struggling to predict individual force components with high levels of accuracy.

The full dataset model with 1062 structures achieves a relatively low R^2 -score of 0.81378. This demonstrates the main difficulty of training a universal force predictor across periodic systems of this scale. Specifically, with the number of atoms per cell spanning two orders of magnitude, the many different cell shapes within the dataset, various levels of padding within the model, the network struggles to transfer learned patterns between all of these differing attributes.

The 64-atom model achieves a decent R^2 -score with the lowest MSE of the results in Table 9.1. With 233 training structures, this represents a reasonable balance between dataset size and structural coherence, with overfitting less likely to occur compared to the 72-atom model.

The 96-atom model gives us insight into the overall models ability to interpret message passed information within the larger systems in our dataset. The R^2 -score of 0.86021 and MSE of 0.71757 demonstrate that message passing pathways struggle to propagate information accurately across extended neighbourhoods, although the caveat of only having 127 training structures may distort this interpretation as it is possibly not enough to train a network on, and good results may be an artifact of overfitting.

Interestingly, the cubic cell model demonstrates the most compelling performance, with an R^2 -score of 0.93033 from 740 training structures. This represents the strongest result across all periodic models and approaches the performance of our high-performing molecular systems. However, even this best-case scenario falls short of the best performing molecules from Part 1. It is reasonable to assume that here we are approaching a performance ceiling with our current architecture and that we have hit a fundamental limitation of the two-body

distance-based representations.

Beyond architectural limitations, we theorise that the symmetries involved in these periodic systems amplify the node alignment failures identified in Section 6. Cubic space groups possess up to 48 symmetry operations, each potentially generating distinct distance matrix arrangements for physically equivalent configurations. These symmetry induced degeneracies compound the representational limitations, and we provide a detailed analysis of this in the following section.

Our results establish that training separate models for structurally homogeneous subsets expectedly outperforms universal models, but gives us limited insight into which aspects of model architecture or dataset features contribute to trainability. Whilst the 64-atom model achieves the best balance of accuracy and dataset size among the size-specific models, the MAE remains too high for practical molecular dynamics applications. The full dataset’s degraded performance further reflects a fundamental insufficiency in the current approach. Progressing beyond the results shown here requires both architectural and potentially larger or more diverse datasets. In Part 3, we address these limitations by transitioning to a reliable Ni-Pd alloy dataset, which provides us with a pathway towards high-entropy alloy force-field development. Furthermore, we implement Echo State Networks (ESNs) as replacements for the standard message passing layers within our model and introduce reservoir computing dynamics in the hopes of capturing higher-order correlations without explicit many-body feature engineering. These modifications aim to overcome the limitations experienced with our periodic research thus far and allow us to scale to the complex spaces required for the metallic systems involved in the active field of high-entropy alloy computational research.

Chapter 10

Node Alignment Failures (Continued)

Whilst it is difficult to pinpoint where exactly the blame lies for the worse performing periodic models when compared to their molecular counterparts, we can extend our previous node alignment failures work to periodic systems to see how this could be an additional contributor to model performance issues. We do not expect this to be the main contributor here due to the large scope of variability involved in analysing periodic systems, but it is nonetheless an important feature to explore and understand.

The extension of graph neural network architectures from molecular systems to periodic systems can introduce fundamental complications to the node alignment problem under certain conditions. Periodic systems are represented through finite unit cells with periodic boundary conditions, where the MIC computes distances as,

$$d_{ij}^{\min} = \min_{\mathbf{n} \in \mathbb{Z}^3} \|\mathbf{r}_i - \mathbf{r}_j - \mathbf{R}(\mathbf{n})\|, \quad (10.1)$$

for lattice translation vectors $\mathbf{R}(\mathbf{n})$. This representation creates novel manifestations of representational ambiguity, beyond the cases observed within isolated molecules.

We can see this by considering a unit cell containing N atoms, represented as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, where all pairwise distances are computed via the minimum image convention. A valid automorphism is a node relabelling $\pi : \mathcal{V} \rightarrow \mathcal{V}$ that satisfies three conditions. Firstly, species preservation requires that $h_i = h_{\pi(i)}$ for all atoms $i \in \mathcal{V}$ and where h_i denotes the species type. Secondly, the distance preservation under MIC requires that $d_{ij}^{\min} = d_{\pi(i)\pi(j)}^{\min}$ for all pairs $i, j \in \mathcal{V}$. Thirdly, the bond preservation requires that bonded pairs remain bonded under the permutation.

The critical distinction between periodic and molecular systems emerges from how the MIC interacts with the unit cell boundaries. Consider a unit cell with two identical atoms of the same species located at positions $\mathbf{r}_1$ and $\mathbf{r}_2$. Under a regular Euclidean distance scheme, the distances within the cell between these two atoms would be $\|\mathbf{r}_1 - \mathbf{r}_2\|$. However, under the MIC, the distances are $\min_{\mathbf{n}} \|\mathbf{r}_1 - \mathbf{r}_2 - \mathbf{R}(\mathbf{n})\|$. This may select a periodic image which is an adjacent cell, thus, swapping the labels of these two atoms via permutation π will only constitute a valid automorphism, if the MIC distances from both atoms to all other atoms in the system remain identical after the swap. This is a highly restrictive constraint that depends on the entire spatial arrangement of the atoms within the overall system and the lattice parameters of the periodic images.

The automorphism group $\text{Aut}(\mathcal{G})$ for a periodic system under the MIC representation is thus determined by the intersection of two constraint sets. These are, the combinatorial constraint of species identity preservation, which yields the previously defined obfuscation measure $O_{\max}(\mathcal{G})$, and the geometric constraint of the MIC distance preservation, which depends on the specific atomic positions and lattice vectors. Unlike molecular systems, periodic systems have distance metrics, which fundamentally alter which permutations preserve the distance matrix structure.

To examine how crystallographic symmetries interact within this representational framework, we need to analyse the various space group symmetries and crystallographic automorphisms that can arise for such a representation.

10.1 Space Group Symmetries and Crystallographic Automorphisms

For a periodic crystal structure, described by the space group $\mathbf{S}$, we expect that the symmetry operations of $\mathbf{S}$ would naturally correspond to the automorphisms of the MIC-based graph representation $\mathcal{G}$. This correspondence is, however, not guaranteed, and depends critically on whether the space group operations preserve the MIC distance structure.

A space group operation $(W, \mathbf{w})$, where W is an orthogonal matrix and $\mathbf{w}$ is a translation vector, maps the position $\mathbf{r}_i$ to $W\mathbf{r}_i + \mathbf{w}$. This generates a valid automorphism in the case

that $d_{ij}^{\min} = d_{\min}(W\mathbf{r}_i) + w$ for all pairs i, j .

For symmorphic space groups, where operations decompose as $(W, \mathbf{t})$ with $\mathbf{t}$ a lattice translation vector, distance preservation is guaranteed [69]. This is due to the lattice translation cancelling out within all MIC calculations, since W is orthogonal and maps lattice vectors to lattice vectors, as seen in,

$$d_{\min}(W\mathbf{r}_i + \mathbf{t}, W\mathbf{r}_j + \mathbf{t}) = \min ||W(\mathbf{r}_i - \mathbf{r}_j) - \mathbf{R}(\mathbf{n})|| = d_{ij}^{\min}. \quad (10.2)$$

For nonsymmorphic space groups containing glide planes or screw axes with fractional translations, the operation which preserves MIC distances depends heavily on the lattice parameters and atomic positions.

We can decompose the automorphism group for periodic systems as,

$$\text{Aut}(\mathcal{G}) = \text{Aut}_{\text{int}}(\mathcal{G}) \times \text{Aut}_{\text{space}}(\mathcal{G}), \quad (10.3)$$

where $\text{Aut}_{\text{int}}(\mathcal{G})$ represents automorphisms from the internal atomic arrangements independent of space group symmetries, akin to the case with molecular automorphisms. $\text{Aut}_{\text{space}}(\mathcal{G})$, instead, captures additional automorphisms from space-group operations, which preserve the MIC distances.

In comparing the automorphisms between molecular and periodic systems, we see that the automorphism group cardinality for periodic systems exhibits fundamental differences from molecular analogues. If we consider a molecular crystal where identical molecules are arranged periodically, the automorphic group of the isolated molecule is defined entirely by its inherent internal molecular symmetries. The periodic unit cell under MIC, however, must additionally account for space group operations, yielding,

$$|\text{Aut}(\mathcal{G}_{\text{cell}})| \geq |\text{Aut}(\mathcal{G}_{\text{mol}})|, \quad (10.4)$$

with equality only when there are no space group operations, beyond the internal symmetries of the system, preserving the MIC distance matrix. This collapse to molecular behaviour occurs in two key scenarios. Firstly, when the crystal structure belongs to low-symmetry

space groups. For example, the triclinic P1 space group, which has $|\text{Aut}_{\text{space}}(\mathcal{G})| = 1$ [70]. This renders the periodic system equivalent to its hypothetical molecular analogue. Secondly, when nonsymmorphic space group operations with fractional translations fail to preserve MIC distances due to specific lattice parameters, leaving only $\text{Aut}_{\text{int}}(\mathcal{G})$ to contribute to the total group automorphism. In these cases, the node alignment problem for periodic systems collapses to exactly that of molecular systems, with the training complexity at the network interfacing stage being determined solely by the internal symmetries rather than crystallographic symmetries.

Amorphous materials represent an extreme limiting case, as these structures lack long-range translational order and space groups entirely [71]. Specifically, for amorphous structures, $|\text{Aut}(\mathcal{G}_{\text{amorph}})| \rightarrow 1$, as irregular atomic arrangements eliminate structural equivalences. Here, node alignment failures arise almost exclusively from species-identical atom relabelings. This suggests that amorphous systems exhibit training characteristics equivalent to asymmetric molecular systems, which would indicate the requirement of comparable training dataset sizes despite being periodic in nature. This is offset by the sheer complexity of a typical amorphous system in comparison to a small molecular system, and additional measures not relevant to this section would be required.

For neural network training on periodic systems, the automorphism group cardinality $|\text{Aut}(\mathcal{G})| = k$ determines the effective training complexity from a network interfacing standpoint. Each configuration can present as k distinct distance matrix arrangements to the network despite representing identical physical structures. High-symmetry crystalline structures with large k values would require proportionally larger training datasets to achieve comparable predictive accuracy to low-symmetry or molecular systems.

**Part 3 - Towards High-Entropy Alloys
and the Energy Surface Interpolation
Scheme**

Chapter 11

Introduction to High-Entropy Alloys

We now move to describe the application of our computational framework to more complex metallic systems. This section utilises Ni-Pd alloys as a test bed for methods that will ultimately, target the analysis and research of *high-entropy alloys* (HEAs).

High-entropy alloys are metallic materials composed of five or more principal elements in near-equi-molar proportions. These were first investigated by Yeh et al. [72], and Cantor et al. in 2004 [73]. Unlike conventional alloys, which add minor elements to a dominant base metal, HEAs leverage configurational entropy to stabilise single-phase solid solutions despite containing multiple principal components. The defining criterion specifically requires that the configurational entropy $S_{\text{conf}} \geq 1.5R$ per mole, where R is the gas constant. This corresponds to at least five equimolar elements with $S_{\text{conf}} = R \ln n$ for n components [74]. This formula represents the ideal configurational entropy. In practice, short-range ordering and non-ideal mixing can reduce the actual entropy below this theoretical maximum.

These materials exhibit remarkable properties that challenge the conventional wisdom of modern metallurgy. HEAs demonstrate exceptional mechanical strength retention at elevated temperatures, often maintaining yield strengths above 500 MPa at temperatures exceeding 800°C, where traditional superalloys would often soften significantly [75]. The resistance to oxidation and corrosion surpasses many conventional stainless steels, with certain refractory HEAs, forming protective oxide scales in environments where nickel-based superalloys can fail [76]. The CoCrFeMnNi Cantor alloy maintains ductility and exhibits fracture toughness exceeding $200 \text{ MPa}\sqrt{\text{m}}$ at cryogenic temperatures approaching 77K, where body-centered cubic steels undergo brittle failure [77]. Additional HEA compositions show superior resistance to radiation-induced damage, with reduced void swelling and helium bubble

formation under neutron irradiation compared to austenitic stainless steels [78].

These properties make HEAs attractive candidates for aerospace turbine components operating at extreme temperatures, next-generation nuclear reactor structural materials requiring radiation tolerance, cryogenic storage vessels for liquified gases, and marine applications, which demand corrosion resistance in chloride-rich environments [77][79][80][81]. The potential applications can extend to the possibility of hydrogen-storage materials, catalytic systems, and thermoelectric devices, though realising these applications requires accurate computational screening of the vast materials space available in order to identify possible formulations, without exhaustive experimental synthesis [82][83][84].

From a computational standpoint, HEAs present extreme challenges, which push machine-learning methodologies to their limits. For an n -component equimolar alloy distributed across N lattice sites, the configuration space scales as n^N , yielding approximately 10^{70} possible atomic configurations for a 100-atom quinary cell. This combinatorial explosion precludes exhaustive sampling via DFT calculations. Furthermore, each atom experiences chemically distinct local environments depending on the composition of its nearest-neighbourhood construction. This chemical heterogeneity means that force fields vary dramatically across geometrically similar lattice sites due to local compositional fluctuations. This creates a highly complex energy landscape, which defies any form of simple analytic description.

The sparsity of available training data exacerbates these issues. DFT calculations for HEA supercells containing hundreds of atoms, considered the minimum size required to achieve a statistically sound representation of all species and coordination environments, approach levels of computational costs which become prohibitive due to the inherent $O(N^3)$ scaling of conventional DFT with respect to the system size [85][86]. Furthermore, the need to sample multiple configurations to capture chemical disorder further compounds this computational burden [7]. Generating training datasets of sufficient size to span the configuration space becomes economically and temporally unfeasible. Thus, the responsibility moves to the development of machine-learning techniques to extract maximal information from limited training examples, and develop methods to accurately generalise to unseen compositional regions.

Chapter 12

Echo State Networks

To advance the capabilities of our networks, we chose to implement echo state architecture into the message passing scheme, with the expectation that this would aid in the predictive power of the calculation of forces on atoms in periodic systems. Echo State Networks (ESNs) represent a specialised class of recurrent neural networks, which were created to address challenges pertaining to traditional recurrent architectures. Introduced by Jäger in 2000, in tandem with Maass’s development of Liquid State Machines, ESNs pioneered what is now collectively termed as ‘reservoir computing’ [87][88]. This framework is built around a large, randomly initialised recurrent network, which acts as a fixed dynamical system, or ‘reservoir’, which is not subject to parameter training. This network projects input signals into a high-dimensional state space, whilst only a single linear readout layer at the end of the network is utilised for training. This architectural choice aims to eliminate notorious difficulties inherent to training fully connected and recurrent networks through gradient-based methods, transforming what was a complex nonlinear optimisation problem into a tractable linear regression task that can be solved in a much more efficient manner due to the minimal number of non-fixed neurons within the model.

Whilst ESNs were originally conceived, and are predominantly applied to, temporal sequence processing tasks (time series prediction, signal processing, speech recognition etc.) the fundamental principles within the underlying use of computational reservoirs extend naturally beyond purely temporal domains. We believe there are strong use cases for ESNs outside of their initially designed use case, and that they can be applied in many cases of high-dimensional function approximation regardless of sequential dependence. Recent investigations have demonstrated that ESNs can effectively process static pattern recognition

tasks, spatial data classification problems and even molecular graph-structured feature extraction problems, the latter of which influenced our decision to implement them within our model [89][90][91]. The reservoir’s ability to perform complex, fixed random projections into higher-dimensional spaces creates diverse representations that can be exploited for various machine-learning tasks and, for our purposes, this adaptability proves particularly relevant. ESN-based message passing can augment, or potentially replace, traditional forms of GNN methods for processing atomic configurations and crystalline structures, providing computationally efficient alternatives to fully trained message passing layers whilst maintaining the full predictive capacity that traditional message passing schemes afford us.

The fundamental architecture of an ESN comprises of three distinct functional components that differ quite substantially from both traditional feedforward neural networks and recurrent networks. The input layer provides a simple linear projection of the input data into the reservoir’s internal dimensionality through randomly initialised weights, which remain fixed during training and application. The reservoir itself consists of a large collection of sparsely connected recurrent neurons, typically on the order of hundreds. The connectivity pattern (which determines the neurons connectivity) and the connection weights are established through random initialisation upon network creation and neither the topology nor the weight values are ever modified during training. Connections are formed with a low probability, often referred to as the ‘density’, and typically created with a chance of 1 – 10% following sparse random graph generation procedures. The weights are randomised according to standard neuron Gaussian initialisation, after which the entire weight matrix undergoes spectral radius scaling (rescaling the weight matrix so that its largest eigenvalue falls below unity) to ensure stability, which is detailed further in Section 12.2. Finally, the output layer represents the sole trainable component of the network, implemented as a simple linear transformation from high-dimensional reservoir state space down to the desired output dimensionality. The weights and biases of this output layer are trained via standard gradient descent procedures.

A diagram displaying the input/output layer interaction with the reservoir is shown in Fig. 12.1.

Beyond temporal sequence processing, the reservoir computing paradigm adapts natu-

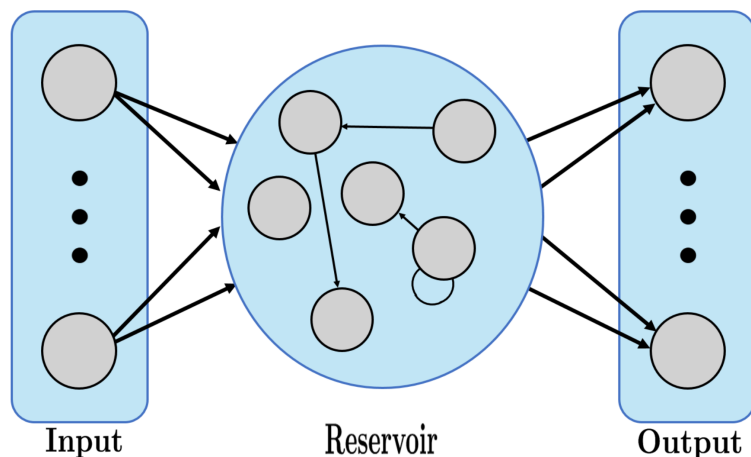


Figure 12.1: Diagram showing the architecture of an Echo State Network and its three functional components. Namely, the input layer, with randomly initialised fixed weights projecting into the reservoir, the reservoir layer consisting of sparsely connected recurrent neurons with fixed random connectivity, and the output layer containing the sole trainable weights. Arrows indicate information flow through the network. Note, the data flow within the reservoir is randomised at initialisation and self loops are possible. The weights within this reservoir to not get updated.

rally to graph-structured data due to the similarity of mappings from iterative state updates in ESNs and message passing operations in GNNs. In graph echo state networks, the reservoir implements a fixed message passing scheme, where node representations undergo recurrent updates through aggregation of neighbouring node states, analogous to standard ESN reservoir dynamics, but operating over graph topology rather than temporal sequences. For molecular and crystalline systems represented as graphs, an ESN-based message passing scheme could aggregate information from neighbouring atoms through randomly weighted combinations, generating diverse node embeddings that encode both local chemical environments and broader structural context. Feeding these representations to trained readout layers for energy or direct force prediction without the computational load of expensive, full network, backpropagation through relatively large message passing architectures across multiple layers and over thousands of epochs saves a huge amount of computational resources alongside the potential increase in overall model predictive power and accuracy.

12.1 Mathematical Formulation

We move on to detail the mathematical specifics of a traditional ESN. The use of temporal phrases remain to aid understanding and display the intended use case, but we stress that the networks are not limited to time-series and temporal data problems.

For a discrete-time echo state network of K input units, N reservoir neurons, and L output units, the reservoir state vector $\mathbf{x}(n) \in \mathbb{R}^N$ at discrete time n evolves in accordance to,

$$\mathbf{x}(n+1) = \sigma(\mathbf{W}^{\text{in}}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{\text{fb}}\mathbf{y}(n)), \quad (12.1)$$

where $\mathbf{u}(n) \in \mathbb{R}^K$ represents the input vector, $\mathbf{y}(n) \in \mathbb{R}^L$ denotes the output vector from the previous time-step, $\mathbf{W}^{\text{in}} \in \mathbb{R}^{N \times K}$ contains the weights, which map the random input to reservoir neurons, $\mathbf{W} \in \mathbb{R}^{N \times N}$ represents the sparse random recurrent weight matrix, $\mathbf{W}^{\text{fb}} \in \mathbb{R}^{N \times L}$ specifies optional output feedback weights, and $\sigma(\cdot) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ denotes a component-wise nonlinear activation function. The three weighted terms each contribute in distinct ways. The input weights $\mathbf{W}^{\text{in}}$ drives the reservoir with external signals, $\mathbf{W}\mathbf{x}(n)$ implements temporal integration through state feedback, and $\mathbf{W}^{\text{fb}}\mathbf{y}(n)$ enables autonomous pattern generation in closed-loop scenarios.

The output computation implements as simple linear readout transformation of the augmented state vector,

$$\mathbf{y}(n+1) = \mathbf{W}^{\text{out}}[\mathbf{u}(n+1); \mathbf{x}(n+1); 1], \quad (12.2)$$

where $\mathbf{W}^{\text{out}} \in \mathbb{R}^{L \times (K+N+1)}$ represents the trainable readout weight matrix, the sole component of the network subject to optimisation. The concatenation $[\mathbf{u}(n+1); \mathbf{x}(n+1); 1]$ combines the current input, current reservoir state, and a bias term into a single vector, thereby allowing the readout to directly access input information alongside the outputted information post-reservoir processing. The inclusion of direct input-to-output connections in the concatenation is beneficial for tasks, where certain output components depend primarily on immediate input values rather than complex temporal integration. This effectively provides us with a learned bypass around the reservoir dynamics for simple input-output relationships whilst still leveraging the power of reservoir states for capturing complex pat-

terms.

The activation function $\sigma(\cdot)$ is typically chosen to be the hyperbolic tangent function.

$$\sigma(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (12.3)$$

The sigmoidal shape, saturating behaviour for large inputs, bounded range, and smooth differentiability provides desirable characteristics for model stability and trainability.

Standard ESNs assume instantaneous state updates, potentially limiting representational capacity for processes across vastly different timescales. Leaky integrator neurons introduce temporal smoothing through first-order low-pass filtering,

$$\mathbf{x}(n+1) = (1 - \alpha)\mathbf{x}(n) + \alpha\sigma(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{fb}\mathbf{y}(n)), \quad (12.4)$$

where $\alpha \in (0, 1]$ represents the leak rate. This controls the balance of state persistence between instances of data flow. Small leak rates produce smoothed data processing trajectories with a slow overall evolution, enabling long-term integration appropriate for low-frequency dynamics. Conversely, large leak rates yield rapid, forgetful, and fast dynamics, suitable for quickly varying data signals. Combined with spectral radius tuning, leaky integrator ESNs span a two-dimensional parameter space for fine tuning the demands of the task at hand. Their inclusion and tweaking as a hyperparameter will be discussed later. Empirically, leaky integrator ESNs have demonstrated improved performance over standard ESNs for tasks requiring multi-timescale dynamics [92].

The performance of echo state networks depends critically on the selection of several global hyperparameters that dictate the dynamics and learning characteristics of the reservoir. Unlike fully trained networks, where suboptimal initialisation settings can be compensated through extensive training via increasing total number of epochs, ESN reservoirs lack this adaptability. Poor hyperparameter choices lead to a fundamental limit of performance and the importance of correct tuning is paramount for obtaining usable models.

12.2 Spectral Radius and the Echo State Property

The spectral radius $\rho(\mathbf{W})$ of the reservoir weight matrix is the main component that ensures control over the reservoir stability and memory capacity. It is defined as the maximum absolute eigenvalue,

$$\rho(\mathbf{W}) = \max |\lambda_i| : \lambda_i \text{ is an eigenvalue of } \mathbf{W}. \quad (12.5)$$

For the network to ensure that reservoir dynamics remain driven by inputs rather than internal instabilities, the condition $\rho(\mathbf{W}) < 1$ is necessary. This ensures autonomous reservoir dynamics converge towards the origin. It is not, however, the only condition which guarantees stability in general, as nonlinear activation functions and input coupling can produce behaviours that cannot be captured by linear eigenvalue analysis. A sufficient condition involves the maximum singular value, which states that the largest singular value of $\mathbf{W}$ must be less than 1. In practice, reservoir matrices are scaled by,

$$\mathbf{W} \leftarrow \frac{\rho_{\text{target}}}{\rho(\mathbf{W})} \mathbf{W}, \quad (12.6)$$

where ρ_{target} is commonly chosen to be near, but slightly below, unity. Values close to the unity position place the reservoir in a state which can exhibit rich complex behaviour, without descending into unstable divergence. These overall sets of conditions and scaling rules are necessary for the optimal operation of an ESN, known collectively as the echo state property.

12.3 Implementation as Message Passing in GNNs

Having established the theoretical foundations of ESNs, we now detail their integration into the graph neural network architectures. The architectural parallels between echo state networks and graph neural networks suggest the existence of a natural integration of ESNs as a message passing technique. Both frameworks implement iterative refinement of node representations through aggregation of neighbourhood information, with ESNs performing

temporal aggregation across sequential inputs whilst GNNs aggregate spatial information from graph-connected neighbours. The mathematical structure of standard message passing GNN layers directly parallels ESN reservoir updates, where current states combine previous states with aggregated external information.

For molecular and crystalline systems represented as graphs with atomic positions, species and bonding connectivity, we implement ESN-inspired message passing through a reservoir-based message computation scheme. Each message passing layer contains an echo state reservoir that processes concatenated node and edge information to generate messages, which are then aggregated and used to update node representations. This is effectively taking the architecture displayed in Fig. 12.1, and inserting it into each message passing block within our equivariant message passing architecture, shown in Fig. 3.1.

The core message passing mechanism operates as follows. For each edge connection atoms i and j with edge features e_{ij} , we construct a message input by concatenating source node features, target node features and edge attributes,

$$\mathbf{m}_{ij}^{\text{input}} = [\mathbf{h}_i; \mathbf{h}_j; \mathbf{e}_{ij}], \quad (12.7)$$

where $\mathbf{h}_i \in \mathbb{R}^{d_{\text{atom}}}$ and $\mathbf{h}_j \in \mathbb{R}^{d_{\text{atom}}}$ represent atom feature vectors and $\mathbf{e}_{ij} \in \mathbb{R}^{d_{\text{edge}}}$ contains edge features, namely, atomic distances. This concatenated input is processed through an echo state reservoir,

$$\mathbf{r}_{ij} = (1 - a)\mathbf{r}_{ij}^{\text{prev}} + a \cdot \tanh(\mathbf{W}^{\text{in}}\mathbf{m}_{ij}^{\text{input}} + \mathbf{W}^{\text{res}}\mathbf{r}_{ij}^{\text{prev}}), \quad (12.8)$$

where $\mathbf{r}_{ij} \in \mathbb{R}^{N_{\text{res}}}$ represents the reservoir state for a particular message, $\mathbf{W}^{\text{in}} \in \mathbb{R}^{N_{\text{res}} \times (2d_{\text{atom}} + d_{\text{edge}})}$ is the randomly initialised input weight matrix, $\mathbf{W}^{\text{res}} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{res}}}$ is the fixed sparse random reservoir weight matrix scaled to the desired spectral radius, and a is the leaking rate.

The reservoir state is then projected to the message space through a trained linear transformation as follows,

$$\mathbf{m}_{ij} = \mathbf{W}^{\text{proj}}\mathbf{r}_{ij}, \quad (12.9)$$

where $\mathbf{W}^{\text{proj}} \in \mathbb{R}^{d_{\text{atom}} \times N_{\text{res}}}$ is a trainable matrix, which creates a map from the high-

dimensional space of the reservoir, down to the dimensionality of the atom feature space. Messages arriving to each atom are then aggregated through a simple summation as follows,

$$\mathbf{m}_i^{\text{agg}} = \sum_j \mathcal{N}_i \cdot \mathbf{m}_{ij}, \quad (12.10)$$

where $\mathcal{N}_i$ denotes the neighbourhood of an atom i , determined by a predetermined cutoff radius and setup of the MIC previously detailed in Section 7.3. The aggregated messages are finally combined with the current atom features through a trained update function, completing the node-wise message passing process,

$$\mathbf{h}_i^{\text{new}} = \mathbf{h}_i + \text{MLP}([\mathbf{h}_i; \mathbf{m}_i^{\text{agg}}]), \quad (12.11)$$

where the MLP (multi-layer perceptron) implements a residual connection, which preserves the original features of a node, whilst incorporating the messages gained from the local neighbourhood of each atom.

This architecture separates the fixed random nonlinear reservoir parameters from the projection and update neuron layers, and concentrates the trainable parameters into components that directly influence task outputs whilst leveraging the reservoir’s high-dimensional dynamics, which are utilised for feature extractions.

The message passing equations described in Eqs. (12.7) and (12.11) constitute a single layer of the network. Multiple layers can be stacked sequentially, as implemented in our model, with the features $\mathbf{h}_i^{\text{new}}$ from layer l acting as the input features for layer $l + 1$. Each layer maintains its own set of unique reservoir weights and trainable projection and update parameters. For our molecular force predictions, we utilise three message passing layers, which provide sufficient scope to capture local chemical environments whilst avoiding any form of over smoothing phenomena that can occur in large deep neural networks. Mirroring this choice, we strike a good balance between meaningful information propagation and computational cost.

After the final message passing layer outputs the finalised features for all atoms, the force prediction process proceeds through a trained linear readout, which maps the atomic

features to three dimensional force vectors,

$$\mathbf{F}_i = \mathbf{W}^{\text{out}} \mathbf{h}_i^L + \mathbf{b}^{\text{out}}, \quad (12.12)$$

where $\mathbf{F}_i \in \mathbb{R}^3$ is the predicted force vector for each atom i , $\mathbf{W}^{\text{out}} \in \mathbb{R}^{3 \times d_{\text{atom}}}$ contains the weights of the readout dense layer, and $\mathbf{b}^{\text{out}} \in \mathbb{R}^3$ is the trainable bias vector. This output layer linearly combines the rich atomic feature representations calculated by the ESN-message passing scheme and transforms them into accurate target predictions.

The complete network is trained via standard backpropagation procedures, minimising over the mean squared error loss function between predicted and reference data, as previously detailed in Section 2.4.

The ESN-based message passing scheme can be seen as a mid-point between fully fixed random features and a fully trained neural message passing scheme. Whereas traditional GNN message passing functions implement learned transformations wherein all parameters involved undergo training, providing maximum flexibility to adapt to task requirements at the cost of computational resources, this approach offers much more training efficiency, at the cost of potentially limited expressivity.

Our approach seeks to combine the positive aspects of both methods. The reservoir provides fixed, random, nonlinear transformations analogous to random features, and trainable projections and updates allow for task-specific adaptation of how the reservoir states map to messages and various different node representations. This hybrid architecture captures benefits from both extremes and yields a practical balance for both molecular and periodic property prediction.

Due to our restricting of model training to only projection matrices, update MLPs and final readout layers, we gain substantial advantages with respect to computational resources compared to our previous standard message passing implementation.

The number of trainable parameters scales as $O(d_{\text{atom}} N_{\text{res}} L)$ for projection matrices across L layers, plus parameters in the final MLP readout layer. Compared to the $O(d_{\text{hidden}}^2 L)$ parameter count for fully trained message passing networks, where hidden layer interactions require learnable weight matrices of dimension $d_{\text{hidden}} \times d_{\text{hidden}}$, the ESN-based approach

offers a massive parameter count reduction.

Beyond parameter efficiency, the fixed reservoir architecture mitigates the gradient pathologies that can plague fully trained recurrent schemes. As reservoir weights are fixed, the optimisation landscape is simplified. Gradients update only the projection matrices and readout layers, rather than propagating through the full system of recurrent dynamics. This reduces vanishing or exploding gradient issues commonly encountered when training deep recurrent architectures. This translates to a more stable training regime with reduced sensitivity to learning rate choices, and needing fewer numbers of epochs to train.

12.4 Ni-Pd as a Binary Foundation

Our research focuses on the binary Ni-Pd system, which gives us a computationally tractable intermediate between pure crystalline materials and true HEAs. The binary case reduces the configurational complexity of the systems we analyse from 5^N to 2^N whilst still introducing the challenge of composition-dependent force fields.

The Ni-Pd system provides several practical advantages for methodological development. Both nickel and palladium adopt face-centered cubic crystal structures with lattice parameters differing by only $\sim 10\%$. As shown in Table 9.1, we expect this to be a relatively optimal form to utilise within model architecture development. Furthermore, to our advantage, the system forms continuous solid-state solutions across all compositions without any phase separation or ordered intermetallic formations [93]. This relative computational simplicity eliminates any form of complications arising from structural phase transitions which would introduce additional variables beyond the core challenge of approximating properties of the underlying binary system. Furthermore, the electronic structures of Ni ($4s^23d^8$) and Pd ($4d^{10}$) differ enough to create measurable variations in local bonding environments, whilst still remaining chemically similar, to the extent that both species occupy Face Centered Cubic (FCC) lattice sites without large local distortions.

The progression from silica to Ni-Pd to eventual HEAs represents a systematic escalation in representation difficulty. Each step introduces new challenges whilst building upon established foundations. The minimum image conventions, periodic boundary conditions,

and equivariant message passing schemes developed for silica extend naturally to Ni-Pd. We demonstrate that our computational methods maintain accuracy despite the increased compositional disorder and establish a confidence in the robustness of these approaches for future HEA applications.

Chapter 13

Force Calculations and Energy Surface Interpolations

The prediction of atomic forces represents one of the most critical challenges in machine-learning-driven materials science calculations. Whilst neural networks can achieve remarkable accuracy in predicting potential energies through direct prediction on precalculated training data, the extraction of corresponding forces introduces additional layers of complexity, which merit careful consideration and investigation.

In our prior work on periodic systems in Section 8.3, we trained neural network architectures to predict atomic forces directly from structural configurations. Whilst this approach proved effective for certain subsets, in particular the cubic and size-specific models, the full dataset model exhibited significant performance degradation with force prediction R^2 -scores substantially below acceptable thresholds. The parity plots revealed that the models struggled to generalise across structurally diverse configurations.

These failures motivated a reconsideration of our training strategy for periodic systems specifically. Whilst direct force predictions had proven highly effective for molecular systems, the heterogeneous silica dataset presented unique challenges. Rather than targeting forces as regression outputs, we shift focus to energy prediction as the initial learning objective. The scalar nature of energy predictions simplifies the learning problem for diverse training sets, and the inherent smoothness of well-trained energy functions provides a more stable foundation from which to derive forces through post-processing interpolation methods.

The methods developed in this section represent a comprehensive exploration of techniques for optimising force calculations from machine-learned potential energy surfaces. To

establish this framework rigorously, we need to examine the fundamental relationship between energy and forces within our framework that motivates our interpolation-based approach. Full understanding will reveal both why direct backpropagation through neural networks can be problematic and how careful construction of interpolated surfaces can mitigate these issues.

Given a potential energy surface, atomic forces arise naturally as the negative gradient of the potential energy surface with respect to the atomic coordinates,

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} E(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n) = \frac{\partial E}{\partial \mathbf{r}_i}, \quad (13.1)$$

where $\mathbf{F}_i \in \mathbb{R}^3$ is the force vector acting on an atom i , and $\mathbf{r}_i$ is its corresponding position vector. This relationship establishes that accurate force predictions are directly linked to the local geometric properties of the energy surface. For neural network models that learn energy functionals, parametrised by weights and operating on molecular graphs, the forces can theoretically be obtained through automatic differentiation by,

$$\mathbf{F}_i = \frac{\partial E_\theta}{\partial \mathbf{r}_i} = \sum_{l=1}^L \frac{\partial E_\theta}{\partial a^l} \frac{\partial a^l}{\partial \mathbf{r}_i}, \quad (13.2)$$

where E_θ is an energy functional with network weights θ , a^l denotes the activations at layer l , and L is the total number of layers. This shows that the chain rule can propagate backwards through L layers of the network architecture in an approach known as propagation-based force calculation. This is the standard methodology in most contemporary neural network potential implementations. [94].

Despite the mathematical simplicity and computational efficiency of the above equation, direct force calculation through automatic differentiation of energy predictions presents several challenges that can compromise the reliability and accuracy of the resulting force fields. The most significant of them stems from the inherent sensitivity of derivative operations to small perturbations in the underlying function, specifically, due to differentiation amplifying high-frequency noise components, which can be present in the learned energy function [25][95].

To understand this phenomenon rigorously, consider a one-dimensional energy function

with a small high-frequency perturbation, represented as a sinusoidal function,

$$E_{\text{approx}}(x) = E_{\text{true}}(x) + \epsilon \sin(\omega x), \quad (13.3)$$

where $\epsilon \ll 1$ and $\omega \gg 1$. This has the following derivative,

$$F_{\text{approx}}(x) = F_{\text{true}} + \epsilon \omega \cos(\omega x). \quad (13.4)$$

Whilst the relative error in the energy is bounded by $\frac{\epsilon}{E_{\text{true}}}$, the force error is now amplified by a factor of ω , which could potentially make $\epsilon \omega$ comparable with, or even large than, F_{true} itself. Neural networks, and in particular networks with insufficient regularisation, can exhibit such high-frequency oscillations in their learned representations despite having a highly accurate energy prediction scheme at face value.

The backpropagation process through deep architectures introduces additional complications as, for a GNN with L message passing layers, the force gradient computations require solving via traversal through a network that is not designed to be simply read as a left-to-right set of dense connections, instead being comprised of multiple sub networks with nonlinear data traversal. At each of these layers, the gradient must propagate through the layers transformation, which involves multiplication of a Jacobian matrix corresponding to that layers operation. For deep networks, we are required to compute a product of many Jacobian matrices from a gradient signal that does not follow a standard dataflow path. Each product of Jacobian matrices can introduce numerical instabilities, especially when activation functions are employed as they can saturate regions with vanishingly small gradients [96].

Training exclusively on energy targets can lead to models that can accurately reproduce a global energy function but is open to fail entirely at capturing the local geometric structure of the potential energy surface. This can manifest in several ways, mainly that the model could learn correct relative energies but incorrect surface curvatures, leading to forces with systematically biased magnitudes. These considerations motivate our development of interpolation-based force calculation methods.

We aim to develop a method that can systematically reduce errors accumulated in this

scheme through utilisation of the accurate energy predictor itself with the goal of optimising force calculations derived from machine-learned potential energy surfaces.

13.1 Interpolation Methods

To develop alternative force calculation methodologies, we must first establish a rigorous mathematical framework for the Potential Energy Surface (PES) itself and understand how interpolation-based approaches can address the limitations of direct backpropagation. The PES represents a hypersurface $\mathcal{S} \in \mathbb{R}^{3N}$ embedded in coordinate space, as for a system of N atoms, the configuration space has dimension $3N$, leading to what is commonly termed the “curse of dimensionality” when attempting to sample or represent this surface computationally.

The potential energy surface possesses specific mathematical properties that inform our methodological choices. Away from degeneracies, the energy function $E_{\text{PES}} : \mathbb{R}^{3N} \rightarrow \mathbb{R}$ is continuous everywhere except at the nuclear coalescence points, which are unphysical for molecular systems. This smoothness justifies gradient-based force calculations and, within most regions of chemical interest, allows for a representation based on convergent Taylor series expansions around equilibrium configurations. We can express the energy in the vicinity of some reference configuration $\mathbf{r}^0 = (\mathbf{r}_1^0, \mathbf{r}_2^0, \dots, \mathbf{r}_n^0)$ as,

$$E(\mathbf{r}) = E(\mathbf{r}_0) + \sum_i^N \nabla_i E|_{\mathbf{r}_0} \cdot (\mathbf{r}_i - \mathbf{r}_i^0) + \frac{1}{2} \sum_{i,j}^N (\mathbf{r}_i - \mathbf{r}_i^0)^T \mathbf{H}_{ij}|_{\mathbf{r}_0} (\mathbf{r}_j - \mathbf{r}_j^0) + \mathcal{O}(|\mathbf{r} - \mathbf{r}_0|^3), \quad (13.5)$$

where $\mathbf{H}_{ij} = \frac{\partial^2 E}{\partial \mathbf{r}_i \partial \mathbf{r}_j}$ represents the Hessian matrix components. This expansion reveals that local force information is encoded in the first-order terms, while the Hessian captures force field variations.

Different chemical interactions impose distinct length scales and patterns within the PES, which can lead to a multiscale problem where interpolation schemes must simultaneously capture steep local variations near bonds and smooth long-range behaviours in non-bonded regions. For example, bonded interactions can exhibit steep potential wells with large curvatures, whilst Van der Waals interactions exhibit shallow, long-range potentials with much

smaller curvatures. Electrostatic interactions display long-range $1/r$ decay in non-periodic systems, requiring specific treatment for convergence.

Given a neural network that provides highly accurate energy predictions but potentially unreliable force estimates via backpropagation, we reformulate the force calculation problem to one which involves constructing, and differentiating, an interpolated energy surface. The approach consists of three stages.

- Dense sampling to generate a large number of energy evaluations.
- Interpolate a smooth energy surface from this dense array of sampled points, resulting in a smooth, differentiable function which approximates these discrete, but dense, samples.
- Compute force by means of numerical differentiation or gradient analysis of the interpolated energy surface.

The motivation for this method comes from the separation of concerns that are conflated during the training process of the neural network. The network optimises energy prediction accuracy with no concern for the explicit smoothness constraints. By generating dense samples and applying interpolation with explicit smoothness constraints, we decouple function approximation from derivative computation. If the energy predictions contain high-frequency noise, which could be prone to amplification by direct differentiation, smooth interpolation will act as a low-pass filter. This would suppress these components before force calculation. Furthermore, many interpolation methods provide us with mathematical guarantees about derivative continuity, which we can utilise and benefit from, as direct backpropagation through neural networks cannot ensure continuous derivatives.

For training a dataset with N configurations, we generate an expanded dataset through local perturbations of the given configurations. For each reference configuration within the training set, we sample variations of the type,

$$\mathbf{r}_i^k = \mathbf{r}_0^k + \delta_i, \quad i = 1, \dots, N_{\text{var}}, \quad (13.6)$$

where $\mathbf{r}_0^k$ is the reference configuration within the training set, and perturbations $\delta_i \sim$

$\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ are sampled from a Gaussian distribution. We must take great care in the choice of σ . If it is too large, we will sample in regions where the neural network may not be guaranteed to be accurate due to configurations being potentially unrealistic. If we make it too small, we will fail to adequately sample the local energy surface geometry, leaving large areas of the PES sparsely populated. The number of variations created and evaluated was of the order of one million for the overall resulting PES evaluation grid of dense energy points.

We now need to decide on an interpolation scheme that will allow for all the desired attributes to be utilised in our force calculation scheme, without resulting in a methodology that is computationally intractable. We first look to the K -nearest neighbours interpolation scheme.

13.2 K-Nearest Neighbours for Local Sample Selection

The *K-Nearest Neighbours* (KNN) algorithm serves as a neighbour selection mechanism to identify relevant samples from the densely generated mesh for subsequent interpolation and force calculation. Given a query point $\mathbf{r}_q \in \mathbb{R}^{3N}$ at which we want to compute the forces, the KNN method will identify the K nearest points from the precalculated densely sampled configuration in accordance with a simple Euclidean distance metric $d(\mathbf{r}_q, \mathbf{r}_i) = \sum_{j=1}^N \sqrt{|\mathbf{r}_{qj} - \mathbf{r}_{ij}|^2}$. These K -nearest neighbours $\{\mathbf{r}_{n_1}, \dots, \mathbf{r}_{n_k}\}$ define the local region around $\mathbf{r}_q$, and their corresponding neural network predicted energies $E_\theta(\mathbf{r}_{n_1}, \dots, \mathbf{r}_{n_k})$ form the basis for constructing a local energy surface approximation.

The critical aspect of this implementation is that once the K neighbours are selected, a single interpolation surface is constructed from these samples and subsequently used for all energy evaluations that are needed for the finite difference calculations. Specifically, an interpolation function $f_K(\mathbf{r})$ is built from the K selected neighbours. Then, forces can be computed with the following,

$$\mathbf{F}_i = \frac{f_K(\mathbf{r}_q + h\mathbf{e}_i) - f_K(\mathbf{r}_q - h\mathbf{e}_i)}{2h}, \quad (13.7)$$

where $\mathbf{e}_i$ is a unit vector in the i -th coordinate direction, h is the step size for the finite

difference scheme, and the same function f_K is evaluated at both perturbed points. This approach eliminates concerns about derivative discontinuities arising from neighbourhood changes during finite differencing. This is due to the fact that the neighbourhood is fixed for all evaluations associated with a given query point. We are computing the finite differences of a single function rather than mixing different local approximations.

The central question then becomes one of choosing the optimal form for the interpolation function f_K built from the K selected neural network predictions. The simplest approach would be to construct f_K as a distance-weighted average,

$$f_K(\mathbf{r}) = \frac{\sum_{k=1}^K w_k(\mathbf{r}) E_\theta(\mathbf{r}_{n_k})}{\sum_{k=1}^K w_k(\mathbf{r})}, \quad (13.8)$$

where the weights $w_k(\mathbf{r})$ depend on the distance between $\mathbf{r}$ and the K -th selected neighbour. Common weighting schemes include the inverse distance weighting scheme, where

$$w_k(\mathbf{r}) = \frac{1}{|\mathbf{r} - \mathbf{r}_{n_k}| + \epsilon}, \quad (13.9)$$

with $\epsilon > 0$ being a small regularisation constant for stability. Another method is the Gaussian weighting,

$$w_k(\mathbf{r}) = e^{-\frac{|\mathbf{r} - \mathbf{r}_{n_k}|^2}{2\sigma^2}}. \quad (13.10)$$

Within the local region defined by the K neighbours, this interpolation function is smooth and continuously differentiable. The weights vary continuously as $\mathbf{r}$ moves around within the local domain, overall ensuring that the derivatives we compute via the finite differences scheme approximate the true gradient of f_K .

The derivative of the weighted average interpolation can be computed analytically. To illustrate, the inverse distances weighting within one-dimension is given as,

$$\frac{df_K}{dx} = \frac{\sum_k \frac{\partial w_k}{\partial x} E_k \cdot \sum_j w_j - \sum_k w_k E_k \cdot \sum_j \frac{\partial w_j}{\partial x}}{\sum_k w_k^2}, \quad (13.11)$$

where $\frac{\partial w_k}{\partial x} = -\frac{\text{sign}(x-x_k)}{(x-x_k)^2}$ for the inverse distance weights. When the query point $\mathbf{r}_q$ is positioned near one of the samples within K , that sample dominates the weighting. However,

between samples, the derivative may not accurately capture the true force field if the underlying PES has some form of structure, which cannot be represented by the weighted scheme alone.

The fundamental limitation is that weighted averaging only provides a zeroth-order interpolation. The interpolation is a convex combination of K energy values, constrained to lie within a set range. For K neighbours in d dimensions, the interpolation has K degrees of freedom. However, taking for example a quadratic surface which would require $1 + d + \frac{d(d+1)}{2}$ coefficients, this demands 325 parameters, which is drastically larger than the typical K values, which would normally be used (often in the range of between 5 and 20).

This representation limitation proves problematic when the true energy surface exhibits local curvature, which is expected to be the case. For systems where the energy surfaces exhibit significant local curvatures the weighted averaging method becomes inadequate. Despite the highly accurate energy models, atomic force calculations do not perform better than direct force calculation from backpropagation. We deduce that more sophisticated methods of fitting the PES are needed, and we move to describe, implement, test, and analyse the fitting of polynomial models instead, rather than simply computing weighted averages. In doing so we hope to construct interpolations f_K that can explicitly capture the local geometry of the PES and, in turn, provide more accurate derivatives for our force calculations.

13.3 Smooth Polynomial Interpolation Methods

We consider two polynomial-based approaches. Namely, local polynomial fitting and Gaussian weighted regression.

13.3.1 Local Polynomial Fitting

Local Polynomial Fitting (LPF) constructs the desired interpolation function f_K by fitting a polynomial model to the K selected neighbours using the method of regularised least squares,

$$f_K(\mathbf{r}) = \mathbf{p}(\mathbf{r})^\top \beta^*, \quad (13.12)$$

where,

$$\beta^* = \operatorname{argmin}_{\beta} \sum_{k=1}^K [E_{\theta}(\mathbf{r}_{n_k}) - \mathbf{p}(\mathbf{r}_{n_k})^{\top} \beta]^2 + \lambda |\beta|^2. \quad (13.13)$$

Here, E_{θ} is the neural network energy prediction at the K -th neighbour, and $\mathbf{p}(\mathbf{r}_{n_k})^{\top} \beta$ is the polynomials prediction at the same point. We fit the polynomial using the squared difference between the actual and the predicted energy at these points. A regularisation term $\lambda |\beta|^2$ is used to penalise large coefficients in an attempt to prevent any form of overfitting.

Then, forces are calculated analytically with,

$$\mathbf{F}(\mathbf{r}_q) = -\nabla[\mathbf{p}(\mathbf{r}_q)^{\top} \beta^*] = \left[\frac{\partial \mathbf{p}^{\top}}{\partial \mathbf{r}} \beta^* \right]_{\mathbf{r}=\mathbf{r}_q}, \quad (13.14)$$

which yields forces as linear functions of position within the defined neighbourhood, with no finite difference error beyond the polynomial fit itself.

The choice of polynomial degree can be treated as a hyperparameter and represents a balance between representational capacity and overall computational cost. Linear models for eight-atom systems and polynomial degree one require 25 coefficients, will produce constant forces. Quadratic models for the same system require 325 coefficients but now capture the local Hessian $\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{r}^2}$, enabling the representation $E(\mathbf{r}) \approx E_0 + \mathbf{b}^{\top} \mathbf{r} + \frac{\mathbf{r}^{\top} \mathbf{H} \mathbf{r}}{2}$ with linearly varying forces $\mathbf{F}(\mathbf{r}) \approx -\mathbf{b} - \mathbf{H} \mathbf{r}$. This provides the natural description near stable configurations where the energy surface should be approximately harmonic. Cubic models require nearly 3000 coefficients and risk overfitting to include small noises within the neural network and fall foul of the original issues taken with the forces calculated directly from backpropagation schemes, which we seek to avoid.

13.3.2 Gaussian Weighted Regression

Gaussian Weighted Regression (GWR) extends polynomial fitting by using all samples from the dense mesh with continuous Gaussian weighting,

$$f_{\text{GWR}}(\mathbf{r}) = \mathbf{p}(\mathbf{r})^{\top} \beta(\mathbf{r}_q), \quad (13.15)$$

$$\beta(\mathbf{r}_q) = \operatorname{argmin}_{\beta} \sum_i^M w(\mathbf{r}_q, \mathbf{r}_i) [E_{\theta}(\mathbf{r}_i) - \mathbf{p}(\mathbf{r}_i)^{\top} \beta]^2, \quad (13.16)$$

$$w(\mathbf{r}_q, \mathbf{r}_i) = e^{-\frac{|\mathbf{r}_q - \mathbf{r}_i|^2}{2\sigma^2}}. \quad (13.17)$$

The forces can then be given as,

$$\mathbf{F}(\mathbf{r}_q) = -\nabla_{\mathbf{r}_q} [\mathbf{p}(\mathbf{r}_q)^T \beta(\mathbf{r}_q)]. \quad (13.18)$$

Unlike the LPF's discrete K-neighbour selection, GWR's continuous weighting ensures $\beta(\mathbf{r}_q)$ varies smoothly. However the GWR is significantly more computationally expensive.

13.4 Finite Difference Calculations

Forces are computed from the interpolated surface f_K via finite differences. The central difference formula is,

$$\mathbf{F}_i \approx \frac{\mathbf{f}_K(\mathbf{r}_q + h\mathbf{e}_i) - \mathbf{f}_K(\mathbf{r}_q - h\mathbf{e}_i)}{2h}. \quad (13.19)$$

This provides second-order accuracy with truncation error of $\mathcal{O}(h^2)$. The Taylor expansion yields an error term of $\left(\frac{h^2}{6}\right) \left(\frac{\partial^3 \mathbf{f}_K}{\partial \mathbf{r}_i^3}\right)$, which vanishes for quadratic polynomials.

The overall computational cost scales as $\mathcal{O}(NK)$ for weighted averaging, and scales $\mathcal{O}(N^3)$ for polynomial evaluation. LPF fitting has a computational cost of $\mathcal{O}(Kd^2)$, and GWR has a cost of $\mathcal{O}(Md^2)$.

Chapter 14

Model Architecture - Ni-Pd

The echo state network architecture for Ni-Pd alloy energy predictions employs carefully chosen dimensional hyperparameters, which balance the representational capacity of the network against the overall computational efficiency. The atomic embedding layer maps discrete atomic numbers to continuous feature vectors with dimension 64.

The Radial Basis Function (RBF) encoding employs 50 Gaussian basis functions uniformly distributed across the $[0, r_{\text{cut}}]$, with $r_{\text{cut}} = 5\text{\AA}$. This routinely captures the first and second nearest-neighbour interactions within FCC structures of our scale, whilst limiting edge counts for computational feasibility. The 50 RBF components provide a high resolution in order to distinguish distance variations influencing interatomic forces within the systems.

The message passing architecture comprises $L = 3$ sequential layers, each transforming atomic features with a fixed dimension of 32. This depth allows information to propagate across multiple coordination shells, whilst avoiding over-smoothing, a drawback seen in overly deep networks [97].

The echo state reservoir within each message passing layer employs four critical hyperparameters, which govern its overall dynamical properties. The reservoir dimension is 256, creating a high-dimensional feature space, enabling diverse nonlinear transformations. The message input dimension is the concatenation of the source atom features (namely, the one-hot encoding dimension), and RBF edge features, whilst projection back to message space reduces the overall dimensionality back down to 64 through the learned linear transformation.

The reservoir sparsity parameter controls the connectivity density within the weights of

the reservoir, and it is set to 10%. This sparse connectivity reduces the computational cost, whilst maintaining sufficient pathways for complex dynamics.

The spectral radius value of 0.9 position reservoir dynamics near the edge of stability, whilst maintaining the echo-state property. Following random initialisation and sparsification, the reservoir weight matrix undergoes spectral radius scaling at this value in order to enable rich transient dynamics, whilst ensuring autonomous reservoir dynamics converge toward the origin.

The leak rate of 0.3 is chosen to control the temporal integration throughout the network. This is a moderate rate, and balances state persistence against input responsiveness, allowing the reservoir to integrate information across several message passing iterations, but not to the extent that it is overly reliant on them.

The global pooling network transforms per-atom features to scalar energy predictions via a three-layer architecture with progressive dimensional reduction [64, 32, 16, 1]. The SiLu activation throughout provides smooth nonlinearity with unbounded positive range [98]. There is an absence of activation function within the final layer. This is done to ensure smooth energy prediction facilitating gradient computation for forces.

Model training employs the Adam optimiser with an initial learning rate $\eta_0 = 10^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ [99]. The optimiser operates on a composite loss function balancing energy and force prediction accuracy,

$$L_{\text{total}} = L_{\text{energy}} + \lambda_{\text{force}} \cdot L_{\text{force}}. \quad (14.1)$$

The energy loss for a batch of B structures computes,

$$L_{\text{energy}} = \frac{1}{B} \sum_{k=1}^B (E_{\theta}^k - E_{\text{ref}}^k)^2, \quad (14.2)$$

averaging the squared prediction errors across all structures as expected. The force loss averages overall atoms and Cartesian components.

$$L_{\text{force}} = \frac{1}{BN_{\text{total}}} \sum_{k=1}^B \sum_{i=1}^{N_k} \|\mathbf{F}_{\theta,i}^k - \mathbf{F}_{\text{ref},i}^k\|_2^2, \quad (14.3)$$

where N_k denotes the atoms in structure k , N_{total} is the total atom count and we compute the squared Euclidean norm summing over x , y , and z components.

The force loss weight $\lambda_{\text{force}} = 100$ is critical for ensuring force errors contribute meaningfully to the total loss despite typical force magnitudes being substantially smaller than energy differences. Without this weighting, the optimiser would predominantly minimise energy errors, whilst tolerating large force discrepancies, as gradient contributions from unweighted force losses would be negligible.

One of the most important features of this architecture is the fact that the training can be completed in only 20 epochs. This gives us a drastic time save compared to previous models in this thesis, especially considering the relative lower parameter count compared to previous traditional message passing architectures. With this in mind, we move to show the results of our ESN infused message passing scheme on our Ni-Pd dataset for energy and force prediction.

14.1 Ni-Pd Dataset

The Ni-Pd dataset used for training and evaluation was generated using density functional theory (DFT) calculations as detailed in Ref. [100]. The dataset comprises over one thousand training structures designed to capture the potential energy surface of the Ni-Pd binary alloy system.

It consists of over a thousand structures spanning diverse atomic configurations and strain states. The bulk comprises 731 structures derived from 43 unique Body Centered Tetragonal (BCT) supercells, each containing eight atoms. Each configuration was DFT-relaxed, with additional out-of-equilibrium structures generated by applying random atomic position perturbations. Additionally, 328 structures of Pd and Ni primitive cells are included, each containing two atoms.

All calculations used VASP with spin-polarised PBE functional and PAW potentials [101], and the cutoff energy was 439 eV.

Chapter 15

Results - Ni-Pd

Having established the ESN-based message passing architectures and polynomial interpolation framework, we now present results for the Ni-Pd binary alloy system.

The energy model results demonstrate a substantial level of predictive performance. Table 15.1 presents the metrics for the ESN-based architecture trained on Ni-Pd configurations.

Ni-Pd Energy Prediction Results

Model	MSE	MAE	R^2
ESN Message Passing	0.27160	0.40180	0.99874

Table 15.1: Energy prediction metrics for the ESN-based model on the Ni-Pd binary alloy dataset. MSE and MAE values reported in eV^2 and eV respectively for total system energies.

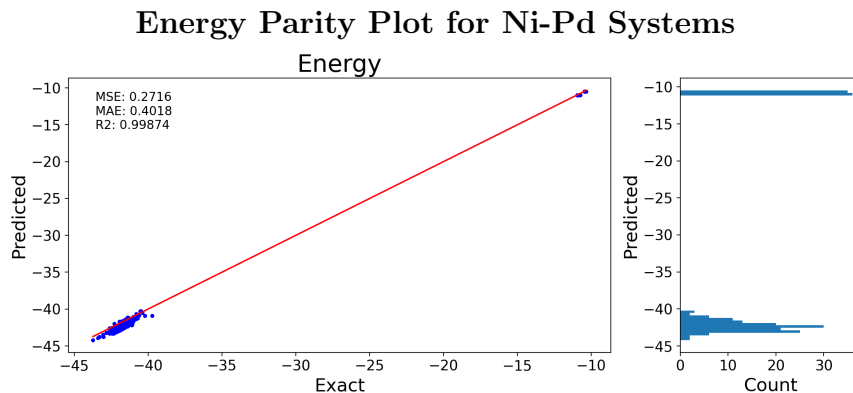


Figure 15.1: Energy parity plot for the Ni-Pd ESN model. The $R^2 = 0.99874$ demonstrates the model's capacity to accurately represent the potential energy surface. The tight clustering along the parity line and narrow histogram distribution indicate consistent prediction quality across the full energy range.

Whilst the comprehensive model shown in Table 15.1 demonstrates robust energy predic-

Specialized Model Energy Prediction Results

Model	MSE	MAE	R^2
2-atom Ni	0.00005	0.00544	0.98565
2-atom Pd	0.00003	0.00470	0.98127
8-atom Ni-Pd	0.00339	0.04465	0.99021

Table 15.2: Energy prediction metrics for the three ESN models trained on 2-atom Ni cells, 2-atom Pd cells, and mixed Ni-Pd 8-atom configurations. MSE and MAE values reported in eV^2 and eV respectively for total system energies.

tion capabilities, we trained three specialised ESN models for force field analysis specifically for each cluster of energies within the dataset, pertaining to the Ni 2-atom cells, Pd 2-atom cells and the mixed Ni-Pd 8-atom supercells. This approach allows each network to focus on the specific electronic structure and bonding characteristics for their respective systems. Each specialised model achieves $r^2 > 0.98$, validating the ESN architectures applicability across diverse metallic system configurations. All subsequent force prediction results are derived from these three specialised models. The results of these models are shown in Fig. 15.2 and Table 15.2.

All three models achieved their respective performance levels after only 20 training epochs. We believe this is due to the ESN architecture’s reduced parameter count. The fixed reservoir dynamics eliminates the need to train the recurrent weights within message passing operations, concentrating optimisation on the projection matrices, update MLPs, and final readout layer. This substantially reduces training time compared to fully trained message passing architectures whilst maintaining representation capacity through the high-dimensional reservoir dynamics.

The binary Ni-Pd system maintains structural coherence through the continuous FCC solid-state solution, with both species adopting similar coordination environments. This geometric uniformity, combined with the moderate compositional complexity, creates a tractable learning problem for the ESN architecture. The energy predictions provide a foundation for the force calculation methodologies detailed in Section 13, wherein polynomial interpolation schemes require accurate energy evaluations as input.

We now move on to examine the force calculations obtained through three differing methodologies using the three specialised models shown in Figure 15.2 and Table 15.2.

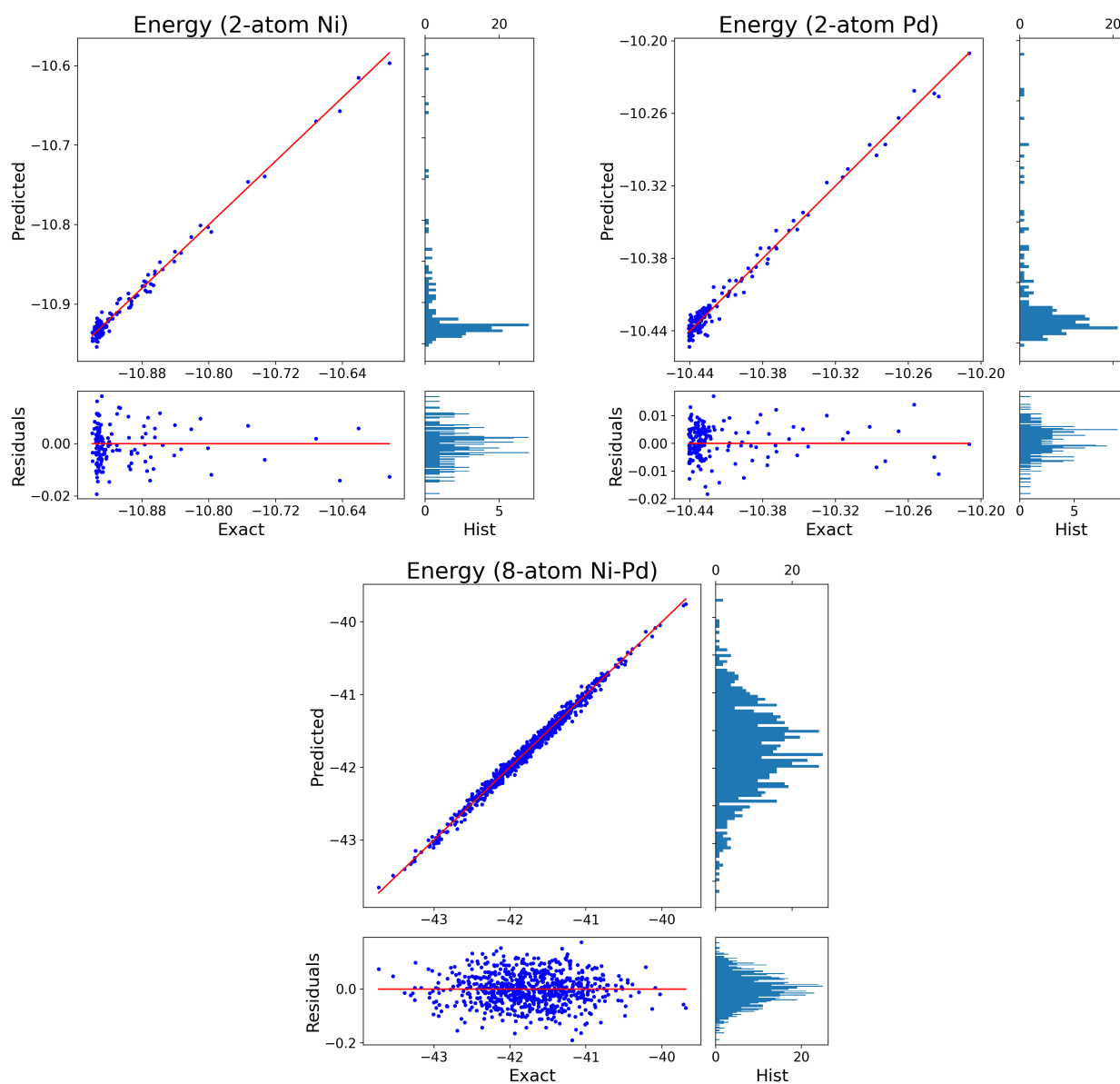


Figure 15.2: Energy parity plots for three separate ESN models trained on different system configurations. They are trained on the three subsections of the energies within the dataset, split into 2-atom Ni cells, 2-atom Pd cells, and the 8-atom Ni-Pd supercells. All three models demonstrate excellent predictive performance, with the ESN architecture successfully capturing the potential energy surfaces across pure elemental systems and mixed binary compositions. The MSE, MAE and R^2 -scores are reported in Table 15.2.

Firstly, the standard backpropagation through the energy models, then through Local Polynomial Fitting (LPF), and finally with Gaussian Weighted Regression (GWR). The latter two approaches, detailed in Section 13.3.2, operate on densely sampled energy surfaces with K -nearest neighbour selection providing local configurations for interpolation construction. Backpropagation serves as the conventional baseline, computing forces directly as analytical gradients of the learned energy function with respect to atomic positions.

Force Prediction Results - Combined Cartesian Components

Method	MSE	MAE	R^2
Backpropagation	0.15768	0.27838	0.80987
LPF	0.11736	0.24552	0.85849
GWR	0.31408	0.36215	0.62128

Table 15.3: Combined force prediction metrics across all atoms and Cartesian components for backpropagation, LPF, and GWR methods. MSE values in $(\text{eV}/\text{\AA})^2$, MAE in $\text{eV}/\text{\AA}$. Results aggregate predictions from all three specialised models from Fig. 15.2.

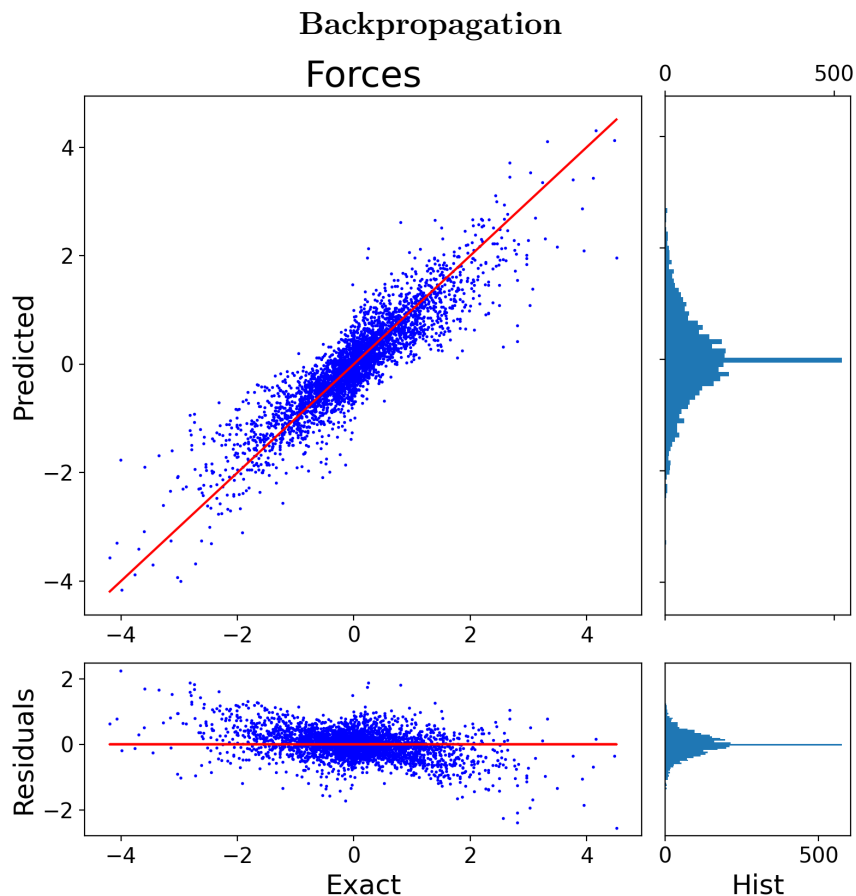


Figure 15.3: Combined force component parity plot for backpropagation showing all Cartesian force components. The R^2 -score of 0.80987 establishes the baseline performance of direct gradient calculation from the energy model. Backpropagation computes forces as analytical gradients of the learned energy function with respect to atomic positions, representing the conventional approach in neural network potential implementations. The moderate scatter and residual distribution reveal limitations in extracting derivative information directly from the energy model, as gradient calculations may amplify high-frequency oscillations present in the learned energy function that contribute negligibly to energy prediction errors, but substantially impact force accuracy. The consistent performance gap between force calculations and the energy model itself suggests an inherent noise amplification when transitioning from energy values to gradients. The performance across all Cartesian components indicates that whilst backpropagation provides a computationally efficient baseline, the method’s inability to filter or smooth these artifacts imposes fundamental accuracy limitations for force field predictions in complex metallic systems. This has an MSE of 0.15768eV^2 , and MAE of 0.27838eV .

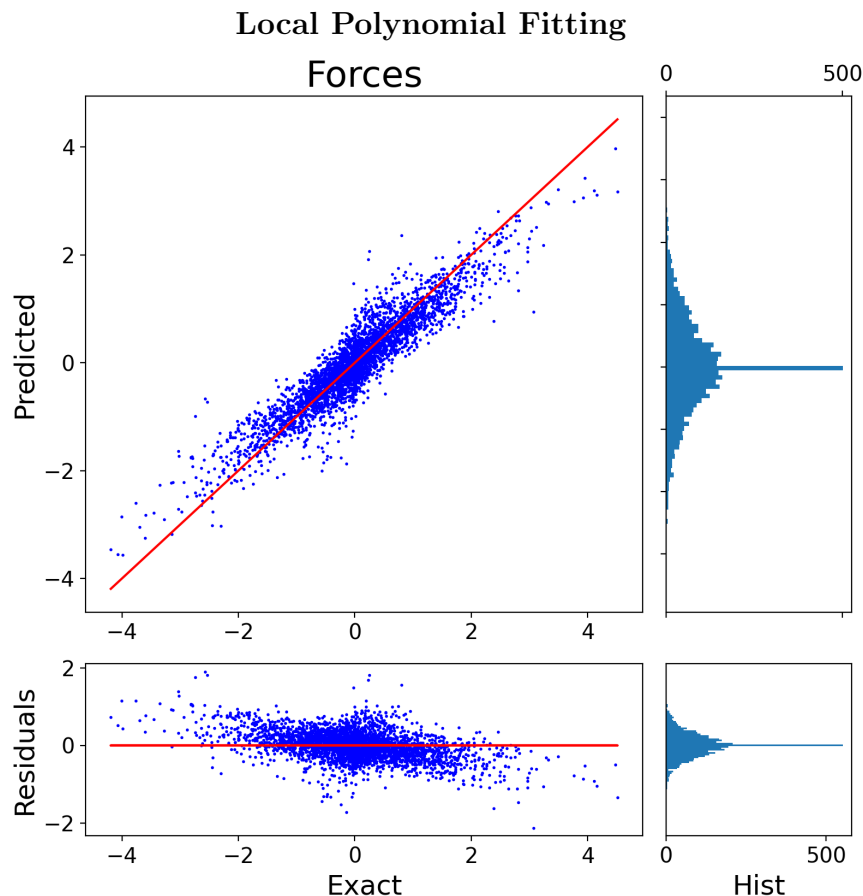


Figure 15.4: Combined force component parity plot for Local Polynomial Fitting. The improved $R^2 = 0.85849$ demonstrates superior predictive capability compared to backpropagation, with tighter clustering along the parity line. The LPF's explicit construction of local quadratic approximations to the densely sampled energy surface provides implicit smoothing that filters high-frequency noise components whilst preserving essential gradient information. The method captures energy surface curvature through polynomial fitting to K -nearest neighbour configurations, enabling more accurate force predictions than direct backpropagation whilst maintaining computational tractability. The slightly reduced scatter indicates that separating function approximation from derivative computation through interpolation successfully mitigates noise amplification to some degree. The slight improvement in performance relative to backpropagation across all components suggests advantages in how local polynomial fits capture the energy surface curvature, with the approach providing a degree of implicit smoothing that filters high-frequency components whilst maintaining the essential gradient structure required for accurate force field representation. From a computational perspective, LPF is substantially more efficient compared to more complex methods, such as GWR, whilst providing greater accuracy. This establishes the method as a potentially viable alternative which warrants more research in future, and could achieve predictive performance improvement with greater computational resources. The MSE is 0.11736eV^2 , and the MAE is 0.24552eV .

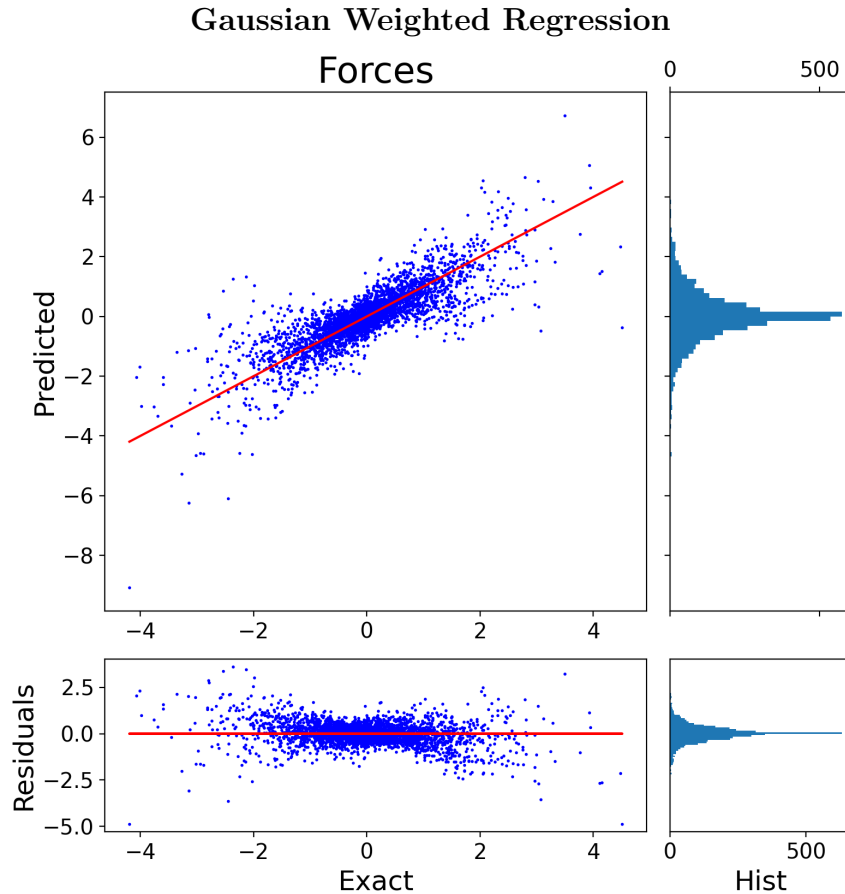


Figure 15.5: Combined force component parity plot for Gaussian Weighted Regression. The degraded $R^2 = 0.62128$ highlights greater difficulty in force field representation and model predictability, with considerably increased scatter and wider residual distribution. Despite the theoretical appeal of continuous coefficient variation through spatial weighting, GWR’s smooth weighting function appears to over-average across dissimilar regions when the local energy landscape may contain sharp features of rapidly varying curvature. The method’s computational expense comes at seemingly no trade-off for accuracy. Furthermore, numerical instabilities in weighted least-squares fitting may introduce systematic errors that vary by components across diverse environments. These results indicate that continuous weighting paradigms may prove to be poorly suited to capturing the complex force field variations in metallic alloy systems, compared to traditional approach already explore and our own LPF technique. The MSE is 0.31408eV^2 , and the MAE is 0.36215eV .

The combined force predictions reveal a notable performance hierarchy across the three methodologies examined. LPF achieves the strongest accuracy with $R^2 = 0.85849$, representing a measurable improvement over the backpropagation baseline at $R^2 = 0.80987$, whilst the GWR approach substantially underperforms both alternatives at $R^2 = 0.62128$. The approximately 5% improvement that LPF provides over backpropagation demonstrates that

Component-wise Force Prediction Results				
Component	Method	MSE	MAE	R^2
F_x	Backprop	0.13916	0.25877	0.86392
	LPF	0.09001	0.21862	0.91198
	GWR	0.33126	0.37521	0.67606
F_y	Backprop	0.13810	0.26390	0.72237
	LPF	0.11450	0.23838	0.76981
	GWR	0.33126	0.37521	0.67606
F_z	Backprop	0.19338	0.30687	0.80022
	LPF	0.14756	0.27956	0.84755
	GWR	0.37743	0.39899	0.61008

Table 15.4: Component-wise force prediction metrics comparing backpropagation, LPF, and GWR methods across Cartesian directions.

the manner in which forces are extracted from the learned energy surface matters critically, not merely the quality of the energy model itself.

Component-wise analysis reveals systematic variations in prediction quality across spatial directions, as detailed in Table 15.4 and Figures 15.6 through 15.8.

15.1 X-Component Analysis

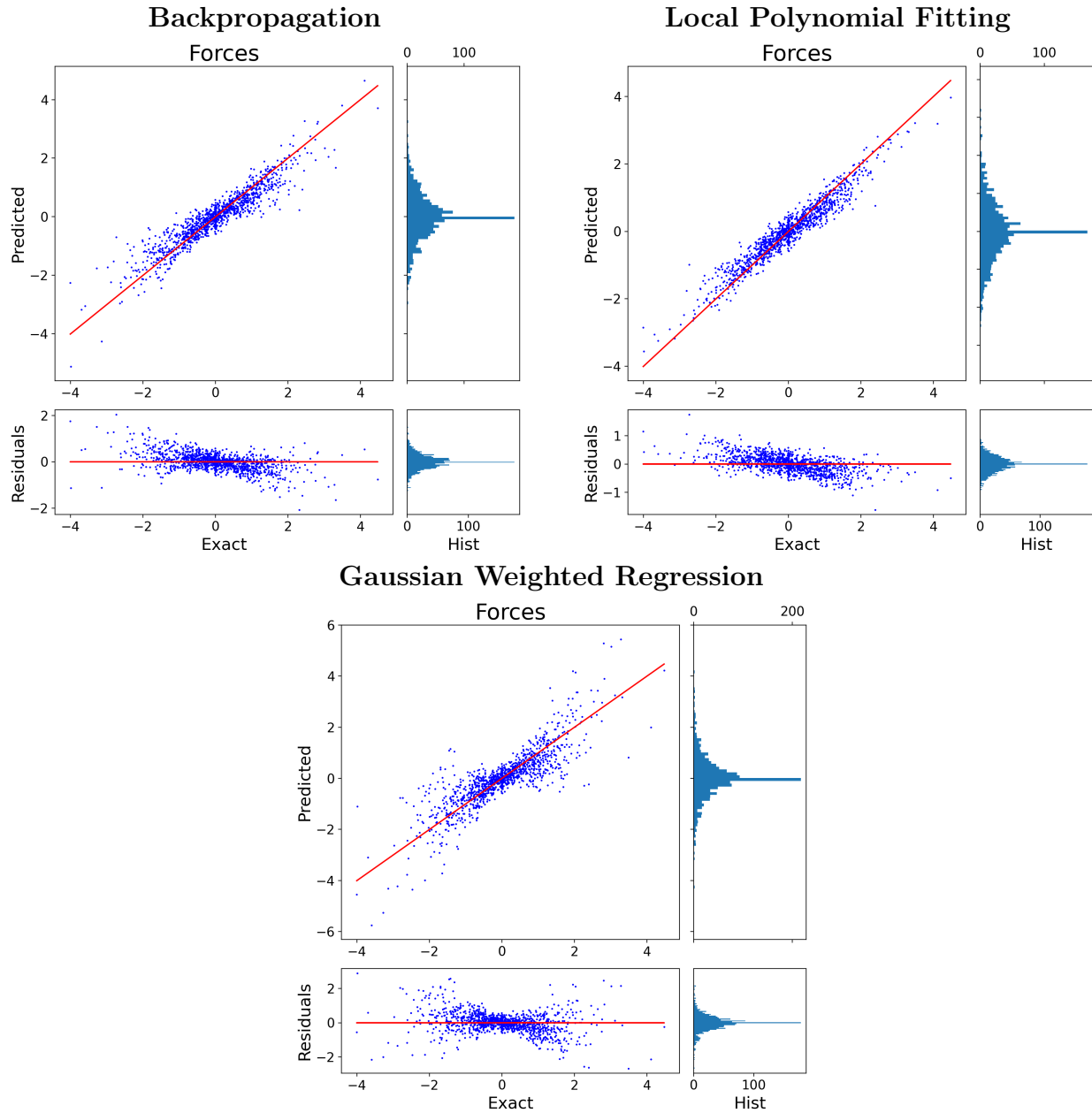


Figure 15.6: F_x component predictions across all three methods. LPF achieves the strongest performance ($R^2 = 0.91198$), substantially outperforming backpropagation ($R^2 = 0.86392$) whilst GWR shows considerable degradation ($R^2 = 0.67606$). The x-component exhibits the best overall prediction quality across all spatial directions, with LPF's local polynomial fitting approach successfully capturing the local energy surface curvature to some degree. The tight clustering along the parity line for LPF demonstrates that explicit quadratic approximations provide better derivative accuracy compared to direct gradient calculations through the energy model. MSE and MAE results are reported in Table 15.4.

15.2 Y-Component Analysis

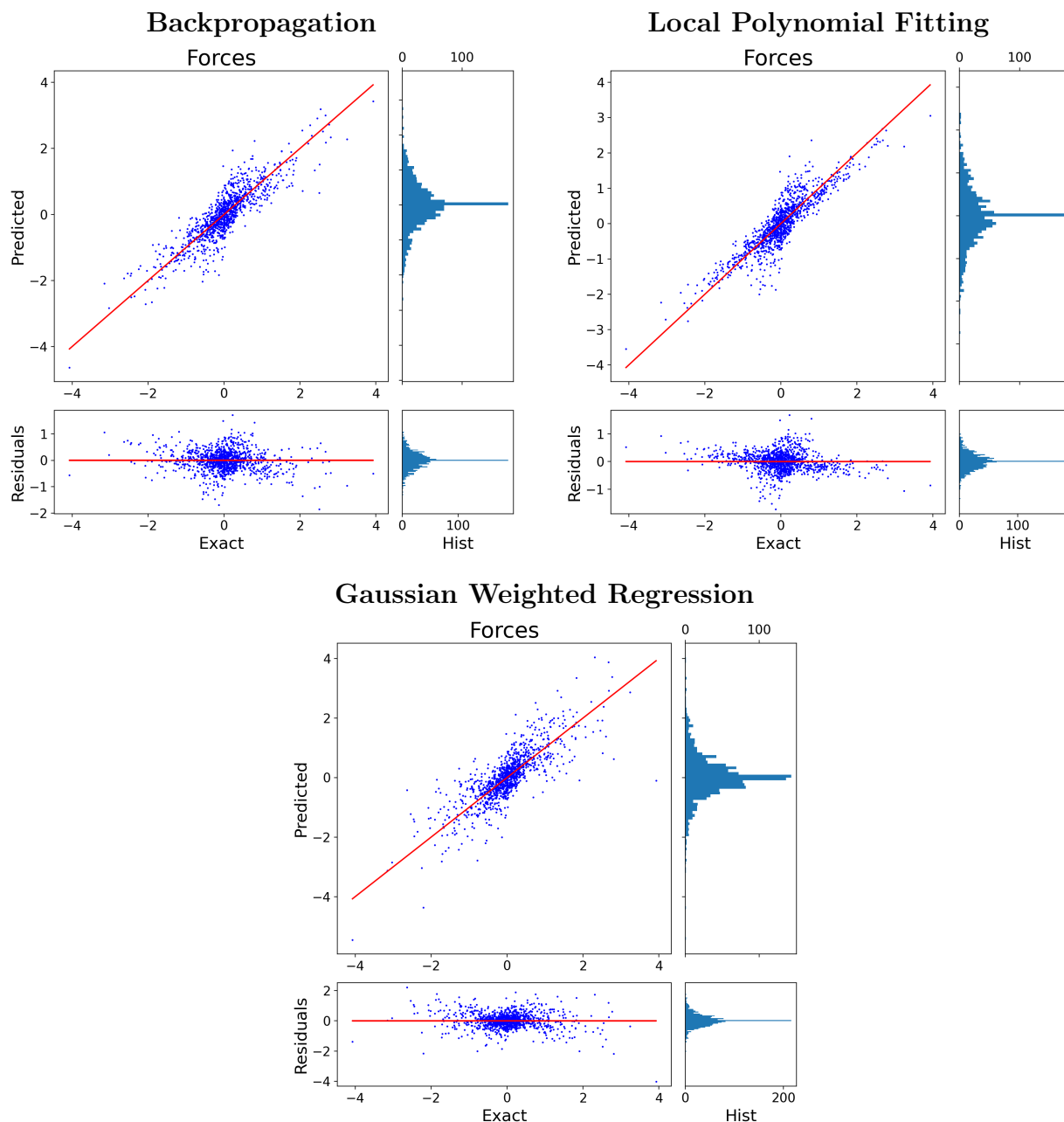


Figure 15.7: F_y component predictions showing degraded performance across all methods. The weakest component for both backpropagation ($R^2 = 0.72237$) and LPF ($R^2 = 0.76981$), suggesting directional anisotropy in the force field representation. This directional bias likely reflects artifacts within the training data distribution or density sampling procedure rather than crystallographic anisotropy, as the cubic FCC structure should exhibit isotropic behaviour. If Gaussian perturbations used to generate the dense energy mesh exhibited directional variance inhomogeneity, force components would sample the energy surface less densely along this axis, leading to degraded interpolation accuracy. MSE and MAE results are reported in Table 15.4.

15.3 Z-Component Analysis

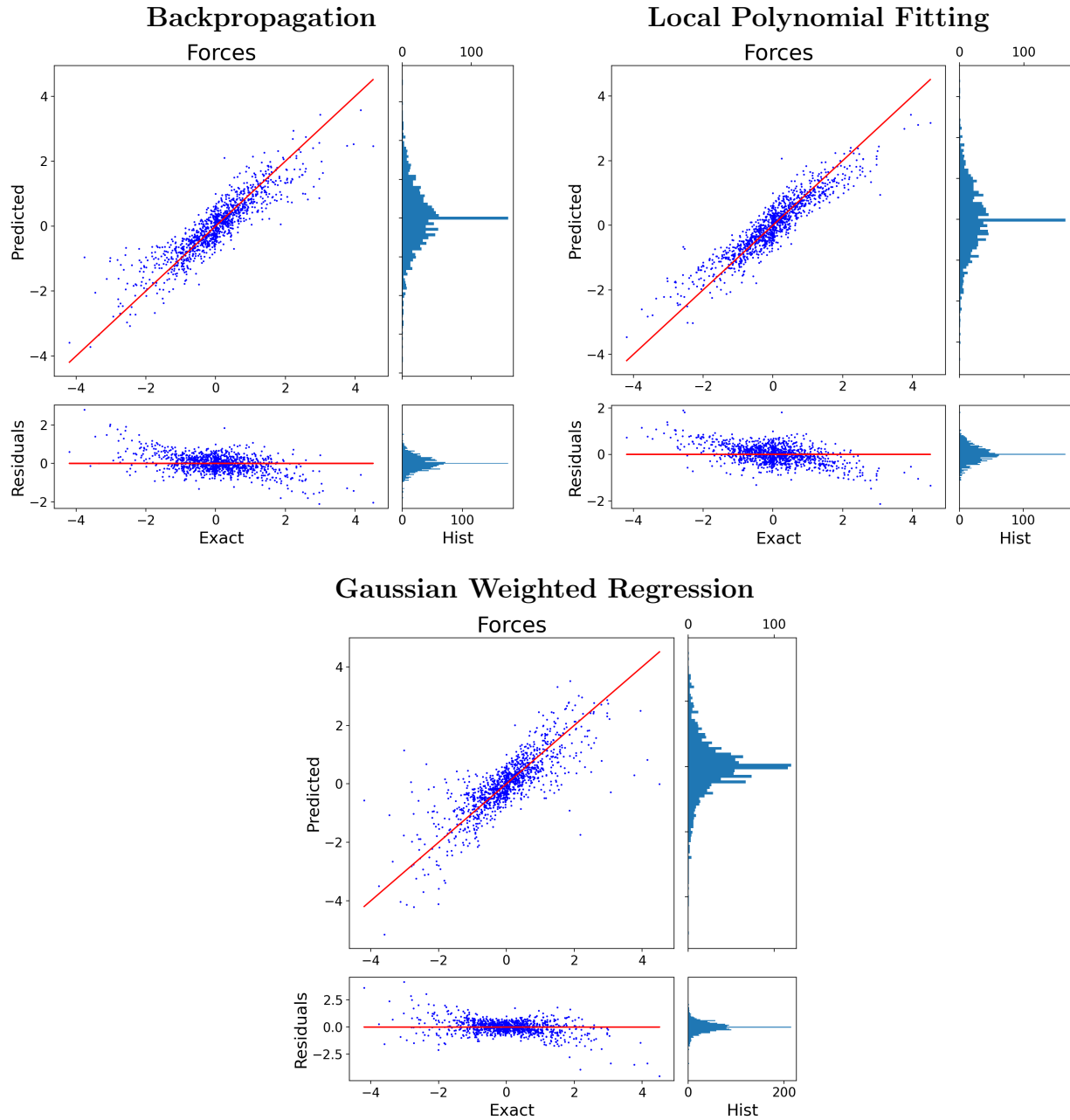


Figure 15.8: F_z component predictions. LPF maintains superior performance ($R^2 = 0.84755$) over backpropagation ($R^2 = 0.80022$), with GWR again showing substantial underperformance ($R^2 = 0.61008$). The z -component performance falls between the two other components in terms of R^2 -scoring. It maintains the pattern of LPF outperforming other techniques tested. We note that GWR performs particularly poorly in this direction, possibly indicating the presence of geometrically challenging features that are incompatible with continuous interpolation schemes based on smooth spatial weighting. MSE and MAE results are reported in Table 15.4.

The component-wise analysis reveals several systematic patterns. The directional anisotropy is overly pronounced in the y -component predictions, where all three methods exhibit degraded performance relative to the x and z components. This directional bias is unlikely to originate from crystallographic anisotropy given the cubic symmetry of the FCC structure. Instead, we hypothesise this likely reflects artifacts within the training data distribution or density sampling procedure. If Gaussian perturbations used to generate the dense energy mesh exhibited directional variance inhomogeneity, force components would sample the energy surface less densely along that axis, leading to degraded interpolation accuracy along that axis. This issue may be rectified with a recalculation of DFT forces within the training data, similar to our approach in 4.2.

These directional variations may also reflect species-dependent differences in local bonding environments. Palladium atoms, with filled d-shells, typically exhibit smoother potential energy surfaces compared to nickel’s partially filled $3d^8$ configuration, which introduces additional electronic structure complexity that manifests as higher-frequency variations within the force field. Such species-specific behaviour could couple with the directional sampling density to produce the observed anisotropy. Furthermore, the relative abundances of Ni versus Pd atoms within particular geometric configurations along different crystallographic directions could amplify these effects, if certain orientations preferentially sample regions dominated by one species.

The slight improvement in performance of LPF relative to backpropagation across all components suggests potential advantages in how local polynomial fits capture the energy surface curvature. Direct gradient calculation through backpropagation may be more susceptible to noise amplification, particularly in regions where the learned energy function exhibits high-frequency oscillations that contribute negligibly to energy prediction errors but substantially impact derivative accuracy. LPF’s explicit construction of local quadratic approximations could provide a degree of implicit smoothing that filters such high-frequency components whilst preserving the essential gradient information.

Conversely, GWR’s substantial underperformance across all components is surprising given the method’s theoretical appeal of continuous coefficient variation through spatial weighting. One potential explanation centres on the possibility that, if the local energy

landscape contains sharp features or rapidly varying curvature, GWR’s smooth weighting function may over-average across dissimilar regions, effectively destroying critical local information required for accurate force prediction. The method’s particularly poor performance in the z-component could indicate that this direction contains the most geometrically challenging features for continuous interpolation schemes. Additionally, numerical instabilities in the weighted least-squares fitting procedure could manifest differently depending on the local configuration geometry, potentially introducing systematic errors that vary by component.

From a computational perspective, the practical implications of these performance differences need to be highlighted. LPF maintains substantially faster evaluation times compared to GWR on standard CPU architectures, operating at speeds orders of magnitude quicker whilst simultaneously providing superior accuracy for our architectures and dataset. GWR’s computationally expensive continuous weighting calculation therefore appears to offer no accuracy advantage that would justify its adoption for this system. Whether LPF’s superior performance would persist for more complex systems with higher-dimensional composition spaces remains an open question requiring further investigation.

The consistent performance gap between all force calculation methods and the energy model itself suggests fundamental limitations in extracting derivative information from the learned potential energy surface. This could arise from insufficient energy surface sampling density, limitations of quadratic local approximations in regions of complex curvature, and inherent noise amplification when transitioning from energy values to gradients. Future work incorporating higher-order polynomial terms or adaptive sampling strategies targeting regions of rapidly varying forces could potentially narrow this performance gap, though care must be taken to avoid overfitting in the interpolation schemes.

Overall, the force prediction results establish LPF as a potential alternative to traditional backpropagation methods, and has the potential to become much more powerful with more research into architectural design, and computational resources available to test the method on. This initial set of results are comparable to those obtained on our silica dataset work in Part 2, and serves as a great foundation for further research into the possible advantages of our method compared to traditional force field predictive methods.

Conclusion

Chapter 16

Conclusion

This thesis presents a comprehensive investigation into the development of machine-learning force fields utilising density functional theory calculations, progressing from molecular systems through periodic crystalline structures to the foundations necessary for high-entropy alloy modelling. Our work establishes both theoretical frameworks, and practical implementations, that address the fundamental challenges in the application of graph neural networks to atomic systems, whilst identifying critical limitations that require detailed attention.

The initial phase of this research focused on the development of a powerful yet lightweight molecular force field model utilising an equivariant message passing architecture. Recognising the severe limitations inherent in the MD17 dataset, we developed custom dataset generation methodologies, which address the temporal correlation issues through strategic *K*-means clustering in internal coordinate space [102], whilst expanding the overall configuration coverage beyond equilibrium geometries through controlled Cartesian perturbations. Our EMPL architecture demonstrated strong predictive performance across ten molecular systems, with parity plots showing well-defined linear correlations between predicted and reference forces.

The identification and mathematical characterisation of node alignment failures represents the most significant theoretical contribution within this thesis. Through rigorous group-theoretic analysis, we demonstrated that molecules exhibiting high symmetry can generate degeneracies within distance-metric based representations that cannot be resolved architecturally. The development of our own obfuscation measures and ambiguity scores provided quantitative measures for the assessment of this representational challenge, and

furthermore gave us methodologies to utilise these limitations in the form of valuable pre-screening methods, allowing us to efficiently determine levels of dataset viability within molecular graph neural networks. Application to the MD17 dataset revealed significant correlations between the molecular symmetry present within a dataset and an overall training difficulty of graph based models, with highly symmetric molecules such as benzene requiring substantially more training data to achieve comparable predictive accuracy achieved by an architecturally identical model trained on less symmetric data instances. This analysis establishes fundamental limitations on efficiency gains promised by symmetry-aware representations and suggests that representational symmetry, rather than being universally advantageous, introduces training complexities that must be carefully considered in dataset design.

The extension to periodic systems introduced substantial additional challenges. The implementation of the minimum image convention, periodic boundary conditions, and padding mechanisms enabled our architecture to process unit cells of varying sizes and symmetries. Training on a silicon-based dataset, containing over one thousand structures with a large span of cell sizes across multiple space groups, revealed a fundamental difficulty in the development of universal force predictors for heterogeneous periodic systems using the previously defined in-house architectures. Training on the full dataset yielded R^2 -scores which were substantially lower than molecular system performance, reflecting the challenge of transferring the learning of patterns across the two orders of magnitude variation in cell sizes and diverse cell arrangements. Size-specific models displayed an improved performance, whilst training only on cubic cells yielded the strongest results. This indicated that dataset homogeneity significantly impacts model performance and suggests architectural limitations within simple equivariant message passing graph architectures for complex periodic systems, and architectural modifications would be needed in order to train on more complex datasets.

The integration of echo state networks and reservoir techniques into our message passing scheme addressed growing computational efficiency concerns whilst improving representational capacity. This was done via restricting the trainable parameters to projection matrices, update functions, and readout layers within the message passing scheme whilst maintaining fixed reservoir weights. The ESN-based approach achieved parameter count

reductions and significantly simplified the backpropagation dynamics, reducing the models overall training sensitivity and sensitivity to input hyperparameters. Implementation within the message passing framework leverages the structural parallels between the temporal aggregation within ESNs and the spatial aggregation within GNNs, with reservoir states processing concatenated node and edge information through fixed random projections. These networks were noted for their highly accurate predictive power relating to the total energy of periodic systems, and served as an effective test bed for the implementation of further force field modelling, analysis, and predictive models.

The development of energy surface interpolation methodologies attempts to address fundamental limitations in force calculations. The amplification of high-frequency noise components inherent to derivative operations can compromise force field reliability. Our interpolation-based approach separates function approximation from the derivative computation through dense sampling of configuration space, smooth surface fitting, and numerical differentiation. K -nearest neighbour selection provides local sample identification, whilst local polynomial fitting and Gaussian weighted regression enable explicit representation of energy surface curvature with guarantees of mathematical smoothness. This methodology trades computational cost for improved derivative continuity and noise suppression, offering an alternative to direct backpropagation when force accuracy is paramount.

16.1 One Year Projection

With one additional year, we would focus research efforts on two primary objectives that build upon the established foundations within this thesis whilst addressing critical knowledge gaps left in this work.

The first objective would involve a comprehensive investigation into how the node alignment failures manifest within periodic systems. Whilst we have established theoretical frameworks connecting space group symmetries to automorphism group cardinalities, empirical validation across diverse crystallographic systems remains incomplete. Development of automated tools for the calculation of the cardinality of automorphism groups for a given structure, even for rough estimates, would help to enable high-throughput analysis of struc-

tural databases and models for the comparison of periodic structures and training complexity, allowing us to evaluate the relationships between group symmetries and model trainability.

Secondly, we would target the expansion of the echo state network message passing framework towards high-entropy alloy applications. The current Ni-Pd binary system work provides a proof-of-concept, but extension to quinary systems requires addressing the configurational explosion and chemical heterogeneity challenges inherent to high-entropy alloys. The FCC crystal structure commonality across many high-entropy alloy systems suggests that lessons from cubic cell silica training can directly inform high-entropy alloy model development. Hyperparameter optimisation for ESN architectures would examine spectral radii, reservoir sizes, and leak rate effects on both computational efficient and predictive accuracy for metallic systems. Benchmarking against available DFT calculations for the Cantor alloy and derivatives thereof would establish baseline performance metrics and identify regions requiring enhanced sampling.

Both objectives would culminate in integrated analysis examining whether ESN-based architectures combined with sophisticated interpolation schemes can overcome the limitations encountered in heterogeneous periodic systems. Success would establish a validated pathway from binary Ni-Pd to ternary and quaternary systems, with clear metrics for the assessment of additional compositional complexity allowing us to track computational intractability.

16.2 Five Year Projection

If we had a longer timeframe for the continuation of this project, we believe we could enable advancement in theoretical foundations and practical applications, and be able to establish a comprehensive framework for machine-learning force field development in complex materials systems.

Theoretical advancement would prioritise the development of the complete mathematical foundations for successful echo state network message passing schemes, specifically in energy prediction in materials science. We would investigate why fixed random projections capture the relevant chemical information, examine connections to kernel methods and random feature approximations. Analysis would establish conditions under which ESN-based

approaches match, or supersede, fully trained traditional neural networks and potentially reveal fundamental limits on trainable parameter requirements. Information-theoretic measures which quantify how reservoir dynamics compress atomic configurational information would provide design principles for the design of optimised architectures.

High-entropy alloy force field development would progress through the systematic escalation in compositional complexity. We would initially focus on well-characterised systems and establish baseline accuracies against experimental measurements. Extension to high-order systems would leverage active learning workflows, identifying high-uncertainty regions and generating targeted DFT calculations to maximise information gain. Validation techniques would extend to molecular dynamics simulations, which would assess the finite temperature behaviour, thermal expansion and diffusion coefficients. Furthermore, we would aim towards the integration of our models with established molecular dynamics engines, such as LAMMPS [103], or ASE [104]. This would enable the seamless deployment in large-scale simulations.

This programme would establish machine-learning force fields as a practical tool for the acceleration of materials discovery in high-entropy systems. Success would be measured through tangible impact on experimental development cycles, reducing time from initial concept to realised samples. The theoretical insights regarding reservoir computing, symmetry effects, and force calculation methodologies would extend far beyond atomic systems and would go on to inform ML approaches within other domains facing similar high-dimensional, symmetry-constrained prediction challenges.

Ultimately, this work would contribute to enabling computational materials discovery at the scale required to address pressing technological challenges, with insights into optimisations and efficiencies available to us from mathematical insights within our choice of datasets and model architectures.

Appendix A

Publications

- E. Monaghan, U. Patil, and S. Sanvito, Neural Obfuscation in Distance-Based Graph Neural Networks and Predicting Predictability with Ambiguity Metrics. *In preparation* (2025)

Bibliography

- ¹ D. R. Hartree, “The wave mechanics of an atom with a non-coulomb central field. part i. theory and methods”, Mathematical Proceedings of the Cambridge Philosophical Society **24**, 89 (1928).
- ² V. Fock, “Näherungsmethode zur lösung des quantenmechanischen mehrkörperproblems”, Zeitschrift für Physik **61**, 126 (1930).
- ³ G. D. Purvis and R. J. Bartlett, “A full coupled-cluster singles and doubles model: the inclusion of disconnected triples”, The Journal of Chemical Physics **76**, 1910 (1982).
- ⁴ K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, “A fifth-order perturbation comparison of electron correlation theories”, Chemical Physics Letters **157**, 479 (1989).
- ⁵ P. Hohenberg and W. Kohn, “Inhomogeneous electron gas”, Physical Review **136**, B864 (1964).
- ⁶ W. Kohn and L. J. Sham, “Self-consistent equations including exchange and correlation effects”, Physical Review **140**, A1133 (1965).
- ⁷ D. R. Bowler and T. Miyazaki, “O(n) methods in electronic structure calculations”, Reports on Progress in Physics **75**, 036503 (2012).
- ⁸ D. M. Ceperley and B. J. Alder, “Ground state of the electron gas by a stochastic method”, Physical Review Letters **45**, 566 (1980).
- ⁹ J. P. Perdew, K. Burke, and M. Ernzerhof, “Generalized gradient approximation made simple”, Physical Review Letters **77**, 3865 (1996).

- ¹⁰ J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais, "Atoms, molecules, solids, and surfaces: applications of the generalized gradient approximation for exchange and correlation", *Physical Review B* **46**, 6671 (1992).
- ¹¹ J. P. Perdew and A. Zunger, "Self-interaction correction to density-functional approximations for many-electron systems", *Physical Review B* **23**, 5048 (1981).
- ¹² L. J. Sham and M. Schlüter, "Density-functional theory of the energy gap", *Physical Review Letters* **51**, 1888 (1983).
- ¹³ A. J. Cohen, P. Mori-Sánchez, and W. Yang, "Insights into current limitations of density functional theory", *Science* **321**, 792 (2008).
- ¹⁴ A. D. Becke, "Density-functional thermochemistry. iii. the role of exact exchange", *The Journal of Chemical Physics* **98**, 5648 (1993).
- ¹⁵ A. D. Becke, "A new mixing of hartree-fock and local density-functional theories", *The Journal of Chemical Physics* **98**, 1372 (1993).
- ¹⁶ A. D. Becke, "Density-functional exchange-energy approximation with correct asymptotic behavior", *Physical Review A* **38**, 3098 (1988).
- ¹⁷ C. Lee, W. Yang, and R. G. Parr, "Development of the colle-salvetti correlation-energy formula into a functional of the electron density", *Physical Review B* **37**, 785 (1988).
- ¹⁸ P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch, "Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields", *The Journal of Physical Chemistry* **98**, 11623 (1994).
- ¹⁹ S. F. Sousa, P. A. Fernandes, and M. J. Ramos, "General performance of density functionals", *The Journal of Physical Chemistry A* **111**, 10439 (2007).
- ²⁰ S. Kristyán and P. Pulay, "Can (semi)local density functional theory account for the london dispersion forces?", *Chemical Physics Letters* **229**, 175 (1994).
- ²¹ E. J. Meijer and M. Sprik, "A density-functional study of the intermolecular interactions of benzene", *The Journal of Chemical Physics* **105**, 8684 (1996).

- ²² D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors”, *Nature* **323**, 533 (1986).
- ²³ L. Bottou, “Large-scale machine learning with stochastic gradient descent”, in *Proceedings of compstat’2010* (2010), pp. 177–186.
- ²⁴ J. Behler and M. Parrinello, “Generalized neural-network representation of high-dimensional potential-energy surfaces”, *Physical Review Letters* **98**, 146401 (2007).
- ²⁵ K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “SchNet – a deep learning architecture for molecules and materials”, *The Journal of Chemical Physics* **148**, 241722 (2018).
- ²⁶ O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, “Spookynet: learning force fields with electronic degrees of freedom and nonlocal effects”, *Nature Communications* **12**, 7273 (2021).
- ²⁷ R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld, “Quantum chemistry structures and properties of 134 kilo molecules”, *Scientific Data* **1**, 1 (2014).
- ²⁸ L. Babai, “Graph isomorphism in quasipolynomial time”, in *Proceedings of the forty-eighth annual acm symposium on theory of computing* (2016), pp. 684–697.
- ²⁹ K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?”, in *International conference on learning representations* (2019).
- ³⁰ J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry”, in *Proceedings of the 34th international conference on machine learning*, Vol. 70, PMLR (2017), pp. 1263–1272.
- ³¹ T. Cohen and M. Welling, “Group equivariant convolutional networks”, in *International conference on machine learning* (PMLR, 2016), pp. 2990–2999.
- ³² V. G. Satorras, E. Hoogeboom, and M. Welling, “E(n) equivariant graph neural networks”, in *Proceedings of the 38th international conference on machine learning*, Vol. 139, PMLR (2021), pp. 9323–9332.

- ³³ S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, “Machine learning of accurate energy-conserving molecular force fields”, *Science Advances* **3**, e1603015 (2017).
- ³⁴ M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, “Fast and accurate modeling of molecular atomization energies with machine learning”, *Physical Review Letters* **108**, 058301 (2012).
- ³⁵ P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions”, *arXiv preprint arXiv:1710.05941* (2017).
- ³⁶ K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.
- ³⁷ S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift”, in *International conference on machine learning* (PMLR, 2015), pp. 448–456.
- ³⁸ J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization”, *arXiv preprint arXiv:1607.06450* (2016).
- ³⁹ V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, “Ab initio molecular simulations with numeric atom-centered orbitals”, *Computer Physics Communications* **180**, 2175 (2009).
- ⁴⁰ A. S. Christensen, L. A. Bratholm, F. A. Faber, and O. A. von Lilienfeld, “Fchl revisited: faster and more accurate quantum machine learning”, *The Journal of Chemical Physics* **152**, 044107 (2020).
- ⁴¹ Y. Leng, Y. Zhang, J. Qi, Y. Liu, J. Zhang, et al., “Comparing machine learning potentials for water: kernel-based regression and behler–parrinello neural networks”, *The Journal of Chemical Physics* **160**, 114107 (2024).
- ⁴² H. E. Sauceda, S. Chmiela, I. Poltavsky, K.-R. Müller, and A. Tkatchenko, “Molecular force fields with gradient-domain machine learning: construction and application to

- dynamics of small molecules with coupled cluster forces”, *The Journal of Chemical Physics* **150**, 114102 (2019).
- ⁴³ V. Zaverkin and J. Kastner, “Gaussian moments as physically inspired molecular descriptors for accurate and scalable machine learning potentials”, *Journal of Chemical Theory and Computation* **16**, 5410 (2020).
- ⁴⁴ Z. Pengmei, J. Liu, and Y. Shu, “Beyond MD17: the reactive xxMD dataset”, *Scientific Data* **11**, 222 (2024).
- ⁴⁵ Q. Sun et al., “Pyscf: the python-based simulations of chemistry framework”, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **8**, e1340 (2018).
- ⁴⁶ S. Grimme, “Semiempirical gga-type density functional constructed with a long-range dispersion correction”, *Journal of Computational Chemistry* **27**, 1787 (2006).
- ⁴⁷ Y. Zhao and D. G. Truhlar, “Density functionals with broad applicability in chemistry”, *Accounts of Chemical Research* **41**, 157 (2008).
- ⁴⁸ T. H. Dunning, “Gaussian basis sets for use in correlated molecular calculations. i. the atoms boron through neon and hydrogen”, *The Journal of Chemical Physics* **90**, 1007 (1989).
- ⁴⁹ D. E. Woon and T. H. Dunning, “Gaussian basis sets for use in correlated molecular calculations. iii. the atoms aluminum through argon”, *The Journal of Chemical Physics* **98**, 1358 (1993).
- ⁵⁰ S. H. Bertz, “The first general index of molecular complexity”, *Journal of the American Chemical Society* **103**, 3599 (1981).
- ⁵¹ J. B. Hendrickson, P. Huang, and A. G. Toczko, “Molecular complexity: a simplified formula adapted to individual atoms”, *Journal of Chemical Information and Computer Sciences* **27**, 63 (1987).
- ⁵² W.-D. Ihlenfeldt, J. H. Voigt, B. Bienfait, F. Oellien, and M. C. Nicklaus, “Enhanced cactvs browser of the open nci database”, *Journal of Chemical Information and Computer Sciences* **42**, 46 (2002).

- ⁵³ S. Kim et al., “Pubchem substance and compound databases”, *Nucleic Acids Research* **44**, D1202 (2016).
- ⁵⁴ K. T. Schütt, O. T. Unke, and M. Gastegger, “Equivariant message passing for the prediction of tensorial properties and molecular spectra”, in *Proceedings of the 38th international conference on machine learning*, Vol. 139, PMLR (2021), pp. 9377–9388.
- ⁵⁵ R. G. Parr and W. Yang, *Density-functional theory of atoms and molecules* (Oxford University Press, Oxford, UK, 1995).
- ⁵⁶ M. Born and T. von Kármán, “Über schwingungen im raumgittern”, *German, Physikalische Zeitschrift* **13**, 297 (1912).
- ⁵⁷ C. Kittel, *Introduction to solid state physics*, 8th (Wiley, New York, 2005).
- ⁵⁸ K. Huang, *Statistical mechanics*, 2nd (Wiley, New York, 1987).
- ⁵⁹ N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculations by fast computing machines”, *Journal of Chemical Physics* **21**, 1078 (1953).
- ⁶⁰ D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, 2nd (Academic Press, San Diego, 2002).
- ⁶¹ J. Behler, “Constructing high-dimensional neural network potentials: a tutorial review”, *International Journal of Quantum Chemistry* **115**, 1032 (2015).
- ⁶² L. Levien, C. T. Prewitt, and D. J. Weidner, “Structure and elastic properties of quartz at pressure”, *American Mineralogist* **65**, 920 (1980).
- ⁶³ K. Vollmayr, W. Kob, and K. Binder, “Cooling-rate effects in amorphous silica: a computer-simulation study”, *Physical Review B* **54**, 15808 (1996).
- ⁶⁴ J. Sanders and E. Kandrot, *Cuda by example: an introduction to general-purpose gpu programming* (Addison-Wesley, Upper Saddle River, NJ, 2010).
- ⁶⁵ I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* (MIT Press, Cambridge, MA, 2016).
- ⁶⁶ K. Zongo, L. K. Béland, and C. Ouellet-Plamondon, “First-principles database for fitting a machine-learning silicon interatomic force field”, *MRS Advances* **7**, 39 (2022).

- ⁶⁷ K. Zongo, H. Sun, C. Ouellet-Plamondon, and L. K. Béland, “A unified moment tensor potential for silicon, oxygen, and silica”, *npj Computational Materials* **10**, 218 (2024).
- ⁶⁸ P. Giannozzi et al., “Quantum espresso: a modular and open-source software project for quantum simulations of materials”, *Journal of Physics: Condensed Matter* **21**, 395502 (2009).
- ⁶⁹ G. Burns and A. M. Glazer, *Space groups for solid state scientists*, 3rd (Academic Press, Amsterdam, 2013).
- ⁷⁰ T. Hahn, *International tables for crystallography, brief teaching edition of volume a: space-group symmetry*, 5th (Springer, Dordrecht, 2005).
- ⁷¹ R. Zallen, *The physics of amorphous solids* (Wiley-VCH, Weinheim, 1998).
- ⁷² J.-W. Yeh, S.-K. Chen, S.-J. Lin, J.-Y. Gan, T.-S. Chin, T.-T. Shun, C.-H. Tsau, and S.-Y. Chang, “Nanostructured high-entropy alloys with multiple principal elements: novel alloy design concepts and outcomes”, *Advanced Engineering Materials* **6**, 299 (2004).
- ⁷³ B. Cantor, I. Chang, P. Knight, and A. Vincent, “Microstructural development in equiatomic multicomponent alloys”, *Materials Science and Engineering: A* **375**, 213 (2004).
- ⁷⁴ Y. Zhang, T. T. Zuo, Z. Tang, M. C. Gao, K. A. Dahmen, P. K. Liaw, and Z. P. Lu, “Microstructures and properties of high-entropy alloys”, *Progress in Materials Science* **61**, 1 (2014).
- ⁷⁵ D. B. Miracle and O. N. Senkov, “A critical review of high entropy alloys and related concepts”, *Acta Materialia* **122**, 448 (2017).
- ⁷⁶ M.-H. Tsai and J.-W. Yeh, “High-entropy alloys: a critical review”, *Materials Research Letters* **2**, 107 (2014).
- ⁷⁷ B. Gludovatz, A. Hohenwarter, D. Catoor, E. H. Chang, E. P. George, and R. O. Ritchie, “A fracture-resistant high-entropy alloy for cryogenic applications”, *Science* **345**, 1153 (2014).

- ⁷⁸ C. Lu, L. Niu, N. Chen, K. Jin, T. Yang, P. Xiu, Y. Zhang, F. Gao, H. Bei, S. Shi, et al., “Enhancing radiation tolerance by controlling defect mobility and migration pathways in multicomponent single-phase alloys”, *Nature Communications* **7**, 13564 (2016).
- ⁷⁹ E. Pickering and N. Jones, “High-entropy alloys: a critical assessment of their founding principles and future prospects”, *International Materials Reviews* **61**, 183 (2016).
- ⁸⁰ M. Dada, P. Popoola, and N. Mathe, “Recent advances of high entropy alloys for aerospace applications: a review”, *World Journal of Engineering* **20**, 43 (2023).
- ⁸¹ O. El-Atwani et al., “Outstanding radiation resistance of tungsten-based high-entropy alloys”, *Science Advances* **5**, eaav2002 (2019).
- ⁸² S. Gorsse, D. B. Miracle, and O. N. Senkov, “From high-entropy alloys to complex concentrated alloys”, *Comptes Rendus Physique* **19**, 721 (2018).
- ⁸³ F. Marques, M. Balcerzak, F. Winkelmann, G. Zepon, and M. Felderhoff, “Review and outlook on high-entropy alloys for hydrogen storage”, *Energy & Environmental Science* **14**, 5191 (2021).
- ⁸⁴ Y. Yao et al., “High-entropy alloys as a platform for catalysis: progress, challenges, and opportunities”, *ACS Catalysis* **10**, 11280 (2020).
- ⁸⁵ F. Körmann and M. H. Sluiter, “Interplay between lattice distortions, vibrations and phase stability in nbmotaw high entropy alloys”, *Entropy* **18**, 403 (2016).
- ⁸⁶ A. Ferrari, B. Dutta, K. Gubaev, Y. Ikeda, P. Srinivasan, B. Grabowski, and F. Körmann, “Frontiers in atomistic simulations of high entropy alloys”, *Journal of Applied Physics* **128**, 150901 (2020).
- ⁸⁷ H. Jaeger, *The “echo state” approach to analysing and training recurrent neural networks*, tech. rep. 148 (German National Research Center for Information Technology GMD Technical Report, 2001).
- ⁸⁸ W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: a new framework for neural computation based on perturbations”, *Neural Computation* **14**, 2531 (2002).

- ⁸⁹ C. Gallicchio and A. Micheli, “Deep reservoir computing: a critical experimental analysis”, *Neurocomputing* **268**, 87 (2017).
- ⁹⁰ M. Lukosečius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training”, *Computer Science Review* **3**, 127 (2009).
- ⁹¹ D. Verstraeten, B. Schrauwen, M. d’Haene, and D. Stroobandt, “An experimental unification of reservoir computing methods”, *Neural Networks* **20**, 391 (2007).
- ⁹² M. Lukoševičius, “A practical guide to applying echo state networks”, *Neural Networks: Tricks of the Trade*, 659 (2012).
- ⁹³ P. Subramanian and D. Laughlin, “The ni-pd (nickel-palladium) system”, *Journal of Phase Equilibria* **12**, 231 (1991).
- ⁹⁴ M. Kulichenko, J. S. Smith, B. Nebgen, Y. W. Li, N. Fedik, A. I. Boldyrev, N. Lubbers, K. Barros, and S. Tretiak, “The rise of neural networks for materials and chemical dynamics”, *Journal of Physical Chemistry Letters* **12**, 6227 (2021).
- ⁹⁵ O. T. Unke and M. Meuwly, “Physnet: a neural network for predicting energies, forces, dipole moments, and partial charges”, *Journal of Chemical Theory and Computation* **15**, 3678 (2019).
- ⁹⁶ S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient flow pathologies in physics-informed neural networks”, *SIAM Journal on Scientific Computing* **43**, A3055 (2021).
- ⁹⁷ Q. Li, Z. Han, and X.-M. Wu, “Deeper insights into graph convolutional networks for semi-supervised learning”, in *Thirty-second aaai conference on artificial intelligence* (2018).
- ⁹⁸ S. Elfving, E. Uchibe, and K. Doya, “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning”, *Neural Networks* **107**, 3 (2018).
- ⁹⁹ D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization”, *arXiv preprint arXiv:1412.6980* (2014).

-
- ¹⁰⁰ R. Hunkao, U. Patil, and S. Sanvito, “Machine-learning potential approach for phase stability and mechanical properties of disordered alloys at finite temperature”, In preparation, 2025.
- ¹⁰¹ G. Kresse and J. Furthmüller, “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set”, *Phys. Rev. B* **54**, 11169 (1996).
- ¹⁰² S. Lloyd, “Least squares quantization in pcm”, *IEEE Transactions on Information Theory* **28**, 129 (1982).
- ¹⁰³ S. Plimpton, “Fast parallel algorithms for short-range molecular dynamics”, *Journal of Computational Physics* **117**, 1 (1995).
- ¹⁰⁴ A. H. Larsen et al., “The atomic simulation environment—a Python library for working with atoms”, *Journal of Physics: Condensed Matter* **29**, 273002 (2017).