Physics-Informed Neural Approaches for Multiscale Molecular Modeling and Design

Thesis by
Zhuoran Qiao

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2023
Defended November 22, 2022

ACKNOWLEDGEMENTS

I would like to begin by thanking the entire Caltech community. When I decided to start pursuing my Ph.D., I never imagined the incredible journey of interdisciplinary research it would lead me to. I sincerely thank all staff and students at Caltech for working together to build a supportive and inclusive environment for international students. I feel fortunate to be part of this great community.

I want to express my thanks to my advisors, Prof. Thomas F. Miller III and Prof. Animashree Anandkumar. Thank you Tom for constantly supporting my research and striving to create an intimate, diverse, and cooperative team that will last for years. Thank you Anima for not only being a great mentor guiding me to confront many technical challenges, but also a leader encouraging me to develop my research taste and to see the bigger scientific picture. I want to also thank my committee members and the professors who helped me along the course of my Ph.D. study: thank you Prof. William Goddard, Prof. Lu Wei, and Prof. Zhen-Gang Wang.

My achievements would not have been possible without the team members, friends, and collaborators who have accompanied me during the past years. Special thanks to Matthew Welborn, Lixue Cheng, and Feizhi Ding for their kind guidance during the early days of my research. Thank you to members from the former Miller group: Tomislav, Emiliano, Tamara, Sebastian, Matthew, Jeongmin, Dan, Jorge, Xuecheng, Roman, Doris, Marta, James, Jiace, Vignesh, Rui and Bo; and to members from Anima Lab: Dani, Or, Rafal, Sahin, Zongyi, Jiawei, Hongkai, and Guanzhi. I thank great collaborators Anders and Krishna from Entos, and Weili, Arash, and Chaowei from Nvidia. Thank you, my friends - Frid Fu, Heyun Li, Hanchen Wang, Leo Wu, Wenjun Xie, and Lin Xu. I wish you all the best in your career and personal life.

Finally, I would like to thank my family. Thank you to my parents for being supportive of my pursuit to a doctoral degree and for reliving my family responsibilities. Thank you Yuanjin; you have turned the past four years into the happiest days of my life and made me always look forward to a better future. This dissertation is dedicated to you.

ABSTRACT

Chemical processes in nature span multiple characteristic length and time scales, and the computational simulation for systems at the intersection of different scales is highly challenging with far-reaching implications for numerous scientific and industrial problems. To facilitate the computational modeling and design for large molecular systems and address the cost-resolution tradeoffs in conventional strategies, in this dissertation we introduce a series of physics-informed machine learning methods for the efficient computational modeling of chemical systems and the accurate prediction of their properties such as energetics, structures, and dynamics. In Chapters 2-3, we introduce a family of orbital-based geometric deep learning methods for the prediction of quantum chemical properties while adhering to the scaling and symmetry constraints of electronic structure theory. The presented methods achieve a chemical accuracy on community-wide benchmarks for molecular property prediction, and are shown to be transferable among diverse main-group molecular systems. In Chapter 4, we introduce a method for the prediction of protein-ligand complex structures based on a finite-time stochastic process parameterized by deep equivariant neural networks. The presented method achieves improved structure prediction accuracy against existing approaches, and is able to rapidly sample protein structures for folding landscapes that are modulated by inter-molecular interactions.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Zhuoran Qiao, Weili Nie, Arash Vahdat, Thomas F. Miller III, and Animashree Anandkumar. “Dynamic-Backbone Protein-Ligand Structure Prediction with Multiscale Generative Diffusion Models”. In: *arXiv preprint arXiv:2209.15171* (2022). *In submission.* To appear at *Machine Learning in Structural Biology workshop at NeurIPS 2022* as a Contributed Talk. doi: 10.48550/ARXIV.2209.15171.
Z.Q. contributed to the conception of the project, implemented the algorithm, prepared and analyzed the data, and wrote the manuscript.

[2] Zhuoran Qiao, Anders S. Christensen, Matthew Welborn, Frederick R. Manby, Animashree Anandkumar, and Thomas F. Miller III. “Informing geometric deep learning with electronic interactions to accelerate quantum chemistry”. In: *Proceedings of the National Academy of Sciences* 119.31 (2022), e2205221119. doi: 10.1073/pnas.2205221119.
Z.Q. conceptualized the project, developed the theoretical results, implemented the algorithm, analyzed the data, and wrote the manuscript.

[3] Zhuoran Qiao, Feizhi Ding, Matthew Welborn, Peter J. Bygrave, Daniel G. A. Smith, Animashree Anandkumar, Frederick R. Manby, and Thomas F. Miller. “Multi-task learning for electronic structure to predict and explore molecular potential energy surfaces”. In: *arXiv preprint arXiv:2011.02680* (2020). Appeared at *Machine Learning for Molecules workshop at NeurIPS 2020* as a Contributed Talk. doi: 10.48550/ARXIV.2011.02680.
Z.Q. contributed to the conception of the project, implemented the algorithm, prepared the data, and participated in the writing of the manuscript.

[4] Zhuoran Qiao, Matthew Welborn, Animashree Anandkumar, Frederick R Manby, and Thomas F Miller III. “OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features”. In: *The Journal of Chemical Physics* 153.12 (2020), p. 124111. doi: 10.1063/5.0021955.
Z.Q. contributed to the conception of the project, implemented the algorithm, prepared the data, and participated in the writing of the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Number *Page*

Chapter 1

INTRODUCTION

The study of chemistry entails many complex phenomena spanning a hierarchy of dynamical scales ranging from the time evolution of electronic wavefunctions and chemical reactions to macroscopic phase transitions and biological homeostasis. Computational chemistry has traditionally been a discipline built upon physical principles to construct simulation algorithms and domain knowledge to empirically select or revise these simulation tools to understand chemical, material, and biology systems at varying scales. Owing to the recent advancements in data-driven deep learning, significant progress in molecular modeling has been made to solve problems that were considered inaccessible by conventional strategies. These successes encompass almost the entire hierarchy of dynamical scales in chemistry, ranging from the representation of correlated many-body quantum states [1, 2], the construction of more accurate density functionals [3], the accelerated simulation for anomalous phase transitions [4], and the accurate prediction of protein and RNA structures [5, 6]. Having witnessed these progresses, one shall prospect for addressing remaining challenges in order to reshape machine-learning-based modeling into versatile tools beyond data-centric models and to ultimately guide the study of exotic chemical phenomena in complement to current domain expertise. With this motivation, a major part of my research has been concerned of integrating physics-informed representations and neural network techniques for chemical systems that existing approximation-based and learning-based approaches struggle to provide a quantitative description at a tractable computational cost. Among the spectrum of learning-based modeling strategies, we have also investigated several theoretical aspects that we believe to have a sustainable impact on computational chemistry. Notably, principles and models that were historically discovered in the context of chemical physics have been found to dramatically benefit the outcomes of learning-based modeling on vision and language data [7, 8], and we anticipate them to open a promising pathway for the efficient simulation of chemical systems themselves. This dissertation represents endeavors during the course of my graduate study to develop and extend such physics-informed neural approaches for machine learning, with applications to molecular modeling problems at multiple scales including the prediction of (i) molecular electronic structure and dynamics, and (ii) structure ensembles of biomolecular complexes.

Here, we seek to provide a primer on some key elements of chemical physics and discuss their contributions to modern deep learning and the author's research.

1.1 Electronic structure methods and equivariant neural networks

An essential task in molecular simulation is the determination of the potential energy surface based on the laws of quantum mechanics. Restricting ourselves to the Born-Oppenheimer approximation, the potential energy E at a molecular configuration is given by the Time Independent Schrödinger Equation (TISE):

$$\hat{H}|\psi\rangle = E|\psi\rangle \tag{1.1}$$

In an *ab-initio* treatment, the Hamiltonian operator $\hat{H} : L^2(\mathbb{R}^{3N}) \to L^2(\mathbb{R}^{3N})$ comprises of one-electron terms, two-electron terms and nuclei repulsion terms; the state $\psi : \mathbb{R}^{3N} \to \mathbb{C}$ is a N-electron wavefunction satisfying exchange anti-symmetry.

The exponential scaling of function space dimension in (1.1) for a general interacting Hamiltonian renders it intractable to exactly solve the TISE on classical computers beyond toy models, and developing polynomial-scaling numerical methods is a core subject of study in theoretical chemistry [9]. Despite many progresses in modern electronic structure methods, quantum chemistry calculations that explicitly treat all electrons of $|\psi\rangle$ in a correlated manner are still inaccessible to most experimental applications beyond fragment-sized molecules. A myriad of methods such as Hartree-Fock (HF) and standard Kohn-Sham Density Functional Theory (KS-DFT) then adopts a mean-field variational ansatz in which the electronic wavefunction is approximated as a single Slater determinant:

$$|\psi\rangle = \prod_{i}^{n_{\text{occ}}} \hat{a}_i^\dagger|0\rangle = \prod_{i}^{n_{\text{occ}}} (\sum_{\mu} C_{i\mu}\hat{b}_\mu^\dagger)|0\rangle \tag{1.2}$$

where each single-electron orbital i excited by the fermionic creation operator $\hat{a}_i^\dagger$ is represented as the linear combination of an atomic orbital basis $|\Phi_\mu\rangle = \hat{b}_\mu^\dagger|0\rangle$ with variational coefficients $\mathbf{C}$. The electronic part of a DFT Hamiltonian reads:

$$\hat{H} = \sum_{i} \Big[-\frac{1}{2}\nabla_i^2 + v_{\text{nuc}}(\mathbf{r}_i) + \int \frac{\rho(\mathbf{r}')}{\|\mathbf{r}' - \mathbf{r}_i\|} d\mathbf{r}' + v_{\text{xc}}[\rho] \Big] \tag{1.3}$$

where the original TISE is mapped to an effective non-interacting system and v_{xc} is called the exchange-correlation (XC) functional approximating for the non-classical two-electron contributions to the electronic energy. Mean-field methods, especially

DFT, have shown great success in many molecular modeling problems often with a quantitative accuracy able to explain and predict experimental measurements. Yet an XC functional involving exact Hartree-Fock exchange terms and a sufficiently large basis set is required in most cases for DFT to produce reliable predictions [10]. This fact has significantly restricted the applicability of DFT for many large molecular systems such as those in enzyme catalysis [11] and condensed-phase systems such as those in battery materials [12], where standard DFT calculations incur a punitive computational cost while it can be also challenging to apply molecular mechanics approximations or embedding-based techniques due to non-local effects. A parallel line of study, semi-empirical quantum mechanics (SEQM) methods [13, 14], instead aims to further coarse-grain mean-field methods to significantly reduce their computational cost while preserving a qualitatively correct description of molecular electronic structure. A class of SEQM methods is based on a tight-binding approximation to the electron density to alleviate the explicit evaluation of molecular integrals or grid-based integrations, in which the electronic energy is expanded as a series of valence orbital density fluctuations $\delta\rho$ around a reference density ρ_0:

$$E[\rho] = E^{(0)}[\rho_0] + \sum_{r=1} E^{(r)}[\rho_0, (\delta\rho)^r] \tag{1.4}$$

which is often truncated such that the electronic Hamiltonian is approximated as linear and bilinear terms involving atomic or shell-resolved charges and distance-based scaling functions, with parameters obtained through a moderate fitting to either KS-DFT functionals or experimental measurements. A main challenge in the development of SEQM approximations is the tradeoff between the ability to achieve chemical accuracy, the computational cost and the applicability to diverse systems. As elaborated in later chapters, we develop learning-based strategies integrating orbital-based representations from SEQM to systematically improve the prediction accuracy of molecular electronic structure and properties while maintaining a physically complete representation of the wavefunction and a transferability to main-group chemical systems out of the training data distribution.

We now take a closer look at the basis $|\Phi_\mu\rangle$ appeared in (1.2). A class of commonly adopted basis functions in quantum chemistry are Slater-type orbitals that resemble solutions of the TISE for hydrogen-like atoms [15]:

$$\Phi_\mu(\mathbf{r}) := \Phi^A_{nlm}(\mathbf{r}) = R^A_{nl}(r) Y_{lm}(\hat{\mathbf{r}}) \tag{1.5}$$

which are eigenstates of the quantum angular momentum operator: $\hat{J}^2|nlm\rangle = l(l+1)|nlm\rangle$, $\hat{J}_\pm|nlm\rangle = \sqrt{l(l+1) - m(m \pm 1)}|nl(m \pm 1)\rangle$. As will be discussed later with greater details, this is equivalent to the fact that the spherical harmonics Y_{lm} forms a basis of SO(3) irreducible representations over the space $L^2(\mathbb{S}^2)$. One remarkable consequence is that the action of the angular momentum operator over a N-particle system decomposes the product state into a linear combination of total angular momentum eigenstates:

$$|LM\rangle = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} |l_1 m_1; l_2 m_2\rangle\langle l_1 m_1; l_2 m_2|LM\rangle \tag{1.6}$$

Although seemingly unrelated to its original context, (1.6) enables the formulation of expressive nonlinear operations in deep neural networks that are *equivariant* to arbitrary Euclidean transformations [16]. We will discuss how this connection improves the accuracy and versatility of orbital-based deep learning techniques.

1.2 Stochastic thermodynamics and score-based generative modeling

Another important discipline in molecular simulation, statistical mechanics, studies the macroscopic thermodynamic properties originated from the collective behavior of many-body systems. A paradigmatic model in classical thermodynamics is the overdamped Langevin process [17] described by the following stochastic differential equation (SDE):

$$d\mathbf{x} = \mathbf{F}(\mathbf{x}, t)dt + \boldsymbol{\sigma}(\mathbf{x}, t)d\mathbf{W}_t \tag{1.7}$$

where $\mathbf{W}_t$ is a standard Brownian motion with diffusion coefficients $\mathbf{D} = \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\sigma}$ corresponding to thermal noises generated from a Markovian bath. It is interesting to study a system initially at thermal equilibrium $p(\mathbf{x}) = \frac{1}{Z}e^{-\beta U(\mathbf{x})}$ and its dissipation behavior when the potential $U(\mathbf{x})$ is erased at $t = 0$. The time evolution of the phase-space density $p(\mathbf{x}, t)$ is dictated by the Fokker-Planck Equation:

$$\partial_t p(\mathbf{x}, t) = -\sum_i \partial_{x_i}[F_i(\mathbf{x}, t)p(\mathbf{x}, t)] + \frac{1}{2}\sum_{i,j} \partial_{x_i}\partial_{x_j}[D_{ij}(\mathbf{x}, t)p(\mathbf{x}, t)] \tag{1.8}$$

Starting from (1.8) and its adjoint, a remarkable result [18] states that the transition kernel for a time-reversed process $p(\mathbf{x}_t|\mathbf{x}_s)$ satisfies the following dynamics:

$$\begin{aligned}\partial_t p(\mathbf{x}_t|\mathbf{x}_s) = -\nabla_{\mathbf{x}}^{\mathrm{T}}\big[\mathbf{F}(\mathbf{x}, t)p(\mathbf{x}_t|\mathbf{x}_s) - \frac{1}{p(\mathbf{x}_t)}(\mathbf{D}(\mathbf{x}, t)\log p(\mathbf{x}_t)) \cdot \nabla_{\mathbf{x}}\big] \\ + \frac{1}{2}\big[\nabla_{\mathbf{x}}^{\mathrm{T}} \cdot (\mathbf{D}(\mathbf{x}, t)p(\mathbf{x}_t|\mathbf{x}_s)) \cdot \nabla_{\mathbf{x}}\big]\end{aligned} \tag{1.9}$$

Based on the Feynman–Kac formula [19], the time evolution of probability distribution described by the PDE (1.9) can be solved by sampling stochastic trajectories from a

SDE of similar form to (1.7). When the diffusion coefficients are position-independent $\mathbf{D}(\mathbf{x}, t) = \mathbf{D}(t)$, this reverse-time SDE reads:

$$d\mathbf{x} = [\mathbf{F}(\mathbf{x}, t) - \mathbf{D}(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}, t)]dt + \boldsymbol{\sigma}(\mathbf{x}, t)d\bar{\mathbf{W}}_t \tag{1.10}$$

In machine learning literatures, the extra current term $\nabla_{\mathbf{x}} \log p(\mathbf{x}, t)$ is called a *score function*. As $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{x}_T)p(\mathbf{x}_T)d\mathbf{x}_T$, (1.10) provides a rigorous scheme to sample from $p(\mathbf{x})$ for high-dimensional distributions without computing the partition function Z. In Chapter 4, we discuss that such score-based generative modeling techniques can enable the prediction of complex molecular structures that are important for both molecular biology research and drug discovery applications.

1.3 Structure of the thesis

Chapter 2 introduces a machine learning framework in which the electronic structure properties of molecular systems are predicted using a symmetry-adapted atomic orbital basis and graph neural networks architecture. The presented method, OrbNet, achieves chemical accuracy at the cost of semi-empirical calculations for thermalized geometries on serveral organic molecule benchmarks, and is transferable to systems larger than the molecules included for model training. We also present the analytic nuclear gradient theory of OrbNet for geometry optimizations and molecular dynamics simulations, as well as strategies to improve data efficiency by incorporating auxiliary information from density matrices computed at DFT level.

Chapter 3 introduces a substantially revised framework for orbital-based deep learning based on the symmetry of the matrix representation of N-reduced operators in an atomic orbital basis. We present theoretical results for equivariant neural networks defined on a generalized class of atomic-orbital-operator representations, implementation of the network in the context of semi-empirical Hamiltonians, and applications to several quantum chemistry problems with comparisons to both conventional electronic structure methods and machine-learning-based methods.

Chapter 4 introduces a method for protein-ligand structure prediction based on score-based generative modeling incorporating techniques from protein biophysics and equivariant neural networks. The NeuralPLexer method outperforms existing physics-based and learning-based methods on benchmarking problems including fixed-backbone blind protein-ligand docking and binding site repacking. Moreover, the predictions agree with compound-specific effects on protein structure distributions in contrast to existing ligand-agnostic protein structure prediction algorithms.

Chapter 2

ORBITAL-BASED DEEP LEARNING FOR MOLECULAR ELECTRONIC STRUCTURE

This chapter is based on the following publications:

[1] Zhuoran Qiao, Matthew Welborn, Animashree Anandkumar, Frederick R Manby, and Thomas F Miller III. "OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features". In: *The Journal of Chemical Physics* 153.12 (2020), p. 124111. DOI: 10.1063/5.0021955.

[2] Zhuoran Qiao, Feizhi Ding, Matthew Welborn, Peter J. Bygrave, Daniel G. A. Smith, Animashree Anandkumar, Frederick R. Manby, and Thomas F. Miller. "Multi-task learning for electronic structure to predict and explore molecular potential energy surfaces". In: *arXiv preprint arXiv:2011.02680* (2020). Appeared at *Machine Learning for Molecules workshop at NeurIPS 2020* as a Contributed Talk. DOI: 10.48550/ARXIV.2011.02680.

Abstract

We introduce a machine learning method in which energy solutions from the Schrodinger equation are predicted using symmetry adapted atomic orbitals features and a graph neural-network architecture. OrbNet is shown to outperform existing methods in terms of learning efficiency and transferability for the prediction of density functional theory results while employing low-cost features that are obtained from semi-empirical electronic structure calculations. Learning efficiency of the method can be further improved by incorporating physically motivated constraints on the electronic structure through multi-task learning. For applications to datasets of drug-like molecules, including QM7b-T, QM9, GDB-13-T, DrugBank, and the conformer benchmark dataset of Folmsbee and Hutchison, OrbNet predicts energies within chemical accuracy of DFT at a computational cost that is thousand-fold or more reduced.

2.1 Introduction

The potential energy surface is the central quantity of interest in the modeling of molecules and materials. Calculation of these energies with sufficient accuracy in chemical, biological, and materials systems is in many – but not all – cases adequately described at the level of density functional theory (DFT). However, due

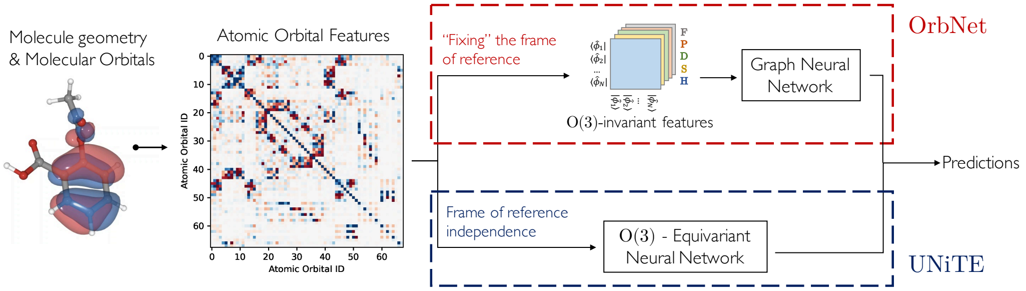

Figure 2.1: A schematic overview of orbital-based deep learning frameworks. A low-level quantum chemistry calculation is first performed on the molecular system, generating the atomic orbital (AO) feature matrices which are constructed from quantum-chemical operators associated with a near-minimal basis set. (Dark red) In the OrbNet approach, a graph neural network is proposed to predict the electronic energy of the molecule. Rotation-inversion invariance is realized through the construction of symmetry-adapted atomic orbitals (SAAOs) and feature matrices re-evaluated in the SAAO basis used as inputs to the neural network. In the followup OrbNet-Equi approach, the prediction is performed in an end-to-end fashion, directly using the AO feature matrices as inputs to the machine learning model; rotation-inversion equivariance is realized through the UNiTE neural network based on O(3)-representation theory. OrbNet-Equi supports the prediction of a broader set of quantum chemistry properties such as multipoles and densities.

to its relatively high cost, the applicability of DFT is limited to either relatively small molecules or modest conformational sampling, at least in comparison to force-field and semi-empirical quantum mechanical theories. A major focus of machine learning (ML) for quantum chemistry has therefore been to improve the efficiency with which potential energies of molecular and materials systems can be predicted while preserving accuracy.

In the context of quantum chemistry, many applications have focused on the use atom- or geometry-specific feature representations and kernel-based[20–28] or neural-network (NN) ML architectures.[29–42] Recent studies focus on the featurization of molecules in abstracted representations — such as quantum mechanical properties obtained from low-cost electronic structure calculations[43–47] — and the utilization of novel graph-based neural network[48–54] techniques to improve transferability and learning efficiency.

In this vein, we present an approach based on the featurization of molecules in terms of symmetry-adapted atomic orbitals (SAAOs) and the use of graph neural network methods for deep-learning quantum-mechanical properties. We demonstrate the performance of the new method for the prediction of molecular properties, including the total and relative conformer energies for molecules in a range of datasets of organic and drug-like molecules. The method enables the prediction of molecular potential energy surfaces with full quantum mechanical accuracy while enabling vast reductions in computational cost; moreover, the method outperforms existing methods in terms of its training efficiency and transferable accuracy across diverse molecular systems.

2.2 Method

The target of the OrbNet machine learning approach is to learn a transferable mapping E_θ from input molecular-orbital-based features $\{\mathbf{f}\}$ to the quantum mechanical energies E.

$$E \approx E_\theta \left[\{\mathbf{f}\}\right]. \tag{2.1}$$

The key elements of OrbNet (Fig. 2.2) include the efficient evaluation of the features $\{\mathbf{f}\}$ in the rotation-inversion-invariant symmetry-adapted atomic orbital (SAAO) basis, the utilization of a graph neural network (GNN) architecture with edge and node attention and message passing layers, and a decoding phase that ensures extensivity of the resulting energies.

Symmetry-adapted atomic orbital (SAAO) features

Let $\{\Phi^A_{n,l,m}\}$ be the set of atomic orbital (AO) basis functions with atom index A and the standard principal and angular momentum quantum numbers, n, l, and m. Let $\mathbf{C}$ be the corresponding molecular orbital coefficient matrix obtained from a mean-field electronic structure calculation, such as HF theory, DFT, or a semi-empirical method. The one-electron density matrix of the molecular system in the AO basis is then

$$P_{\mu\nu} = 2 \sum_{i\in\mathrm{occ}} C_{\mu i} C_{\nu i} \tag{2.2}$$

(for a closed-shell system). We construct a rotationally invariant symmetry-adapted atomic-orbital (SAAO) basis $\{\hat{\Phi}^A_{n,l,m}\}$ by diagonalizing diagonal density-matrix blocks associated with indices A, n, and l, such that

$$\mathbf{P}^A_{nl}\mathbf{Y}^A_{nl} = \mathbf{Y}^A_{nl}\,\mathrm{diag}(\lambda^A_{nlm}) \tag{2.3}$$

where $[\mathbf{P}^A_{nl}]_{mm'} = P^A_{nlm,nlm'}$. For s orbitals ($l = 0$), this symmetrization procedure is obviously trivial, and can be skipped. By construction, SAAOs are localized and consistent with respect to geometric perturbations of the molecule, and in contrast with localized molecular orbitals (LMOs) obtained from minimizing a localization objective function (Pipek-Mezey, Boys, etc.), SAAOs are obtained by a series of

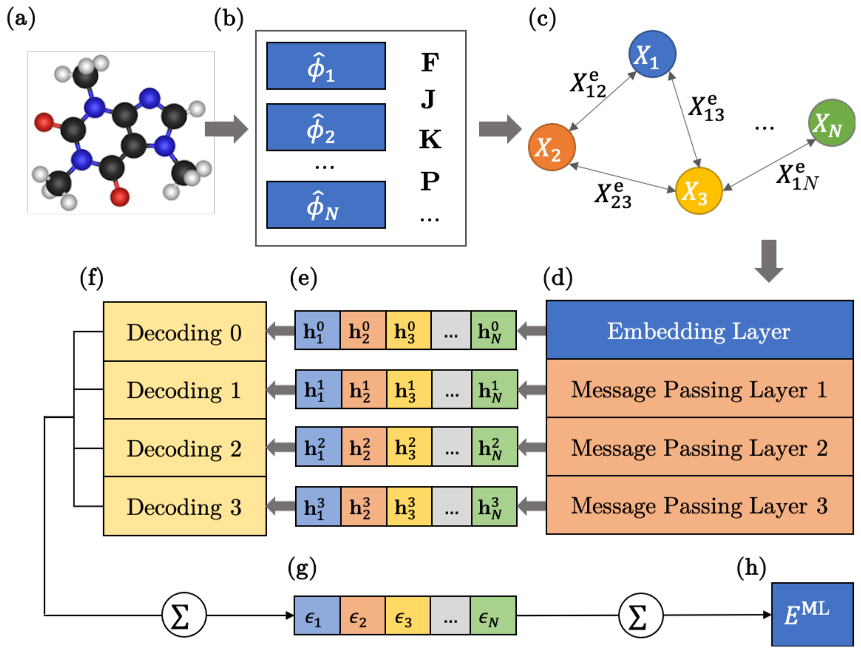
(a)
(b)
$\hat{\phi}_1$
$\hat{\phi}_2$
…
$\hat{\phi}_N$
F
J
K
P
…
(c)
X_1
X_2
X_3
X_N
X^{e}_{12}
X^{e}_{13}
X^{e}_{23}
X^{e}_{1N}
…
(d)
Embedding Layer
Message Passing Layer 1
Message Passing Layer 2
Message Passing Layer 3
(e)
$\mathbf{h}^0_1$ $\mathbf{h}^0_2$ $\mathbf{h}^0_3$ ... $\mathbf{h}^0_N$
$\mathbf{h}^1_1$ $\mathbf{h}^1_2$ $\mathbf{h}^1_3$ ... $\mathbf{h}^1_N$
$\mathbf{h}^2_1$ $\mathbf{h}^2_2$ $\mathbf{h}^2_3$ ... $\mathbf{h}^2_N$
$\mathbf{h}^3_1$ $\mathbf{h}^3_2$ $\mathbf{h}^3_3$ ... $\mathbf{h}^3_N$
(f)
Decoding 0
Decoding 1
Decoding 2
Decoding 3
(g)
Σ
ϵ_1 ϵ_2 ϵ_3 ... ϵ_N
Σ
(h)
E^{ML}

Figure 2.2: Summary of the OrbNet workflow. (a) A low-cost mean-field electronic structure calculation is performed for the molecular system, and (b) the resulting the one-electron quantum operators and the SAAOs are constructed. (c) An attributed graph representation is built with node and edge attributes corresponding to the diagonal and off-diagonal elements of the SAAO tensors. (d) The attributed graph is processed by the embedding layer and message passing layers to produce transformed node and edge attributes. (e) The transformed node attributes for the encoding layer and each message passing layer are extracted and (f) passed to MPL-specific decoding networks. (g) The node-resolved energy contributions ϵ_u are obtained by summing the decoding networks outputs node-wise, and (h) the final extensive energy prediction is obtained from a one-body summation over the nodes.

very small diagonalizations, without the need for an iterative procedure. The SAAO eigenvectors $\mathbf{Y}^{A}_{nl}$ are aggregated to form a block-diagonal transformation matrix $\mathbf{Y}$ that specifies the full transformation from AOs to SAAOs:

$$|\hat{\Phi}_p\rangle = \sum_{\mu} Y_{\mu p}|\Phi_\mu\rangle \tag{2.4}$$

where μ and p index the AOs and SAAOs, respectively.

We employ ML features $\{\mathbf{f}\}$ comprised of matrices obtained by evaluating (one-electron-reduced-) quantum-chemical operators in the SAAO basis. In Sections 2.2-2.5, all quantum mechanical matrices will be assumed to represented in the SAAO basis.

The features include expectation values of the Fock ($\mathbf{F}$), density ($\mathbf{P}$), core Hamiltonian ($\mathbf{H}$), and overlap ($\mathbf{S}$) operators in the SAAO basis. For the models employed in Section 2.2, Coulomb ($\mathbf{J}$), exchange ($\mathbf{K}$), and the orbital centroid distance ($\mathbf{D}$) matrices in the SAAO basis are also employed as features; other quantum-mechanical matrix elements are also possible for featurization.

Approximated Coulomb and exchange for SAAO features

When a semi-empirical quantum chemical theory is employed, the computational bottleneck of SAAO feature generation becomes the $\mathbf{J}$ and $\mathbf{K}$ terms, due to the need to compute four-index electron-repulsion integrals. We address this problem by introducing a generalized form of the Mataga–Nishimoto–Ohno–Klopman formula, as in the sTDA-xTB method,[55, 56]

$$(pq|rs)^{\mathrm{MNOK}} = \sum_{A}\sum_{B} Q^{A}_{pq} Q^{B}_{rs} \gamma_{AB} \tag{2.5}$$

Here, A and B are atom indices, p, q, r, s are SAAO indices, and

$$\gamma^{\{\mathbf{J},\mathbf{K}\}}_{AB} = \left(\frac{1}{R_{AB}^{y_{\{\mathbf{J},\mathbf{K}\}}} + \eta^{-y_{\{\mathbf{J},\mathbf{K}\}}}}\right)^{1/y_{\{\mathbf{J},\mathbf{K}\}}} \tag{2.6}$$

where R_{AB} is the distance between atoms A and B, η is the average chemical hardness for the atoms A and B, and $y_{\{\mathbf{J},\mathbf{K}\}}$ are empirical parameters specifying the decay behavior of the damped interaction kernels, $\gamma^{\{\mathbf{J},\mathbf{K}\}}_{AB}$. In this work, we used $y_{\mathbf{J}} = 4$ and $y_{\mathbf{K}} = 10$ similar to which employed in the sTDA-RSH method[57]. The transition density Q^{A}_{pq} is calculated from a Löwdin population analysis,

$$Q^{A}_{pq} = \sum_{\mu \in A} Y'_{\mu p} Y'_{\mu q} \tag{2.7}$$

where the pth column of $\mathbf{Y}' = \mathbf{Y}\mathbf{S}^{1/2}$ contains the expansion coefficients for the pth SAAO in the symmetrically orthgonalized AO basis. This yields approximated $\mathbf{J}$ and $\mathbf{K}$ matrices for featurization:

$$J_{pq}^{\mathrm{MNOK}} = (pp|qq)^{\mathrm{MNOK}} = \sum_{A,B} Q_{pp}^{A} Q_{qq}^{B} \gamma_{AB}^{\mathbf{J}} \tag{2.8}$$

$$K_{pq}^{\mathrm{MNOK}} = (pq|pq)^{\mathrm{MNOK}} = \sum_{A,B} Q_{pq}^{A} Q_{pq}^{B} \gamma_{AB}^{\mathbf{K}} \tag{2.9}$$

A naive implementation of Eqs. 2.8 and 2.9 is $O(N^4)$, the leading asymptotic cost. However, this scaling may be reduced to $O(N^2)$ with negligible loss of accuracy through a tight-binding approximation; for molecules in this study, computation of $\mathbf{J}^{\mathrm{MNOK}}$ and $\mathbf{K}^{\mathrm{MNOK}}$ is not the leading order cost for feature generation and such tight-binding approximation is thus not employed.

The OrbNet model architecture

OrbNet encodes the molecular system as graph-structured data and utilizes a graph neural network (GNN) machine-learning architecture. The GNN represents data as an attributed graph $G(\mathbf{V}, \mathbf{E}, \mathbf{X}, \mathbf{X}^{\mathbf{e}})$, with nodes $\mathbf{V}$, edges $\mathbf{E}$, node attributes $\mathbf{X} : \mathbf{V} \longrightarrow \mathbb{R}^{n \times d}$, and edge attributes $\mathbf{X}^{\mathbf{e}} : \mathbf{E} \longrightarrow \mathbb{R}^{n_{\mathrm{e}} \times e}$, where $n = |V|$, $n_{\mathrm{e}} = |E|$, and d and e are the number of attributes per node and edge, respectively.

Specifically, OrbNet employs a graph representation for a molecular system in which node attributes correspond to diagonal SAAO features

$$X_u = [F_{uu}, J_{uu}, K_{uu}, P_{uu}, H_{uu}] \tag{2.10}$$

and edge attributes correspond to off-diagonal SAAO features

$$X_{uv}^{\mathrm{e}} = [F_{uv}, J_{uv}, K_{uv}, D_{uv}, P_{uv}, S_{uv}, H_{uv}] \tag{2.11}$$

By introducing an edge attribute cutoff value for edges to be included, non-interacting molecular systems separated at infinite distance are encoded as disconnected graphs, thereby satisfying size-consistency.

The model capacity is enhanced by introducing nonlinear input-feature transformations to the graph representation via radial basis functions to generate node embeddings $\mathbf{h}^{\mathrm{RBF}}$ and edge embeddings $\mathbf{e}^{\mathrm{RBF}}$, with infinite-order-differentiable 'auxiliary edge' attributes $\mathbf{e}_{uv}^{\mathrm{aux}}$ to enforce size-consistency, as detailed in Appendix 2.7. The radial basis function embeddings are transformed by neural network modules to yield 0-th order node and edge attributes,

$$\mathbf{h}_u^0 = \mathrm{Enc_h}(\mathbf{h}_u^{\mathrm{RBF}}), \ \mathbf{e}_{uv}^0 = \mathrm{Enc_e}(\mathbf{e}_{uv}^{\mathrm{RBF}}) \tag{2.12}$$

where $\mathrm{Enc_h}$ and $\mathrm{Enc_e}$ are residual blocks[58] comprising 3 dense NN layers. In contrast to atom-based message passing neural networks, this additional embedding transformation captures the interactions among the physical operators.

The node and edge attributes are updated via the Transformer-motivated[59] message passing mechanism in Fig. 2.6. For a given message passing layer (MPL) $t+1$, the information carried by each edge is encoded into a message function $\mathbf{m}_{uv}^t$ and associated attention weight $\mathbf{m}_{uv}^t$, and is accumulated into node features through a graph convolution operation. The overall message passing mechanism is given by:

$$\mathbf{h}_u^{t+1} = \mathbf{h}_u^t + \sigma\Big(\mathbf{W}_\mathrm{h}^t \cdot \big[\bigoplus_j \big(\sum_{v \in N(u)} w_{uv}^{t,j} \cdot \mathbf{m}_{uv}^t\big)\big] + \mathbf{b}_\mathrm{h}^t\Big) \tag{2.13}$$

where $\mathbf{m}_{uv}^t$ is the message function computed on each edge

$$\mathbf{m}_{uv}^t = \sigma(\mathbf{W}_\mathrm{m}^t \cdot [\mathbf{h}_u^t \odot \mathbf{h}_v^t \odot \mathbf{e}_{uv}^t] + \mathbf{b}_\mathrm{m}^t) \tag{2.14}$$

and the convolution kernel weights, $w_{uv}^{t,j}$, are evaluated as (multi-head) attention scores[49] to characterize the relative importance of an orbital pair,

$$w_{uv}^{t,j} = \sigma_\mathrm{a}\big(\sum [(\mathbf{W}_\mathrm{a}^{t,j} \cdot \mathbf{h}_u^t) \odot (\mathbf{W}_\mathrm{a}^{t,j} \cdot \mathbf{h}_v^t) \odot \mathbf{e}_{uv}^t \odot \mathbf{e}_{uv}^\mathrm{aux}]/n_\mathrm{e}\big) \tag{2.15}$$

where the summation is applied over the elements of the vector in the summand. Here, the index j specifies a single attention head, and n_e is the dimension of hidden edge features $\mathbf{e}_{uv}^t$, $\bigoplus$ denotes a vector concatenation operation, $\odot$ denotes the Hadamard product, and $\cdot$ denotes the matrix-vector product.

The edge attributes are updated according to

$$\mathbf{e}_{uv}^{t+1} = \sigma(\mathbf{W}_\mathrm{e}^t \cdot \mathbf{m}_{uv}^t + \mathbf{b}_\mathrm{e}^t) \tag{2.16}$$

$\mathbf{W}_\mathrm{m}^t$, $\mathbf{W}_\mathrm{h}^t$, $\mathbf{W}_\mathrm{e}^t$, $\mathbf{b}_\mathrm{m}^t$, $\mathbf{b}_\mathrm{h}^t$, $\mathbf{b}_\mathrm{e}^t$ are MPL-specific trainable parameter matrices, $\mathbf{W}_\mathrm{a}^{t,i}$ are MPL- and attention-head-specific trainable parameter matrices, $\sigma(\cdot)$ is an activation function with a normalization layer, and $\sigma_\mathrm{a}(\cdot)$ is the activation function used for generating attention scores.

The decoding phase of OrbNet (Fig. 2.2f-h) is designed to ensure the size-extensivity of energy predictions. The employed mechanism outputs node-resolved energy contributions for the embedding layer ($t=0$) and all MPLs ($t = 1, 2, ..., T$) to predict the energy components associated with all nodes and MPLs. The final energy prediction E_θ is obtained by first summing over t (Fig. 2.2g) for each node u and then performing a one-body sum over nodes (i.e., orbitals) (Fig. 2.2h), such that

$$E_\theta = \sum_{u \in \mathbf{V}} \epsilon_u = \sum_{u \in \mathbf{V}} \sum_{t=0}^{T} \mathrm{Dec}^t(\mathbf{h}_u^t) \tag{2.17}$$

where the decoding networks Dec^t are multilayer perceptrons.

2.3 Numerical results

We present results that focus on the prediction of accurate DFT energies using input features obtained from the GFN1-xTB method [60], a member of the family of semi-empirical quantum mechanics (SEQM) methods for molecular electronic structure simulations. The GFN family of methods[60–62] have proven to be extremely useful for the simulation of large molecular system (1000s of atoms or more) with time-to-solution for energies and forces on the order of seconds. However, this applicability can be limited by the accuracy of the semi-empirical method,[63, 64] thus creating a natural opportunity for "delta-learning" the difference between the GFN1 and DFT energies on the basis of the GFN1 features. Specifically, we consider regression labels associated with the difference between high-level DFT and the GFN1-xTB total atomization energies,

$$E_\theta \approx E^{\mathrm{DFT}} - E^{\mathrm{GFN1}} - \Delta E_{\mathrm{atoms}}^{\mathrm{fit}} \tag{2.18}$$

where the last term is the sum of differences for the isolated-atom energies between DFT and GFN1 as determined by a linear model. This approach yields the direct ML prediction of total DFT energies, given the results of a GFN1-xTB calculation.

The QM9 dataset

We begin with a broad comparison of recently introduced ML methods for the total energy task, U_0, from the widely studied QM9 dataset.[65] QM9 is composed of organic molecules with up to nine heavy atoms at locally optimized geometries, so this test (Table 2.1) examines the expressive power of the ML models for systems in similar chemical environments. Results for OrbNet are presented both without ensemble averaging of independently trained models (i.e., predicting only on the basis of the first of trained model) and with ensemble averaging the results of five independently trained models (OrbNet-ens5). As observed previously,[52] ensembling helps in this and other learning tasks, reducing the OrbNet prediction error by approximately 10-20%.

Table 2.1: MAEs (reported in meV) for predicting the QM9 dataset of total energies at the B3LYP/6-31G(2df,p) level of theory. Results from the current work are reported for a single model (OrbNet) and with ensembling over 5 models (OrbNet-ens5).

Training size	SchNet[51]	PhysNet[52]	PhysNet-ens5[52]	DimeNet[53]	DeepMoleNet[54]	OrbNet	OrbNet-ens5
25,000	-	-	-	-	-	**11.6**	**10.4**
50,000	15	13	10	-	-	**8.22**	**6.80**
110,000	14	8.2	6.1	8.02	6.1	**5.01**	**3.92**

Also included in the table are previously published methods utilizing graph representations of atom-based features, including SchNet[51], PhysNet[52], DimeNet[53], and DeepMoleNet[54]. We note that DimeNet employs a directional message passing mechanism and PhysNet and DeepMoleNet employ supervision based on prior physical information to improve the model transferability, which could also be employed within OrbNet; it is clear that without these additional strategies and even without model ensembling, OrbNet provides greater accuracy and learning efficiency than all previous deep-learning methods.

Transferability and Conformer Energy Predictions

A more realistic and demanding test of ML methods is to train them on datasets of relatively small molecules (for which high-accuracy data is more readily available) and then to test on datasets of larger and more diverse molecules. This provides useful insight into the transferability of the ML methods and the general applicability of the trained models.

To this end, we investigate the performance of OrbNet on a series of dataset containing organic and drug-like molecules. Fig. 2.3 presents results in which OrbNet models are trained with increasing amounts of data. Using the training-test splits described in Section 2.7, Model 1 is trained using data from only the QM7b-T dataset; Model 2 is trained using data from the QM7b-T, GDB13-T, and DrugBank-T datasets; Model 3 is trained using data from the QM7b-T, QM9, GDB13-T, and DrugBank-T datasets; and Model 4 is obtained by ensembling five independent training runs with the same data as used for Model 3. Predictions are made for total energies (Fig. 2.3A) and relative conformer energies (Fig. 2.3B) for held-out molecules from each of these datasets, as well as for the Hutchison conformer dataset.

As expected, it is seen from Fig. 2.3 that the OrbNet predictions improve with additional data and with ensemble modeling. The median and mean of the absolute errors consistently decrease from Model 1 to Model 4 except for a non-monotonicity in the DrugBank-T MAE, likely due to the relatively small size of that dataset. It is nonetheless striking that Model 1, which includes only data from QM7b-T yields relative conformer energy predictions on the DrugBank-T and Hutchison datasets (which include molecules with up to 50 heavy atoms) with an accuracy that is comparable to the more heavily trained models. Note that all of the OrbNet models predict relative conformer energies with MAE and median prediction errors that are well within the 1 kcal/mol threshold of chemical accuracy, across all four test datasets. Predictions for QM9 using Models 1 and 2 are not included, since QM9 includes F atoms whereas the training data in those models do not; relative conformer energies are not predicted for QM9 since they are not available in this dataset. Although total energy prediction error for the OrbNet is slightly larger per heavy atom on the Hutchison dataset than for the other datasets, the relative conformer energy prediction error for the Hutchison dataset is slightly smaller than for GDB13-T and DrugBank-T; this is due to the fact that the Hutchison dataset involves locally minimized conformers that are less spread in energy per heavy atom than the conformers of the thermalized datasets. This relatively small energy spread among conformers in the Hutchison dataset is a realistic and challenging aspect of drug-molecule conformer-ranking prediction, which we next consider.

Figure 2.4 presents a direct comparison of the accuracy and computational cost of OrbNet in comparison to a variety of other force-field, semiempirical, machine-learning, DFT, and wavefunction methods, as compiled in Ref. 64. For the Hutchison conformer dataset of drug-like molecules which range in size from nine to 50 heavy atoms, the accuracy of the various methods was evaluated using the median R^2 of the predicted conformer energies in comparison to DLPNO-CCSD(T) reference data and with computation time evaluated on a single CPU core.[64]

The OrbNet conformer energy predictions (Fig. 2.4, black) are reported using Model 4 (i.e., with training data from QM7b-T, GDB13-T, DrugBank-T, and QM9 and with ensemble averaging over five independent training runs). The solid black circle indicates the median R^2 value (0.81) of the OrbNet predictions relative to the DLPNO-CCSD(T) reference data, as for the other methods; this point provides the most direct comparison to the accuracy of the other methods. The open black circle indicates the median R^2 value (0.90) of the OrbNet predictions relative to the ωB97X-D/Def2-TZVP reference data against which the model was trained; this point indicates the accuracy that would be expected of the Model 4 implementation of OrbNet if it had employed coupled-cluster training data rather than DFT training data. We performed timings for OrbNet on a single core of an Intel Core i5-1038NG7 CPU @ 2.00GHz, finding that the OrbNet computational cost is dominated by the GFN1-xTB calculation for the feature generation. In contrast to Ref. 64 which used the XTB code of Grimme and coworkers[66], we used ENTOS QCORE for the GFN1-xTB calculation calculations. We find the reported timings for GFN1-xTB to be surprisingly slow in Ref. 64, particularly in comparison to the GFN0-xTB timings. For GFN0-xTB, our timings with ENTOS QCORE are very similar to those reported in Ref. 64, which is sensible given that the method involves no self-consistent field (SCF) iteration. However, whereas Ref. 64 indicates GFN1-xTB timings that are 43-fold slower than GFN0-xTB, we find this ratio to be only 4.5 with Entos Qcore, perhaps due to differences of SCF convergence. To account for the issue of code efficiency in the GFN1-xTB implementation and to control for the details of the single CPU core used in the timings for this work versus in Ref. 64, we normalize the OrbNet timing reported in Fig. 2.4 with respect to the GFN0-xTB timing from Ref. 64. The CPU neural-network inference costs for OrbNet are negligible contribution to this timing.

The results in Fig. 2.4 make clear that OrbNet enables the prediction of relative conformer energies for drug-like molecules with an accuracy that is comparable to DFT but with a computational cost that is 1000-fold reduced from DFT to realm of semiempirical methods. Alternatively viewed, the results indicate that OrbNet provides dramatic improvements in prediction accuracy over currently available ML and semiempirical methods for realistic applications, without significant increases in computational cost.

2.4 Analytical nuclear gradient theory

In this section, we introduce and numerically demonstrate the analytical gradient theory for OrbNet, which is essential for the calculation of inter-atomic forces and other response properties, such as dipoles and linear-response excited states.

OrbNet is constructed to be end-to-end differentiable by employing input features (i.e., the SAAO matrix elements) that are smooth functions of both atomic coordinates and external fields. We derive the analytic gradients of the total energy E^{out} with respect to the atom coordinates, and we employ local energy minimization with respect to molecular structure as an exemplary task to demonstrate the quality of the learned potential energy surface (Section 2.4).

Analytical gradient formulation

The derivation of analytical nuclear gradients for OrbNet is nontrivial due to the response of molecular orbitals and the density matrix when perturbing atomic nuclei coordinates. In our work, this challenge is addressed using a Lagrangian formalism [67, 68], and the analytic gradient of the predicted energy with respect to an atom coordinate x can be expressed in terms of contributions from the tight-binding model,

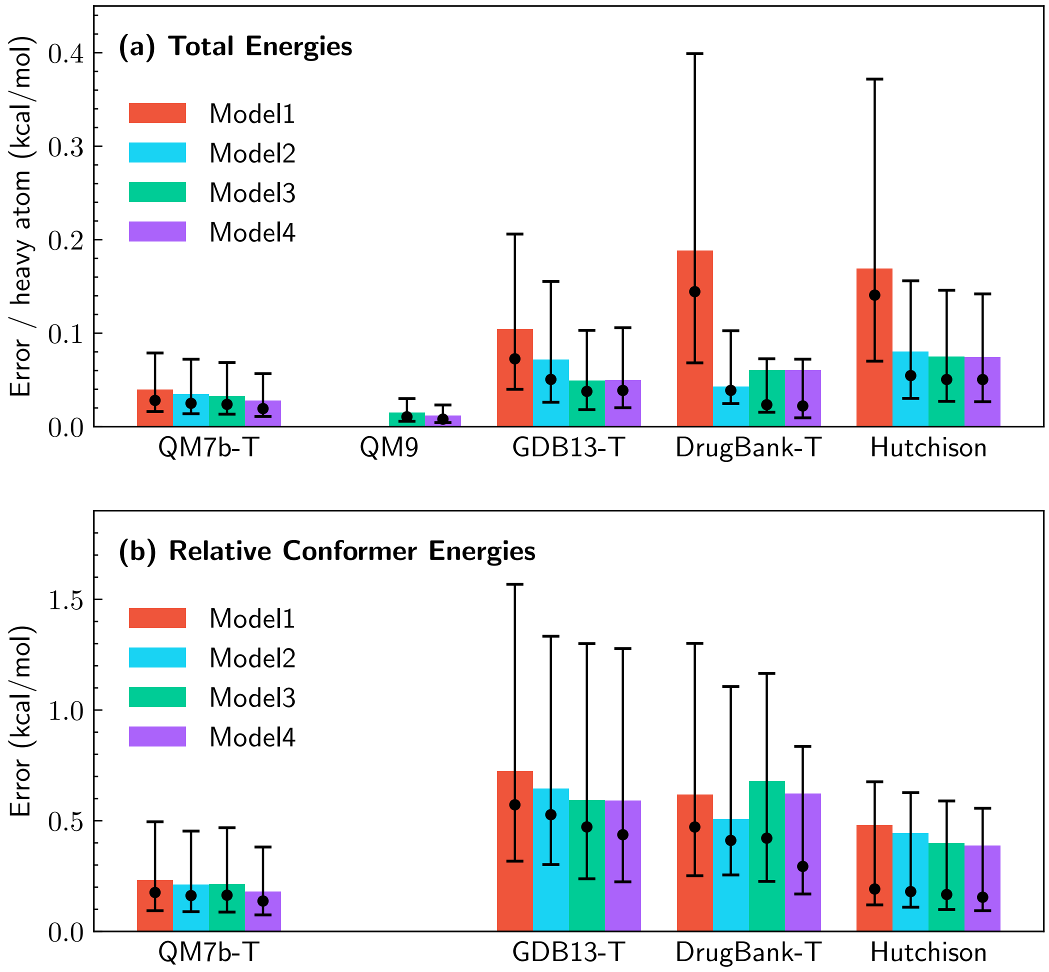

Figure 2.3: Prediction errors for (a) molecule total energies and (b) relative conformer energies performed using OrbNet models trained using various datasets. The mean absolute error (MAE) is indicated by the bar height, the median of the absolute error is indicated by a black dot, and the the first and third quantiles for the absolute error are indicated as the lower and upper bars. Model 1 uses training data from QM7b-T; Model 2 additionally includes training data from GDB13-T and DrugBank-T; Model 3 additionally includes training data from QM9; and Model 4 additionally includes ensemble averaging over five independent training runs. Testing is performed on data that is held-out from training in all cases. Training and prediction employs energies at the ωB97X-D/Def2-TZVP level of theory. All energies in kcal/mol.

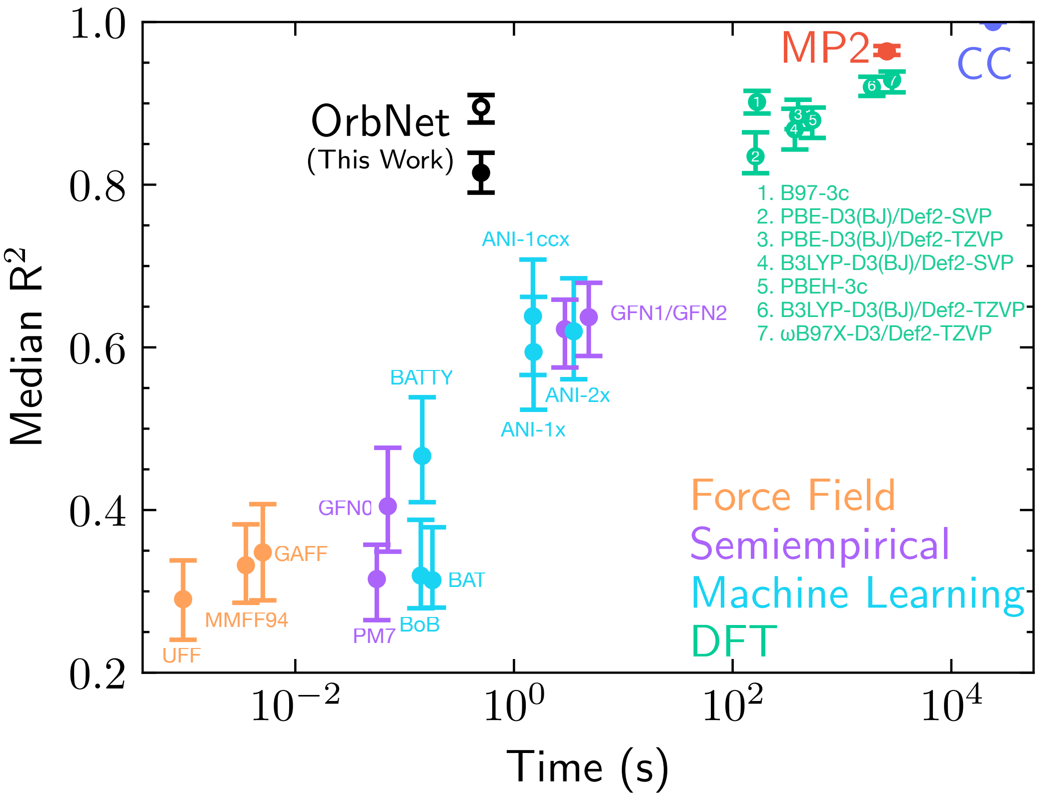

Figure 2.4: Comparison of the accuracy/computational-cost tradeoff for a range of potential energy methods for the Hutchison conformer benchmark dataset. Aside from the OrbNet results (black), all data was previously reported in Ref. 64, with median R^2 values for the predicted conformer energies computed relative to DLPNO-CCSD(T) reference data and with computation time evaluated on a single CPU core. The OrbNet results (black) are obtained using Model 4 (i.e., with training data from QM7b-T, GDB13-T, DrugBank-T, and QM9 and with ensemble averaging over five independent training runs). The solid black circle plots the median R^2 value from the OrbNet predictions relative to DLPNO-CCSD(T) reference data, as for the other methods. The open black circle plots the median R^2 value from the OrbNet predictions relative to the ωB97X-D/Def2-TZVP reference data against which the OrbNet model was trained. Error bars correspond to the 95% confidence interval, determined by statistical bootstrapping.

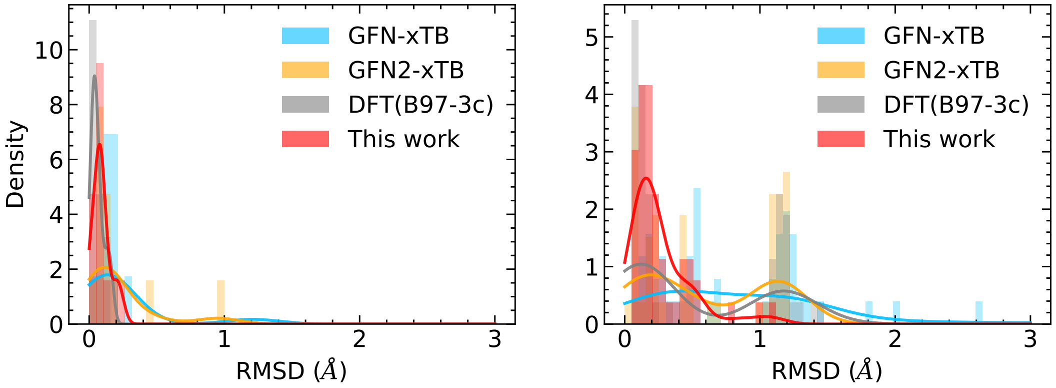

Figure 2.5: The molecular geometry optimization accuracy for the ROT34 (left) and MCONF (right) datasets, reported as the best-alignment root-mean-square-deviation (RMSD) compared to the reference DFT geometries at the ωB97X-D3/Def2-TZVP level. The distribution of errors are plotted as histograms (with overlaying kernel density estimations). Timings correspond to the average cost for a single force evaluation for the MCONF dataset on a single Intel Xeon Gold 6130 @ 2.10GHz CPU core.

the neural network, and additional constraint terms:

$$\frac{dE_{\text{out}}}{dx} = \frac{dE_{\text{TB}}}{dx} + \sum_{\mathbf{f}\in\{\mathbf{F},\mathbf{D},\mathbf{P},\mathbf{S},\mathbf{H}\}} \text{Tr}\left[\frac{\partial E_{\text{NN}}}{\partial \mathbf{f}}\frac{\partial \mathbf{f}}{\partial x}\right] + \text{Tr}[\mathbf{W}\frac{\partial \mathbf{S}^{\text{AO}}}{\partial x}] + \text{Tr}[\mathbf{z}\frac{\partial \mathbf{F}^{\text{AO}}}{\partial x}] \quad (2.19)$$

Here, the third and fourth terms on the right-hand side are gradient contributions from the orbital orthogonality constraint and the Brillouin condition, respectively, where $\mathbf{F}^{\text{AO}}$ and $\mathbf{S}^{\text{AO}}$ are the Fock matrix and orbital overlap matrix in the atomic orbital (AO) basis. An overview of the expressions for $\frac{\partial \mathbf{f}}{\partial x}$, $\mathbf{W}$, and $\mathbf{z}$ and derivations are provided in Appendix 2.7. The gradient for the GFN-xTB model $\frac{dE_{\text{TB}}}{dx}$ has been previously reported [60], and the neural network gradients with respect to the input features $\frac{\partial E_{\text{NN}}}{\partial \mathbf{f}}$ are obtained using reverse-mode automatic differentiation [69].

Results: Molecular geometry optimizations

A practical application of energy gradient (i.e., force) calculations is to optimize molecule structures by locally minimizing the energy. Here, we use this application as a test of the accuracy of the OrbNet potential energy surface in comparison to other widely used methods of comparable and greater computational cost. Test are performed for the ROT34 [70] and MCONF [71] datasets, with initial structures that are locally optimized at the high-quality level of ωB97X-D3/Def2-TZVP DFT with tight convergence parameters. Dataset and geometry optimization details

Table 2.2: The mean geometry optimization errors and the percentage of optimized structures that correspond to incorrect geometries (i.e., RMSD > 0.6 Angstrom).

Method	Mean RMSD (Å)		Incorrect geometries		Time/step
	ROT34	MCONF	ROT34	MCONF	MCONF
GFN-xTB	0.23	0.90	8%	52%	< 1 s
GFN2-xTB	0.21	0.60	8%	44%	< 1 s
DFT (B97-3c)	**0.06**	0.51	**0%**	37%	> 100 s
This work	0.09	**0.26**	**0%**	**6%**	< 1 s
Ref. DFT (ωB97X-D3)	-	-	-	-	> 1,000 s

are provided in Appendix 2.7. This test investigates whether the potential energy landscape for each method is locally consistent with a high-quality DFT description.

Fig. 2.5 presents the resulting distribution of errors for the various methods over each dataset, with results summarized in the accompanying table. It is clear that while the GFN semi-empirical methods provide a computational cost that is comparable to OrbNet, the resulting geometry optimizations are substantially less accurate, with a significant (and in some cases very large) fraction of the local geometry optimizations relaxing into structures that are inconsistent with the optimized reference DFT structures (i.e., with RMSD in excess of 0.6 Angstrom). In comparison to DFT using the B97-3c functional, OrbNet provides optimized structures that are of comparable accuracy for ROT34 and that are significantly more accurate for MCONF; this should be viewed in light of the fact that OrbNet is over 100-fold less computationally costly. On the whole, OrbNet is the best approximation to the reference DFT results, at a computational cost that is over 1,000-fold reduced.

2.5 Improved data efficiency via multi-task learning

To improve data efficiency, we introduce a multi-task learning strategy in which OrbNet is trained with respect to both molecular energies and other computed properties of the quantum mechanical wavefunction.

Refinements on the OrbNet architecture for multi-task learning

In this extension to the OrbNet framework, the feature embedding and neural message-passing mechanism employed for the node and edge attributes is largely unchanged. However, to enable multi-task learning and to improve the capacity of the model, we introduce atom-specific attributes $\mathbf{f}^t_A$, and molecule-level attributes $\mathbf{q}^t$, where t is the message passing layer index and A is the atom index. The whole-molecule and atom-specific attributes allow for the prediction of auxiliary targets through multi-task learning, thereby providing physically motivated constraints on the electronic structure of the molecule that can be used to refine the representation at the SAAO level.

For the prediction of both the electronic energies and the auxiliary targets, only the final atom-specific attributes, $\mathbf{f}_A^L$, are employed, since they self-consistently incorporate the effect of the whole-molecule and node- and edge-specific attributes. The electronic energy is obtained by combining the approximate energy E_{TB} from the extended tight-binding calculation and the model output E_{NN}, the latter of which is a one-body sum over atomic contributions; the atom-specific auxiliary targets $\mathbf{d}_A$ are predicted from the same attributes.

$$\hat{E}_{\text{out}} = E_{\text{TB}} + E_{\text{NN}} = E_{\text{TB}} + \sum_A [\text{Dec}(\mathbf{f}_A^T) + E_A^{\text{c}}] \; ; \quad \hat{\mathbf{d}}_A = \text{Dec}^{\text{aux}}(\mathbf{f}_A^T) \qquad (2.20)$$

Here, the energy decoder Dec and the auxiliary-target decoder Dec^{aux} are residual neural networks [58] built with fully connected and normalization layers, and E_A^{c} are element-specific, constant shift parameters for the isolated-atom contributions to the total energy. The GradNorm algorithm [72] is used to adaptively adjust the weight of the auxiliary target loss based on the gradients of the last fully-connected layer before the decoding networks.

Auxiliary targets from density matrix projection

The utility of graph- and atom-level auxiliary tasks to improve the generalizability of the learned representations for molecules has been highlighted for learning molecular properties in the context of graph pre-training [73, 74] and multi-task learning [54]. Here, we employ multi-task learning with respect to the total molecular energy and atom-specific auxiliary targets. The atom-specific targets that we employ are similar to the features introduced in the DeePHF model [47], obtained by projecting the density matrix into a basis set that does not depend upon the identity of the atomic element,

$$\mathbf{d}_{nl}^A = [\text{EigenVals}_{m,m'}([\,{}^{\text{O}}\mathscr{D}_{nl}^A]_{m,m'})||\text{EigenVals}_{m,m'}([\,{}^{\text{V}}\mathscr{D}_{nl}^A]_{m,m'})] \qquad (2.21)$$

Here, the projected density matrix is given by $[\,{}^{\text{O}}\mathscr{D}_{nl}^A]_{m,m'} = \sum_{i \in \text{occ}} \langle \alpha_{nlm}^A | \psi_i \rangle \langle \psi_i | \alpha_{nlm'}^A \rangle$, and the projected valence-occupied density matrix is given by $[\,{}^{\text{V}}\mathscr{D}_{nl}^A]_{m,m'} = \sum_{j \in \text{valocc}} \langle \alpha_{nlm}^A | \psi_j \rangle \langle \psi_j | \alpha_{nlm'}^A \rangle$, where $|\psi_{\{i,j\}}\rangle$ are molecular orbitals from the reference DFT calculation, $|\alpha_{nlm}^A\rangle$ is a basis function centered at atom A with radial index

n and spherical-harmonic degree l and order m. The indices i and j runs over all occupied orbitals and valence-occupied orbital indices, respectively, and $||$ denotes a vector concatenation operation. The auxiliary target vector $\mathbf{d}_A$ for each atom A in the molecule is obtained by concatenating $\mathbf{d}_{nl}^A$ for all n and l. The parameters for the projection basis $|\alpha_{nlm}^A\rangle$ can be found in the Appendix of the original publication [75]. Additional attributes, such as such as partial charges and reactivities, could also be naturally included within this framework.

Results: Atomization energy predictions

We perform a standard benchmark test of predicting molecular energies for the QM9 dataset. Table 2.3 presents results from current work, as well as previously published results using SchNet [51], PhysNet [52], DimeNet [53], DeepMoleNet [54], and OrbNet [76]. The approach proposed in this work significantly outperforms existing methods [51–54] in terms of both data efficiency and prediction accuracy. In particular, it is seen that the use of multi-task learning in the current study leads to significant improvement over the OrbNet results obtained via directly training on energies (single-task), which already exhibited the smallest errors among published methods.

2.6 Conclusions

Electronic structure methods typically face a punishing trade-off between the prediction accuracy of the method and its computational cost, across all areas of the chemical, biological, and materials sciences. We present a new machine-learning method with the potential to substantially shift that trade-off in favor of *ab initio*-quality accuracy at low computational cost. OrbNet utilizes a graph neural network architecture to predict high-quality electronic-structure energies on the basis of features obtained from low-cost/minimal-basis mean-field electronic structure methods. The method is demonstrated for the case of predicting ωB97X-D/Def2-TZVP energies using GFN1-xTB input features, although it is completely general with respect to both the choice of high-level (including correlated wavefunction) method used for generating reference data and the choice of mean-field method used for feature generation. In comparison to state-of-the-art GNN methods for the prediction of total molecule energies for the QM9 dataset, it is shown that OrbNet provides a 33% improvement in prediction accuracy with the same amount of data relative to the next-most accurate method (DeepMoleNet).[54] And in comparison to the wide array of methods used for predicting relative conformer energies in a realistic and

Table 2.3: MAEs (reported in meV) for predicting the QM9 dataset of total molecular energies. The employed labels are the published values [65] calculated at the B3LYP/6-31G(2df,p) level of theory.

Training size	SchNet	PhysNet	DimeNet	DeepMoleNet	OrbNet	OrbNet (+**d**)
25k	-	-	-	-	11.6	**8.08**
50k	15	13	-	-	8.22	**5.89**
110k	14	8.2	8.02	6.1	5.01	**3.87**

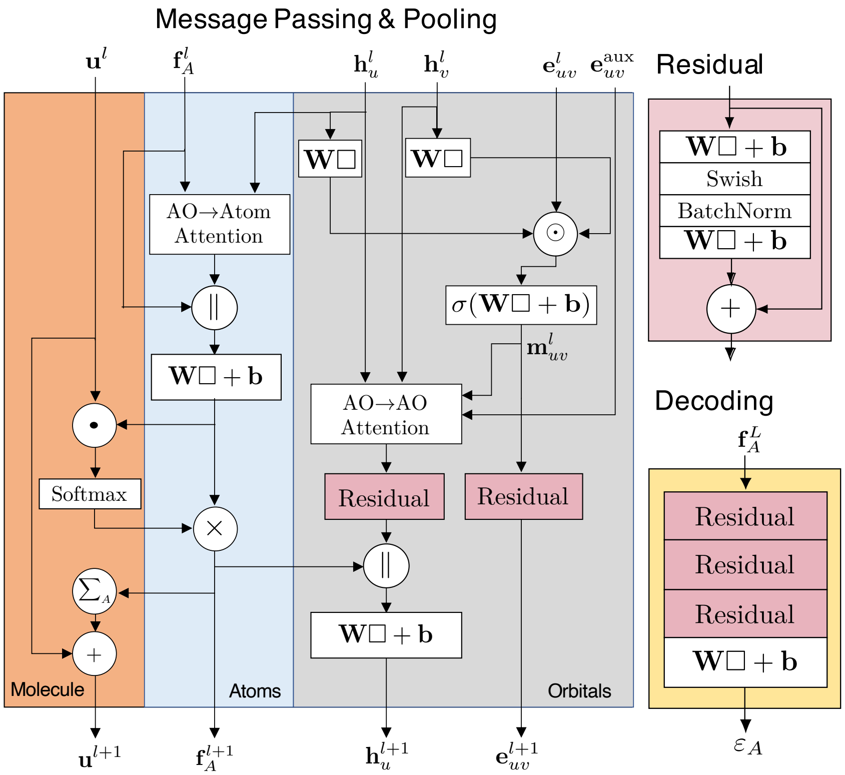

Figure 2.6: Detail of a single message-passing and pooling layer ("Message Passing Layer" in Fig. 2.2), and a decoding network ("Decoding" in Fig. 2.2). At message passing and pooling layer $l + 1$, the whole-molecule, atom-specific, node-specific, and edge-specific attributes are updated. The atom-specific attributes $\mathbf{f}_A^l$ are updated with input from node- and edge-specific attributes $\mathbf{h}_u^l$ and $\mathbf{e}_{uv}^l$ and likewise includes the back-propagation from the whole-molecule attributes; finally, the whole-molecule attributes $\mathbf{q}^l$ are updated with input from the atom-specific attributes. The final atom-specific attributes are passed into separate decoding networks to generate the energy prediction and auxiliary target predictions. A decoding network is composed of multiple residual blocks ("Residual") and a linear output layer, as illustrated above.

diverse dataset of drug-like molecules, as compiled by Folmsbee and Hutchison,[64] it is shown that OrbNet provides a prediction accuracy that is similar to DFT and much improved over existing ML methods, but at a computational cost that is reduced by at least three orders of magnitude relative to DFT. Natural future directions for development will include the expansion of OrbNet to a broader set of chemical elements, incorporation of directional message-passing and model supervision using prior physical information,[52–54] and end-to-end refitting of the semi-empirical method used for feature generation.[33, 77]

2.7 Appendix

Feature and architecture details

We employ the following feature embedding scheme where the SAAO feature matrices are transformed by radial basis functions,

$$\mathbf{h}_u^{\mathrm{RBF}} = [\phi_1^{\mathrm{h}}(\tilde{X}_u), \phi_2^{\mathrm{h}}(\tilde{X}_u), ..., \phi_{n_{\mathrm{r}}}^{\mathrm{h}}(\tilde{X}_u)] \tag{2.22}$$

$$\mathbf{e}_{uv}^{\mathrm{RBF}} = [\phi_1^{\mathrm{e}}(\tilde{X}_{uv}^{\mathrm{e}}), \phi_2^{\mathrm{e}}(\tilde{X}_{uv}^{\mathrm{e}}), ..., \phi_{m_{\mathrm{r}}}^{\mathrm{e}}(\tilde{X}_{uv}^{\mathrm{e}})] \tag{2.23}$$

where $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}^{\mathrm{e}}$ are pre-normalized SAAO feature matrices, $\phi_n^{\mathrm{h}}(r) = \sin(\pi n r)$ is a sine function used for node (SAAO) embedding; to improve the smoothess of the potential energy surface, we used the real Morlet wavelet functions for edge embedding:

$$\phi_m^{\mathrm{e}}(r) = \exp(-(\frac{r}{\sigma \cdot c_{\mathbf{X}}})^2) \cdot \sin(\pi m r / c_{\mathbf{X}}) \tag{2.24}$$

and $c_{\mathbf{X}}$ ($\mathbf{X} \in \{\, \mathbf{F}, \mathbf{D}, \mathbf{P}, \mathbf{S}, \mathbf{H} \,\}$) is the operator-specific upper cutoff value to $\tilde{X}_{uv}^{\mathrm{e}}$. To ensure size-consistency for energy predictions, a mollifier $I_{\mathbf{X}}(r)$ with the auxiliary edge attribute $\mathbf{e}_{uv}^{\mathrm{aux}}$ is introduced:

$$\mathbf{e}_{uv}^{\mathrm{aux}} = \mathbf{W}^{\mathrm{aux}} \cdot I_{\mathbf{X}}(\tilde{X}_{uv}^{\mathrm{e}}) \tag{2.25}$$

where

$$I_{\mathbf{X}}(r) = \begin{cases} \exp\left(\frac{c_{\mathbf{X}}}{|r| - c_{\mathbf{X}}} + 1\right) \cdot \exp(-(\frac{r}{\sigma \cdot c_{\mathbf{X}}})^2) & \text{if } 0 \le |r| < c_{\mathbf{X}} \\ 0 \quad \text{if } |r| \ge c_{\mathbf{X}} \end{cases} \tag{2.26}$$

In the revised OrbNet architecure 2.5, the radial basis function embeddings of the SAAOs and a one-hot encoding of the chemical element of the atoms ($\mathbf{f}_A^{\mathrm{onehot}}$) are transformed by neural network modules to yield 0-th order SAAO, SAAO-pair, and atom attributes,

$$\mathbf{h}_u^0 = \mathrm{Enc_h}(\mathbf{h}_u^{\mathrm{RBF}}),\ \mathbf{e}_{uv}^0 = \mathrm{Enc_e}(\mathbf{e}_{uv}^{\mathrm{RBF}}),\ \mathbf{f}_A^0 = \mathrm{Enc_f}(\mathbf{f}_A^{\mathrm{onehot}}) \tag{2.27}$$

where $\mathrm{Enc_h}$ and $\mathrm{Enc_e}$ are residual blocks[58] comprising 3 dense NN layers, and $\mathrm{Enc_f}$ is a single dense NN layer. In contrast to atom-based message passing neural networks, this additional embedding transformation captures the interactions among the physical operators.

The update of the node- and edge-specific attributes (gray block in Fig. 2.6) is unchanged from Ref. [76], except with the additional information back-propagation from the atom-specific attributes. The node and edge attributes at step $l + 1$ are updated via the following neural message passing mechanism (corresponding to "AO-AO attention" in Fig. 2.6):

$$\tilde{\mathbf{h}}_u^{l+1} = \mathbf{h}_u^l + \mathbf{W}_{\mathrm{h},2}^l \cdot \mathrm{Swish}\Big(\mathrm{BatchNorm}(\mathbf{W}_{\mathrm{h},1}^l \cdot [\bigoplus_i (\sum_{v\in N(u)} w_{uv}^{l,i} \cdot \mathbf{m}_{uv}^l)] + \mathbf{b}_{\mathrm{h},1}^l)\Big) + \mathbf{b}_{\mathrm{h},2}^l \tag{2.28a}$$

$$\mathbf{m}_{uv}^l = \mathrm{Swish}(\mathbf{W}_\mathrm{m}^l \cdot [\mathbf{h}_u^l \odot \mathbf{h}_v^l \odot \mathbf{e}_{uv}^l] + \mathbf{b}_\mathrm{m}^l) \tag{2.28b}$$

$$w_{uv}^{l,i} = \mathrm{Tanh}(\sum[(\mathbf{W}_\mathrm{a}^{l,i} \cdot \mathbf{h}_u^l) \odot (\mathbf{W}_\mathrm{a}^{l,i} \cdot \mathbf{h}_v^l) \odot \mathbf{e}_{uv}^l \odot \mathbf{e}_{uv}^{\mathrm{aux}}]/n_\mathrm{e}) \tag{2.28c}$$

$$\mathbf{e}_{uv}^{l+1} = \mathbf{e}_{uv}^l + \mathbf{W}_{\mathrm{e},2}^l \cdot \big(\mathrm{Swish}(\mathbf{W}_{\mathrm{e},1}^l \cdot \mathbf{m}_{uv}^l + \mathbf{b}_{\mathrm{e},1}^l)\big) + \mathbf{b}_{\mathrm{e},2}^l \tag{2.28d}$$

where $\mathbf{m}_{uv}^l$ is the message function on each edge, $w_{uv}^{l,i}$, are multi-head attention scores [49] for the relative importance of SAAO pairs (i indexes attention heads), $\bigoplus$ denotes a vector concatenation operation, $\odot$ denotes the Hadamard product, and $\cdot$ denotes the matrix-vector product.

The SAAO attributes are accumulated into the atoms on which the corresponding SAAOs are centered, using an attention-based pooling operation ("AO-Atom attention" in Fig. 2.6) inspired by the set transformer [78] architecture:

$$a_{A,u}^l = \mathrm{Softmax}(\mathbf{f}_A^l \cdot (\mathbf{h}_u^l)^\mathrm{T}/\sqrt{n_\mathrm{h}}) \tag{2.29a}$$

$$\tilde{\mathbf{f}}_A^{l+1} = \mathbf{W}_{\mathrm{f},1}^l \cdot \big[\mathbf{f}_A^l || (\sum_{u\in A} a_{A,u}^l \mathbf{h}_u^l)\big] + \mathbf{b}_{\mathrm{f},1}^l \tag{2.29b}$$

where the Softmax operation is taken over all SAAOs u centered on atom A. Then the global attention α_A^l is calculated for all atoms in the molecule to update the molecule-level attribute $\mathbf{q}^{l+1}$:

$$\alpha_A^{l+1} = \mathrm{Softmax}(\mathbf{q}^l \cdot (\tilde{\mathbf{f}}_A^{l+1})^\mathrm{T}/\sqrt{n_\mathrm{h}}) \tag{2.30a}$$

$$\mathbf{q}^{l+1} = \mathbf{q}^l + \sum_A \alpha_A^{l+1} \tilde{\mathbf{f}}_A^{l+1} \tag{2.30b}$$

where the Softmax is taken over all atoms in the molecule, and the initial global attribute $\mathbf{q}^0$ is a molecule-independent, trainable parameter vector.

Finally, the molecule- and atom-level information is propagated back to the SAAO attributes:

$$\mathbf{f}_A^{l+1} = \alpha_A^{l+1} \tilde{\mathbf{f}}_A^{l+1} \tag{2.31a}$$

$$\mathbf{h}_u^{l+1} = \mathbf{W}_{\mathrm{f},2}^l \cdot \big[\mathbf{f}_A^{l+1} || \tilde{\mathbf{h}}_u^{l+1}\big] + \mathbf{b}_{\mathrm{f},2}^l. \tag{2.31b}$$

The list of trainable model parameters is: $\mathbf{W}^{\mathrm{aux}}$, $\mathbf{W}_{\mathrm{h},1}^l$, $\mathbf{W}_{\mathrm{h},2}^l$, $\mathbf{b}_{\mathrm{h},1}^l$, $\mathbf{b}_{\mathrm{h},2}^l$, $\mathbf{W}_\mathrm{m}^l$, $\mathbf{b}_\mathrm{m}^l$, $\mathbf{W}_\mathrm{a}^{l,i}$, $\mathbf{W}_{\mathrm{e},1}^l$, $\mathbf{W}_{\mathrm{e},2}^l$, $\mathbf{b}_{\mathrm{e},1}^l$, $\mathbf{b}_{\mathrm{e},2}^l$, $\mathbf{W}_{\mathrm{f},1}^l$, $\mathbf{W}_{\mathrm{f},2}^l$, $\mathbf{b}_{\mathrm{f},1}^l$, $\mathbf{b}_{\mathrm{f},2}^l$, $\mathbf{q}^0$, and the parameters of Enc_h, Enc_e, Enc_f, Dec, and $\mathrm{Dec}^{\mathrm{aux}}$.

Dataset and computational details

Datasets used in Sec. 2.2

Results are presented for the QM7b-T dataset[44, 79] (which has seven conformations for each of 7211 molecules[37] with up to seven heavy atoms of type C, O, N, S, and Cl), the QM9 dataset[65] (which has locally optimized geometries for 133885 molecules with up to nine heavy atoms of type C, O, N, and F), the GDB-13-T dataset[44, 79] (which has six conformations for each of 1000 molecules from the GDB-13 dataset[80] with up to thirteen heavy atoms of type C, O, N, S, and Cl), DrugBank-T (which has six conformations for each of 168 molecules from the DrugBank database[81] with between fourteen and 30 heavy atoms of type C, O, N, S, and Cl), and the Hutchison conformer dataset from Ref. 64 (which has up to 10 conformations for each of 622 molecules with between nine and 50 heavy atoms of type C, O, N, F, P, S, Cl, Br, and I). Except for DrugBank-T, all of these datasets have been described previously; thermalized geometries from the DrugBank dataset are sampled at 50 fs intervals from *ab initio* molecular dynamics trajectories performed using the B3LYP[82–85]/6-31g*[86] level of theory and a Langevin thermostat[87] at 350 K. The structures for the datasets are provided in the Supporting Information of our published works. [76] For results reported in Section 2.3, the pre-computed DFT labels from Ref. 65 were employed. For results reported in Section 2.3, all DFT labels were computed using the ωB97X-D functional[88] with a Def2-TZVP AO basis set[89] and using density fitting[90] for both the Coulomb and exchange integrals using the Def2-Universal-JKFIT basis set;[91] these calculations are performed using Psi4.[92] Semi-empirical calculations are performed using the GFN1-xTB method[60] using the Entos Qcore[93] package, which is also employed for the SAAOs feature generation.

For the results presented in this work, we train OrbNet models using the following training-test splits of the datasets. For results on the QM9 dataset, we removed 3054 molecules due to a failed a geometric consistency check, as recommended in Ref. 65; we then randomly sampled 110000 molecules for training and used 10831 molecules for testing. The training sets of 25000 and 50000 molecules in section 2.3 are subsampled from the 110000-molecule dataset. For the QM7b-T dataset, two sets of training-test splits are generated; for the model trained on the QM7b-T dataset only (Model 1 in Section 2.3), we randomly selected 6500 different molecules (with 7 geometries for each) from the total 7211 molecules for training, holding out 500 molecules (with 7 geometries for each) for testing; for Models 2-4 in Section 2.3, we used a 361-molecule subset of this 500-molecules set for testing, and we used the remaining 6850 molecules of QM7b-T for training. For the GDB13-T dataset, we randomly sampled 948 different molecules (with 6 geometries for each) for training, holding out 48 molecules (with 6 geometries for each) for testing. For the DrugBank-T dataset, we randomly sampled 158 different molecules (with 6 geometries for each) for training, holding out 10 molecules (with 6 geometries for each) for testing. No training on the Hutchison conformer dataset was performed, as it was only used for transferability testing. Since none of the training datasets for OrbNet included molecules with elements of type P, Br, and I, we excluded the molecules in the Hutchison dataset that included elements of these types for the reported tests (as was also done in Ref. 64 and in Fig. 2.4 for the ANI methods). Moreover, following Ref. 64, we excluded sixteen molecules due to missing DLPNO-LCCSD(T) reference data; an additional eight molecules were excluded on the basis of DFT convergence issues for at least one conformer using Psi4. The specific molecules that appear in all training-test splits are listed in the Supporting Information of the original publication [76].

Datasets used in Sec. 2.5

For results reported in Section 2.5, we employ the QM9 dataset[65] with pre-computed DFT labels. From this dataset, 3054 molecules were excluded as recommended in Ref. [65]; we sample 110000 molecules for training and 10831 molecules for testing. The training sets of 25000 and 50000 molecules are subsampled from the 110000-molecule dataset.

To train the model reported in Section 2.4, we employ the published geometries from Ref. [76], which include optimized and thermalized geometries of molecules up to 30 heavy atoms from the QM7b-T, QM9, GDB13-T, and DrugBank-T datasets. We perform model training using the dataset splits of Model 3 in Ref. [76]. DFT labels are computed using the ωB97X-D3 functional [94] with a Def2-TZVP AO basis set[89] and using density fitting[90] for both the Coulomb and exchange integrals using the Def2-Universal-JKFIT basis set.[91]

For results reported in Section 2.4, we perform geometry optimization for the DFT, OrbNet, and GFN-xTB calculations by minimizing the potential energy using the BFGS algorithm with the Translation-rotation coordinates (TRIC) of Wang and Song[95]; geometry optimizations for GFN2-xTB are performed using the default algorithm in the xtb package [66].

ROT34 includes conformers of 12 small organic molecules with up to 13 heavy atoms; MCONF includes 52 conformers of the melatonin molecule which has 17 heavy atoms. All local geometry optimizations are initialized from pre-optimized structures at the ωB97X-D3/Def2-TZVP level of theory. From these initial structures, we performed a local geometry optimization using the various energy methods, including OrbNet from the current work, the GFN semi-empirical methods [60, 62], and the relatively low-cost DFT functional B97-3c [96]. For the B97-3c method, the mTZVP basis set is employed. The error in the resulting structure with respect to the reference structures optimized at the ωB97X-D3/Def2-TZVP level was computed as root mean squared distance (RMSD) following optimal molecular alignment.

All DFT and GFN-xTB calculations are performed using Entos Qcore [93]; GFN2-xTB calculation are performed using xtb package [66].

Hyperparameters and training details

Hyperparameters and model training for results in Sec. 2.2

Table 2.4 summarizes the hyperparameters used for OrbNet in Sec. 2.2 for the reported results. We perform a pre-transformation on the input features from **F**, **J**, **K**, **D**, **P**, **H**, and **S** to obtain $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}^{\mathrm{e}}$: We normalize all diagonal SAAO tensor values X_{uu} to range $[0, 1)$ for each operator type to obtain $\tilde{X}_u$; for off-diagonal SAAO tensor values, we take $\tilde{X}_{uv} = -\ln(|X_{uv}|)$ for $\mathbf{X} \in \{\mathbf{F}, \mathbf{J}, \mathbf{K}, \mathbf{P}, \mathbf{S}, \mathbf{H}\}$, and $\tilde{D}_{uv} = D_{uv}$. The model hyperparameters are selected within a limited search space; the cutoff hyperparameters $c_{\mathbf{X}}$ are obtained by examining the overlap between feature element distributions between the QM7b-T and GDB13-T datasets. The same set of hyperparameters is used throughout this work, thereby providing a universal model.

To provide additional regularization for predicting energy variations from the configurational degree of freedom, we performed training on loss function of the form

$$\begin{aligned} \mathcal{L}(\hat{\mathbf{E}}, \mathbf{E}) &= (1-\alpha) \sum_i \mathcal{L}_2(\hat{E}_i, E_i) \\ &+ \alpha \sum_i \mathcal{L}_2(\hat{E}_i - \hat{E}_{t(i)}, E_i - E_{t(i)}). \end{aligned} \tag{2.32}$$

For a conformer i in a minibatch, we randomly sample another conformer $t(i)$ of the same molecule to be paired with i to evaluate the relative conformer loss $\mathcal{L}_2(\hat{E}_i - \hat{E}_{t(i)}, E_i - E_{t(i)})$, putting additional penalty on the prediction errors for configurational energy variations. **E** denotes the ground truth energy values of the

Hyperparameter	Meaning	Value or name
n_r	Number of basis functions for node embedding	8
m_r	Number of basis functions for edge embedding	8
n_h	Dimension of hidden node attributes	256
n_e	Dimension of hidden edge attributes	64
n_a	Number of attention heads	4
L	Number of message passing layers	3
L_enc	Number of dense layers in $\mathrm{Enc_h}$ and $\mathrm{Enc_e}$	3
L_dec	Number of dense layers in a decoding network	4
	Hidden dimensions of a decoding network	128, 64, 32, 16
σ	Activation function	Swish
σ_a	Activation function for attention generation	TanhShrink
γ	Batch normalization momentum	0.4
$c_\mathbf{F}$	Cutoff value for $\tilde{F}_{uv}$	8.0
$c_\mathbf{J}$	Cutoff value for $\tilde{J}_{uv}$	1.6
$c_\mathbf{K}$	Cutoff value for $\tilde{K}_{uv}$	20.0
$c_\mathbf{D}$	Cutoff value for $\tilde{D}_{uv}$	9.45
$c_\mathbf{P}$	Cutoff value for $\tilde{P}_{uv}$	14.0
$c_\mathbf{S}$	Cutoff value for $\tilde{S}_{uv}$	8.0
$c_\mathbf{H}$	Cutoff value for $\tilde{H}_{uv}$	8.0

Table 2.4: Model hyperparameters employed in the OrbNet model. All cutoff values are in atomic units.

minibatch, $\hat{\mathbf{E}}$ denotes the model prediction values of the minibatch, and $\mathcal{L}_2$ denotes the L2 loss function $\mathcal{L}_2(\hat{y}, y) = ||\hat{y} - y||_2^2$. For all models in Section 2.3, we choose $\alpha = 0$ as only the optimized geometries are available; for models in Section 2.3, we choose $\alpha = 0.9$ for all training setups.

All models are trained on a single Nvidia Tesla V100-SXM2-32GB GPU using the Adam optimizer.[97] For all training runs, we set the minibatch size to 64 and use a cyclical learning rate schedule[98] that performs a linear learning rate increase from 3×10^{-5} to 3×10^{-3} for the initial 100 epochs, a linear decay from 3×10^{-3} to 3×10^{-5} for the next 100 epochs, and an exponential decay with a factor of 0.9 every epoch for the final 100 epochs. Batch normalization[99] is employed before every activation function σ except for that used in the attention heads, σ_a.

Hyperparameters and model training for results in Sec. 2.4-2.5

Table 2.5 summarizes the hyperparameters for OrbNet employed in Sec. 2.4-2.5. We perform a pre-transformation on the input features from **F**, **D**, **P**, **H**, and **S** to obtain

Hyperparameter	Meaning	Value
n_r	Number of basis functions for node embedding	8
m_r	Number of basis functions for edge embedding	8
n_h	Dimension of hidden node attributes	256
n_e	Dimension of hidden edge attributes	64
n_a	Number of attention heads	4
L	Number of message passing & pooling layers	2
L_enc	Number of dense layers in Enc_h and Enc_e	3
L_dec	Number of residual blocks in a decoding network	3
n_d	Hidden dimension of a decoding network	256
γ	Batch normalization momentum	0.4
$c_\mathbf{F}$	Cutoff value for $\tilde{F}_{uv}$	6.0
$c_\mathbf{D}$	Cutoff value for $\tilde{D}_{uv}$	9.45
$c_\mathbf{P}$	Cutoff value for $\tilde{P}_{uv}$	6.0
$c_\mathbf{S}$	Cutoff value for $\tilde{S}_{uv}$	6.0
$c_\mathbf{H}$	Cutoff value for $\tilde{H}_{uv}$	6.0
σ	Morlet wavelet RBF scale	1/3

Table 2.5: Model hyperparameters employed in the revised OrbNet model. All cutoff values are in atomic units.

$\tilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}^\mathrm{e}$: We normalize all diagonal SAAO tensor values X_{uu} to the range $[0, 1)$ for each operator type to obtain $\tilde{X}_u$; for off-diagonal SAAO tensor values, we take $\tilde{X}_{uv} = -\ln(|X_{uv}|)$ for $\mathbf{X} \in \{\, \mathbf{F}, \mathbf{P}, \mathbf{S}, \mathbf{H} \,\}$, and $\tilde{D}_{uv} = D_{uv}$.

Training is performed on a loss function of the form

$$\begin{aligned}\mathcal{L}(\hat{\mathbf{E}}, \mathbf{E}, \hat{\mathbf{d}}, \mathbf{d}) &= (1-\alpha)\sum_i \mathcal{L}_2(\hat{E}_i, E_i) \\ &+ \alpha \sum_i \mathcal{L}_2(\hat{E}_i - \hat{E}_{t(i)}, E_i - E_{t(i)}) \qquad (2.33) \\ &+ \beta \sum_i \sum_{A\in i} \mathcal{L}_2(\hat{\mathbf{d}}_A, \mathbf{d}_A). \qquad (2.34)\end{aligned}$$

$\sum_i$ denotes summation over a minibatch of molecular geometries i. For each geometry i, we randomly sample another conformer of the same molecule $t(i)$ to evaluate the relative conformer loss $\mathcal{L}_2(\hat{E}_i - \hat{E}_{t(i)}, E_i - E_{t(i)})$; $\mathbf{E}$ denotes the ground truth energy values of the minibatch, $\hat{\mathbf{E}}$ denotes the model prediction values of the minibatch; $\hat{\mathbf{d}}_A$ and $\mathbf{d}_A$ denote the predicted and reference auxiliary target vectors for each atom A in molecule i, and $\mathcal{L}_2(\hat{y}, y) = ||\hat{y} - y||_2^2$ denotes the L2 loss function. For the model used in Section 2.5, we choose $\alpha = 0$ as only the optimized geometries are available; for models in Section 2.4, we choose $\alpha = 0.95$. β is adaptively updated using the

GradNorm[72] method.

All models are trained on a single Nvidia Tesla V100-SXM2-32GB GPU using the Adam optimizer [97]. For all training runs, we set the minibatch size to 64 and use a cosine annealing with warmup learning rate schedule [100] that performs a linear learning rate increase from 3×10^{-6} to 3×10^{-4} for the initial 100 epochs, and a cosine decay from 3×10^{-4} to 0 for 200 epochs.

Analytical nuclear gradients for atomic orbital features

The electronic energy in the OrbNet model is given by

$$E_{\mathrm{out}}[\mathbf{f}] = E_{\mathrm{xTB}} + E_{\mathrm{NN}}[\mathbf{f}] \tag{2.35}$$

Here, $\mathbf{f}$ denotes the features, which correspond to the matrix elements of the single-electron quantum mechanical operators $\{\mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{H}, \mathbf{S}\}$ evaluated in the AO or the SAAO basis. We provide an overview of the derivation of analytical nuclear gradients for the SAAO features used in OrbNet. The full details of the gradient expressions can be found in our published work [75].

Generation of SAAOs

We denote $\{\phi^A_{n,l,m}\}$ as the set of atomic basis functions with atom indices A, with principle, angular and magnetic quantum numbers n, l, m, and $\{\psi_i\}$ as the set of canonical molecular orbitals obtained from a low-level electronic structure calculation.

We define the transformation matrix $\mathbf{X}$ between AOs and SAAOs as eigenvectors of the local density matrices (in covariant form):

$$\tilde{\mathbf{P}}^A_{n,l}\mathbf{X}^A_{n,l} = \mathbf{X}^A_{n,l}\Sigma^A_{n,l} \tag{2.36}$$

where $\tilde{\mathbf{P}}$ is the covariant density matrix in AO basis and is defined as

$$\tilde{\mathbf{P}} = \mathbf{SP}^{\mathrm{AO}}\mathbf{S} \tag{2.37}$$

Neural network gradient

The Lagrangian for OrbNet is

$$\mathcal{L} = E_{\mathrm{NN}}[\mathbf{f}] + \sum_{pq} W_{pq}\left(\mathbf{C}^{\dagger}\mathbf{SC} - \mathbf{I}\right)_{pq} + \sum_{ai} z_{ai}F_{ai} \tag{2.38}$$

The second term of right-hand-side corresponds to the orbitals orthogonality constraint, and the third term corresponds to the Brillion conditions.

Stationary condition for the Lagrangian with respect to the MOs

The Lagrangian is stationary with respect to variations of the MOs:

$$\frac{\partial \mathcal{L}}{\partial V_{pq}} = 0 \tag{2.39}$$

where V_{pq} is a variation of the MOs in terms of the orbital rotation between MO pair p and q and is defined as

$$\tilde{\mathbf{C}} = \mathbf{C}(\mathbf{I} + \mathbf{V}) \tag{2.40}$$

This leads to the following expressions for each term on the right-hand-side of Eq. 2.38:

$$A_{pq} = \frac{\partial E_{\text{NN}}[\mathbf{f}]}{\partial V_{pq}}\bigg|_{\mathbf{V}=0} = \frac{\partial E_{\text{NN}}[\mathbf{f}]}{\partial \mathbf{f}}\frac{\partial \mathbf{f}}{\partial V_{pq}}\bigg|_{\mathbf{V}=0} \tag{2.41}$$

$$W_{pq} = \frac{\partial \sum_{pq} W_{pq}(\mathbf{C}^{\dagger}\mathbf{SC} - \mathbf{I})_{pq}}{\partial V_{pq}}\bigg|_{\mathbf{V}=0} \tag{2.42}$$

$$(\mathbf{A}[\mathbf{z}])_{pq} = \frac{\partial \sum_{ai} z_{ai}F_{ai}}{\partial V_{pq}}\bigg|_{\mathbf{V}=0} = (\mathbf{Fz})_{pq}\bigg|_{q\in\text{occ}} + \left(\mathbf{Fz}^{\dagger}\right)_{pq}\bigg|_{q\in\text{vir}} + 2\left(\mathbf{g}[\bar{\mathbf{z}}]\right)_{pq}\bigg|_{q\in\text{occ}} \tag{2.43}$$

SAAO derivatives

The OrbNet energy gradient involves the derivatives of the SAAO transformation matrix $\mathbf{X}^{A}_{n,l}$ with respect to orbital rotations and nuclear coordinates. As detailed in Appendix D.5 of [75], the derivatives of the SAAOs $\mathbf{X}$ with respect to nuclear coordinates x can be expressed as

$$\frac{\partial \mathbf{X}}{\partial x} = \mathbf{X}\mathbf{T}^{x} \tag{2.44}$$

where $\mathbf{T}^{x}$ is defined as

$$T^{x}_{I,\kappa\lambda} = \frac{\mathbf{X}^{T}_{I,\kappa}\tilde{\mathbf{P}}^{x}_{I}\mathbf{X}_{I,\lambda}}{\epsilon_{I,\lambda} - \epsilon_{I,\kappa}} \quad \text{for } \kappa \neq \lambda, \quad T^{x}_{I,\kappa\kappa} = 0 \tag{2.45}$$

where $\tilde{\mathbf{P}}^{x}_{I}$ is defined as

$$\begin{aligned} \tilde{P}^{x}_{\mu\nu\in I} \equiv \frac{\partial \tilde{P}^{I}_{\mu\nu}}{\partial x} &= \frac{\partial(\mathbf{SPS})_{\mu\nu}}{\partial x} \\ &= \sum_{\kappa\lambda} \frac{\partial S_{\mu\kappa}}{\partial x} P_{\kappa\lambda} S_{\lambda\nu} + S_{\mu\kappa} P_{\kappa\lambda} \frac{\partial S_{\lambda\nu}}{\partial x} \\ &= \sum_{\kappa\lambda} S^{x}_{\mu\kappa} P_{\kappa\lambda} S_{\lambda\nu} + S_{\mu\kappa} P_{\kappa\lambda} S^{x}_{\lambda\nu} \end{aligned} \tag{2.46}$$

Define $N = \mathbf{PS}$, then

$$\tilde{\mathbf{P}}_I^x = \left[\mathbf{S}^x\mathbf{N} + \mathbf{N}^\dagger\mathbf{S}^x\right]_I \tag{2.47}$$

The nuclear derivatives of the OrbNet energy usually involve the term $\mathrm{Tr}[\mathbf{BT}^x]$, which can be re-written as

$$\mathrm{Tr}[\mathbf{BT}^x] = \sum_I \mathrm{Tr}[\tilde{\mathbf{B}}_I\tilde{\mathbf{P}}_I^x] = \sum_I \mathrm{Tr}\left(\tilde{\mathbf{B}}_I\left[\mathbf{S}^x\mathbf{N} + \mathbf{N}^\dagger\mathbf{S}^x\right]_I\right) = \mathrm{Tr}[\overline{\mathbf{W}}\mathbf{S}^x]$$

where $\tilde{\mathbf{B}}$ defined as

$$\bar{B}_{I,\kappa\lambda} = \frac{B_{I,\kappa\lambda}}{\epsilon_\kappa - \epsilon_\lambda} \quad \text{for } \kappa \neq \lambda, \quad \bar{B}_{I,\kappa\kappa} = 0 \tag{2.48}$$

$$\tilde{\mathbf{B}}_I = \frac{1}{2}\mathbf{X}_I(\bar{\mathbf{B}}_I + \bar{\mathbf{B}}_I^\dagger)\mathbf{X}_I^\dagger \tag{2.49}$$

and $\overline{\mathbf{W}}$ is defined as

$$\left.\overline{W}_{\mu\nu}^I\right|_{\nu\in I} = 2\left(\mathbf{N}\tilde{\mathbf{B}}_I\right)_{\mu\nu} \tag{2.50}$$

Derivatives of OrbNet energy with respect to MOs

Define the derivatives of the OrbNet energy with respect to feature $\mathbf{f}$ as:

$$\mathbf{Q}^f = \frac{\partial E_{\mathrm{NN}}[\mathbf{f}]}{\partial \mathbf{f}} \tag{2.51}$$

where $\mathbf{f} \in \{\mathbf{F}, \mathbf{P}, \mathbf{D}, \mathbf{H}, \mathbf{S}\}$.

Note that $\mathbf{Q}^f$ has the same dimension as $\mathbf{f}$, and is symmetrized.

The derivatives of OrbNet energy with respect to the MO variations, Eq. 2.41, can be rewritten as

$$A_{pq} = \left.\frac{\partial E_{\mathrm{NN}}[\mathbf{f}]}{\partial V_{pq}}\right|_{\mathbf{V}=0} = \left.\frac{\partial E_{\mathrm{NN}}[\mathbf{f}]}{\partial \mathbf{f}}\frac{\partial \mathbf{f}}{\partial V_{pq}}\right|_{\mathbf{V}=0} = \sum_f \left[\mathbf{Q}^f \cdot \frac{\partial \mathbf{f}}{\partial V_{pq}}\right] \tag{2.52}$$

Define

$$A_{pq}^f = \mathbf{Q}^f \cdot \frac{\partial \mathbf{f}}{\partial V_{pq}} \tag{2.53}$$

which corresponds to the contribution to OrbNet energy derivatives with respect to MOs from a specific feature $\mathbf{f}$.

Derivatives of OrbNet energy with respect to nuclear coordinates

The derivatives of OrbNet energy with respect to nuclear coordinates can be written as

$$\frac{\partial E_{\mathrm{NN}}}{\partial x} = \frac{\partial E_{\mathrm{NN}}[\mathbf{f}]}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial x} = \sum_{f} \left[\mathbf{Q}^{f} \frac{\partial \mathbf{f}}{\partial x} \right] \tag{2.54}$$

Define

$$A_x^f = \mathbf{Q}^f \cdot \frac{\partial \mathbf{f}}{\partial x} \tag{2.55}$$

which corresponds to the contribution to OrbNet energy derivatives with respect to MOs from a specific feature **f**.

xTB generalized Fock matrix

The xTB generalized Fock matrix is defined as

$$(\mathbf{g}[\mathbf{Y}])_{\mu\nu} = \sum_{\kappa\lambda} \frac{\partial F_{\mu\nu}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} \tag{2.56}$$

where **Y** is an arbitrary symmetric matrix with the same dimension as the AO density matrix **P**.

The xTB Fock matrix is defined as

$$F_{\mu\nu} = H_{\mu\nu} + \frac{1}{2} S_{\mu\nu} \sum_{C,l''} (\gamma_{AC,ll''} + \gamma_{BC,l'l''}) p_{l''}^{C} + \frac{1}{2} S_{\mu\nu} (q_A^2 \Gamma_A + q_B^2 \Gamma_B) \quad (\mu \in A, l; \nu \in B, l') \tag{2.57}$$

which is a functional of the shell-resolved charges, i.e. $\mathbf{F}[p_{l''}^{C}]$.

With the above expression, the xTB generalized Fock matrix can be computed as

$$(\mathbf{g}[\mathbf{Y}])_{\mu\nu} = \sum_{\kappa\lambda} \frac{\partial F_{\mu\nu}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} = \sum_{C,l''} \sum_{\kappa\lambda} \frac{\partial F_{\mu\nu}}{\partial p_{l''}^{C}} \frac{\partial p_{l''}^{C}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} \tag{2.58}$$

The shell-resolved charges $p_{l''}^{C}$ are defined as

$$p_{l''}^{C} = p_{l''}^{C\,0} - \sum_{\kappa \in C, l''} \sum_{\lambda} S_{\kappa\lambda} P_{\kappa\lambda} \tag{2.59}$$

Define

$$\tilde{p}_{l''}^{C} \equiv \sum_{\kappa\lambda} \frac{\partial p_{l''}^{C}}{\partial P_{\kappa\lambda}} Y_{\kappa\lambda} = - \sum_{\kappa \in C, l''} \sum_{\lambda} S_{\kappa\lambda} Y_{\kappa\lambda} \tag{2.60}$$

The final expression for the xTB generalized Fock matrix is

$$
\begin{aligned}
(\mathbf{g}[\mathbf{Y}])_{\mu\nu} &= \sum_{C,l''}\sum_{\kappa\lambda}\frac{\partial F_{\mu\nu}}{\partial p^C_{l''}}\frac{\partial p^C_{l''}}{\partial P_{\kappa\lambda}}Y_{\kappa\lambda} = \sum_{C,l''}\frac{\partial F_{\mu\nu}}{\partial p^C_{l''}}\tilde{p}^C_{l''} \\
&= \frac{1}{2}S_{\mu\nu}\sum_{C,l''}(\gamma_{AC,ll''} + \gamma_{BC,l'l''})\tilde{p}^C_{l''} + S_{\mu\nu}(q_A\tilde{q}_A\Gamma_A + q_B\tilde{q}_B\Gamma_B) \qquad (2.61)
\end{aligned}
$$

where $\tilde{q}_A = \sum_l \tilde{p}^A_l$.

Coupled-perturbed z-vector equation for xTB

Combining the stationary condition of the Lagrangian, Eq. 2.39 and the condition $\mathbf{x} = \mathbf{x}^\dagger$ leads to the coupled-perturbed z-vector equation for xTB:

$$
(\varepsilon_a - \varepsilon_i)z_{ai} + 2[\mathbf{g}(\bar{\mathbf{z}})]_{ai} = -(A_{ai} - A_{ia}) \qquad (2.62)
$$

where $\varepsilon_a, \varepsilon_i$ are the xTB orbital energies, $\mathbf{z}$ is the Lagrange multiplier defined in Eq. 2.38. $\bar{\mathbf{z}} = \mathbf{z} + \mathbf{z}^\dagger$.

$\mathbf{g}(\bar{\mathbf{z}})$ is the generalized xTB Fock matrix and is defined in Eq. 2.61.

Expression for W

The stationary condition of the Lagrangian, Eq. 2.39 also leads to the expression for the weight matrix $\mathbf{W}$:

$$
W_{pq} = -\frac{1}{4}(1 + \hat{P}_{pq})[\mathbf{A} + \mathbf{A}(\mathbf{z})]_{pq} \qquad (2.63)
$$

where $\hat{P}_{pq}$ is the permutation operator that permutes indices p and q.

Final OrbNet gradient expression

With all intermediate quantities obtained in the previous sections, we can now write the expression for the OrbNet energy gradient:

$$
\frac{dE_{\text{out}}}{dx} = \frac{\partial E_{\text{out}}}{\partial x} + \mathrm{Tr}[\mathbf{W}\mathbf{S}^x] + \mathrm{Tr}[\mathbf{z}\mathbf{F}^{(x)}] \qquad (2.64)
$$

where the first term on the right-hand-side can be computed as

$$
\begin{aligned}
\frac{\partial E_{\text{out}}}{\partial x} &= \frac{dE_{\text{xTB}}}{dx} + \sum_{f}\left[\mathbf{Q}^{f}\frac{\partial \mathbf{f}}{\partial x}\right] && (2.65a)\\
&= \frac{dE_{\text{xTB}}}{dx} + \sum_{f}\left[\mathbf{Q}^{f}\frac{\partial \mathbf{f}}{\partial x}\right] + \text{Tr}[\mathbf{W}\mathbf{S}^{x}] + \text{Tr}[\mathbf{z}^{\text{AO}}\frac{\partial \mathbf{F}^{\text{AO}}}{\partial x}] && (2.65b)\\
&= \frac{dE_{\text{xTB}}}{dx} + \text{Tr}[\mathbf{W}\mathbf{S}^{x}] + \text{Tr}[\mathbf{z}^{\text{AO}}\mathbf{F}^{x}] && (2.65c)\\
&\quad + 2\text{Tr}\left[\overline{\mathbf{W}}^{H}\mathbf{S}^{x}\right] + \text{Tr}\left[\mathbf{Q}^{H,\text{AO}}\mathbf{H}^{x}\right]\\
&\quad + 2\text{Tr}\left[\overline{\mathbf{W}}^{S}\mathbf{S}^{x}\right] + \text{Tr}\left[\mathbf{Q}^{S,\text{AO}}\mathbf{S}^{x}\right]\\
&\quad + 2\text{Tr}\left[\overline{\mathbf{W}}^{F}\mathbf{S}^{x}\right] + \text{Tr}\left[\mathbf{Q}^{F,\text{AO}}\mathbf{F}^{x}\right]\\
&\quad + 2\text{Tr}\left[\overline{\mathbf{W}}^{P}\mathbf{S}^{x}\right] \quad + 4\text{Tr}\left[\overline{\mathbf{W}}^{D}\mathbf{S}^{x}\right] + 2\text{Tr}\left[\bar{\mathbf{d}}^{L}\cdot\mathbf{r}^{x}\right]
\end{aligned}
$$

The GFN-xTB gradient is written as [60]

$$
\frac{dE_{\text{xTB}}}{dx} = \text{Tr}[\mathbf{P}\mathbf{H}^{x}] + E^{x}_{\text{h2}} + E^{x}_{\text{h3}} \qquad (2.66)
$$

Chapter 3

EQUIVARIANT NEURAL NETWORKS FOR ORBITAL-BASED DEEP LEARNING

This chapter is based on the following publication:

[1] Zhuoran Qiao, Anders S. Christensen, Matthew Welborn, Frederick R. Manby, Animashree Anandkumar, and Thomas F. Miller III. "Informing geometric deep learning with electronic interactions to accelerate quantum chemistry". In: *Proceedings of the National Academy of Sciences* 119.31 (2022), e2205221119. DOI: `10.1073/pnas.2205221119`.

Abstract

Predicting electronic energies, densities, and related chemical properties can facilitate the discovery of novel catalysts, medicines, and battery materials. By developing a physics-inspired equivariant neural network, we introduce a method to learn molecular representations based on the electronic interactions among atomic orbitals. Our method, OrbNet-Equi, leverages efficient tight-binding simulations and learned mappings to recover high fidelity quantum chemical properties. OrbNet-Equi models a wide spectrum of target properties with an accuracy consistently better than standard machine learning methods and a speed several orders of magnitude greater than density functional theory. Despite only using training samples collected from readily available small-molecule libraries, OrbNet-Equi outperforms traditional methods on comprehensive downstream benchmarks that encompass diverse main-group chemical processes. Our method also describes interactions in challenging charge-transfer complexes and open-shell systems. We anticipate that the strategy presented here will help to expand opportunities for studies in chemistry and materials science, where the acquisition of experimental or reference training data is costly.

3.1 Introduction

Discovering new molecules and materials is central to tackling contemporary challenges in energy storage and drug discovery [101, 102]. As the experimentally uninvestigated chemical space for these applications is immense, large-scale computational design and screening for new molecule candidates has the potential to vastly reduce the burden of laborious experiments and to accelerate discovery [103–

105]. A crucial task is to model the quantum chemical properties of molecules by solving the many-body Schrödinger equation, which is commonly addressed by *ab initio* electronic structure methods [9, 106] such as density functional theory (DFT) (Figure 3.1a). While very successful, *ab initio* methods are laden with punitive computational requirements that makes it difficult to achieve a throughput on a scale of the unexplored chemical space.

In contrast, machine learning (ML) approaches are highly flexible as function approximators, and thus are promising for modelling molecular properties at a drastically reduced computational cost. A large class of ML-based molecular property predictors includes methods that use atomic-coordinate-based input features which closely resemble molecular mechanics (MM) descriptors [22, 40, 51–54, 107–111]; these methods will be referred to as Atomistic ML methods in the current work (Figure 3.1b). Atomistic ML methods have been employed to solve challenging problems in molecular sciences such as RNA structure prediction [6] and anomalous phase transitions [4]. However, there remains a key discrepancy between Atomistic ML and *ab initio* approaches regarding the modelling of quantum chemical properties, as Atomistic ML approaches typically neglect the electronic degrees of freedom which are central for the description of important phenomena such as electronic excitations, charge transfer, and long-range interactions. Moreover, recent work shows that Atomistic ML can struggle with transferability on downstream tasks where the molecules may chemically deviate from the training samples [64, 112] as is expected to be common for under-explored chemical spaces.

Recent efforts to embody quantum mechanics (QM) into molecular representations based on electronic structure theory have made breakthroughs in improving both the chemical and electronic transferability of ML-based molecular modelling [3, 33, 44, 47, 76, 113, 114]. Leveraging a physical feature space extracted from QM simulations, such QM-informed ML methods have attained data efficiency that significantly surpass Atomistic ML methods, especially when extrapolated to systems with length scales or chemical compositions unseen during training. Nevertheless, QM-informed ML methods still fall short in terms of the flexibility of modelling diverse molecular properties unlike their atomistic counterparts, as they are typically implemented for a limited set of learning targets such as the electronic energy or the exchange-correlation potential. A key bottleneck hampering the broader applicability of QM-informed approaches is the presence of unique many-body symmetries necessitated by an explicit treatment on electron-electron interactions. Heuristic schemes have been used to enforce invariance [43, 46, 47, 76, 115, 116] at a potential loss of information in their input features or expressivity in their ML models. Two objectives remain elusive for QM-informed machine learning: (a) incorporate the underlying physical symmetries with maximal data efficiency and model flexibility, and (b) accurately infer downstream molecular properties for large chemical spaces, at a computational resource requirement on par with existing empirical and Atomisic ML methods.

Herein, we introduce an end-to-end ML method for QM-informed molecular representations, OrbNet-Equi, in fulfillment of these two objectives. OrbNet-Equi featurizes a mean-field electronic structure via the atomic orbital basis, and learns molecular representations through a machine learning model that is equivariant with respect to isometric basis transformations (Figure 3.1c-e). By the virtue of equivariance, OrbNet-Equi respects essential physical constraints of symmetry conservation so that the target quantum chemistry properties are learned independent of a reference frame. Underpinning OrbNet-Equi is a neural network designed with insights from recent advances in geometric deep learning [7, 16, 117–121], but with key architectural innovations to achieve equivariance based on the tensor-space algebraic structures entailed in atomic-orbital-based molecular representations.

We demonstrate the data efficiency of OrbNet-Equi on learning molecular properties using input features obtained from tight-binding QM simulations which are efficient and scalable to systems with thousands of atoms [14]. We find that OrbNet-Equi consistently achieves lower prediction errors than existing Atomistic ML methods and our previous QM-informed ML method [76] on diverse target properties such as electronic energies, dipole moments, electron densities, and frontier orbital energies. Specifically, our study on learning frontier orbital energies illustrates an effective strategy to improve the prediction of electronic properties by incorporating molecular-orbital-space information.

To showcase its transferability to complex real-world chemical spaces, we trained an OrbNet-Equi model on single-point energies of 236k molecules curated from readily available small-molecule libraries. The resulting model, OrbNet-Equi/SDC21, achieves a performance competitive to state-of-the-art composite DFT methods when tested on a wide variety of main-group quantum chemistry benchmarks, while being up to thousand-fold faster at runtime. As a particular case study, we found that OrbNet-Equi/SDC21 substantially improved the prediction accuracy of ionization potentials relative to semi-empirical QM methods, even though no radical species was included for training. Thus, our method has the potential to accelerate simulations for challenging problems in organic synthesis [122], battery design [123], and molecular biology [124]. Detailed data analysis pinpoints viable future directions to systematically improve its chemical space coverage, opening a plausible pathway towards a generic hybrid physics-ML paradigm for the acceleration of molecular modelling and discovery.

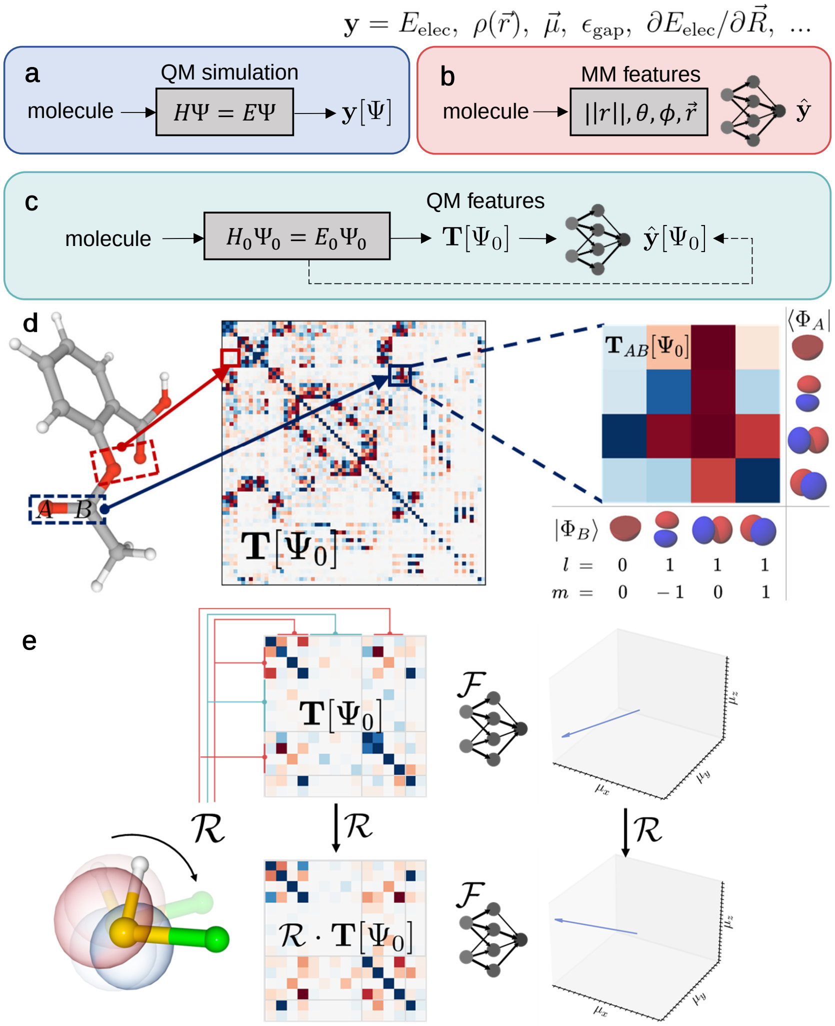

Figure 3.1: QM-informed machine learning for modelling molecular properties. (a) Conventional *ab initio* quantum chemistry methods predict molecular properties based on electronic structure theory through computing molecular wavefunctions and interaction terms, with general applicability but at high computational cost. (b) Atomistic machine learning approaches use geometric descriptors such as interatomic distances, angles, and directions to bypass the procedure of solving the electronic structure problem, but often requires vast amounts of data to generalize toward new chemical species. (c) In our approach, features are extracted from a highly coarse-grained QM simulation to capture essential physical interactions. An equivariant neural network efficiently learns the mapping, yielding improved transferability at an evaluation speed that is competitive to Atomistic ML methods. (d) Characteristics of the atomic orbital features considered in OrbNet-Equi. Every pair of atoms (A, B) is mapped to a block in the feature matrix, with the row dimension of the block matching the atomic orbitals of the source atom A and the column dimension matching the atomic orbitals of the destination atom B. (e) OrbNet-Equi is equivariant with respect to isometric basis transformations on the atomic orbitals (Equations 3.3-3.4), yielding consistent predictions (illustrated as the dipole moment vector of a HSF molecule) at different viewpoints.

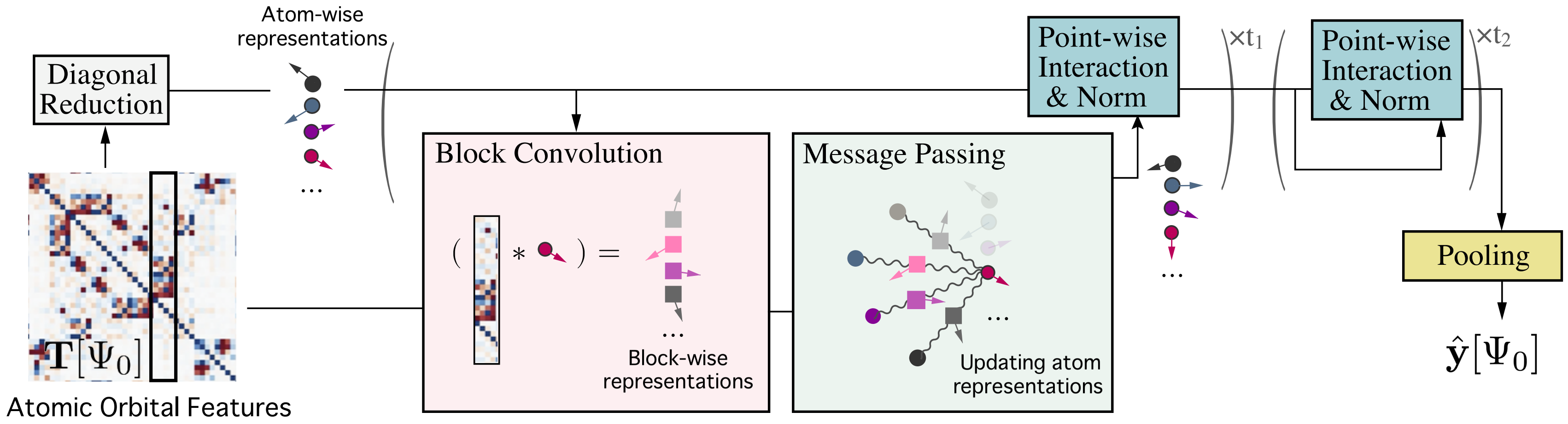
Diagonal Reduction
Atom-wise representations
...
$\mathbf{T}[\Psi_0]$
Atomic Orbital Features
Block Convolution
$(\; * \;) =$
...
Block-wise representations
Message Passing
...
Updating atom representations
Point-wise Interaction & Norm
...
×t₁
Point-wise Interaction & Norm
×t₂
Pooling
$\hat{\mathbf{y}}[\Psi_0]$

Figure 3.2: Schematic illustration of the OrbNet-Equi method. The input atomic orbital features $\mathbf{T}[\Psi_0]$ are obtained from a low-fidelity QM simulation. A neural network termed UNiTE first initializes atom-wise representations through the diagonal reduction module, and then updates the representations through stacks of block convolution, message passing, and point-wise interaction modules. A programmed pooling layer reads out high-fidelity property predictions $\hat{\mathbf{y}}$ based on the final representations. Neural network architecture details are provided in Methods 3.7.

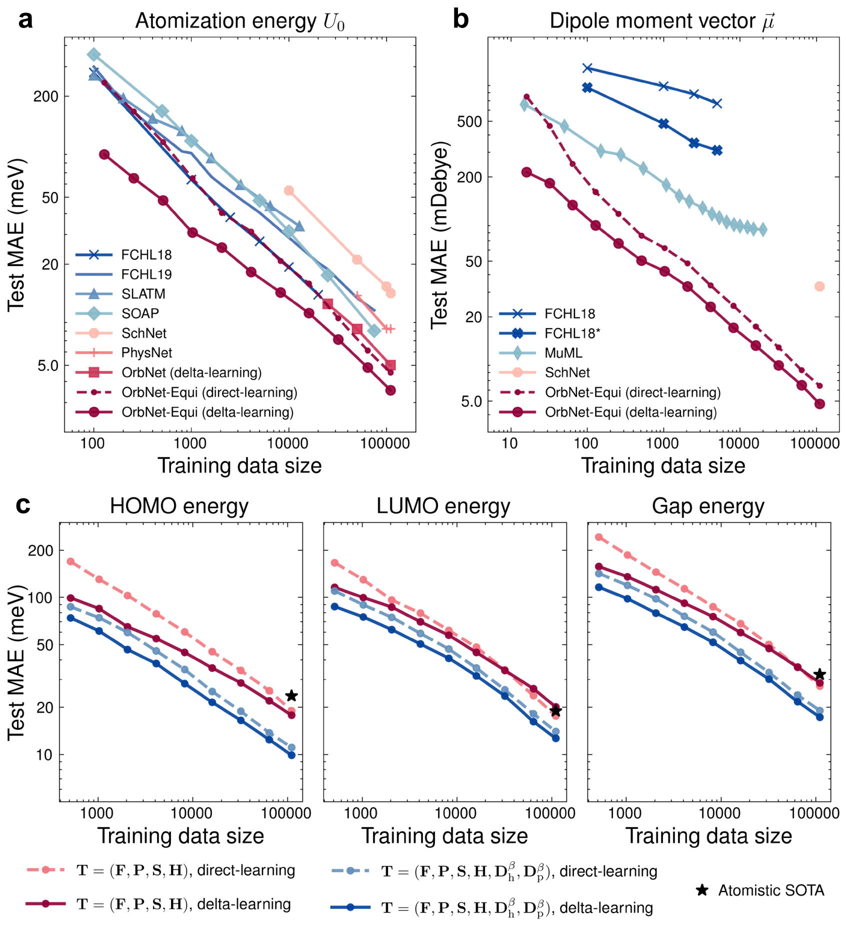

Figure 3.3: Model performance on the QM9 dataset. (a-b) Test mean absolute error (MAE) of OrbNet-Equi is shown as functions of the number of training samples, along with previously reported results from task-specific ML methods (FCHL18[125], FCHL19[126], SLATM[127], SOAP[128], FCHL18*[129], MuML[130]) and deep-learning-based methods (SchNet[51], PhysNet[52], OrbNet [76]) for targets (a) electronic energy U_0 and (b) molecular dipole moment vector $\vec{\mu}$ on the QM9 dataset. Results for OrbNet-Equi models trained with direct-learning and delta-learning are shown in dashed and solid lines, respectively. (c) Incorporating energy-weighted density matrices to improve data efficiency on learning frontier orbital properties. The HOMO, LUMO, and HOMO-LUMO gap energy test MAEs of OrbNet-Equi are shown as functions of the number of training samples. For models with the default feature set (red curves), the reduction in test MAE for delta-learning over direct-learning models gradually diminishes as the training data size grows. The LUMO and gap energy MAE curves exhibit a crossover around 32k-64k training samples, thereafter direct-learning models outperform delta-learning models. In contrast, when the energy-weighted density matrix features are supplied (blue curves), the test MAE curves between direct-learning and delta-learning models remain gapped when the training data size is varied. The black stars indicate the lowest test MAEs achieved by Atomistic ML methods (SphereNet [108]) trained with 110k samples.

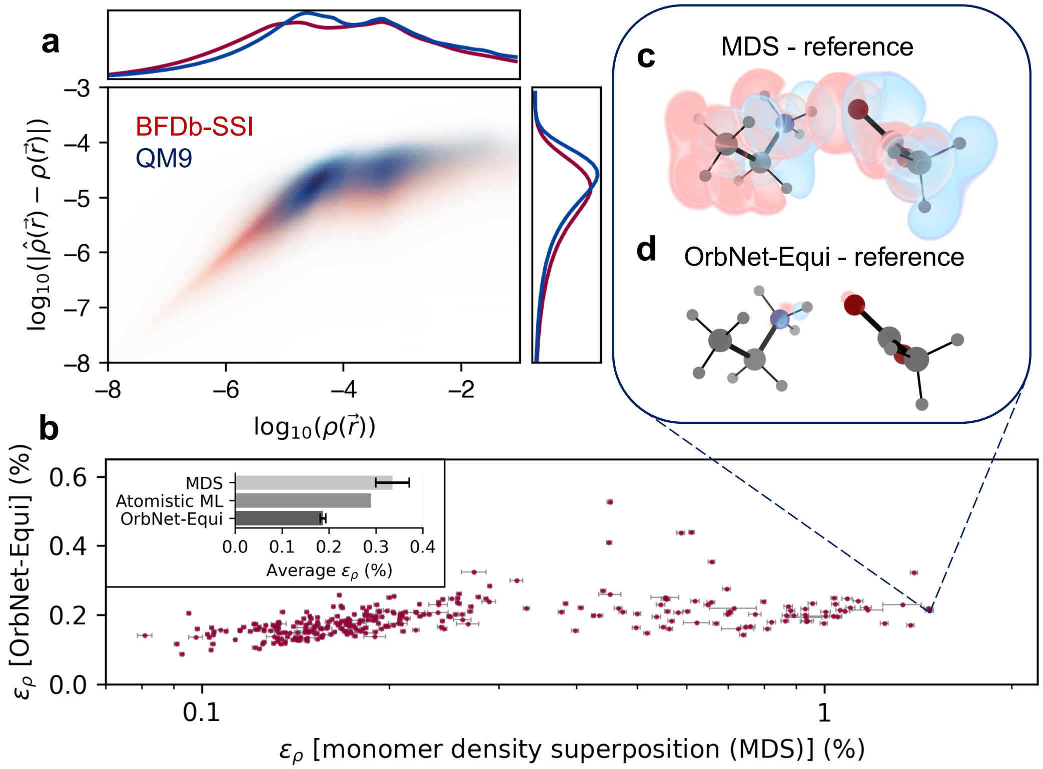

Figure 3.4: Learning electron charge densities for organic and biological motif systems. (a) 2D heatmaps of the log-scale reference density $\rho(\vec{r})$ and the log-scale OrbNet-Equi density prediction error $|\hat{\rho}(\vec{r}) - \rho(\vec{r})|$ (both in a_0^{-3}). The heatmaps are calculated by sampling real-space query points $\vec{r} \in \mathbb{R}^3$ for all molecules in the (red) BfDB-SSI test set and (blue) QM9 test set. The nearly-linear relationship for $\log_{10}(\rho(\vec{r})) < -4$ low-density regions reveals that OrbNet-Equi-predicted densities possess a physical long-range decay behavior. Distributions of $\log_{10}(\rho(\vec{r}))$ and $\log_{10}(|\hat{\rho}(\vec{r}) - \rho(\vec{r})|)$ are plotted within the marginal charts. (b) The L^1 density errors ε_ρ of OrbNet-Equi are plotted against the ε_ρ of densities obtained through monomer density superposition (MDS), across the BfDB-SSI test set. Error bars mark the 99% confidence intervals of ε_ρ for individual samples. The inset figure shows the average ε_ρ for MDS, an Atomistic ML method [131], and OrbNet-Equi predictions on the BfDB-SSI test set. OrbNet-Equi yields the lowest average prediction error and consistently produces accurate electron densities for cases where inter-molecular charge transfer is substantial. (c-d) Visualization of density deviation maps for (c) MDS and (d) OrbNet-Equi-predicted densities on the $\mathrm{Glu}^-/\mathrm{Lys}^+$ system (SSI-139GLU-144LYS-1), a challenging example from the BfDB-SSI test set. Red isosurfaces correspond to $\Delta\rho = -0.001\ a_0^{-3}$ and blue isosurfaces correspond to $\Delta\rho = +0.001\ a_0^{-3}$, where $\Delta\rho$ is the model density subtracted by the DFT reference density.

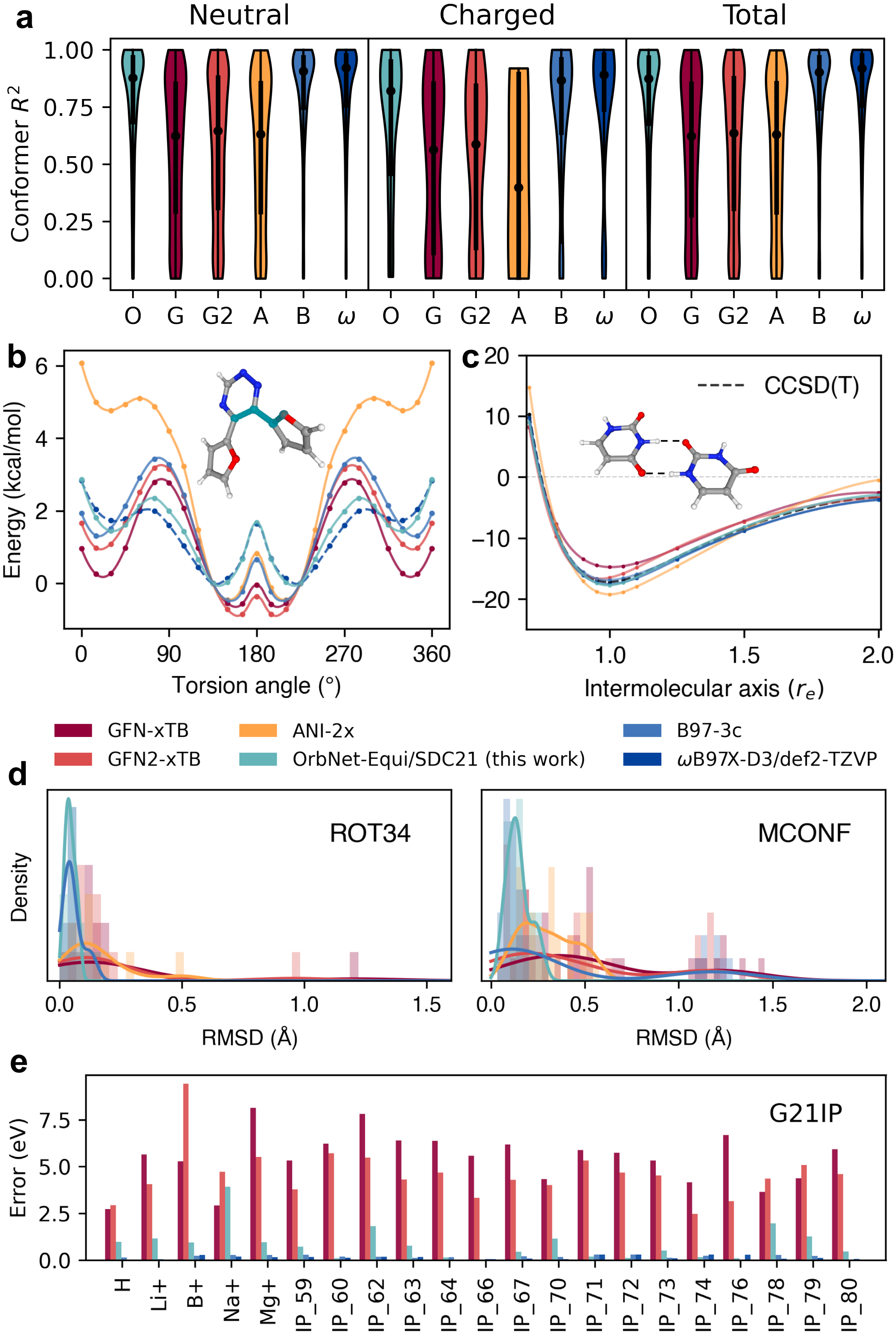
a
Neutral
Charged
Total
Conformer R^2
1.00
0.75
0.50
0.25
0.00
O G G2 A B ω O G G2 A B ω O G G2 A B ω
b
Energy (kcal/mol)
6
4
2
0
0 90 180 270 360
Torsion angle (°)
c
20
10
0
−10
−20
CCSD(T)
1.0 1.5 2.0
Intermolecular axis (r_e)
GFN-xTB
GFN2-xTB
ANI-2x
OrbNet-Equi/SDC21 (this work)
B97-3c
ωB97X-D3/def2-TZVP
d
Density
ROT34
0.0 0.5 1.0 1.5
RMSD (Å)
MCONF
0.0 0.5 1.0 1.5 2.0
RMSD (Å)
e
G21IP
Error (eV)
7.5
5.0
2.5
0.0
H
Li+
B+
Na+
Mg+
IP_59
IP_60
IP_62
IP_63
IP_64
IP_66
IP_67
IP_70
IP_71
IP_72
IP_73
IP_74
IP_76
IP_78
IP_79
IP_80

Figure 3.5: OrbNet-Equi/SDC21 infers diverse downstream properties.(a) Conformer energy ranking on the Hutchison dataset of drug-like molecules. The horizontal axis is labelled with acronyms indicating each method (O: OrbNet-Equi/SDC21 (this work); G: GFN-xTB; G2: GFN2-xTB; A: ANI-2x; B: B97-3c; ω: ωB97X-D3/def2-TZVP). The y-axis corresponds to the molecule-wise R^2 between predictions and the reference (DLPNO-CCSD(T)) conformer energies. Violin plots display the distribution of R^2 scores for each method over the (left) neutral, (middle) charged, and (right) all molecules from the Hutchison dataset. Medians and first/third quantiles are shown as black dots and vertical bars. (b) A torsion profiles example from the TorsionNet500 benchmark. All predicted torsion scans surfaces are aligned to the true global minima of the highest level of theory (ωB97X-D3/def2-TZVP) results, with spline interpolations. (c) A uracil-uracil base pair example for non-covalent interactions. The dimer binding energy curves are shown as functions of the intermolecular axis (r_e) where $r_e = 1.0$ corresponds to the distance of optimal binding energy. (d) Geometry optimization results on the (left) ROT34 and (right) MCONF datasets. Histograms and kernel density estimations of the symmetry-corrected RMSD scores (Methods 3.8) with respect to the reference DFT geometries are shown for each test dataset. (e) Evidence of zero-shot model generalization on radical systems. OrbNet-Equi/SDC21 yields prediction errors drastically lower than semi-empirical QM methods for adiabatic ionization potential on the G21IP dataset, achieving accuracy comparable to DFT on 7 out of 21 test cases.

3.2 Method

OrbNet-Equi featurizes a molecular system through mean-field QM simulations. Semi-empirical tight-binding models [14] are used through this study since they can be solved rapidly for both small-molecules and extended systems, which enables deploying OrbNet-Equi to large chemical spaces. In particular, we employ the recently reported GFN-xTB [60] QM model in which the mean-field electronic structure Ψ_0 is obtained through self-consistently solving a tight-binding model system (Figure 3.1c). Built upon Ψ_0, the inputs to the neural network comprises a stack of matrices $\mathbf{T}[\Psi_0]$ defined as single-electron operators $\hat{O}[\Psi_0]$ represented in the atomic orbitals (Figure 3.1d),

$$\big(\mathbf{T}[\Psi_0]\big)_{AB}^{n,l,m;n',l',m'} = \langle \Phi_A^{n,l,m} | \hat{O}[\Psi_0] | \Phi_B^{n',l',m'} \rangle \tag{3.1}$$

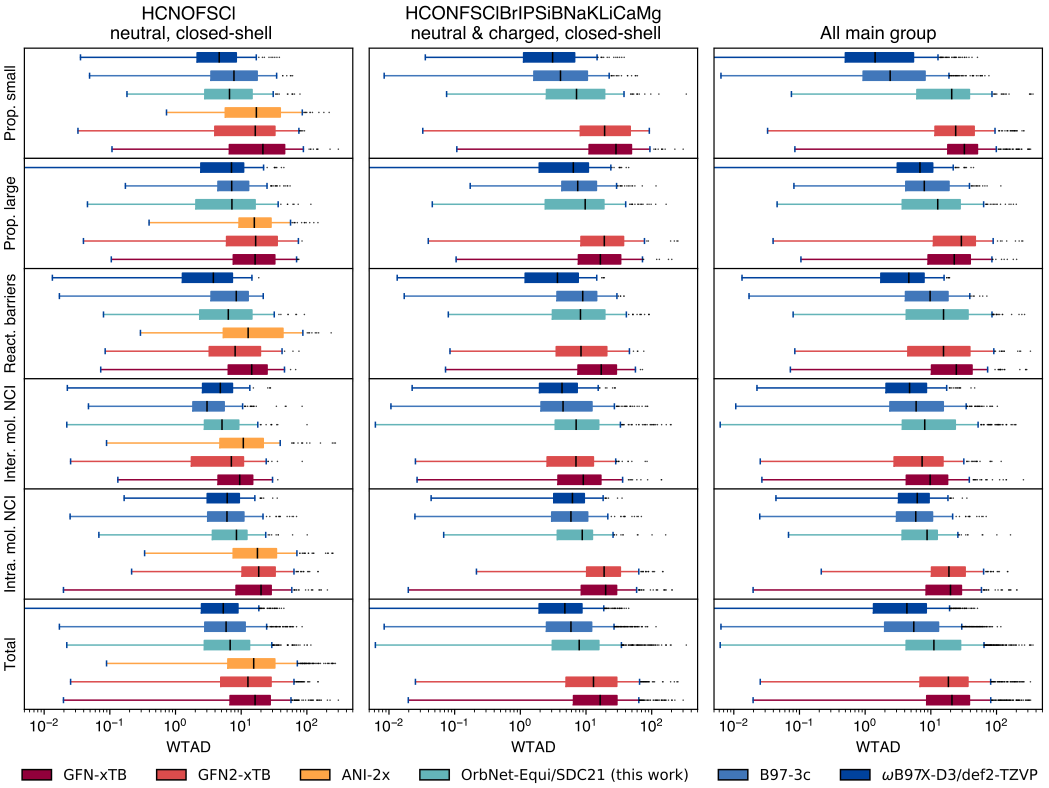

Figure 3.6: Assessing model performance on tasks from the GMTKN55 challenge. Box plots depict the distributions of task-difficulty-weighted absolution deviations (WTAD, see Methods 3.8) filtered by chemical elements and electronic states (a) supported by the ANI-2x model; (b) appeared in the dataset used for training OrbNet-Equi/SDC21; (c) all reactions. Statistics are categorized by each class of tasks in the GMTKN55 benchmark, as shown in y-axis labels. Prop. small: Basic properties and reaction energies for small systems; Prop. large: Reaction energies for large systems and isomerisation reactions; React. barriers: Reaction barrier heights; Inter. mol. NCI: Intermolecular noncovalent interactions; Intra. mol. NCI: Intramolecular noncovalent interactions; Total: total statistics of all tasks.

where A and B are both atom indices; (n, l, m) and (n', l', m') indicate a basis function in the set of atomic orbitals $\{\Phi\}$ centered at each atom. Motivated by mean-field electronic energy expressions, the input atomic orbital features are selected as $\mathbf{T} = (\mathbf{F}, \mathbf{P}, \mathbf{H}, \mathbf{S})$ using the Fock $\mathbf{F}$, density $\mathbf{P}$, core-Hamiltonian $\mathbf{H}$, and overlap $\mathbf{S}$ matrices of the tight-binding QM model (see Methods 3.7), unless otherwise specified.

OrbNet-Equi learns a map $\mathcal{F}$ to approximate the target molecular property $\mathbf{y}$ of high-fidelity electronic structure simulations or experimental measurements,

$$\min_{\mathcal{F}} \mathcal{L}\Big(\mathbf{y}, \mathcal{F}\big(\mathbf{T}[\Psi_0]\big)\Big) \tag{3.2}$$

where $\mathcal{L}$ denotes a cost functional between the reference and predicted targets over training data. The learning problem described by (3.2) requires a careful treatment on isometric coordinate transformations imposed on the molecular system, because the coefficients of $\mathbf{T}[\Psi_0]$ are defined up to a given viewpoint (Figure 3.1e). Precisely, the atomic orbitals $\{\Phi_A^{n,l,m}\}$ undergo a unitary linear recombination subject to 3D rotations: $\mathcal{R}\cdot|\Phi_A^{n,l,m}\rangle = \sum_{m'} \mathcal{D}^l_{m,m'}(\mathcal{R})|\Phi_A^{n,l,m'}\rangle$, where $\mathcal{D}^l(\mathcal{R})$ denotes the Wigner-D matrix of degree l for a rotation operation $\mathcal{R}$. As a consequence of the basis changing induced by $\mathcal{R}$, $\mathbf{T}[\Psi_0]$ is transformed block-wise:

$$\left(\mathcal{R}\cdot\mathbf{T}[\Psi_0]\right)_{AB}^{l;l'} = \mathcal{D}^l(\mathcal{R})\left(\mathbf{T}[\Psi_0]\right)_{AB}^{l;l'}\mathcal{D}^{l'}(\mathcal{R})^{\dagger} \tag{3.3}$$

where the dagger symbol denotes a Hermitian conjugate. To account for the roto-translation symmetries, the neural network $\mathcal{F}$ must be made equivariant with respect to all such isometric basis rotations, that is,

$$\mathcal{R}\cdot\mathcal{F}\big(\mathbf{T}[\Psi_0]\big) \equiv \mathcal{F}\big(\mathcal{R}\cdot\mathbf{T}[\Psi_0]\big) \tag{3.4}$$

which is fulfilled through our delicate design of the neural network in OrbNet-Equi (Figure 3.2). The neural network iteratively updates a set of representations $\mathbf{h}^t$ defined at each atom through its neural network modules, and reads out predictions using a pooling layer located at the end of the network. During its forward pass, diagonal blocks of the inputs $\mathbf{T}[\Psi_0]$ are first transformed into components that are isomorphic to orbital-angular-momentum eigenstates, which are then cast to the initial representations $\mathbf{h}^{t=0}$. Each subsequent module exploits off-diagonal blocks of $\mathbf{T}[\Psi_0]$ to propagate non-local information among atomic orbitals and refine the representations $\mathbf{h}^t$, which resembles a process of applying time-evolution operators on quantum states. We provide a technical introduction to the neural network architecture in Methods 3.7. We incorporate other constraints on the learning task such as size-consistency solely through programming the pooling layer (Methods 3.10), therefore achieving task-agnostic modelling for diverse chemical properties. Additional details and theoretical results are provided in Appendix 3.10-3.9.

3.3 Performance on benchmark datasets

We begin with benchmarking OrbNet-Equi on the QM9 dataset [65] which has been widely adopted for assessing ML-based molecular property prediction methods. QM9 contains 134k small organic molecules at optimized geometries, with target properties computed by DFT. Following previous works [51, 52, 108, 109, 119, 132], we take 110000 random samples as the training set and 10831 samples as the test set.

We present results for both the “direct-learning” training strategy which corresponds to training the model directly on the target property, and, whenever applicable, the “delta-learning” strategy [24] which corresponds to training on the residual between output of the tight-binding QM model and the target level of theory.

We first trained OrbNet-Equi on two representative targets, the total electronic energy U_0 and the molecular dipole moment vector $\vec{\mu}$ (Figure 3.3a-b), for which a plethora of task-specific ML models has previously been developed [76, 125, 128–130, 133]. The total energy U_0 is predicted through a sum over atom-wise energy contributions and the dipole moment $\vec{\mu}$ is predicted through a combination of atomic partial charges and dipoles (Appendix 3.10). For U_0 (Figure 3.3a), the direct-learning results of OrbNet-Equi match the state-of-the-art kernel-based ML method FCHL18/GPR [125] in terms of the test mean absolute error (MAE), while being scalable to large data regimes (Figure 3.3a, training data size > 20,000) where no competitive result has been reported before. With delta-learning, OrbNet-Equi outperforms our previous QM-informed ML approach OrbNet [76] by $\sim$ 45% in the test MAE. Because OrbNet also uses the GFN-xTB QM model for featurization and the delta-learning strategy for training, this improvement underscores the strength of our neural network design which seamlessly integrates the underlying physical symmetries. Moreover, for dipole moments $\vec{\mu}$ (Figure 3.3b), OrbNet-Equi exhibits steep learning curve slopes regardless of the training strategy, highlighting its capability of learning rotational-covariant quantities at no sacrifice of data efficiency.

We then targeted on the learning task of frontier molecular orbital (FMO) properties, in particular energies of the highest occupied molecular orbital (HOMO), the lowest unoccupied molecular orbital (LUMO) and the HOMO-LUMO gaps which are important in the prediction of chemical reactivity and optical properties [134, 135]. Because the FMOs are inherently defined in the electron energy space and are often spatially localized, it is expected to be challenging to predict FMO properties based on molecular representations in which a notion of electronic energy levels is absent. OrbNet-Equi overcame this obstacle by breaking the orbital filling degeneracy of its input features to encode plausible electron excitations near the FMO energy levels, that is, adding energy-weighted density matrices of ‘hole-excitation’ $\mathbf{D}_{\mathrm{h}}^{\beta}$ and that of ‘particle-excitation’ $\mathbf{D}_{\mathrm{p}}^{\beta}$:

$$(D_{\mathrm{h}}^{\beta})_{\mu\nu} = \sum_{i} C_{\mu i}^{*} C_{\nu i} \cdot \exp\big(-\beta(\epsilon_{\mathrm{HOMO}} - \epsilon_i)\big) \cdot n_i \tag{3.5}$$

$$(D_{\mathrm{p}}^{\beta})_{\mu\nu} = \sum_{i} C_{\mu i}^{*} C_{\nu i} \cdot \exp\big(\beta(\epsilon_{\mathrm{LUMO}} - \epsilon_i)\big) \cdot (1 - n_i) \tag{3.6}$$

where ϵ_i and n_i are the orbital energy and occupation number of the i-th molecular orbital from tight-binding QM, and $C_{\mu i}, C_{\nu i}$ denotes the molecular orbital coefficients with μ and ν indexing the atomic orbital basis. Here the effective temperature parameters β are chosen as $\beta = [4, 16, 64, 256]$ (atomic units), and a global-attention based pooling is used to ensure size-intensive predictions (Appendix 3.10). Figure 3.3c shows that the inclusion of energy-weighted density matrices $(\mathbf{D}_{\mathrm{h}}^{\beta}, \mathbf{D}_{\mathrm{p}}^{\beta})$ indeed greatly enhanced model generalization on FMO energies, as evident from the drastic test MAE reduction against the model with default ground-state features $(\mathbf{F}, \mathbf{P}, \mathbf{S}, \mathbf{H})$ as well as the best result from Atomistic ML methods. Remarkably, for models using default ground-state features (Figure 3.3c, red lines) we noticed a rank reversal behavior between direct-learning and delta-learning models as more training samples become available, mirroring similar observations from a recent Atomistic ML study [136]. The absence of this crossover when $(\mathbf{D}_{\mathrm{h}}^{\beta}, \mathbf{D}_{\mathrm{p}}^{\beta})$ are provided (Figure 3.3c, blue) suggests that the origin of such a learning slow-down is the incompleteness of spatially-degenerate descriptors, and the gap between delta-learning and direct-learning curves can be restored by breaking the energy-space degeneracy. This analysis reaffirms the role of identifying the dominant physical degrees of freedom in the context of ML-based prediction of quantum chemical properties, and is expected to benefit the modelling of relevant electrochemical and optical properties such as redox potentials.

Furthermore, OrbNet-Equi is benchmarked on 12 targets of QM9 using the 110k full training set (Appendix Table 3.1), for which we programmed its pooling layer to reflect the symmetry constraint of each target property (Appendix 3.10). We observed top-ranked performance on all targets with average test MAE around two-fold lower than atomistic deep learning methods. In addition, we tested OrbNet-Equi on fitting molecular potential energy surfaces by training on multiple configurations of a molecule (Appendix 3.10). Results (Appendix Table 3.2-3.3) showed that OrbNet-Equi obtained energy and force prediction errors that match state-of-the-art machine learning potential methods [137, 138] on the MD17 dataset [137, 139], suggesting that our method also efficiently generalizes over the conformation degrees of freedom apart from being transferable across the chemical space. These extensive benchmarking studies confirm that our strategy is consistently applicable to a wide range of molecular properties.

3.4 Accurate modeling for electron charge densities

We next focus on the task of predicting the electron density $\rho(\vec{r}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ which plays an essential role in both the formulation of DFT and in the interpretation of molecular interactions. It is also more challenging than predicting the energetic properties from a machine learning perspective, due to the need of preserving its real-space continuity and rotational covariance. OrbNet-Equi learns to output a set of expansion coefficients $\hat{d}_A^{nlm}$ to represent the predicted electron density $\hat{\rho}(\vec{r})$ through a density fitting basis set $\{\chi\}$ (Methods 3.8, Appendix 3.10),

$$\hat{\rho}(\vec{r}) = \sum_{A}^{N_{\text{atom}}} \sum_{l}^{l_{\max}(z_A)} \sum_{m=-l}^{l} \sum_{n}^{n_{\max}(z_A,l)} \hat{d}_A^{nlm} \, \chi_A^{nlm}(\vec{r}) \tag{3.7}$$

where $l_{\max}(z_A)$ is the maximum angular momentum in the density fitting basis set for atom type z_A, and $n_{\max}(z_A, l)$ denotes the cardinality of basis functions with angular momentum l. We train OrbNet-Equi to learn DFT electron densities on the QM9 dataset of small organic molecules and the BfDB-SSI [140] dataset of amino-acid side-chain dimers (Figure 3.4) using the direct-learning strategy. OrbNet-Equi results are substantially better than Atomistic ML baselines in terms of the average L^1 density error $\varepsilon_\rho = \frac{\int |\rho(\vec{r}) - \hat{\rho}(\vec{r})| d\vec{r}}{\int |\rho(\vec{r})| d\vec{r}}$ (Methods 3.8); specifically, OrbNet-Equi achieves an average ε_ρ of 0.191±0.003% on BfDB-SSI using 2000 training samples compared to 0.29% of SA-GPR [131], and an average ε_ρ of 0.206±0.001% on QM9 using 123835 training samples as compared to 0.28%-0.36% of DeepDFT [141]. Figure 3.4a confirms that OrbNet-Equi predicts densities at consistently low errors across the real-space and maintains a robust asymptotic decay behavior within low-density ($\rho(\vec{r}) < 10^{-4}\ a_0^{-3}$) regions that are far from the molecular system.

To understand whether the model generalizes to cases where charge transfer is significant, as in donor-acceptor systems, we introduce a simple baseline predictor termed monomer density superposition (MDS). The MDS electron density of a dimeric system is taken as the sum of independently-computed DFT electron densities of the two monomers. OrbNet-Equi yields accurate predictions in the presence of charge redistribution induced by non-covalent effects, as identified by dimeric examples from the BfDB-SSI test set for which the MDS density (Figure 3.4b, x-axis) largely deviates from the DFT reference density of the dimer due to intermolecular interactions. One representative example is a strongly interacting Glutamic acid - Lysine system (Figure 3.4, c-d) whose salt-bridge formation is known to be essential for the helical stabilization in protein folding [142], for which OrbNet-Equi predicts $\rho(\vec{r})$ with $\varepsilon_\rho = 0.211 \pm 0.001‰$ significantly lower than that of monomer density superposition ($\varepsilon_\rho = 1.47 \pm 0.02‰$). The accurate modelling of $\rho(\vec{r})$ offers an opportunity for constructing transferable DFT models for extended systems by learning on both energetics and densities, while at a small fraction of expense relative to solving the Kohn-Sham equations from scratch.

3.5 Transferability on downstream tasks

Beyond data efficiency on established datasets in train-test split settings, a crucial but highly challenging aspect is whether the model accurately infers downstream properties after being trained on data that are feasible to obtain. To comprehensively evaluate whether OrbNet-Equi can be transferred to unseen chemical spaces without any additional supervision, we have trained an OrbNet-Equi model on a dataset curated from readily available small-molecule databases (Methods 3.8). The training dataset contains 236k samples with chemical space coverage for drug-like molecules and biological motifs containing chemical elements C, O, N, F, S, Cl, Br, I, P, Si, B, Na, K, Li, Ca, and Mg, and thermalized geometries. The resulting OrbNet-Equi/SDC21 potential energy model is solely trained on DFT single-point energies using the delta-learning strategy. Without any fine-tuning, we directly apply OrbNet-Equi/SDC21 to downstream benchmarks that are recognized for assessing the accuracy of physics-based molecular modelling methods.

The task of ranking conformer energies of drug-like molecules is benchmarked via the Hutchison dataset of conformers of ~700 molecules [64] (Figure 3.5a; Table 3.5, row 1-2). On this task, OrbNet-Equi/SDC21 achieves a median R^2 score of 0.87±0.02 and R^2 distributions closely matching the reference DFT theory on both neutral and charged systems. On the other hand, we notice that the median R^2 of OrbNet-Equi/SDC21 with respect to the reference DFT theory (ωB97X-D3/def2-TZVP) is 0.96±0.01, suggesting that the current performance on this task is saturated by the accuracy of DFT and can be systematically improved by applying fine-tuning techniques on higher-fidelity labels [143, 144]. Timing results on the Hutchison dataset (Table 3.4) confirms that the neural network inference time of OrbNet-Equi/SDC21 is on par with the GFN-xTB QM featurizer, resulting in an overall computational speed that is 10^{2-3} fold faster relative to existing cost-efficient composite DFT methods [64, 96, 145]. To understand the model's ability to describe dihedral energetics which are crucial for virtual screening tasks, we benchmark OrbNet-Equi on the prediction of intra-molecular torsion energy profiles using the

TorsionNet500 [146] dataset, the most diverse benchmark set available for this problem (Table 3.5, row 3). Although no explicit torsion angle sampling was performed during training data generation, OrbNet-Equi/SDC21 exhibits a barrier MAE of 0.173±0.003 kcal/mol much lower than the 1 kcal/mol threshold commonly considered for chemical accuracy. On the other hand, we notice a MAE of 0.7 kcal/mol for the TorsionNet model [146] which was trained on ∼1 million torsion energy samples. As shown in Figure 3.5b, OrbNet-Equi/SDC21 robustly captures the torsion sectors of potential energy surface on an example challenging for both semi-empirical QM [60, 62] and cost-efficient composite DFT [96] methods, precisely resolving both the sub-optimal energy minima location at ∼ 30° dihedral angle as well as the barrier energy between two local minimas within a 1 kcal/mol chemical accuracy. Next, the ability to characterize non-covalent interactions is assessed on the S66x10 dataset [147] of inter-molecular dissociation curves (Table 3.6), on which OrbNet-Equi achieves an equilibrium-distance binding energy MAE of 0.35 ± 0.09 kcal/mol with respect to the reference DFT theory compared against 1.55 ± 0.17 kcal/mol of the GFN-xTB baseline. As shown from a Uracil-Uracil base-pair example (Figure 3.5c) for which high-fidelity wavefunction-based reference calculations have been reported, the binding energy curve along the inter-molecular axis predicted by OrbNet-Equi/SDC21 agrees well with both DFT and the high-level CCSD(T) results. To further understand the accuracy and smoothness of the energy surfaces and the applicability on dynamics tasks, we perform geometry optimizations on the ROT34 dataset of 12 small organic molecules and the MCONF dataset of 52 conformers of melatonin [70, 71] (Figure 3.5d; Table 3.5, row 4-5). Remarkably, OrbNet-Equi/SDC21 consistently exhibits the lowest average RMSD among all physics-based and ML-based approaches (Table 3.5) including the popular cost-efficient DFT method B97-3c [96]. Further details regarding the numerical experiments and error metrics are provided in Methods 3.8.

Remarkably, on the G21IP dataset [148] of adiabatic ionization potentials, we find that the OrbNet-Equi/SDC21 model achieves prediction errors substantially lower than semi-empirical QM methods (Figure 3.5e, Table 3.7) even though samples of open-shell signatures are expected to be rare from the training set (Methods 3.8). Such an improvement cannot be solely attributed to structure-based corrections, since there is no or negligible geometrical changes between the neutral and ionized species for both the single-atom systems and several poly-atomic systems (e.g., IP_66, a Phosphanide anion) in the G21IP dataset. This reveals that our method has the potential to be transferred to unseen electronic states in a zero-shot manner, which represents an early evidence that a hybrid physics-ML strategy may unravel novel chemical processes such as unknown electron-catalyzed reactions [149].

To comprehensively study the transferability of OrbNet-Equi on complex under-explored main-group chemical spaces, we evaluate OrbNet-Equi/SDC21 on the challenging, community-recognized benchmark collection of the General Thermochemistry, Kinetics, and Non-covalent Interactions (GMTKN55)[150] datasets (Figure 3.6). Prediction error statistics on the GMTKN55 benchmark are reported with three filtration schemes. First, we evaluate the WTMAD error metrics (Methods 3.8) on reactions that only consist of neutral and closed-shell molecules with chemical elements CHONFSCl (Figure 3.6a), as is supported by an Atomistic-ML-based potential method, ANI-2x [151], which is trained on large-scale DFT data. OrbNet-Equi/SDC21 predictions are found to be highly accurate on this subset, as seen from the WTMAD with respect to CCSD(T) being on par with the DFT methods on all five reaction classes and significantly outperforming ANI-2x and the GFN family of semi-empirical QM methods [60, 62]. It is worth noting that OrbNet-Equi/SDC21 uses much fewer number of training samples than the ANI-2x training set, which signifies the effectiveness of combining physics-based and ML-based modelling.

The second filtration scheme includes reactions that consist of closed-shell — but can be charged — molecules with chemical elements that have appeared in the SDC21 training dataset (Figure 3.6b). Although all chemical elements and electronic configurations in this subset are contained in the training dataset, we note that unseen types of physical interactions or bonding types are included, such as in alkali metal clusters from the ALK8 subset [150] and short strong hydrogen bonds in the AHB21 subset [152]. Therefore, assessments of OrbNet-Equi with this filtration strategy reflect its performance on cases where examples of atom-level physics are provided but the chemical compositions are largely unknown. Despite this fact, the median WTMADs of OrbNet-Equi/SDC21 are still competitive with DFT methods on the tasks of small-system properties, large-system properties and intra-molecular interactions. On reaction barriers and inter-molecular non-covalent interactions (NCIs), OrbNet-Equi/SDC21 results fall behind DFT but still show improvements against the GFN-xTB baseline and match the accuracy of GFN2-xTB which is developed with physic-based schemes to improve the descriptions on NCI properties against its predecessor GFN-xTB.

The last scheme includes all reactions in the GMTKN55 benchmarks containing chemical elements and spin states never seen during training (Figure 3.6c), which represents on the most stringent test and reflects the performance of OrbNet-Equi/SDC21 when being indiscriminately deployed as a quantum chemistry method. When evaluated on the collection of all GMTKN55 tasks (Figure 3.6, 'Total' panel), OrbNet-Equi/SDC21 maintains the lowest median WTMAD among methods considered here that can be executed at the computational cost of semi-empirical QM calculations. Moreover, we note that failure modes on a few highly extrapolative subsets can be identified to diagnose cases that are challenging for the QM model used for featurization (Table 3.7). For example, the fact that predictions being inaccurate on the W4-11 subset of atomization energies [153] and the G21EA subset of electron affinities [148] parallels the absence of an explicit treatment of triplet or higher-spin species within the formulation of GFN family of tight-binding models. On the population level, the distribution of prediction WTMADs across GMTKN55 tasks also differ from that of GFN2-xTB, which implies that further incorporating physics-based approximations into the QM featurizer can complement the ML model, and thus the accuracy boundary of semi-empirical methods can be pushed to a regime where no known physical approximation is feasible.

3.6 Discussion

We have introduced OrbNet-Equi, a QM-informed geometric deep learning framework for learning molecular or material properties using representations in the atomic orbital basis. OrbNet-Equi shows excellent data efficiency for learning related to both energy-space and real-space properties, expanding the diversity of molecular properties that can be modelled by QM-informed machine learning. Despite only using readily available small-molecule libraries as training data, OrbNet-Equi offers an accuracy alternative to DFT methods on comprehensive main-group quantum chemistry benchmarks at a computation speed on par with semi-empirical methods, thus offering a possible replacement for conventional *ab initio* simulations for general-purpose downstream applications. For example, OrbNet-Equi could immediately facilitate applications such as screening electrochemical properties of electrolytes for the design of flow batteries [123], and performing accurate direct or hybrid QM/MM simulations for reactions in transition-metal catalysis [122, 154]. The method can also improve the modelling for complex reactive biochemical processes [11] using multi-scale strategies that have been demonstrated in our previous study [124], while conventional *ab initio* reference calculations can be prohibitively expensive even on a minimal sub-system.

The demonstrated transferability of OrbNet-Equi to seemingly dissimilar chemical species identifies a promising future direction of improving the accuracy and chemical space coverage through adding simple model systems of the absent types of physical interactions to the training data, a strategy that is consistent with using synthetic data to improve ML models [155] which has been demonstrated for improving the accuracy of DFT functionals [3]. Additionally, OrbNet-Equi may provide valuable perspectives for the development of physics-based QM models by relieving the burden of parameterizing Hamiltonian parameters against specific target systems, potentially expanding their design space to higher energy-scales without sacrificing model accuracy. Because the framework presented here can be readily extended to alternative quantum chemistry models for either molecular or material systems, we expect OrbNet-Equi to broadly benefit studies in chemistry, materials science, and biotechnology.

3.7 The UNiTE neural network architecture

This section introduces Unitary N-body Tensor Equivariant Network (UNiTE), the neural network model developed for the OrbNet-Equi method to enable learning equivariant maps between the input atomic orbital features $\mathbf{T}[\Psi_0]$ and the property predictions $\hat{\mathbf{y}}[\Psi_0]$. Given the inputs $\mathbf{T}$, UNiTE first generates initial representations $\mathbf{h}^{t=0}$ through its diagonal reduction module (Methods 3.7). Then UNiTE updates the representations $\mathbf{h}^{t=0} \mapsto \mathbf{h}^{t=1} \mapsto \cdots \mapsto \mathbf{h}^{t=t_f}$ with t_1 stacks of block convolution (Methods 3.7), message passing (Methods 3.7), and point-wise interaction (Methods 3.7) modules, followed by t_2 stacks of point-wise interaction modules. A pooling layer (Methods 3.7) outputs predictions $\hat{\mathbf{y}}$ using the final representations $\mathbf{h}^{t=t_f}$ at $t_f = t_1 + t_2$ as inputs.

$\mathbf{h}^t$ is a stack of atom-wise representations, i.e., for a molecular system containing d atoms, $\mathbf{h}^t := [\mathbf{h}^t_1, \mathbf{h}^t_2, \cdots, \mathbf{h}^t_d]$. The representation for the A-th atom, $\mathbf{h}^t_A$, is a concatenation of neurons which are associated with irreducible representations of group $\mathrm{O}(3)$. Each neuron in $\mathbf{h}^t_A$ is identified by a channel index $n \in \{1, 2, \cdots, n_{\max}\}$, a "degree" index $l \in \{0, 1, 2, \cdots, l_{\max}\}$, and a "parity" index $p \in \{+1, -1\}$. The neuron $\mathbf{h}^t_{A,nlp}$ is a vector of length $2l + 1$ and transforms as the l-th irreducible representation of group SO(3); i.e., $\mathbf{h}^t_{A,nlp} = \bigoplus_m h^t_{A,nlpm}$ where $\oplus$ denotes a vector concatenation operation and $m \in \{-l, -l+1, \cdots, l-1, l\}$. We use N_{lp} to denote the number of neurons with degree l and parity p in $\mathbf{h}^t$, and $N := \sum_{l,p} N_{lp}$ to denote the total number of neurons in $\mathbf{h}^t$.

For a molecular/material system with atomic coordinates $\mathbf{x} \in \mathbb{R}^{d\times 3}$, the following equivariance properties with respect to isometric Euclidean transformations are fulfilled for any input gauge-invariant and Hermitian operator $\mathcal{O}[\Psi_0]$; for all allowed indices A, n, l, p, m:

- Translation invariance:

$$\mathbf{h}^t_{A,nlpm} \mapsto \mathbf{h}^t_{A,nlpm} \quad \text{for} \quad \mathbf{x} \mapsto \mathbf{x} + \mathbf{x}_0 \tag{3.8}$$

  where $\mathbf{x}_0 \in \mathbb{R}^3$ is an arbitrary global shift vector;

- Rotation equivariance:

$$\mathbf{h}^t_{A,nlpm} \mapsto \sum_{m'} \mathbf{h}^t_{A,nlpm'} \mathcal{D}^l_{m,m'}(\alpha, \beta, \gamma) \tag{3.9}$$

  for $\mathbf{x} \mapsto \mathbf{x} \cdot \mathcal{R}(\alpha, \beta, \gamma)$ where $\mathcal{R}(\alpha, \beta, \gamma)$ denotes a rotation matrix corresponding to standard Euler angles α, β, γ;

- Parity inversion equivariance:

$$\mathbf{h}^t_{A,nlpm} \mapsto (-1)^l \cdot p \cdot \mathbf{h}^t_{A,nlpm} \quad \text{for} \quad \mathbf{x} \mapsto -\mathbf{x} \tag{3.10}$$

The initial vector representations $\mathbf{h}^{t=0}$ are generated by decomposing diagonal sub-tensors of the input $\mathbf{T}$ into a spherical-tensor representation without explicitly solving tensor factorization, based on the tensor product property of group SO(3). The intuition behind this operation is that the diagonal sub-tensors of $\mathbf{T}$ can be viewed as isolated systems interacting with an effective external field, whose rotational symmetries are described by the Wigner-Eckart Theorem [15] which links tensor operators to their spherical counterparts and applies here within a natural generalization. Each update step $\mathbf{h}^t \mapsto \mathbf{h}^{t+1}$ is composed of (a) block convolution, (b) message passing, and (c) point-wise interaction modules which are all equivariant with respect to index permutations and basis transformations. In an update step $\mathbf{h}^t \mapsto \mathbf{h}^{t+1}$, each off-diagonal block of $\mathbf{T}$ corresponding to a pair of atoms is contracted with $\mathbf{h}^t$. This block-wise contraction operation can be interpreted as performing local convolutions using the blocks of $\mathbf{T}$ as convolution kernels, and therefore is called *block convolution* module. The output block-wise representations are then passed into a *message passing* module, which is analogous to a message-passing operation on edges in graph neural networks [156]. The message passing outputs are then fed into a *point-wise interaction* module with the previous-step representation $\mathbf{h}^t$ to finish the update $\mathbf{h}^t \mapsto \mathbf{h}^{t+1}$. The point-wise interaction modules are constructed as a stack of multi-layer perceptrons (MLPs), Clebsch-Gordan product operations, and skip connections. Within those modules, a *matching layer* assigns the channel indices of $\mathbf{h}^t$ to indices of the atomic orbital basis.

We also introduce a normalization layer termed Equivariant Normalization (EvNorm, see Methods 3.7) to improve training and generalization of the neural network. EvNorm normalizes scales of the representations $\mathbf{h}$, while recording the direction-like information to be recovered afterward. EvNorm is fused with a point-wise interaction module by first applying EvNorm to the module inputs, then using an MLP to transform the normalized frame-invariant scale information, and finally by multiplying the recorded direction vector to the MLP's output. Using EvNorm within the point-wise interaction modules is found to stabilize training and eliminate the need for tuning weight initializations and learning rates across different tasks.

The explicit expressions for the neural network modules are provided for quantum operators O being one-electron operators and therefore the input tensors $\mathbf{T}$ is a stack of matrices (i.e., order-2 tensors). Without loss of generality, we also assume that $\mathbf{T}$ contains only one feature matrix. Additional technical aspects regarding the case of multiple input features, the inclusion of geometric descriptors, and implementation details are discussed in Appendix 3.10. The proofs regarding equivariance and theoretical generalizations to order-N tensors are provided in Appendix 3.9.

The Diagonal Reduction module

We define the shorthand notations $\mu := (n_1, l_1, m_1)$ and $\nu := (n_2, l_2, m_2)$ to index atomic orbitals. The initialization scheme for $\mathbf{h}^{t=0}$ is based on the following proposition: for each diagonal block of $\mathbf{T}$, $\mathbf{T}_{AA}$, defined for an on-site atom pair (A, A),

$$T_{AA}^{\mu,\nu} := \langle \Phi_A^{\mu} | \hat{O} | \Phi_A^{\nu} \rangle \tag{3.11}$$

there exists a set of $\mathbf{T}$-independent coefficients $Q_{nlpm}^{\mu,\nu}$ such that the following linear transformation ψ

$$\psi(\mathbf{T}_{AA})_{nlpm} := \sum_{\mu,\nu} T_{AA}^{\mu,\nu} \, Q_{nlpm}^{\mu,\nu} \tag{3.12}$$

is injective and yields $\mathbf{h}_A := \psi(\mathbf{T}_{AA})$ that satisfy equivariance ((3.8)-(3.10)).

The existence of $\mathbf{Q}$ is discussed in Appendix, Corollary 3. For the sake of computational feasibility, a physically-motivated scheme is employed to tabulate $\mathbf{Q}$ and produce order-1 equivariant embeddings $\mathbf{h}_A$, using *on-site 3-index overlap integrals*

$\tilde{\mathbf{Q}}$:

$$\begin{aligned}\tilde{Q}^{\mu,\nu}_{nlm} &:= \tilde{Q}^{n_1,l_1,m_1;n_2,l_2,m_2}_{nlm} \\ &= \int_{\mathbf{r}\in\mathbb{R}^3} (\Phi_A^{n_1,l_1,m_1}(\mathbf{r}))^* \Phi_A^{n_2,l_2,m_2}(\mathbf{r}) \tilde{\Phi}_A^{n,l,m}(\mathbf{r}) d\mathbf{r} \end{aligned} \tag{3.13}$$

where Φ_A are the atomic orbital basis, and $\tilde{\Phi}_A$ are auxiliary Gaussian-type basis functions defined as (for conciseness, at $\mathbf{x}_A = 0$):

$$\tilde{\Phi}^{n,l,m}(\mathbf{r}) := c_{n,l} \cdot \exp(-\gamma_{n,l} \cdot r^2)\, r^l\, Y_{lm}(\frac{\mathbf{r}}{r}) \tag{3.14}$$

where $c_{n,l}$ is a normalization constant such that $\int_{\mathbf{r}} ||\tilde{\Phi}_A^{n,l,m}(\mathbf{r}))||^2 d\mathbf{r} = 1$ following standard conventions [157]. For numerical experiments considered in this work the scale parameters γ are chosen as (in atomic units):

$$\begin{aligned} \gamma_{n,l=0} &:= 128 \cdot (0.5)^{n-1} \quad \text{where} \quad n \in \{1, 2, \cdots, 16\} \\ \gamma_{n,l=1} &:= 32 \cdot (0.25)^{n-1} \quad \text{where} \quad n \in \{1, 2, \cdots, 8\} \\ \gamma_{n,l=2} &:= 4.0 \cdot (0.25)^{n-1} \quad \text{where} \quad n \in \{1, 2, 3, 4\} \end{aligned}$$

$\tilde{\mathbf{Q}}$ adheres to equivariance constraints due to its relation to SO(3) Clebsch-Gordan coefficients $C^{lm}_{l_1m_1;l_2m_2} \propto \int_{\mathbf{r}\in\mathbb{S}^2} Y_{l_1m_1}(\mathbf{r}) Y_{l_2m_2}(\mathbf{r}) (Y_{lm}(\mathbf{r}))^* d\mathbf{r}$ [15]. Note that the auxiliary basis $\tilde{\Phi}_A$ is independent of the atomic numbers, and thus the resulting $\mathbf{h}_A$ are of equal length for all chemical elements. $\tilde{\mathbf{Q}}$ can be efficiently generated using electronic structure programs, here done with [93]. The resulting $\mathbf{h}_A$ in explicit form are

$$\begin{aligned} \mathbf{h}_A &:= \bigoplus_{n,l,p,m} h_{A,nlpm} \quad \text{where} \\ h_{A,nl(p=+1)m} &= \sum_{\mu,\nu} T^{\mu,\nu}_{AA}\, \tilde{Q}^{\mu,\nu}_{nlm} \\ h_{A,nl(p=-1)m} &= 0 \end{aligned}$$

$\mathbf{h}_A$ are then projected by learnable linear weight matrices such that the number of channels for each (l, p) matches the model specifications. The outputs are regarded as the initial representations $\mathbf{h}^{t=0}$ to be passed into other modules.

The Block Convolution module

In an update step $\mathbf{h}^t \mapsto \mathbf{h}^{t+1}$, sub-blocks of $\mathbf{T}$ are first contracted with a stack of linearly-transformed order-1 representations $\mathbf{h}^t$.

$$\mathbf{m}^{t,i}_{AB,\nu} = \sum_{\mu} \left(\rho_i(\mathbf{h}^t_A)\right)_\mu T^{\mu,\nu}_{AB} \tag{3.15}$$

which can be viewed as a 1D convolution between each block $\mathbf{T}_{AB}$ (as convolution kernels) and the $\rho(\mathbf{h}^t_A)$ (as the signal) in the i-th channel where $i \in \{1, 2, \cdots, I\}$ is the convolution channel index. The block convolution produces block-wise representations $\mathbf{m}^t_{AB}$ for each block index (A, B). ρ_i is called a *matching layer* at atom A and channel i, defined as:

$$\big(\rho_i(\mathbf{h}^t_A)\big)_\mu = \mathrm{Gather}\big(\mathbf{W}^i_l \cdot (\mathbf{h}^t_A)_{l(p=+1)m}, n[\mu, z_A]\big) \tag{3.16}$$

$\mathbf{W}^i_l \in \mathbb{R}^{M_l \times N_{l,+1}}$ are learnable linear weight matrices specific to each degree index l, where M_l is the maximum principle quantum number for shells of angular momentum l within the atomic orbital basis used for featurization. The Gather operation maps the feature dimension to valid atomic orbitals by indexing $\mathbf{W}^i_l \cdot (\mathbf{h}^t_A)_{l(p=+1)m}$ using $n[\mu, z_A]$, the principle quantum numbers of atomic orbitals μ for atom type z_A.

The Message Passing module

Block-wise representations $\mathbf{m}^t_{AB}$ are then aggregated into each atom index A by summing over the indices B, analogous to a 'message-passing' between nodes and edges in common realizations of graph neural networks [156],

$$\tilde{\mathbf{m}}^t_A = \sum_B \bigoplus_{i,j} \mathbf{m}^{t,i}_{BA} \cdot \alpha^{t,j}_{AB} \tag{3.17}$$

up to a non-essential symmetrization and inclusion of point-cloud geometrical terms ((3.66)). $\alpha^{t,j}_{AB}$ in (3.17) are scalar-valued weights parameterized as *SE(3)-invariant multi-head attentions*:

$$\boldsymbol{\alpha}^t_{AB} = \mathrm{MLP}\big((\mathbf{z}^t_{AB} \cdot \mathbf{W}^t_\alpha) \odot \boldsymbol{\kappa}(||\mathbf{T}_{AB}||)/\sqrt{N}\big) \tag{3.18}$$

where $\odot$ denotes an element-wise (Hadamard) product, and

$$\mathbf{z}^t_{AB} = \bigoplus_{n,l,p} \sum_{m=-l}^{l} h^t_{A,nlpm} \cdot h^t_{B,nlpm} \tag{3.19}$$

where MLP denotes a 2-layer MLP, $\mathbf{W}^t_\alpha$ are learnable linear functions and $j \in \{1, 2, \cdots, J\}$ denotes an attention head (one value in $\boldsymbol{\alpha}^t_{AB}$). $\boldsymbol{\kappa}(\cdot)$ is chosen as Morlet wavelet basis functions:

$$\boldsymbol{\kappa}(||\mathbf{T}_{AB}||) := \mathbf{W}_\kappa\Big(\bigoplus_k \sum_{n,l} \sum_{n',l'} \xi_k(\log\big(||\mathbf{T}^{n,l;n',l'}_{AB}||)\big)\Big) \tag{3.20}$$

$$\xi_k(x) := \exp(-\gamma_k \cdot x^2) \cdot \cos(\pi\gamma_k \cdot x) \tag{3.21}$$

where $\mathbf{W}^t_\kappa$ are learnable linear functions and γ_k are learnable frequency coefficients initialized as $\gamma_k = 0.3 \cdot (1.08)^k$ where $k \in \{0, 1, \cdots, 15\}$. Similar to the scheme proposed in SE(3)-transformers [121], the attention mechanism (3.18) improves the network capacity without increasing memory costs as opposed to explicitly expanding $\mathbf{T}$.

The aggregated message $\tilde{\mathbf{m}}^t_A$ is combined with the representation of current step $\mathbf{h}^t_A$ through a point-wise interaction module ϕ (see Methods 3.7) to complete the update $\mathbf{h}^t_A \mapsto \mathbf{h}^{t+1}_A$.

Equivariant Normalization (EvNorm)

We define EvNorm : $\mathbf{h} \mapsto (\bar{\mathbf{h}}, \hat{\mathbf{h}})$ where $\bar{\mathbf{h}}$ and $\hat{\mathbf{h}}$ are given by

$$\bar{\mathbf{h}}_{nlp} := \frac{\|\mathbf{h}_{nlp}\| - \mu^h_{nlp}}{\sigma^h_{nlp}} \quad \text{and} \quad \hat{\mathbf{h}}_{nlpm} := \frac{\mathbf{h}_{nlpm}}{\|\mathbf{h}_{nlp}\| + 1/\beta_{nlp} + \epsilon} \tag{3.22}$$

where $\|\cdot\|$ denotes taking a neuron-wise regularized L^2 norm:

$$\|\mathbf{h}_{nlp}\| := \sqrt{\sum_m \mathbf{h}^2_{nlpm} + \epsilon^2} - \epsilon \tag{3.23}$$

μ^h_{klp} and σ^h_{klp} are mean and variance estimates of the invariant content $\|\mathbf{h}\|$ that can be obtained from either batch or layer statistics as in normalization schemes developed for scalar neural networks [99, 158]; β_{klp} are positive, learnable scalars controlling the fraction of vector scale information from $\mathbf{h}$ to be retained in $\hat{\mathbf{h}}$, and ϵ is a numerical stability factor. The EvNorm operation (3.22) decouples $\mathbf{h}$ to the normalized frame-invariant representation $\bar{\mathbf{h}}$ suitable for being transformed by an MLP, and a 'pure-direction' $\hat{\mathbf{h}}$ that is later multiplied to the MLP-transformed normalized invariant content to finish updating $\mathbf{h}$. Note that in (3.22), $\mathbf{h} = \mathbf{0}$ is always a fixed point of the map $\mathbf{h} \mapsto \hat{\mathbf{h}}$ and the vector directions information $\mathbf{h}$ is always preserved.

The Point-wise Interaction module and representation updates

A *point-wise interaction module* ϕ ((3.24)-(3.26)) nonlinearly updates the atom-wise representations through $\mathbf{h}^{t+1} = \phi(\mathbf{h}^t, \mathbf{g})$

$$\mathbf{f}^t_{lpm} = \big(\mathrm{MLP}_1(\bar{\mathbf{h}}^t)\big)_{lp} \odot (\hat{\mathbf{h}}^t_{lpm} \cdot \mathbf{W}^{\mathrm{in},t}_{l,p}) \quad \text{where} \quad (\bar{\mathbf{h}}^t, \hat{\mathbf{h}}^t) = \mathrm{EvNorm}(\mathbf{h}^t) \tag{3.24}$$

$$\mathbf{q}_{lpm} = \mathbf{g}_{lpm} + \sum_{l_1,l_2} \sum_{m_1,m_2} \sum_{p_1,p_2} (\mathbf{f}^t_{l_1p_1m_1} \odot \mathbf{g}_{l_2p_2m_2})\, C^{lm}_{l_1m_1;l_2m_2}\, \delta^{(-1)^{l_1+l_2+l}}_{p_1\cdot p_2\cdot p} \tag{3.25}$$

$$\mathbf{h}^{t+1}_{lpm} = \mathbf{h}^t_{lpm} + \big(\mathrm{MLP}_2(\bar{\mathbf{q}})\big)_{lp} \odot (\hat{\mathbf{q}}_{lpm} \cdot \mathbf{W}^{\mathrm{out},t}_{l,p}) \quad \text{where} \quad (\bar{\mathbf{q}}, \hat{\mathbf{q}}) = \mathrm{EvNorm}(\mathbf{q}) \tag{3.26}$$

which consist of coupling another O(3)-equivariant representation $\mathbf{g}$ with $\mathbf{h}^t$ and performing normalizations. In (3.24)-(3.26), $C^{lm}_{l_1 m_1; l_2 m_2}$ are Clebsch-Gordan coefficients of group SO(3), δ^j_i is a Kronecker delta function, and MLP$_1$ and MLP$_2$ denote multi-layer perceptrons acting on the feature (nlp) dimension. $\mathbf{W}^{\text{in},t}_{l,p} \in \mathbb{R}^{N_{l,p} \times N_{l,p}}$ and $\mathbf{W}^{\text{out},t}_{l,p} \in \mathbb{R}^{N_{l,p} \times N_{l,p}}$ correspond to learnable linear weight matrices specific to the update step t and each (l, p).

For $t < t_1$, the updates are performed by combining $\mathbf{h}^t$ with the aggregated messages $\tilde{\mathbf{m}}^t$ from step t:

$$\mathbf{h}^{t+1}_A = \phi\big(\mathbf{h}^t_A, \rho^\dagger(\tilde{\mathbf{m}}^t_A)\big) \tag{3.27}$$

where $\rho^\dagger$ is called a reverse matching layer, defined as:

$$\big(\rho^\dagger(\tilde{\mathbf{m}}^t_A)\big)_{l(p=+1)m} = \mathbf{W}^\dagger_l \cdot \sum_\mu \text{Scatter}\big(\tilde{\mathbf{m}}^t_{A,\mu}, n[\mu, z_A]\big) \tag{3.28}$$

$$\big(\rho^\dagger(\tilde{\mathbf{m}}^t_A)\big)_{l(p=-1)m} = \mathbf{0} \tag{3.29}$$

the Scatter operation maps the atomic-orbital dimension in $\tilde{\mathbf{m}}^t$ to a feature dimension with fixed length M_l using $n[\mu, z_A]$ as the indices and flattens the outputs into shape $(N_{\text{atoms}}, M_l IJ)$. $\mathbf{W}^\dagger_l \in \mathbb{R}^{N_{l,+1} \times M_l IJ}$ are learnable linear weight matrices to project the outputs into the shape of $\mathbf{h}^t$.

For $t_1 \le t < t_2$, the updates are based on local information:

$$\mathbf{h}^{t+1}_A = \phi\big(\mathbf{h}^t_A, \mathbf{h}^t_A\big) \tag{3.30}$$

Pooling layers and training

A programmed pooling layer reads out the target prediction $\hat{\mathbf{y}}$ after the representations $\mathbf{h}^t$ are updated to the last step $\mathbf{h}^{t_f}$. Pooling operations employed for obtaining main numerical results are detailed in Appendix 3.10; hyperparameter, training, and loss function details are provided in Appendix 3.10. As a concrete example, the dipole moment vector is predicted as $\vec{\mu} = \sum_A (\vec{x}_A \cdot q_A + \vec{\mu}_A)$ where $\vec{x}_A$ is the 3D coordinate of atom A, and atomic charges q_A and atomic dipoles $\vec{\mu}_A$ are predicted respectively using scalar ($l = 0$) and Cartesian-coordinate vector ($l = 1$) components of $\mathbf{h}^{t_f}_A$.

QM-informed featurization details and gradient calculations

The QM-informed representation employed in this work is motivated by a pair of our previous works [75, 76], but in this study the features are directly evaluated in the atomic orbital basis without the need of heuristic post-processing algorithms to enforce rotational invariance.

In particular, this work (as well as [76] and [75]) constructs features based on the GFN-xTB semi-empirical QM method [60]. As a member of the class of *mean field* quantum chemical methods, GFN-xTB centers around the self-consistent solution of the Roothaan-Hall equations,

$$\mathbf{FC} = \mathbf{SC}\boldsymbol{\epsilon} \tag{3.31}$$

All boldface symbols are matrices represented in the atomic orbital basis. For the particular case of GFN-xTB, the atomic orbital basis is similar to STO-6G and comprises a set of hydrogen-like orbitals. **C** is the molecular orbital coefficients which defines Ψ_0, and $\boldsymbol{\epsilon}$ is a diagonal eigenvalue matrix of the molecular orbital energies. **S** is the overlap matrix and is given by

$$S_{\mu\nu} = \langle \Phi^{\mu} | \Phi^{\nu} \rangle \tag{3.32}$$

where μ and ν index the atomic orbital basis $\{\Phi\}$. **F** is the *Fock matrix* and is given by

$$\mathbf{F} = \mathbf{H} + \mathbf{G}\,[\mathbf{P}] \tag{3.33}$$

H is the one-electron integrals including electron-nuclear attraction and electron kinetic energy. **G** is the two-electron integrals comprising the electron-electron repulsion. Approximation of **G** is the key task for self-consistent field methods, and GFN-xTB provides an accurate and efficient tight-binding approximation for **G**. Finally, **P** is the (one-electron-reduced) density matrix, and is given by

$$P_{\mu\nu} = \sum_{i=1}^{n_{\text{elec}}/2} C^{*}_{\mu i} C_{\nu i} \tag{3.34}$$

n_{elec} is the number of electrons, and a closed-shell singlet ground state is assumed for simplicity. Equations 3.31 and 3.33 are solved for **P**. The electronic energy E is related to the Fock matrix by

$$\mathbf{F} = \frac{\delta E}{\delta \mathbf{P}} \tag{3.35}$$

The particular form of the GFN-xTB electronic energy can be found in [60].
UNiTE is trained to predict the quantum chemistry properties of interest based on the inputs $\mathbf{T} = (\mathbf{F}, \mathbf{P}, \mathbf{S}, \mathbf{H})$ with possible extensions (e.g., the energy-weighted density matrices). For the example of learning the DFT electronic energy with the "delta-learning" training strategy:

$$E_{\text{DFT}} \approx E_{\text{TB}} + \mathcal{F}(\mathbf{T}) \tag{3.36}$$

Note that **F**, **P**, **S**, and **H** all implicitly depend on the atomic coordinates **x** and charge/spin state specifications.

In addition to predicting E it is also common to compute its gradient with respect to atomic nuclear coordinates **x** to predict the forces used for geometry optimization and molecular dynamics simulations. We directly differentiate the energy (3.36) to obtain energy-conserving forces. The partial derivatives of the UNiTE energy with respect to **F**, **P**, **S**, and **H** is determined through automatic differentiation. The resulting forces are computed through an adjoint approach developed in Appendix D of our previous work [75], with the simplification that the SAAO transformation matrix **X** is replaced by the identity.

3.8 Dataset and computational details

Training datasets

The molecule datasets used in Section 3.3-3.4 are all previously published. Following Section 2.1 of [131], the 2291 BFDb-SSI samples for training and testing are selected as the sidechain–sidechain dimers in the original BFDb-SSI dataset that contain ≤ 25 atoms and no sulfur element to allow for comparisons among methods.

The Selected Drug-like and biofragment Conformers (SDC21) dataset used for training the OrbNet-Equi/SDC21 model described in Section 3.5 is collected from several publicly-accessible sources. First 11,827 neutral SMILES strings were extracted from the ChEMBL database [159]. For each SMILES string, up to four conformers were generated by Entos Breeze, and optimized at the GFN-xTB level. Non-equilibrium geometries of the conformers were generated using either normal mode sampling [160] at 300K or *ab initio* molecular dynamics for 200fs at 500K in a ratio of 50%/50%, resulting in a total of 178,836 structures. An additional number 2,549 SMILES string were extracted from ChEMBL, and random protonation states for these were selected using Dimorphite-DL [161], as well as another 2,211 SMILES strings which were augmented by adding randomly selected salts from the list of common salts in the ChEMBL Structure Pipeline [162]. For these two collections of modified ChEMBL SMILES strings, non-equilibrium geometries were created using the same protocol described earlier, resulting in 21,141 and 27,005 additional structures for the two sets, respectively. To compensate for the bias towards large drug-like molecules, ~45,000 SMILES strings were enumerated using common bonding patterns, from which 9,830 conformers were generated from a randomly sampled subset. Lastly, molecules in the BFDb-SSI and JSCH-2005 datasets were added to the training data set [140, 163]. In total, the data set consists of 237,298 geometries spanning the elements C, O, N, F, S, Cl, Br, I, P, Si, B, Na, K, Li, Ca, and Mg. For each geometry DFT single point energies were calculated on the dataset at the ωB97X-D3/def2-TZVP level of theory in Entos Qcore version 0.8.17.[88, 93, 164] Lastly, we additionally filtered the geometries for which DFT calculation failed to converge or broken bonds between the equilibirum and non-equilibrium geometries are detected, resulting in 235,834 geometries used for training the OrbNet-Equi/SDC21 model.

Electronic structure computational details

The dipole moment labels $\vec{\mu}$ for QM9 dataset used in Section 3.3 were calculated at the B3LYP level of DFT theory with def2-TZVP AO basis set to match the level of theory used for published QM9 labels, using Entos Qcore version 1.1.0 [83, 89, 93]. The electron density labels $\rho(\vec{r})$ for QM9 and BFDb-SSI were computed at the ωB97X-D3/def2-TZVP level of DFT theory using def2-TZVP-JKFIT [165] for Coulomb and Exchange fitting, also as the electron charge density expansion basis $\{\chi\}$. The density expansion coefficients $\mathbf{d}$ are calculated as

$$d_\gamma = \sum_{\xi} \sum_{\mu,\nu} \left((\mathbf{S}^\rho)^{-1} \right)_{\gamma\xi} S_{\mu\nu;\xi} P_{\mu\nu} \tag{3.37}$$

where μ, ν are AO basis indices, ξ, γ are density fitting basis indices. Note that γ stands for the combined index (A, n, l, m) in (3.7). $\mathbf{P}$ is the DFT AO density matrix, $\mathbf{S}^\rho$ is the density fitting basis overlap matrix, and $S_{\mu\nu;\xi}$ are 3-index overlap integrals between the AO basis and the density fitting basis $\{\chi\}$.

Benchmarking details and summary statistics

For the mean L^1 electronic density error over the test sets reported in Section 3.4, we use 291 dimers as the test set for the BFDb-SSI dataset, and 10000 molecules as the test set for the QM9 dataset, following literature [131, 141]. ε_ρ for each molecule in the test sets is computed using a 3D cubic grid of voxel spacing $(0.2, 0.2, 0.2)$ Bohr for BFDb-SSI test set and voxel spacing $(1.0, 1.0, 1.0)$ Bohr for the QM9 test set, both with cutoff at $\rho(\vec{r}) = 10^{-5}\ a_0^{-3}$. We note that two baseline methods used slightly different normalization conventions when computing the dataset-averaged L^1 density errors ε_ρ, (a) computing ε_ρ for each molecule and normalizing over the number of molecules in the test set [141] or (b) normalizing over the total number of electrons in the test set [131]. We found the average ε_ρ computed using normalization (b) is higher than (a) by around 5% for our results. We follow their individual definitions for average ε_ρ for the quantitative comparisons described in the main text, that is, using scheme (a) for QM9 but scheme (b) for BfDB-SSI.

For downstream task statistics reported in Figure 3.5 and Table 3.5, the results on the Hutchison dataset in Figure 3.5a are calculated as the R^2 correlation coefficients comparing the conformer energies of multiple conformers from a given model to the energies from DLPNO-CCSD(T). The median R^2 in Table 3.5 with respect to both DLPNO-CCSD(T) and ωB97X-D3/def2-TZVP are calculated over the R^2-values for every molecule, and error bars are estimated by bootstrapping the pool of molecules. The error bars for TorsionNet500 and s66x10 are computed as 95% confidence intervals. Geometry optimization experiments are performed through relaxing the reference geometries until convergence. Geometry optimization accuracies in Figure 3.5d and Table 3.5 are reported as the symmetry-corrected root mean square deviation (RMSD) of the minimized geometry versus the reference level of theory (ωB97X-D3/def2-TZVP) calculated over molecules in the benchmark set. Additional computational details for this task are provided in Appendix 3.10.

For the GMTKN55 benchmark dataset collection, the reported CCSD(T)/CBS results are used as reference values. The WTAD scores for producing Figure 3.6 is defined similar to the updated weighted mean absolute deviation (WTMAD-2) in [150], but computed for each reaction in GMTKN55:

$$\mathrm{WTAD}_{i,j} = \frac{56.84}{\frac{1}{N_i}\sum_j |\Delta E|_{i,j}} \cdot |\Delta E|_{i,j} \tag{3.38}$$

for j-th reaction in the i-th task subset. Note that the subset-wise WTMAD-2 metric in Appendix Table 3.7 is given by

$$\text{WTMAD-2}_i = \frac{1}{N_i}\sum_j \mathrm{WTAD}_{i,j} \tag{3.39}$$

and the overall WTMAD-2 is reproduced by

$$\text{WTMAD-2} = \frac{1}{\sum_i^{55} N_i}\sum_{i,j} \mathrm{WTAD}_{i,j} \tag{3.40}$$

3.9 Additional theoretical results

We formally introduce the problem of interest, restate the definitions of the building blocks of UNiTE (Methods 3.7) using more formal notations, and prove the theoretical results claimed in this work. We first generalize the input data domain to a generic class of tensors beyond quantum chemistry quantities; for brevity we call such inputs *N-body tensors*.

N-body tensors (informal)

We are interested in a class of tensors $\mathbf{T}$, for which each sub-tensor $\mathbf{T}(u_1, u_2, \cdots, u_N)$ describes relation among a collection of N geometric objects defined in an n-dimensional physical space. For simplicity, we will first introduce the tensors of interest using a special case based on point clouds embedded in the n-dimensional Euclidean space, associating a (possibly different) set of orthogonal basis with each point's neighbourhood. In this setting, our main focus is the change of the order-N tensor's coefficients when applying n-dimensional rotations and reflections to the local reference frames.

Definition 1 (N-body tensor)**.** *Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_d\}$ be d points in $\mathbb{R}^n$ for each $u \in \{1, 2, \cdots, d\}$. For each point index u, we define an orthonormal basis (local reference frame) $\{\mathbf{e}_{u;v_u}\}$ centered at $\mathbf{x}_u$ [1], and denote the space spanned by the basis as $V_u := \mathrm{span}(\{\mathbf{e}_{u;v_u}\}) \subseteq \mathbb{R}^n$. We consider a tensor $\hat{\mathbf{T}}$ defined via N-th direct products of the 'concatenated' basis $\{\mathbf{e}_{u;v_u}; (u, v_u)\}$:*

$$\hat{\mathbf{T}} := \sum_{\vec{u},\vec{v}} T\big((u_1; v_1), (u_2; v_2), \cdots, (u_N; v_N)\big)\, \mathbf{e}_{u_1;v_1} \otimes \mathbf{e}_{u_2;v_2} \otimes \cdots \otimes \mathbf{e}_{u_N;v_N} \tag{3.41}$$

$\hat{\mathbf{T}}$ is a tensor of order-N and is an element of $(\bigoplus_{u=1}^{d} V_u)^{\otimes N}$. We call its coefficients $\mathbf{T}$ an N-body tensor if $\mathbf{T}$ is invariant to global translations ($\forall\, \mathbf{x}_0 \in \mathbb{R}^n, \mathbf{T}[\mathbf{x}] = \mathbf{T}[\mathbf{x}+\mathbf{x}_0]$, and is symmetric:

$$T\big((u_1; v_1), (u_2; v_2), \cdots, (u_N; v_N)\big) = T\big((u_{\sigma_1}; v_{\sigma_1}), (u_{\sigma_2}; v_{\sigma_2}), \cdots, (u_{\sigma_N}; v_{\sigma_N})\big) \tag{3.42}$$

where σ denotes arbitrary permutation on its dimensions $\{1, 2, \cdots, N\}$. Note that each sub-tensor, $\mathbf{T}_{\vec{u}}$, does not have to be symmetric. The shorthand notation $\vec{u} := (u_1, u_2, \cdots, u_N)$ indicates a subset of N points in $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_d\}$ which then identifies a sub-tensor [2] $\mathbf{T}_{\vec{u}} := \mathbf{T}(u_1, u_2, \cdots, u_N)$ in the N-body tensor $\mathbf{T}$; $\vec{v} := (v_1, v_2, \cdots, v_N)$ index a coefficient $T_{\vec{u}}(v_1, v_2, \cdots, v_N) := T\big((u_1; v_1), (u_2; v_2), \cdots, (u_N; v_N)\big)$ in a sub-tensor $\mathbf{T}_{\vec{u}}$, where each index $v_j \in \{1, 2, \cdots, \dim(V_{u_j})\}$ for $j \in \{1, 2, \cdots, N\}$.

[1] We additionally allow for $\mathbf{0} \in \{\mathbf{e}_{u;v_u}\}$ to represent features in $\mathbf{T}$ that transform as scalars.

[2] For example, if there are $d = 5$ points defined in the 3-dimensional Euclidean space $\mathbb{R}^3$ and each point is associated with a standard basis (x, y, z), then for the example of N=4, there are 5^4 sub-tensors and each sub-tensor $\mathbf{T}_{(u_1,u_2,u_3,u_4)}$ contains $3^4 = 81$ elements with indices spanning from $xxxx$ to $zzzz$. In total, $\mathbf{T}$ contains $(5 \times 3)^4$ coefficients. The coefficients of $\mathbf{T}$ are in general complex-valued as formally discussed in Definition S2, but are real-valued for the special case introduced in Definition 1 .

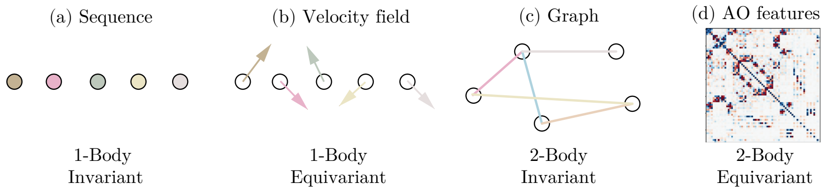

Figure 3.7: Examples of N-body tensors.

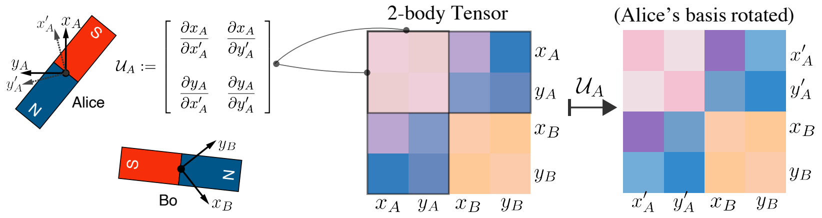
x'_A
x_A
y_A
y'_A
S
N
Alice
$\mathcal{U}_A := \begin{bmatrix} \frac{\partial x_A}{\partial x'_A} & \frac{\partial x_A}{\partial y'_A} \\ \frac{\partial y_A}{\partial x'_A} & \frac{\partial y_A}{\partial y'_A} \end{bmatrix}$
S
N
Bo
y_B
x_B
2-body Tensor
x_A
y_A
x_B
y_B
$\mathcal{U}_A$
(Alice's basis rotated)
x'_A
y'_A
x_B
y_B

Figure 3.8: Illustrating an N-body tensor with $N = 2$. Imagine Alice and Bo are doing experiments with two bar magnets without knowing each other's reference frame. The magnetic interactions depend on both bar magnets' orientations and can be written as a 2-body tensor. When Alice make a rotation on her reference frame, sub-tensors containing index A are transformed by a unitary matrix $\mathcal{U}_A$, giving rise to the 2-body tensor coefficients in the transformed basis. We design neural network to be equivariant to all such local basis transformations.

We aim to build neural networks $\mathcal{F}_\theta : (\bigoplus_{u=1}^{d} V_u)^{\otimes N} \to \mathcal{Y}$ that map $\hat{\mathbf{T}}$ to order-1 tensor- or scalar-valued outputs $\mathbf{y} \in \mathcal{Y}$. While $\hat{\mathbf{T}}$ is independent of the choice of local reference frame $\mathbf{e}_u$, its coefficients $\mathbf{T}$ (i.e. the N-body tensor) vary when rotating or reflecting the basis $\mathbf{e}_u := \{\mathbf{e}_{u;v_u}; v_u\}$, i.e. acted by an element $\mathcal{U}_u \in \mathrm{O(n)}$. Therefore, the neural network $\mathcal{F}_\theta$ should be constructed equivariant with respect to those reference frame transformations.

Equivariance

For a map $f : \mathcal{V} \to \mathcal{Y}$ and a group G, f is said to be G-equivariant if for all $g \in G$ and $\mathbf{v} \in \mathcal{V}$, $\varphi'(g) \cdot f(\mathbf{v}) = f(\varphi(g) \cdot \mathbf{v})$ where $\varphi(g)$ and $\varphi'(g)$ are the group representation of element g on $\mathcal{V}$ and $\mathcal{Y}$, respectively. In our case, the group G is composed of (a) Unitary transformations $\mathcal{U}_u$ locally applied to basis: $\mathbf{e}_u \mapsto \mathcal{U}_u^\dagger \cdot \mathbf{e}_u$, which are rotations and reflections for $\mathbb{R}^n$. $\mathcal{U}_u$ induces transformations on tensor coefficients: $\mathbf{T}_{\vec{u}} \mapsto (\mathcal{U}_{u_1} \otimes \mathcal{U}_{u_2} \otimes \cdots \otimes \mathcal{U}_{u_N})\mathbf{T}_{\vec{u}}$, and an intuitive example for infinitesimal basis rotations in $N = 2, n = 2$ is shown in Figure 3.8; (b) Tensor index permutations: $(\vec{u}, \vec{v}) \mapsto \sigma(\vec{u}, \vec{v})$; (c) Global translations: $\mathbf{x} \mapsto \mathbf{x} + \mathbf{x}_0$. For conciseness, we borrow the term G-equivariance to say $\mathcal{F}_\theta$ is equivariant to all the symmetry transformations listed above.

N-body tensors

Here we generalize the definition of N-body tensors to the basis of irreducible group representations instead of a Cartesian basis. The atomic orbital features discussed in the main text fall into this class, since the angular parts of atomic orbitals (i.e., spherical harmonics Y_{lm}) form the basis of the irreducible representations of group SO(3).

Definition 2. *Let $G_1, G_2, \cdots, G_d$ denote unitary groups where $G_u \subset \mathrm{U(n)}$ are closed subgroups of $\mathrm{U(n)}$ for each $u \in \{1, 2, \cdots, d\}$. We denote $G := G_1 \times G_2 \times \cdots \times G_d$. Let $(\pi_L, \mathbb{V}^L)$ denote a irreducible unitary representation of $\mathrm{U(n)}$ labelled by L. For each $u \in \{0, 1, \cdots, d\}$, we assume there is a finite-dimensional Banach space $V_u \simeq \bigoplus_L (\mathbb{V}^L)^{\oplus K_L}$ where $K_L \in \mathbb{N}$ is the multiplicity of $\mathbb{V}^L$ (e.g. the number of feature channels associated with representation index L), with basis $\{\boldsymbol{\pi}_{L,M}\}_u$ such that $\mathrm{span}(\{\boldsymbol{\pi}_{L,M,u}; k, L, M\}) = V_u$ for each $u \in \{1, 2, \cdots, d\}$ and $k \in \{1, 2, \cdots, K_L\}$, and $\mathrm{span}(\{\boldsymbol{\pi}_{L,M,u}; M\}) \simeq \mathbb{V}^L$ for each u, L. We denote $\mathcal{V} := \bigoplus_u V_u$, and index notation $v := (k, L, M)$. For a tensor $\hat{\mathbf{T}} \in \mathcal{V}^{\otimes N}$, we call the coefficients $\mathbf{T}$ of $\hat{\mathbf{T}}$ in the N-th direct products of basis $\{\boldsymbol{\pi}_{L,M,u}; L, M, u\}$ an N-body tensor, if $\hat{\mathbf{T}} = \sigma(\hat{\mathbf{T}})$ for any permutation $\sigma \in \mathrm{Sym}(N)$ (i.e. permutation invariant).*

Note that the vector spaces V_u do not need to be embed in the same space $\mathbb{R}^n$ as in the special case from Definition S1, but can be originated from general 'parameterizations' $u \mapsto V_u$, e.g., coordinate charts on a manifold.

Corollary 1. *If $V_u = \mathbb{C}^n$, $G_u = \mathrm{U(n)}$ and $\boldsymbol{\pi}_{L,M,u} = \mathbf{e}_M$ where $\{\mathbf{e}_M\}$ is a standard basis of $\mathbb{C}^n$, then $\mathbf{T}$ is an N-body tensor if $\hat{\mathbf{T}}$ is permutation invariant.*

Proof. For $V_u = \mathbb{C}^n$, $\pi : G_u \to \mathrm{U}(\mathbb{C}^n)$ is a fundamental representation of U(n). Since the fundamental representations of a Lie group are irreducible, it follows that $\{\mathbf{e}_M\}$ is a basis of a irreducible representation of U(n), and $\mathbf{T}$ is an N-body tensor. □

Similarly, when $V_u = \mathbb{R}^n$ and $G_u = \mathrm{O(n)} \subset \mathrm{U(n)}$, $\mathbf{T}$ is an N-body tensor if $\hat{\mathbf{T}}$ is permutation invariant. Then we can recover the special case based on point clouds in $\mathbb{R}^n$ in Definition 1.

Procedures for constructing complete bases for irreducible representations of U(n) with explicit forms are established [166]. A special case is $G_u = \mathrm{SO}(3)$, for which a common construction of a complete set of $\{\boldsymbol{\pi}_{L,M}\}_u$ is using the spherical harmonics $\boldsymbol{\pi}_{L,M,u} := Y_{lm}$; this is an example that polynomials Y_{lm} can be constructed as a basis of square-integrable functions on the 2-sphere $L^2(S^2)$ and consequently as a basis of the irreducible representations $(\pi_L, \mathbb{V}^L)$ for all L [167].

Decomposition of diagonals $\mathbf{T}_u$

We consider the algebraic structure of the diagonal sub-tensors $\mathbf{T}_u$, which can be understood from tensor products of irreducible representations.

First we note that for a sub-tensor $\mathbf{T}_{\vec{u}} \in V_{u_1} \otimes V_{u_2} \otimes \cdots \otimes V_{u_N}$, the action of $g \in G$ is given by

$$g \cdot \mathbf{T}_{\vec{u}} = (\pi(g_{u_1}) \otimes \pi(g_{u_2}) \otimes \cdots \otimes \pi(g_{u_N}))\mathbf{T}_{\vec{u}} \tag{3.43}$$

for diagonal sub-tensors $\mathbf{T}_u$, this reduces to the action of a diagonal sub-group

$$g \cdot \mathbf{T}_u = (\pi(g_u) \otimes \pi(g_u) \otimes \cdots \otimes \pi(g_u))\mathbf{T}_u \tag{3.44}$$

which forms a representation of $G_u \in \mathrm{U}(n)$ on $V_u^{\otimes N}$. According to the isomorphism $V_u \simeq \bigoplus_L (\mathbb{V}^L)^{\oplus K_L}$ in Definition S2 we have $\pi(g_u) \cdot \mathbf{v} = \bigoplus_L U^L_{g_u} \cdot \mathbf{v}_L$ for $\mathbf{v} \in V_u$ where $\mathbf{v}_L \in \mathbb{V}^L$, more explicitly

$$g \cdot \mathbf{T}_u(\vec{k}, \vec{L}) = (U^{L_1}_{g_u} \otimes U^{L_2}_{g_u} \otimes \cdots \otimes U^{L_N}_{g_u})\, \mathbf{T}_u(\vec{k}, \vec{L}) \tag{3.45}$$

where we have used the shorthand notation $\mathbf{T}_u(\vec{k}, \vec{L}) := \mathbf{T}_u\big((k_1, L_1), (k_2, L_2), \cdots, (k_N, L_N)\big)$ and $U^L_{g_u}$ denotes the unitary matrix representation of $g_u \in \mathrm{U(n)}$ on $\mathbb{V}^L$ expressed in the basis $\{\boldsymbol{\pi}_{L,M,u}; M\}$, on the vector space $\mathbb{V}^L$ for the irreducible representation labelled by L. Therefore $\mathbf{T}_u(\vec{k}, \vec{L}) \in \mathbb{V}^{L_1} \otimes \mathbb{V}^{L_2} \otimes \cdots \otimes \mathbb{V}^{L_N}$ is the representation space of an N-fold tensor product representations of U(n). We note the following theorem for the decomposition of $\mathbf{T}_u(\vec{k}, \vec{L})$:

Theorem 1 (Theorem 2.1 and Lemma 2.2 of [168]). *The representation of* $\mathrm{U(n)}$ *on the direct product of* $\mathbb{V}^{L_1}, \mathbb{V}^{L_2}, \cdots, \mathbb{V}^{L_N}$ *decomposes into direct sum of irreducible representations:*

$$\mathbb{V}^{L_1} \otimes \mathbb{V}^{L_2} \otimes \cdots \otimes \mathbb{V}^{L_N} \simeq \bigoplus_L \bigoplus_{\nu}^{\mu(L_1,L_2,\cdots,L_N;L)} \mathbb{V}^{L;\nu} \tag{3.46}$$

and

$$\sum_L \mu(L_1, L_2, \cdots, L_N; L) \dim(\mathbb{V}^L) = \prod_{u=1}^{N} \dim(\mathbb{V}^{L_u}) \tag{3.47}$$

where $\mu(L_1, L_2, \cdots, L_N; L)$ *is the multiplicity of* L *denoting the number of replicas of* $\mathbb{V}^L$ *being present in the decomposition of* $\mathbb{V}^{L_1} \otimes \mathbb{V}^{L_2} \otimes \cdots \otimes \mathbb{V}^{L_N}$.

Note that we have abstracted the labeling details for U(n) irreducible representations into the index L. See [168] for proof and details on representation labeling. We now state the following result for generating order-1 representations (Materials and methods, 3.7):

Corollary 2. *There exists an invertible linear map* $\psi : V_u^{\otimes N} \rightarrow V_u^\star := \bigoplus (\mathbb{V}^L)^{\oplus \mu(L;V_u)}$ *where* $\mu(L;V_u) \in \mathbb{N}$*, such that for any* $\mathbf{T}_u$*,* L *and* $\nu \in \{1,2,\cdots,\mu(L;V_u)\}$*,* $\psi(g_u \cdot \mathbf{T}_u)_{\nu,L} = U_{g_u}^L \cdot \psi(\mathbf{T}_u)_{\nu,L}$ *if* $\mu(L;V_u) > 0$*.*

Proof. First note that each block $\mathbf{T}_u(\vec{k},\vec{L})$ of $\mathbf{T}_u$ is an element of $\mathbb{V}^{L_1} \otimes \mathbb{V}^{L_2} \otimes \cdots \otimes \mathbb{V}^{L_N}$ up to an isomorphism. (3.47) in Theorem S1 states there is an invertible linear map $\psi_{\vec{L}} : \mathbb{V}^{L_1} \otimes \mathbb{V}^{L_2} \otimes \cdots \otimes \mathbb{V}^{L_N} \rightarrow \bigoplus_L (\mathbb{V}^L)^{\oplus \mu(L_1,L_2,\cdots,L_N;L)}$, such that $\tau(g_u) = (\psi_{\vec{L}})^{-1} \circ \pi(g_u) \circ \psi_{\vec{L}}$ for any $g_u \in G_u$, where $\tau : G_u \rightarrow \mathrm{U}(\mathbb{V}^{L_1} \otimes \mathbb{V}^{L_2} \otimes \cdots \otimes \mathbb{V}^{L_N})$ and $\pi : G_u \rightarrow \mathrm{U}(\bigoplus_L (\mathbb{V}^L)^{\oplus \mu(L_1,L_2,\cdots,L_N;L)})$ are representations of G_u. Note that π is defined as a direct sum of irreducible representations of U(n), i.e. $\pi(g_u)\psi_{\vec{L}}(\mathbf{T}_u(\vec{k},\vec{L})) := \bigoplus_{\nu,L} U_{g_u}^L \big(\psi_{\vec{L}}(\mathbf{T}_u(\vec{k},\vec{L}))\big)_{\nu,L}$. Note that $\psi(\mathbf{T}_u) := \bigoplus_L \bigoplus_{\vec{k},\vec{L}} \psi_{\vec{L}}(\mathbf{T}_u(\vec{k},\vec{L}))_L$ and $\mu(L,V_u) := \sum_{\vec{k},\vec{L}} \mu(L_1,L_2,\cdots,L_N;L)$ directly satisfies $\psi(g_u \cdot \mathbf{T}_u)_{\nu,L} = \psi(\tau(g_u)\mathbf{T}_u)_{\nu,L} = U_{g_u}^L \psi(\mathbf{T}_u)_{\nu,L}$ for $\nu \in \{1,2,\cdots,\mu(L;V_u)\}$. Since each $\psi_{\vec{L}}$ are finite-dimensional and invertible, it follows that the finite direct sum ψ is invertible. □

For Hermitian tensors, we conjecture the same result for SU(2), O(2) and O(3) as each irreducible representation is isomorphic to its complex conjugate.

We then formally restate the proposition in Methods 3.7 which was originally given for orthogonal representations of O(3) (i.e., the real spherical harmonics):

Corollary 3. *For each* L *where* $\mu(L;V_u) > 0$*, there exist* $n_L \times \dim(V_u)^N$ $\mathbf{T}$*-independent coefficents* $Q_{\nu,L,M}^{\vec{v}}$ *parameterizing the linear transformation* ψ *that performs* $\mathbf{T}_{\vec{u}} \mapsto \mathbf{h}_u := \psi(\mathbf{T}_u)$*, if* $u_1 = u_2 = \cdots = u_N = u$*:*

$$\big(\psi(\mathbf{T}_u)\big)_{\nu,L,M} := \sum_{\vec{v}} T_u(v_1,v_2,\cdots v_N)\, Q_{\nu,L,M}^{\vec{v}} \quad \textit{for} \quad \nu \in \{1,2,\cdots,n_L\} \tag{3.48}$$

such that the linear map ψ *is injective,* $\sum_L n_L \leq \dim(V_u)^N$*, and for each* $g_u \in G_u$*:*

$$\psi\big(g_u \cdot \mathbf{T}_u\big)_{\nu,L} = U_{g_u}^L \big(\psi(\mathbf{T}_u)\big)_{\nu,L} \tag{3.49}$$

Proof. According to Definition 2, a complete basis of $V_u^{\otimes N}$ is given by $\{\boldsymbol{\pi}_{L_1,M_1,u} \otimes \boldsymbol{\pi}_{L_2,M_2,u} \otimes \cdots \otimes \boldsymbol{\pi}_{L_N,M_N,u}; (\vec{k},\vec{L},\vec{M})\}$ and a complete basis of $(V_u^\star)_L$ is $\{\boldsymbol{\pi}_{L,M,u}; M\}$. Note that V_u and $(V_u^\star)_L$ are both finite dimensional. Therefore an example of $\mathbf{Q}_L$

is the $\dim(V_u)^N \times \mu(L; V_u)$ matrix representation of the bijective map ψ in the two basis, which proves the existence. □

Note that Corollary S3 does not guarantee the resulting order-1 representations $\mathbf{h}_u := \psi(\mathbf{T}_u)$ (i.e. vectors in $V_u^\star$) to be invariant under permutations σ, as the ordering of $\{\nu\}_L$ may change under $\mathbf{T} \mapsto \sigma(\mathbf{T})$. Hence, the symmetric condition on $\mathbf{T}$ is important to achieve permutation equivariance for the decomposition $V_u^{\otimes N} \longrightarrow V_u^\star$; we note that $\mathbf{T}_u$ has a symmetric tensor factorization and is an element of $\mathrm{Sym}^N(V_u)$, then algebraically the existence of a permutation-invariant decomposition is ensured by the Schur-Weyl duality [169] giving the fact that all representations in the decomposition of $\mathrm{Sym}^N(V_u)$ must commute with the symmetric group S_N. With the matrix representation $\mathbf{Q}$ in (3.48), clearly for any σ, $\psi(\sigma(\mathbf{T}_u)) = \sigma(\mathbf{T}_u)\cdot\mathbf{Q} = \mathbf{T}_u\cdot\mathbf{Q} = \psi(\mathbf{T}_u)$. For general asymmetric N-body tensors, we expect the realization of permutation equivariance to be sophisticated and may be achieved through tracking the Schur functors from the decomposition of $V_u^{\otimes N} \longrightarrow V_u^\star$, which is considered out of scope of the current work. Additionally, the upper bound $\sum_L n_L \leq \dim(V_u)^N$ is in practice often not saturated and the contraction (3.48) can be simplified. For example, when $N > 2$ it suffices to perform permutation-invariant decomposition on symmetric $\mathbf{T}_u$ recursively through Clebsch-Gordan coefficients $\mathbf{C}$ which has the following property:

$$\mathbf{C}^{\nu,L}_{L_1;L_2}\cdot(U^{L_1}_{g_u}\otimes U^{L_2}_{g_u})\cdot(\mathbf{C}^{\nu,L}_{L_1;L_2})^\dagger = U^{L}_{g_u} \quad \text{for} \quad \nu\in\{1,2,\cdots,\mu(L_1,L_2;L)\} \tag{3.50}$$

i.e., $\mathbf{C}$ parameterizes the isomorphism $\psi_{\vec{L}}$ of Theorem S2 for $N=2$, $\vec{L}=(L_1,L_2)$. Then ψ can be constructed with the procedure $(V_u)^{\otimes N} \mapsto V_u' \otimes (V_u)^{\otimes(N-2)} \mapsto V_u'' \otimes (V_u)^{\otimes(N-3)} \mapsto V_u^\star$ without explicit order-$N+1$ tensor contractions, where each reduction step can be parameterized using $\mathbf{C}$.

Procedures for computing $\mathbf{C}$ in general are established [170, 171]. For the main results reported in this work $\mathrm{O}(3) \simeq \mathrm{SO}(3)\times\mathbb{Z}_2$ is considered, where $\mu(L_1,L_2;L)\leq 1$ and the basis of an irreducible representation $\boldsymbol{\pi}_{L,M}$ can be written as $\boldsymbol{\pi}_{L,M} := |l,m,p\rangle$ where $p\in\{1,-1\}$ and $m\in\{-l,-l+1,\cdots,l-1,l\}$. $|l,m,p\rangle$ can be thought as a spherical harmonic Y_{lm} but may additionally flips sign under point reflections $\mathcal{I}$ depending on the parity index p: $\mathcal{I}\,|l,m,p\rangle = p\cdot(-1)^l\,|l,m,p\rangle$ where $\forall\mathbf{x}\in\mathbb{R}^3, \mathcal{I}(\mathbf{x}) = -\mathbf{x}$. Clebsch-Gordan coefficients $\mathbf{C}$ for O(3) is given by:

$$C^{\nu=1,lpm}_{l_1p_1m_1;l_2p_2m_2} = C^{lm}_{l_1m_1;l_2m_2}\,\delta^{((-1)^{l_1+l_2+l})}_{p_1\cdot p_2\cdot p} \tag{3.51}$$

where $C^{lm}_{l_1m_1;l_2m_2}$ are SO(3) Clebsch-Gordan coefficients. For $N=2$, the problem reduces to using Clebsch-Gordan coefficients to decompose $\mathbf{T}_u$ as a combination of matrix representations of *spherical tensor operators* which are linear operators transforming under irreducible representation of SO(3) based on the the Wigner-Eckart Theorem (see [15] for formal derivations).

Both V_u and $V_u^\star$ are defined as direct sums of the representation spaces $\mathbb{V}^L$ of irreducible representations of G_u, but each L may be associated with a different multiplicity K_L or $K_L^\star$ (e.g. different numbers of feature channels). We also allow for the case that the definition basis $\{\mathbf{e}_{u;v}\}$ for the N-body tensor $\mathbf{T}$ differ from $\{\boldsymbol{\pi}_{u;L,M}\}$ by a known linear transformation $\mathbf{D}_u$ such that $\mathbf{e}_{u;v} := \sum_{L,M} (D_u)_v^{L,M} \boldsymbol{\pi}_{u;L,M}$, or $(D_u)_v^{L,M} := \langle \mathbf{e}_{u;v}, \boldsymbol{\pi}_{u;L,M} \rangle$ where $\langle \cdot, \cdot \rangle$ denotes a Hermitian inner product, and we additionally define if $K_L = 0$, $\langle \mathbf{e}_{u;v}, \boldsymbol{\pi}_{u;L,M} \rangle := 0$. We then give a natural extension to Definition S2:

Definition 3. *We extend the basis in Definition 2 for N-body tensors to* $\{\mathbf{e}_{u;v}\}$ *where* $\mathrm{span}(\{\mathbf{e}_{u;v}; v\}) = V_u$, *if*

$$\mathbf{D}_u \cdot \pi^2(g_u) = \pi^1(g_u) \cdot \mathbf{D}_u \quad \forall g_u \in G_u \tag{3.52}$$

where π^1 *and* π^2 *are matrix representations of* g_u *on* $V_u \subset V_u^\star$ *in basis* $\{\mathbf{e}_{u;v}\}$ *and in basis* $\{\boldsymbol{\pi}_{u;L,M}\}$. *Note that* $\pi^2(g_u) \cdot \mathbf{v} = U_{g_u}^L \cdot \mathbf{v}$ *for* $\mathbf{v} \in \mathbb{V}^L$.

Generalized neural network building blocks

We clarify that in all the sections below n refers to a feature channel index within a irreducible representation group labelled by L, which should not be confused with $\dim(V_u)$. More explicitly, we note $n \in \{1, 2, \cdots, N_L^{\mathrm{h}}\}$ where N_L^{h} is the number of vectors in the order-1 tensor $\mathbf{h}_u^t$ that transforms under the L-th irreducible representation G_u (i.e. the multiplicity of L in $\mathbf{h}_u^t$). $M \in \{1, 2, \cdots, \dim(\mathbb{V}^L)\}$ indicates the M-th component of a vector in the representation space of the L-th irreducible representation of G_u, corresponding to a basis vector $\boldsymbol{\pi}_{L,M,u}$. We also denote the total number of feature channels in h as $N^{\mathrm{h}} := \sum_L N_L^{\mathrm{h}}$.

For a simple example, if the features in the order-1 representation $\mathbf{h}^t$ are specified by $L \in \{0, 1\}$, $N_{L=0}^{\mathrm{h}} = 8$, $N_{L=1}^{\mathrm{h}} = 4$, $\dim(\mathbb{V}^{L=0}) = 1$, and $\dim(\mathbb{V}^{L=1}) = 5$, then $N^{\mathrm{h}} = 8{+}4 = 12$ and $\mathbf{h}_u^t$ is stored as an array with $\sum_L N_L^{\mathrm{h}} {\cdot} \dim(\mathbb{V}^L) = (8{\times}1{+}4{\times}5) = 28$ elements.

We reiterate that $\vec{u} := (u_1, u_2, \cdots, u_N)$ is a sub-tensor index (location of a sub-tensor in the N-body tensor $\mathbf{T}$), and $\vec{v} := (v_1, v_2, \cdots, v_N)$ is an element index in a sub-tensor $\mathbf{T}_{\vec{u}}$.

Convolution and message passing. We generalize the definition of a block convolution module (3.15) to order-N and complex numbers:

$$(\mathbf{m}^t_{\vec{u}})^i_{v_1} = \sum_{v_2,\cdots,v_N} T_{\vec{u}}(v_1, v_2, \cdots, v_N) \prod_{j=2}^{N} \left(\rho_{u_j}(\mathbf{h}^t_{u_j})^*\right)^i_{v_j} \tag{3.53}$$

Message passing modules (3.17)-(3.27) is generalized to order N:

$$\tilde{\mathbf{m}}^t_{u_1} = \sum_{u_2,u_3,\cdots,u_N} \bigoplus_{i,j} (\mathbf{m}^t_{\vec{u}})^i \cdot \alpha^{t,j}_{\vec{u}} \tag{3.54}$$

$$\mathbf{h}^{t+1}_{u_1} = \phi\big(\mathbf{h}^t_{u_1}, \rho^\dagger_{u_1}(\tilde{\mathbf{m}}^t_{u_1})\big) \tag{3.55}$$

EvNorm. We write the EvNorm operation (3.22) as EvNorm : $\mathbf{h} \mapsto (\bar{\mathbf{h}}, \hat{\mathbf{h}})$ where

$$\bar{h}_{nL} := \frac{||\mathbf{h}_{nL}||-\mu^x_{nL}}{\sigma^x_{nL}} \quad \text{and} \quad \hat{h}_{nLM} := \frac{x_{nLM}}{||\mathbf{h}_{nL}||+1/\beta_{nL}+\epsilon} \tag{3.56}$$

Point-wise interaction ϕ. We adapt the notations and explicitly expand (3.24)-(3.26) for clarity. The operations within a point-wise interaction block $\mathbf{h}^{t+1}_u = \phi(\mathbf{h}^t_u, \mathbf{g}_u)$ are defined as:

$$(\mathbf{f}^t_u)_{nLM} = \big(\mathrm{MLP}_1(\bar{\mathbf{h}}^t_A)\big)_{nL} (\hat{\mathbf{h}}^t_u)_{nLM} \quad \text{where} \quad (\bar{\mathbf{h}}^t_u, \hat{\mathbf{h}}^t_u) = \mathrm{EvNorm}(\mathbf{h}^t_u) \tag{3.57}$$

$$(\mathbf{q}_u)_{nLM} = (\mathbf{g}_u)_{nLM} + \sum_{L_1,L_2} \sum_{M_1,M_2} (\mathbf{f}^t_u)_{nL_1M_1} (\mathbf{g}_u)_{nL_2M_2}\, C^{\nu(n),LM}_{L_1M_1;L_2M_2} \tag{3.58}$$

$$(\mathbf{h}^{t+1}_u)_{nLM} = (\mathbf{h}^t_u)_{nLM} + \big(\mathrm{MLP}_2(\bar{\mathbf{q}}_u)\big)_{nL} (\hat{\mathbf{q}}_u)_{nLM} \quad \text{where} \quad (\bar{\mathbf{q}}_u, \hat{\mathbf{q}}_u) = \mathrm{EvNorm}(\mathbf{q}_u) \tag{3.59}$$

where $\nu : \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ assigns an output multiplicity index ν to a group of feature channels n.

For the special example of O(3) where the output multiplicity $\mu(L_1, L_2; L) \leq 1$ (see Theorem S1 for definitions), we can restrict $\nu(n) \equiv 1$ for all values of n, and (3.58) can be rewritten as

$$(\mathbf{q}_u)_{nlpm} = (\mathbf{g}_u)_{nlpm} + \sum_{l_1,l_2} \sum_{m_1,m_2} \sum_{p_1,p_2} (\mathbf{f}^t_u)_{nl_1p_1m_1} (\mathbf{g}_u)_{nl_2p_2m_2}\, C^{lm}_{l_1m_1;l_2m_2}\, \delta^{((-1)^{l_1+l_2+l})}_{p_1\cdot p_2\cdot p} \tag{3.60}$$

which is based on the construction of $C^{\nu(n),LM}_{L_1M_1;L_2M_2}$ in (3.51). The above form recovers (3.25).

Matching layers. Based on Definition S3, we can define generalized matching layers ρ_u and $\rho_u^\dagger$ as

$$\big(\rho_u(\mathbf{h}_u^t)\big)_v^i = \sum_{L,M} \big(\mathbf{W}_L^i \cdot (\mathbf{h}_u^t)_{LM}\big) \cdot \langle \mathbf{e}_{u;v}, \boldsymbol{\pi}_{u;L,M}\rangle \tag{3.61a}$$

$$\big(\rho_u^\dagger(\tilde{\mathbf{m}}_u^t)\big)_{LM} = \sum_{v} \mathbf{W}_L^\dagger \cdot (\tilde{\mathbf{m}}_u^t)_v \cdot \langle \boldsymbol{\pi}_{u;L,M}, \mathbf{e}_{u;v}\rangle \tag{3.61b}$$

where $\mathbf{W}_L^i$ are learnable $(1 \times N_L^\mathrm{h})$ matrices; $\mathbf{W}_L^\dagger$ are learnable $(N_L^\mathrm{h} \times (N^\mathrm{i}\, N^\mathrm{j}))$ matrices where N^i denotes the number of convolution channels (number of allowed i in (3.15)).

G-equivariance

With main results from Corollary S2 and Corollary S3 and basic linear algebra, the equivariance of UNiTE can be straightforwardly proven. G-equivariance of the Diagonal Reduction layer ψ is stated in Corollary S3, and it suffices to prove the equivariance for other building blocks.

Proof of G-equivariance for the convolution block (3.53). For any $g \in G$:

$$\sum_{v_2,\cdots,v_N} \big(g \cdot T_{\vec{u}}(v_1, v_2, \cdots, v_N)\big) \prod_{j=2}^{N} \big(\rho_{u_j}(g \cdot \mathbf{h}^t_{u_j})^*\big)^i_{v_j} \tag{3.62a}$$

$$= \sum_{v_2,\cdots,v_N} ((\bigotimes_{\vec{u}} \pi^1(g_{u_j}) \cdot T_{\vec{u}})(v_1, v_2, \cdots, v_N)) \prod_{j=2}^{N} \big(\rho_{u_j}(\pi^2(g_{u_j}) \cdot \mathbf{h}^t_{u_j})^*\big)^i_{v_j} \tag{3.62b}$$

$$= \sum_{v_2,\cdots,v_N} ((\bigotimes_{\vec{u}} \pi^1(g_{u_j}) \cdot T_{\vec{u}})(v_1, v_2, \cdots, v_N)) \prod_{j=2}^{N} \big(\sum_{L,M} (\mathbf{D}_{u_j})^{L,M}_{v_j} \cdot \big(\mathbf{W}^i_L \cdot (\pi^2(g_{u_j}) \cdot \mathbf{h}^t_{u_j})_{LM}\big)\big)^*\big)^i \tag{3.62c}$$

$$= \sum_{v_2,\cdots,v_N} ((\bigotimes_{\vec{u}} \pi^1(g_{u_j}) \cdot T_{\vec{u}})(v_1, v_2, \cdots, v_N)) \prod_{j=2}^{N} \big((\sum_{L,M} \mathbf{W}^i_L \cdot (\mathbf{D}_{u_j})^{L,M}_{v_j} \cdot (\pi^2(g_{u_j}) \cdot \mathbf{h}^t_{u_j})_{LM}\big)\big)^*\big)^i \tag{3.62d}$$

$$\overset{\text{(S35)}}{=} \sum_{v_2,\cdots,v_N} \big((\bigotimes_{\vec{u}} \pi^1(g_{u_j}) \cdot T_{\vec{u}})(v_1, v_2, \cdots, v_N)\big) \prod_{j=2}^{N} \big(\sum_{L,M} \big(\mathbf{W}^i_L \cdot (\pi^1(g_{u_j}) \cdot \mathbf{D}^{L,M}_{u_j} \cdot \mathbf{h}^t_{u_j})_{v_j})\big)^*\big)^i \tag{3.62e}$$

$$= \sum_{v_2,\cdots,v_N} ((\bigotimes_{\vec{u}} \pi^1(g_{u_j}) \cdot T_{\vec{u}})(v_1, v_2, \cdots, v_N)) \prod_{j=2}^{N} \big(\pi^1(g_{u_j})^* \cdot (\rho_{u_j}(\mathbf{h}^t_{u_j}))^*\big)^i_{v_j} \tag{3.62f}$$

$$= \sum_{v_2,\cdots,v_N} \sum_{v'_1,v'_2,\cdots,v'_N} ((\pi^1(g_{u_j}))_{v_j,v'_j} \cdot T_{\vec{u}}(v'_1, v'_2, \cdots, v'_N)) \prod_{j=2}^{N} \sum_{v''_j} (\pi^1(g_{u_j})^*)_{v_j,v''_j} \cdot \big((\rho_{u_j}(\mathbf{h}^t_{u_j}))^*\big)^i_{v''_j} \tag{3.62g}$$

$$= \sum_{v'1,v'_2,\cdots,v'_N} (\pi^1(g_{u_1}))_{v_1,v'_1} T_{\vec{u}}(v'_1, v'_2, \cdots, v'_N) \prod_{j=2}^{N} \sum_{v_j,v''_j} (\pi^1(g_{u_j}))_{v_j,v'_j} (\pi^1(g_{u_j})^*)_{v_j,v''_j} \cdot \big((\rho_{u_j}(\mathbf{h}^t_{u_j}))^*\big)^i_{v''_j} \tag{3.62h}$$

$$= \sum_{v'_1,v'_2,\cdots,v'_N} (\pi^1(g_{u_1}))_{v_1,v'_1} T_{\vec{u}}(v'_1, v'_2, \cdots, v'_N) \prod_{j=2}^{N} \sum_{v''_j} \delta^{v''_j}_{v'_j} \cdot \big((\rho_{u_j}(\mathbf{h}^t_{u_j}))^*\big)^i_{v''_j} \tag{3.62i}$$

$$= \sum_{v'_1} (\pi^1(g_{u_1}))_{v_1,v'_1} \sum_{v'_2,\cdots,v'_N} T_{\vec{u}}(v'_1, v'_2, \cdots, v'_N) \prod_{j=2}^{N} \big((\rho_{u_j}(\mathbf{h}^t_{u_j}))^*\big)^i_{v'_j} \tag{3.62j}$$

$$= \sum_{v'_1} (\pi^1(g_{u_1}))_{v_1,v'_1} (\mathbf{m}^t_{\vec{u}})^i_{v'_1} \tag{3.62k}$$

$$= \pi^1(g_{u_1}) \cdot (\mathbf{m}^t_{\vec{u}})^i_{v'_1} = (g \cdot (\mathbf{m}^t_{\vec{u}})^i)_{v_1} \tag{3.62l}$$

Proof of G-equivariance for the message passing block (3.54)-(3.55). From the invariance condition $g \cdot \alpha^{t,j}_{\vec{u}} = \alpha^{t,j}_{\vec{u}}$, clearly

$$\sum_{u_2,u_3,\cdots,u_N} \bigoplus_{i,j} (g \cdot \mathbf{m}^t_{\vec{u}})^i \cdot \alpha^{t,j}_{\vec{u}} = \sum_{u_2,u_3,\cdots,u_N} \bigoplus_{i,j} (\pi^1(g_{u_1}) \cdot (\mathbf{m}^t_{\vec{u}})^i) \cdot \alpha^{t,j}_{\vec{u}} \tag{3.63a}$$

$$= \pi^1(g_{u_1}) \cdot \sum_{u_2,u_3,\cdots,u_N} \bigoplus_{i,j} ((\mathbf{m}^t_{\vec{u}})^i) \cdot \alpha^{t,j}_{\vec{u}} \tag{3.63b}$$

$$= \pi^1(g_{u_1}) \cdot \tilde{\mathbf{m}}^t_{u_1} = g \cdot \tilde{\mathbf{m}}^t_{u_1} \tag{3.63c}$$

Proof of G-equivariance for EvNorm (3.56). Note that the vector norm $||\mathbf{x}_{nL}||$ is invariant to unitary transformations $\mathbf{x}_{nL} \mapsto U^L_{g_u} \cdot \mathbf{x}_{nL}$. Then $\overline{(g \cdot \mathbf{x})} = \frac{||\mathbf{x}_{nL}|| - \mu^x_{nL}}{\sigma^x_{nL}} = \bar{\mathbf{x}}$, and $\widehat{(g\, \mathbf{x}_{nL})} = \frac{(\pi^2(g_u) \cdot x)_{nLM}}{||\mathbf{x}_{nL}|| + 1/\beta_{nL} + \epsilon} = \pi^2(g_u) \cdot \hat{\mathbf{x}}_{nL} = g \cdot \hat{\mathbf{x}}_{nL}$.

Proof of G-equivariance for the point-wise interaction block (3.57)-(3.59). Equivariances for (3.57) and (3.59) are direct consequences of the equivariance of EvNorm $\overline{(g\cdot\mathbf{x})} = \bar{\mathbf{x}}$ and $\widehat{(g\,\mathbf{x}_{nL})} = g\cdot\hat{\mathbf{x}}_{nL}$, if $g_u\cdot\mathbf{x}_{nL} = \pi^2(g_u)\cdot\mathbf{x}_{nL} \equiv U^L_{g_u}\cdot\mathbf{x}_{nL}$. Then it suffices to prove $g\cdot(\mathbf{q}_u)_{nL} = U^L_{g_u}\cdot(\mathbf{q}_u)_{nL}$, which is ensured by (3.50):

$$(g_u\cdot\mathbf{g}_u)_{nLM} + \sum_{L_1,L_2}\sum_{M_1,M_2}(g_u\cdot\mathbf{f}^t_u)_{nL_1M_1}(g_u\cdot\mathbf{g}_u)_{nL_2M_2}\,C^{\nu(n),LM}_{L_1M_1;L_2M_2} \tag{3.64a}$$

$$= (U^L_{g_u}\cdot\mathbf{g}_u)_{nLM} + \sum_{L_1,L_2}\sum_{M_1,M_2}(U^{L_1}_{g_u}\cdot\mathbf{f}^t_u)_{nL_1M_1}(U^{L_2}_{g_u}\cdot\mathbf{g}_u)_{nL_2M_2}\,C^{\nu(n),LM}_{L_1M_1;L_2M_2} \tag{3.64b}$$

$$= (U^L_{g_u}\cdot\mathbf{g}_u)_{nLM} + \sum_{L_1,L_2}\sum_{M_1,M_2}\sum_{M_1',M_2'}(U^{L_1}_{g_u}\otimes U^{L_2}_{g_u})^{M_2,M_2'}_{M_1,M_1'}\cdot(\mathbf{f}^t_u)_{nL_1M_1'}(\mathbf{g}_u)_{nL_2M_2'}\,C^{\nu(n),LM}_{L_1M_1;L_2M_2} \tag{3.64c}$$

$$\overset{\text{(S33)}}{=} (U^L_{g_u}\cdot\mathbf{g}_u)_{nLM} + \sum_{L_1,L_2}\sum_{M_1',M_2'}\sum_{M'}(U^L_{g_u})_{M,M'}\cdot\big((\mathbf{f}^t_u)_{nL_1M_1'}(\mathbf{g}_u)_{nL_2M_2'}\,C^{\nu(n),LM'}_{L_1M_1';L_2M_2'}\big) \tag{3.64d}$$

$$= U^L_{g_u}\cdot\big(\mathbf{g}_u + \sum_{L_1,L_2}\sum_{M_1',M_2'}(\mathbf{f}^t_u)_{nL_1M_1'}(\mathbf{g}_u)_{nL_2M_2'}\,C^{\nu(n),LM'}_{L_1M_1';L_2M_2'}\big)_{nLM} \tag{3.64e}$$

$$= U^L_{g_u}\cdot(\mathbf{q}_u)_{nLM} = \pi^2(g_u)\cdot(\mathbf{q}_u)_{nLM} = g\cdot(\mathbf{q}_u)_{nLM} \tag{3.64f}$$

For permutation equivariance, it suffices to realize $\sigma(\mathbf{T}) \equiv \mathbf{T}$ due to the symmetric condition in Definition S2 so (3.53) is invariant under σ, the permutation invariance of ψ (see Equation 3.48), and the actions of σ on network layers in ϕ defined for a single dimension $\{(u;v)\}$ are trivial (since $\sigma(u)\equiv u$).

3.10 Appendix

Additional neural network details

Efficient GPU evaluation of spherical harmonics and Clebsch-Gordan coefficients

All $\mathrm{O}(3)$-representation related operations are implemented through element-wise operations on arrays and gather-scatter operations, without the need of recursive computations that can be difficult to parallelize on GPUs at runtime. The real spherical harmonics (RSHs) are computed based on Equations 6.4.47-6.4.50 of [172], which reads:

$$Y_{lm}(\vec{r}) = N^{\mathrm{S}}_{lm} \sum_{t=0}^{[(l-|m|/2)]} \sum_{u=0}^{t} \sum_{v=v_m}^{[|m|/2-v_m]+v_m} C^{lm}_{tuv} (\frac{x}{\|r\|})^{2t+|m|-2(u+v)} (\frac{y}{\|r\|})^{2(u+v)} (\frac{z}{\|r\|})^{l-2t-|m|} \tag{3.65a}$$

$$C^{lm}_{tuv} = (-1)^{t+v-v_m} (\frac{1}{4})^t \binom{l}{t} \binom{l-t}{|m|+t} \binom{t}{u} \binom{|m|}{2v} \tag{3.65b}$$

$$N^{\mathrm{S}}_{lm} = \frac{1}{2^{|m|} l!} \sqrt{\frac{2(l+|m|)!(l-|m|)!}{2^{\delta_{0m}}}} \tag{3.65c}$$

$$v_m = \begin{cases} 0 & \text{if } m \geq 0 \\ \frac{1}{2} & \text{if } m < 0 \end{cases} \tag{3.65d}$$

where $[\cdot]$ is the floor function. The above scheme only requires computing element-wise powers of 3D coordinates and a linear combination with pre-tabulated coefficients. The Clebsch-Gordan (CG) coefficients are first tabulated using their explicit expressions for complex spherical harmonics (CSHs) based on Equation 3.8.49 of Ref. 15, and are then converted to RSH CG coefficients with the transformation matrix between RSHs and CSHs [173].

Multiple input channels

UNiTE is naturally extended to inputs that possess extra feature dimensions, as in the case of AO features **T** described in Section 3.7 the extra dimension equals the cardinality of selected QM operators. Those stacked features is processed by a learnable linear layer $\mathbf{W}^{\mathrm{in}}$ resulting in a fixed-size channel dimension. Each channel is then shared among a subset of convolution channels (indexed by i), instead of using one convolution kernel for all channels i. For the numerical experiments of this work, **T** are mixed into I input channels by $\mathbf{W}^{\mathrm{in}}$ and we assign a convolution channel to each input channel.

Restricted summands in Clebsch-Gordan coupling

For computational efficiency, in the Clebsch-gordan coupling (3.25) (i.e., (3.60)) of a point-wise interaction block, we further restrict the angular momentum indices (l_1, l_2) within the range $\{(l_1, l_2); l_1 + l_2 \leq l_{\mathrm{max}}, l_1 \leq n, l_2 \leq n\}$ where l_{max} is the maximum angular momentum considered in the implementation.

Incorporating geometric information

Because the point cloud of atomic coordinates $\mathbf{x}$ is available in addition to the atomic-orbital-based inputs $\mathbf{T}$, we incorporated such geometric information through the following modified message-passing scheme to extend (3.17):

$$(\tilde{\mathbf{m}}_A^t)_{lm} = \sum_{B \neq A} \bigoplus_{i,j} \Big((\mathbf{m}_{AB}^{t,i})_{lm} + Y_{lm}(\hat{x}_{AB}) \big(\mathbf{W}_i^{l,t} || \mathbf{m}_{AB}^{t,i} || \big) \Big) \cdot \alpha_{AB}^{t,j} \tag{3.66}$$

where Y_{lm} denotes a spherical harmonics of degree l and order m, $\hat{x}_{AB} := \frac{\vec{x}_{AB}}{||\vec{x}_{AB}||}$ denotes the direction vector between atomic centers A and B, and $\mathbf{W}_i^{l,t}$ are learnable linear functions.

Pooling layers

We define schemes for learning different classes of chemical properties with OrbNet-Equi without modifying the base UNiTE model architecture. We use A to denote an atom index, $|A|$ to denote the total number of atoms in the molecule, $z_A \in \mathbb{N}^+$ to denote the atomic number of atom A, and $\vec{x}_A \in \mathbb{R}^3$ to denote the atomic coordinate of atom A.

Energetic properties

A representative target in this family is the molecular electronic energy $E(\mathbf{x})$ (i.e., U_0 in the convention of QM9), which is rotation-invariant and proportional to the system size (i.e., extensive). The pooling operation is defined as:

$$y_\theta = \sum_A \mathbf{W}_\text{o} \cdot ||\mathbf{h}_A^{t_f}|| + b_{z_A}^\text{o} \tag{3.67}$$

which is a direct summation over atom-wise contributions. $\mathbf{W}_\text{o}$ is a learnable linear layer and $b_{z_A}^\text{o}$ are learnable biases for each atomic number z. To account for nuclei contributions to molecular energies, we initialize b_z^o from a linear regression on the training labels with respect to $\{z_A\}$ to speed up training on those tasks. This scheme is employed for learning U_0, U, H, G, ZPVE, and c_v on QM9, the energies part in MD17 and for the OrbNet-Equi/SDC21 model.

Dipole moment $\vec{\mu}$

The dipole moment $\vec{\mu}$ can be thought as a vector in $\mathbb{R}^3$. It is modelled as a combination of atomic charges q_A and atomic dipoles $\vec{\mu}_A$, and the pooling operation is defined as

$$\vec{\mu}_\theta = \sum_A (\vec{R}_A \cdot q_A + \vec{\mu}_A) \tag{3.68}$$

$$q_A = q'_A - \Delta q \quad \text{where} \quad \Delta q := \frac{\sum_A q'_A}{|A|} \tag{3.69}$$

$$q'_A := \mathbf{W}_{\mathrm{o},0} \cdot (\mathbf{h}_A^{tf})_{l=0,p=1} + b^{\mathrm{o}}_{z_A} \tag{3.70}$$

$$(\vec{\mu}_A)_m := \mathbf{W}_{\mathrm{o},1} \cdot (\mathbf{h}_A^{tf})_{l=1,p=1,m} \quad \text{where} \quad m \in \{x, y, z\} \tag{3.71}$$

where $\mathbf{W}_{\mathrm{o},0}$ and $\mathbf{W}_{\mathrm{o},1}$ are learnable linear layers. Equation 3.69 ensures the translation invariance of the prediction through charge neutrality.
Note that OrbNet-Equi is trained by directly minimizing a loss function $\mathcal{L}(\vec{\mu}, \vec{\mu}_\theta)$ between the ground truth and the predicted molecular dipole moment vectors. For the published QM9 reference labels [65] only the dipole norm $\mu := ||\vec{\mu}||$ is available; we use the same pooling scheme to readout $\vec{\mu}_\theta$ but train on $\mathcal{L}(\mu, ||\vec{\mu}_\theta||)$ instead to allow for comparing to other methods in Table 3.1.

Polarizability α

For isotropic polarizability α, the pooling operation is defined as

$$\alpha_\theta = \sum_A (\alpha_A + \vec{R}_A \cdot \vec{p}_A) \tag{3.72}$$

$$\alpha_A := \mathbf{W}_{\mathrm{o},0} \cdot (\mathbf{h}_A^{tf})_{l=0,p=1} + b^{\mathrm{o}}_{z_A} \tag{3.73}$$

$$\vec{p}_A = \vec{p'}_A - \Delta\vec{p} \quad \text{where} \quad \Delta\vec{p} := \frac{\sum_A \vec{p'}_A}{|A|} \tag{3.74}$$

$$(\vec{p'}_A)_m := \mathbf{W}_{\mathrm{o},1} \cdot (\mathbf{h}_A^{tf})_{l=1,p=1,m} \quad \text{where} \quad m \in \{x, y, z\} \tag{3.75}$$

Molecular orbital properties

For frontier molecular orbital energies, a *global-attention* based pooling is employed to produce intensive predictions:

$$a_A = \mathrm{Softmax}(\mathbf{W}_\mathrm{a} \cdot ||\mathbf{h}_A^{tf}||) := \frac{\mathbf{W}_\mathrm{a} \cdot ||\mathbf{h}_A^{tf}||}{\sum_A \mathbf{W}_\mathrm{a} \cdot ||\mathbf{h}_A^{tf}||} \tag{3.76}$$

$$y_\theta = \sum_A a_A \cdot (\mathbf{W}_\mathrm{o} \cdot ||\mathbf{h}_A^{tf}|| + b^{\mathrm{o}}_{z_A}) \tag{3.77}$$

where $\mathbf{W}_\mathrm{a}$ and $\mathbf{W}_\mathrm{o}$ are learnable linear layers and $b^\mathrm{o}_{z_A}$ are learnable biases for each atomic number z. Similar to energy tasks, we initialize b^o_z from a linear fitting on the targets to precondition training.
We take the difference between the predicted HOMO energies (ϵ_HOMO) and LUMO energies (ϵ_LUMO) as the HOMO-LUMO Gap ($\Delta\epsilon$) predictions.

Electronic spatial extent $\langle R^2 \rangle$

The pooling scheme for $\langle R^2 \rangle$ is defined as:

$$\langle R^2 \rangle_\theta = \sum_A (||\vec{R}_A - \vec{R}_0||^2 \cdot q_A + s_A) \tag{3.78}$$

$$\vec{R}_0 := \frac{\sum_A (\vec{R}_A \cdot q_A + \vec{\mu}_A)}{\sum_A q_A} \tag{3.79}$$

$$q_A := \mathbf{W}_{\mathrm{o},0} \cdot (\mathbf{h}^{t_f}_A)_{l=0,p=1} + b^\mathrm{o}_{z_A} \tag{3.80}$$

$$(\vec{\mu}_A)_m := \mathbf{W}_{\mathrm{o},1} \cdot (\mathbf{h}^{t_f}_A)_{l=1,p=1,m} \quad \text{where} \quad m \in \{x, y, z\} \tag{3.81}$$

$$s_A := \mathbf{W}_{\mathrm{o},2} \cdot (\mathbf{h}^{t_f}_A)_{l=0,p=1} \tag{3.82}$$

where $\mathbf{W}_{\mathrm{o},0}$, $\mathbf{W}_{\mathrm{o},1}$, and $\mathbf{W}_{\mathrm{o},2}$ are learnable linear layers.

Electron densities $\rho(\vec{r})$

Both the ground truth and predicted electron densities $\rho(\vec{r})$ are represented as a superposition of atom-centered density fitting basis $\{\chi\}$,

$$\rho(\vec{r}) = \sum_A^{N_\mathrm{atom}} \sum_l^{l_\mathrm{max}(z_A)} \sum_{m=-l}^{l} \sum_n^{n_\mathrm{max}(z_A,l)} d^{nlm}_A \, \chi^{nlm}_A(\vec{r}) \tag{3.83}$$

similar to the approach employed in [131]; here we use the def2-TZVP-JKFIT density fitting basis for $\{\chi\}$. Computational details regarding obtaining the reference density coefficients d^{nlm}_A are given in Section 3.8, and the training loss function is defined in SI 3.10. The pooling operation to predict $\rho(\vec{r})$ from UNiTE is defined as

$$\hat{d}^{nlm}_A := \left(\mathbf{W}^\mathrm{d}_{z_A,l} \cdot (\mathbf{h}^{t_f}_A)_{l,p=1,m}\right)_n \tag{3.84}$$

where $\mathbf{W}^\mathrm{d}_{z,l}$ are learnable weight matrices specific to each atomic number z and angular momentum index l, and z_A denotes the atomic number of atom A. This atom-centered expansion scheme compactly parameterizes the model-predicted density $\hat{\rho}(\vec{r})$. We stress that all UNiTE neural network parameters except for this density pooling layer (3.84) are independent of the atomic numbers z.

Time complexity

The asymptotic time complexity of UNiTE model inference is $\mathcal{O}(BNI)$, where B is the number of non-zero elements in $\mathbf{T}$, and I denotes the number of convolution channels in a convolution block (3.15). This implies UNiTE scales as $\mathcal{O}(N(nd)^N)$ if the input is dense, but can achieve a lower time complexity for sparse inputs, e.g., when long-range cutoffs are applied. We note that in each convolution block (3.15) the summand $T_{\vec{u},\vec{v}} \cdot \prod_{j=2}^{N} \left(\rho_{u_j}(\mathbf{h}^t_{u_j})\right)^i_{v_j} \neq 0$ only if the tensor coefficient $T_{\vec{u},\vec{v}} \neq 0$; therefore (3.15) can be exactly evaluated using $((N-1)BI)$ arithmetic operations. In each message passing block (3.17) the number of arithmetic operations scales as $\mathcal{O}(B'I)$ where B' is the number of indices $\vec{u}$ such that $\mathbf{m}^t_{\vec{u}} \neq 0$, and $B' \leq B$. The embedding block ϕ and the point-wise interaction block ψ has $\mathcal{O}(d)$ time complexities since they act on each point independently and do not contribute to the asymptotic time complexity.

Additonal numerical results

The QM9 dataset

We provide the QM9 MAEs on all 12 target properties as reported in Table 3.1. The standardized MAE and standardized log MAE in Table 3.1 are computed following the Appendix C of [53]. Uncertainties for test MAEs are obtained by statistical bootstrapping with sample size 5000 and 100 iterations.

MD17 and rMD17 datasets

The MD17 dataset [139] contains energy and force labels from molecular dynamics trajectories of small organic molecules, and is used to benchmark ML methods for modelling a single instance of a molecular potential energy surface. Recently the revised-MD17 (rMD17) dataset[137] was reported with improved label fidelity. For both the MD17 and the rMD17 dataset, We train OrbNet-Equi simultaneously on energies and forces of 1000 geometries of each molecule and test on another 1000 geometries of the same molecule, using previously reported dataset splits (Section 3.10). All results are obtained without performing system-specific hyperparameter selections and without using additional regularization techniques such as model ensembling or stochastic weight averaging [174].

As shown in Table 3.2 and Table 3.3, OrbNet-Equi with direct learning achieves competitive accuracy when compared against the best results reported by kernel methods [137, 175] and graph neural networks (GNNs) [53, 109, 138]. Additional

Table 3.1: Test mean absolute errors (MAEs) on QM9 for atomistic deep learning models and OrbNet-Equi trained on 110k samples. OrbNet-Equi results on first 8 tasks are obtained by training on the residuals between the DFT reference labels and the tight-binding QM model estimations (delta-learning), because the tight-binding results for these targets can be directly obtained from the single-point calculation that featurizes the molecule. Results on the last 4 targets are obtained through directly training on the target properties (direct-learning).

Target	Unit	SchNet	Cormorant	DimeNet++	PaiNN	SphereNet	OrbNet-Equi
μ	mD	33	38	29.7	12	26.9	6.3±0.2
ϵ_{HOMO}	meV	41	32.9	24.6	27.6	23.6	9.9±0.02
ϵ_{LUMO}	meV	34	38	19.5	20.4	18.9	12.7±0.3
$\Delta\epsilon$	meV	63	38	32.6	45.7	32.3	17.3±0.3
U_0	meV	14	22	6.3	5.9	6.3	3.5±0.1
U	meV	19	21	6.3	5.8	7.3	3.5±0.1
H	meV	14	21	6.5	6.0	6.4	3.5±0.1
G	meV	14	20	7.6	7.4	8.0	5.2±0.1
α	a_0^3	0.235	0.085	0.044	0.045	0.047	0.036±0.002
$\langle R^2 \rangle$	a_0^2	0.073	0.961	0.331	0.066	0.292	0.030±0.001
ZPVE	meV	1.7	2.0	1.2	1.3	1.1	1.11±0.04
c_v	$\frac{\text{cal}}{\text{molK}}$	0.033	0.026	0.023	0.024	0.022	0.022±0.001
std. MAE	%	1.76	1.44	0.98	1.01	0.94	0.47
log. MAE	-	-5.2	-5.0	-5.7	-5.8	-5.7	-6.4

Table 3.2: Test MAEs on the rMD17 dataset in terms of energies (in meV) and forces (in meV/Å) for models trained on 1000 sample geometries for each molecular system. For OrbNet-Equi, both direct learning and delta learning results are reported.

Molecule		FCHL19 [137]	NequIP (l = 3) [138]	OrbNet-Equi (direct learning)	OrbNet-Equi (delta learning)
Aspirin	Energy	6.2	2.3	2.4	1.8
	Forces	20.9	8.5	7.6	6.1
Azobenzene	Energy	2.8	0.7	1.1	0.63
	Forces	10.8	3.6	4.2	2.7
Ethanol	Energy	0.9	0.4	0.62	0.42
	Forces	6.2	3.4	3.7	2.6
Malonaldehyde	Energy	1.5	0.8	1.2	0.80
	Forces	10.2	5.2	7.1	4.6
Naphthalene	Energy	1.2	0.2	0.46	0.27
	Forces	6.5	1.2	2.6	1.5
Paracetamol	Energy	2.9	1.4	1.9	1.2
	Forces	12.2	6.9	7.1	4.5
Salicylic Acid	Energy	1.8	0.7	0.73	0.52
	Forces	9.5	4.0	3.8	2.9
Toluene	Energy	1.6	0.3	0.45	0.27
	Forces	8.8	1.6	2.5	1.6
Uracil	Energy	0.4	0.4	0.58	0.35
	Forces	4.2	3.2	3.8	2.4
Benzene	Energy	0.3	0.04	0.07	0.02
	Forces	2.6	0.3	0.73	0.27

Table 3.3: OrbNet-Equi test force MAEs (in kcal/mol/Å) on the original MD17 dataset using 1000 training geometries.

Molecule	OrbNet-Equi (direct learning)	OrbNet-Equi (delta learning)
Aspirin	0.156	0.118
Ethanol	0.092	0.069
Malonaldehyde	0.159	0.128
Naphthalene	0.064	0.048
Salicylic Acid	0.097	0.067
Toluene	0.072	0.057
Uracil	0.098	0.072

Table 3.4: OrbNet-Equi inference time breakdowns (mean/std in milliseconds) for the calculation of energy and forces on the Hutchison dataset [64].

Feature generation	NN inference	NN back propagation	Nuclear gradients calculation
85.8 ± 40.1	181 ± 83	273 ± 73	33.2 ± 1.8

performance gains are observed when the models are trained with the delta learning strategy, resulting in the lowest test MAEs for both energy and forces on most of the test molecules. We note that MD17 represents an ideal case scenario where abundant high-level reference calculations are available on the potential energy surface of a single chemical system and the configurations of interest only spans a thermally-accessible energy scale. Despite the highly-interpolative nature of this learning task, OrbNet-Equi results still matches the accuracy that can be achieved with state-of-the-art neural network potentials for which system-dependent optimizations are often employed.

Inference timings

Wall-clock timing results for evalulating the pretrained OrbNet-Equi/SDC21 model are reported on the Hutchison dataset [64] which represents the distribution of realistic drug-like molecules,. All timing results are obtained using 16 cores of an Intel Xeon Gold 6130 CPU. We note that due to the use of a Lagrangian formalism we previous developed [75] to efficiently compute the analytic nuclear gradients in which the operator derivatives with respect to atomic coordinates are not explicitly evaluated, for such medium-sized organic molecules the overhead for computing energy and forces in addition to only computing the energies is still dominated by neural network back propagation.

Table 3.5: Summary statistics of representative semi-empirical quantum mechanics (GFN-xTB and GFN2-xTB), machine learning (ANI-2x), density functional theory (B97-3c) methods and OrbNet-Equi/SDC21 on down-steam tasks. See section 3.10 regarding results on geometry optimization tasks.

Task	Dataset	Metric	GFN-xTB	GFN2-xTB	ANI-2x	B97-3c	OrbNet-Equi/SDC21
Conformer ordering	Hutchison [64]	Med. R^2 / DLPNO-CCSD(T)	0.62±0.04	0.64±0.04	0.63±0.06	0.90±0.01	0.87±0.02
Conformer ordering	Hutchison [64]	Med. R^2 / ωB97X-D3/def2-TZVP	0.64±0.04	0.69±0.04	0.68±0.04	0.97±0.01	0.96±0.01
Torsion profiles	TorsionNet [146]	MAE[3] (kcal/mol)	0.948±0.017	0.731±0.013	0.893±0.017	0.284±0.006	0.173±0.003
Geometry optimization	ROT34 [70]	Avg. RMSD (Å)	0.227±0.087	0.210±0.072	-	0.063±0.013	0.045±0.005
Geometry optimization	MCONF [71]	Avg. RMSD (Å)	0.899±0.106	0.603±0.064	-	0.511±0.072	0.227±0.042

[3] With respect to ωB97X-D3/def2-TZVP. Note that ANI-2x is trained on a different DFT theory and the number is provided for reference only.

Table 3.6: MAEs (in kcal/mol) of binding energy predictions on the S66x10 [147] dataset, computed for different inter-molecular distances in r_e unit with r_e being the equilibrium inter-molecular distance. ωB97X-D3/def2-TZVP binding energy results are used as the reference level of theory. Standard errors of the mean are reported in the parentheses.

Distance (r_e)	GFN-xTB	GFN2-xTB	ANI-2x	B97-3c	OrbNet-Equi/SDC21
0.7	6.7584(2.1923)	6.8887(2.2193)	2.3236(0.5964)	1.7856(0.6036)	1.6443(0.4657)
0.8	2.6225(0.7901)	2.8569(0.8791)	1.1433(0.2438)	0.9751(0.2456)	0.9241(0.2836)
0.9	1.4087(0.1956)	1.3301(0.2715)	1.0103(0.1603)	0.5922(0.1034)	0.5336(0.1515)
0.95	1.4365(0.1694)	1.2018(0.1807)	0.9752(0.1589)	0.5018(0.0837)	0.4341(0.1124)
1.0	1.5552(0.1730)	1.1927(0.1773)	0.9688(0.1484)	0.4433(0.0673)	0.3540(0.0881)
1.05	1.5962(0.1740)	1.1960(0.1845)	0.9501(0.1461)	0.3756(0.0525)	0.3090(0.0872)
1.1	1.5577(0.1751)	1.1800(0.1848)	0.9404(0.1620)	0.3049(0.0435)	0.3328(0.0946)
1.25	1.2690(0.1694)	0.9764(0.1715)	0.9645(0.1706)	0.1344(0.0241)	0.4432(0.0736)
1.5	0.8270(0.1533)	0.5764(0.1211)	0.8503(0.1362)	0.0610(0.0165)	0.4697(0.0687)
2.0	0.3346(0.0899)	0.1664(0.0370)	0.7139(0.2071)	0.0294(0.0078)	0.2820(0.0744)

Table 3.7: Subset-averaged WTMAD-2 (WTMAD-$2_i = \frac{1}{N_i}\sum_j \text{WTAD}_{i,j}$, see Methods 3.8) on the GMTKN55 collection of benchmarks, reported for all potential energy methods considered in this study against the CCSD(T)/CBS reference values. Standard errors of the mean are reported in the parentheses. For cases in which no reaction within a subset is supported by a method, results are marked as "-". The OrbNet-Equi/SDC21 (filtered) column corresponds to OrbNet-Equi/SDC21 evaluated on reactions that consist of chemical elements and electronic states appeared in the SDC21 training dataset, as shown in Figure 3.6b.

Group	Subset	GFN1-xTB	GFN2-xTB	ANI-2x	B97-3c	ωB97xD3	OrbNet-Equi/SDC21	OrbNet-Equi/SDC21 (filtered)
Prop. small	W4-11	32.5(1.5)	22.0(1.2)	-	1.4(0.1)	0.7(0.1)	31.1(1.5)	-
	G21EA	392.6(185.7)	158.3(12.3)	-	13.9(1.3)	12.0(1.4)	254.1(182.7)	-
	G21IP	31.2(2.0)	26.4(2.0)	-	0.8(0.1)	0.7(0.1)	8.1(1.3)	-
	DIPCS10	26.2(2.6)	23.9(5.4)	-	0.4(0.1)	0.5(0.1)	4.4(2.2)	4.4(2.5)
	PA26	48.8(0.5)	49.0(0.6)	-	1.7(0.2)	1.0(0.1)	2.5(0.4)	2.5(0.4)
	SIE4x4	163.6(27.8)	108.5(14.2)	-	38.0(5.2)	20.5(3.1)	133.0(30.1)	-
	ALKBDE10	39.2(8.7)	35.5(9.0)	-	4.5(1.0)	3.0(0.8)	42.6(7.3)	-
	YBDE18	18.6(2.7)	22.1(2.7)	-	5.9(1.0)	2.5(0.4)	23.9(2.0)	20.3(1.7)
	AL2X6	24.0(6.5)	23.2(3.7)	-	3.5(1.2)	4.9(0.5)	20.1(4.8)	-
	HEAVYSB11	23.6(2.8)	6.0(1.6)	-	2.5(0.5)	2.5(0.3)	28.1(5.2)	-
	NBPRC	22.5(6.4)	21.6(6.1)	-	3.2(1.0)	3.4(1.0)	23.8(4.5)	23.8(4.5)
	ALK8	47.7(19.0)	21.7(7.4)	-	3.2(0.8)	3.7(1.2)	68.2(40.9)	68.2(40.9)
	RC21	35.1(4.5)	37.7(4.4)	-	10.2(1.3)	5.2(0.7)	36.2(4.9)	-
	G2RC	32.5(5.7)	24.3(4.3)	34.3(7.9)	9.2(1.4)	5.1(0.8)	16.2(2.9)	16.2(3.0)
	BH76RC	56.2(8.7)	49.2(9.6)	218.5(-)	9.7(2.4)	6.1(0.8)	46.0(7.0)	45.1(14.8)
	FH51	22.1(2.8)	20.9(3.3)	24.5(4.2)	8.1(1.1)	4.5(0.5)	9.4(2.1)	9.4(2.1)
	TAUT15	108.1(21.9)	18.3(4.8)	46.9(9.9)	31.9(4.8)	19.6(3.1)	21.8(5.5)	21.8(5.5)
	DC13	38.9(10.1)	33.9(7.6)	23.5(13.3)	11.7(2.0)	7.0(1.5)	32.8(13.7)	13.3(3.7)
Prop. large	MB16-43	18.5(2.2)	31.8(3.1)	-	3.3(0.4)	4.9(0.4)	16.5(2.6)	31.3(18.6)
	DARC	27.7(1.6)	31.1(2.2)	9.6(1.7)	7.6(1.0)	2.2(0.7)	1.3(0.3)	1.3(0.3)
	RSE43	50.7(3.7)	56.9(3.9)	-	26.1(2.0)	10.7(0.8)	44.2(7.3)	-
	BSR36	8.2(0.7)	9.7(1.6)	31.0(3.3)	6.7(0.5)	15.3(1.6)	24.1(2.4)	24.1(2.4)
	CDIE20	28.6(4.8)	25.3(4.1)	50.9(9.5)	27.8(3.1)	10.1(2.3)	16.0(3.6)	16.0(3.6)
	ISO34	24.6(3.7)	26.9(4.2)	50.5(39.8)	7.3(1.4)	4.6(0.7)	7.3(3.3)	7.3(3.3)
	ISOL24	28.3(4.4)	30.3(4.7)	18.9(4.1)	13.5(2.8)	7.1(1.2)	7.6(1.3)	7.6(1.3)
	C60ISO	4.6(1.0)	3.4(0.9)	26.0(2.9)	3.6(1.1)	7.8(1.2)	2.3(0.4)	2.3(0.4)
	PArel	55.8(13.2)	72.0(18.5)	-	22.1(6.2)	8.2(2.0)	43.5(9.3)	43.5(9.3)

Table 3.8: Subset-averaged WTMAD-2 on the GMTKN55 collection of benchmarks, continued.

Group	Subset	GFN1-xTB	GFN2-xTB	ANI-2x	B97-3c	ωB97xD3	OrbNet-Equi/SDC21	OrbNet-Equi/SDC21 (filtered)
React. barriers	BH76	64.2(6.6)	59.9(6.6)	68.9(55.2)	21.0(1.7)	6.9(0.6)	57.0(6.0)	35.7(6.7)
	BHPERI	25.4(2.1)	27.9(2.3)	65.7(9.6)	12.5(0.8)	7.8(0.9)	10.5(2.4)	10.5(2.4)
	BHDIV10	10.5(2.3)	10.2(2.6)	13.8(7.2)	7.3(1.5)	1.3(0.3)	8.2(2.1)	8.2(2.1)
	INV24	10.4(2.2)	5.9(1.0)	26.2(5.7)	3.5(0.8)	2.9(0.8)	20.1(4.9)	20.1(4.9)
	BHROT27	21.5(3.2)	10.6(1.6)	12.9(3.2)	5.5(1.0)	4.3(0.7)	5.4(0.9)	5.4(0.9)
	PX13	14.1(3.1)	4.7(1.1)	22.7(7.4)	12.1(0.8)	5.4(0.7)	22.1(5.7)	22.1(5.7)
	WCPT18	8.6(1.3)	6.2(1.1)	10.0(1.6)	8.9(1.2)	3.5(0.6)	12.1(1.6)	12.1(1.6)
Inter. mol. NCI	RG18	31.8(7.1)	11.0(3.1)	-	11.8(3.0)	11.1(1.8)	53.6(13.1)	-
	ADIM6	17.1(2.8)	19.5(4.2)	5.8(1.2)	8.9(2.1)	6.2(2.2)	4.5(1.2)	4.5(1.2)
	S22	10.4(1.7)	5.9(0.9)	11.7(2.8)	2.2(0.4)	2.8(0.5)	4.1(0.6)	4.1(0.6)
	S66	11.2(0.8)	7.6(0.6)	11.5(1.2)	3.4(0.4)	5.4(0.4)	5.1(0.5)	5.1(0.5)
	HEAVY28	30.0(9.6)	27.8(5.0)	-	36.8(4.0)	12.1(2.2)	54.3(8.8)	-
	WATER27	5.2(0.7)	2.1(0.3)	33.4(8.1)	6.6(0.9)	10.0(1.6)	12.0(1.5)	12.0(1.5)
	CARBHB12	6.3(1.4)	16.9(6.7)	58.3(17.1)	19.5(4.3)	7.8(1.2)	19.7(3.9)	19.7(3.9)
	PNICO23	31.0(6.7)	14.7(2.7)	251.8(7.3)	21.8(2.6)	5.0(0.7)	39.1(9.5)	39.1(9.5)
	HAL59	16.6(2.6)	15.8(1.7)	74.4(41.7)	20.1(2.9)	4.2(0.4)	33.5(6.0)	33.5(6.0)
	AHB21	11.8(2.6)	7.5(1.2)	-	8.3(1.2)	8.6(1.2)	18.6(3.7)	18.6(3.7)
	CHB6	8.4(3.9)	11.5(2.3)	-	2.9(1.2)	2.8(0.9)	24.0(10.7)	24.0(10.7)
	IL16	3.0(0.6)	2.2(0.3)	-	1.2(0.3)	1.1(0.2)	2.5(0.4)	2.5(0.4)
Intra. mol. NCI	IDISP	26.1(14.4)	27.1(17.0)	82.5(45.7)	15.6(5.3)	11.1(3.6)	19.7(8.9)	19.7(8.9)
	ICONF	45.7(13.5)	28.3(4.8)	73.3(27.2)	6.6(1.5)	5.9(1.4)	29.8(10.3)	29.8(10.3)
	ACONF	20.5(3.4)	6.0(1.3)	4.8(1.1)	6.6(0.8)	2.7(0.3)	1.6(0.4)	1.6(0.4)
	Amino20x4	26.0(2.1)	22.2(2.2)	23.2(2.2)	7.6(0.7)	6.1(0.5)	7.0(0.6)	7.0(0.6)
	PCONF21	76.0(14.1)	61.6(10.1)	77.2(18.5)	29.0(5.4)	11.7(2.0)	17.8(2.5)	17.8(2.5)
	MCONF	16.5(1.2)	19.7(1.5)	9.4(0.9)	3.8(0.4)	5.5(0.4)	5.4(0.5)	5.4(0.5)
	SCONF	30.9(12.0)	20.3(6.2)	30.5(7.0)	9.5(2.1)	3.7(1.2)	6.6(1.3)	6.6(1.3)
	UPU23	12.3(1.6)	28.9(2.9)	-	5.0(0.8)	9.4(1.0)	10.7(1.5)	10.7(1.5)
	BUT14DIOL	19.4(1.6)	25.4(1.8)	28.8(1.7)	8.4(0.5)	8.3(0.4)	11.9(0.5)	11.9(0.5)

Downstream benchmarks

Table 3.5-3.7 provide summary statistics of method performances on downstream main-group quantum chemistry benchmarks considered in this study.

Additional computational details for geometry optimization experiments are provided as follows. The symmetry-corrected root-mean-square-deviations (RMSDs) are computed between the test geometries and the reference DFT (ωB97X-D3/def2-TZVP) geometries following a Hungarian algorithm [176] to account for equivalent atoms. OrbNet-Equi/SDC21 and GFN-xTB results are obtained using Entos Qcore version 1.1.0 with the L-BFGS algorithm with tight thresholds (energy change after iteration < 1E-6 a.u., gradient RMS < 4.5E-4 a.u, max element of gradient < 3E-4 a.u., optimization step RMS < 1.8E-3 a.u., and max element of optimization step < 1.2E-3 a.u.). GFN2-xTB results are obtained with the XTB [66] package with default settings. ANI-2x energy calculations are performed with the TorchANI [177] package and geometry optimizations are performed with the geomeTRIC optimizer [95] with default settings. Using this software setting, ANI-2x optimizations are found unable to converge on 21 out of 52 conformers on MCONF and 1 out of 12 conformers on ROT34, and the average errors are reported as “-”. On the subsets that ANI-2x geometry optimizations converged, the average RMSD for ANI-2x is 0.154±0.038 on ROT34 and 0.324±0.026 on MCONF versus the reference geometries. The histograms and kernel density estimations displayed in Figure 3.5d are computed on the subsets where all methods successfully converged, that is, 11 conformers from ROT34 and 31 conformers from MCONF.

Hyperparameters and training details

Model hyperparameters

We use the same set of model hyperparameters to obtain all numerical experiment results on open benchmarks reported in this work. The hyperparameters are summarized in Table 3.9 and Table 3.10. The hyperparameters for the OrbNet-Equi/SDC21 model is locally optimized based on a 4219-sample validation set from the training data distribution, with the additional difference of using LayerNorm [158] for mean μ and variance σ estimates, and $\epsilon = 0.5$ in all EvNorm layers. This choice is made to improve model robustness when applied to extrapolative molecular geometries.

Table 3.9: The model hyperparameters for OrbNet-Equi used for benchmarking studies.

Symbol	Meaning	Defined in	Value(s)
N	Total number of feature channels in $\mathbf{h}^t$	Methods. 3.7, (3.12)	256
N_{lp}	Number of feature channels for each (l, p) in $\mathbf{h}^t$	Methods. 3.7, (3.12)	See Table 3.10
t_1	Number of convolution-message-passing update steps	Methods. 3.7	4
t_2	Number of post-update point-wise interaction modules	Methods. 3.7	4
I	Number of convolution channels i	Methods. 3.7, (3.15)	8
J	Number of attention heads j	Methods. 3.7, (3.17)	8
d^{MLP}	Depth of MLPs	MLPs in (3.24)-(3.26)	2
	Activation function	MLPs in (3.24)-(3.26)	Swish [178]
N^{ξ}	Number of Radial basis functions ξ	Methods. 3.7, (3.21)	16
	Estimation scheme for (μ, σ) for EvNorm, $t < t_1$	Methods. 3.7, (3.22)	BatchNorm [99]
	Estimation scheme for (μ, σ) for EvNorm, $t \geq t_1$	Methods. 3.7, (3.22)	LayerNorm [158]
	Initialization of β_{nlp} in EvNorm layers	Methods. 3.7, (3.22)	$\text{Uniform}\big([0.5, 1.5)\big)$
ϵ	Stability factor ϵ in EvNorm layers	Methods. 3.7, (3.22)	0.1
	Total number of parameters	-	2.1M

Table 3.10: The number of feature channels N^{h}_{lp} for each representation group (l, p) of $\mathbf{h}^t$ used in this work across all values of t. Note that $l_{\max} = 4$.

N_{lp}	$l = 0$	$l = 1$	$l = 2$	$l = 3$	$l = 4$
$p = +1$	128	48	24	12	6
$p = -1$	24	8	4	2	0

Training

For all training setups we use the Adam optimizer [97] with maximal learning rate 5×10^{-4} and parameters $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-4}$. The loss function denoted as $\mathcal{L}$ below refers to a SmoothL1Loss function [179]. Batch sizes and the total number of epochs are adjusted for different benchmarks to account for their vastly different training set sizes, as detailed below. No additional regularization such as weight decay or early stopping is employed.

QM9

For QM9 tasks we optimize the model using the loss $\mathcal{L}(y, y_\theta)$ for each target y. We use a batch size of 64 when the training sample size > 1000, and a batch size of 16 for smaller training sizes. We employ a learning rate schedule of first performing linear warmup for 100 epochs to the maximal learning rate followed by a cosine learning rate annealing [180] for 200 epochs. Models are trained on a single Nvidia Tesla V100-SXM2-32GB GPU, taking around 36 hours for training runs with 110k training samples.

MD17

For MD17, we optimize the model by simultaneously training on energies $E(\mathbf{R})$ and forces $\mathbf{F}(\mathbf{R})$, using the following loss function:

$$\mathcal{L}_{\mathrm{E+F}}(E, \mathbf{F}; E_\theta, \mathbf{F}_\theta) := c_{\mathrm{E}} \cdot \mathcal{L}(E(\mathbf{R}); E_\theta(\mathbf{R})) + c_{\mathrm{F}} \cdot \frac{1}{3|A|} \sum_{A}^{|A|} \sum_{m \in \{x,y,z\}} \mathcal{L}(-\frac{\partial E_\theta(\mathbf{R}))}{\partial R_{A,m}} - F_{A,m}(\mathbf{R})) \tag{3.85}$$

Following previous works [51, 53, 138], we set $c_{\mathrm{E}} = 1$, $c_{\mathrm{F}} = 1000$ for training on the rMD17 labels, and $c_{\mathrm{E}} = 0$, $c_{\mathrm{F}} = 1000$ for training on the original MD17 labels. For each molecular system, we use the 1000 geometries of the ‘train 01’ subset given by [137] for training and the 1000 geometries of the ‘test 01’ subset for testing. We use a batch size of 8, and train the model on a single Nvidia Tesla V100-SXM2-32GB

GPU for 1500 epochs using a step decay learning rate schedule, taking around 30 hours for each training run.

For proof-of-principle purposes, the gradients of tight-binding features with respect to atomic coordinates are obtained using finite difference with a 5-point stencil for each degree of freedom. The grid spacing between the stencil points is set to 0.01 Bohr. We note that in principle, this cost of evaluating and storing feature gradients can be avoided if the electronic structure method is implemented with back-propagation and the model can be trained end-to-end on both energy and force labels.

Electron densities

For electron density, we train the models on the analytic L^2 density loss following [27]:

$$\mathcal{L}_\rho(\rho, \hat{\rho}) := \int \|\rho(\vec{r}) - \hat{\rho}(\vec{r})\|^2 d\vec{r} = (\mathbf{d} - \hat{\mathbf{d}})^T \, \mathbf{S}^\rho \, (\mathbf{d} - \hat{\mathbf{d}}) \tag{3.86}$$

where the density coefficients $\mathbf{d} := \bigoplus_{A,n,l,m} d_A^{nlm}$ are defined in (3.7), and $\mathbf{S}^\rho$ is the overlap matrix of the density fitting basis $\{\chi\}$. A sparse COO format is used for $\mathbf{S}^\rho$ to efficiently compute $\mathcal{L}_\rho$ during batched training. We use a batch size of 64 and a cosine annealing learning schedule for training; the models are trained on a single Nvidia Tesla V100-SXM2-32GB GPU for 2000 epochs on the BFDb-SSI dataset taking 10 hours, and for 500 epochs on the QM9 dataset taking 120 hours.

The OrbNet-Equi/SDC21 model

The training dataset (see Methods 3.8) contains different geometries b_η for each molecule η in the dataset. We train on a loss function following [76]:

$$\begin{aligned} \mathcal{L}_\mathrm{G}(E(\eta, b_\eta), E_\theta(\eta, b_\eta)) := {} & \mathcal{L}(E(\eta, b_\eta), E_\theta(\eta, b_\eta) \\ & + c_\mathrm{G} \cdot \mathcal{L}(E(\eta, b_\eta) - E(\eta, \hat{b}_\eta), E_\theta(\eta, b_\eta) - E_\theta(\eta, \hat{b}_\eta)) \end{aligned} \tag{3.87}$$

where $\hat{b}_\eta$ is a geometry randomly sampled from all the geometries $\{b_\eta\}$ of each molecule η within each mini-batch during training. We use $c_\mathrm{G} = 10$ in this work. We train the model for 125 epochs on a Nvidia Tesla V100-SXM2-32GB GPU using a batch size of 64 and a cosine annealing learning rate schedule taking 64 hours.

Chapter 4

MULTISCALE EQUIVARIANT SCORE-BASED GENERATIVE MODELING FOR DYNAMIC-BACKBONE PROTEIN-LIGAND STRUCTURE PREDICTION

This chapter is based on the following publication:

[1] Zhuoran Qiao, Weili Nie, Arash Vahdat, Thomas F. Miller III, and Animashree Anandkumar. "Dynamic-Backbone Protein-Ligand Structure Prediction with Multiscale Generative Diffusion Models". In: *arXiv preprint arXiv:2209.15171* (2022). *In submission.* To appear at *Machine Learning in Structural Biology workshop at NeurIPS 2022* as a Contributed Talk. DOI: `10.48550/ARXIV.2209.15171`.

Abstract

Molecular complexes formed by proteins and small-molecule ligands are ubiquitous, and predicting their 3D structures can facilitate both biological discoveries and the design of novel enzymes or drug molecules. Here we propose NeuralPLexer, a deep generative model framework to rapidly predict protein-ligand complex structures and their fluctuations using protein backbone template and molecular graph inputs. NeuralPLexer jointly samples protein and small-molecule 3D coordinates at an atomistic resolution through a generative model that incorporates biophysical constraints and inferred proximity information into a time-truncated diffusion process. The reverse-time generative diffusion process is learned by a novel stereochemistry-aware equivariant graph transformer that enables efficient, concurrent gradient field prediction for all heavy atoms in the protein-ligand complex. NeuralPLexer outperforms existing physics-based and learning-based methods on benchmarking problems including fixed-backbone blind protein-ligand docking and ligand-coupled binding site repacking. Moreover, we identify preliminary evidence that NeuralPLexer enriches bound-state-like protein structures when applied to systems where protein folding landscapes are significantly altered by the presence of ligands. Our results reveal that a data-driven approach can capture the structural cooperativity among protein and small-molecule entities, showing promise for the computational identification of novel drug targets and the end-to-end differentiable design of functional small-molecules and ligand-binding proteins.

4.1 Introduction

Protein structures are dynamically modulated by their interactions with small-molecule ligands, triggering downstream responses that are crucial to the regulation of biological functions [181–183]. Proposing ligands that selectively target protein conformations has become an increasingly important strategy in small-molecule-based therapeutics [184–186]. However, computational prediction of protein-ligand structures that are coupled to receptor conformational responses is still hampered by the prohibitive cost of physically simulating slow protein state transitions [187, 188], as well as the static nature of existing protein folding prediction algorithms [5, 189]. While several schemes have been proposed to remedy these issues [190–199], such methods often require case-specific expert interventions and lack a unified framework to predict 3D structures in a systematic and cooperative fashion.

A data-driven approach may facilitate the study of protein-small-molecule interactions on many aspects. One important category of under-addressed problem in structural biology is the prediction of protein-ligand complex structures with significant binding-induced protein conformational changes, which is common for those involved in allosteric regulations. The identification of an unobserved non-native proteome for proposing new drugs with unconventional action mechanism. Traditional structure-based drug design are largely limited to inhibitors for proteins with well-characterized binding pockets, but recent experimental evidences suggest that a large fraction of the human proteome are potential drug targets because of under-characterized ligand-specific conformational changes. Such allosteric modulators raise immense opportunities for small-molecule-based therapeautics, but the discovery of such functional molecules has been mostly serendipitous due to the lack of experimental approaches to systematically resolve dynamic protein structures, the prohibitive cost of predicting milisecond-scale protein state transition behaviors using physical simulations, as well as the static nature of existing protein structure prediction algorithms such as AlphaFold. A computational method that rapidly generates protein-ligand complex structures can therefore significantly aid the process of unconventional target identification and rational allosteric modulator design.

Here we propose NeuralPLexer, a Neural framework for Protein-Ligand complex structure prediction. NeuralPLexer leverages diffusion-based generative modeling [200, 201] to sample 3D structures from a learned statistical distribution. We demonstrate that the multi-scale inductive bias in biomolecular complexes can be feasibly integrated with diffusion models by designing a finite-time stochastic differential equation (SDE) with structured drift terms. Owing to this formulation, NeuralPLexer can generalize to ligand-unbound or predicted protein structure inputs once trained solely on experimental protein-ligand complex structures that are not paired to alternative protein conformations. When applied to blind protein-ligand docking, NeuralPLexer improves both the geometrical accuracy and structure quality compared to baseline methods; when applied to ligand binding site design, an inpainting version of NeuralPLexer can accurately repack 44% of failed AlphaFold2 [5] binding sites with up to 60% success rate improvements compared to the method in Rosetta [202]. Furthermore, NeuralPLexer only requires molecular graphs as ligand inputs, therefore can enable end-to-end gradient-based design for functional small-molecules and ligand-binding proteins when coupled to recently-proposed differentiable protein sequence [203–205] and molecular graph generators [206, 207].

4.2 Method

We assume the model inputs are a receptor protein backbone template containing the amino acid sequence $\mathbf{s}$ and (N, Cα, C) atomic coordinates $\tilde{\mathbf{x}} \in \mathbb{R}^{n_{\text{res}}\times 3\times 3}$, and a set of ligand molecular graphs $\{\mathcal{G}_k\}_{k=1}^{K}$ containing atom/bond types and stereochemistry labels (e.g., tetrahedral or E/Z isomerism [208]). We aim to sample $(\mathbf{x}, \mathbf{y}) \sim q_\phi(\cdot|\mathbf{s}, \tilde{\mathbf{x}}, \{\mathcal{G}\})$ from a generative model q_ϕ with predicted 3D heavy-atom coordinates of the protein $\mathbf{x} \in \mathbb{R}^{n\times 3}$ and that of the ligands $\mathbf{y} \in \mathbb{R}^{m\times 3}$. It can be understood as a conditional generative modeling problem for partially-observed systems.

NeuralPLexer adopts a two-stage architecture for protein-ligand structure prediction (Figure4.1a). The input protein backbone template and molecule graphs are first encoded and passed into a *contact predictor* that iteratively samples binding interface spatial proximity distributions for each ligand in $\{\mathcal{G}\}$; the output contact map parameterizes the *geometry prior*, a finite-time marginal of a designed SDE that progressively injects structured noise into the data distribution. An *equivariant structure diffusion module* (ESDM) then jointly generates 3D protein and ligand structures by denoising the atomic coordinates sampled from the geometry prior through a learned reverse-time SDE (Figure4.1b).

Protein-ligand structure generation with biophysics-informed diffusion processes

Diffusion models [201] introduce a forward SDE that diffuses data into a noised distribution and a neural-network-parameterized reverse-time SDE that generate data by reverting the noising process. To motivate the design principles for our

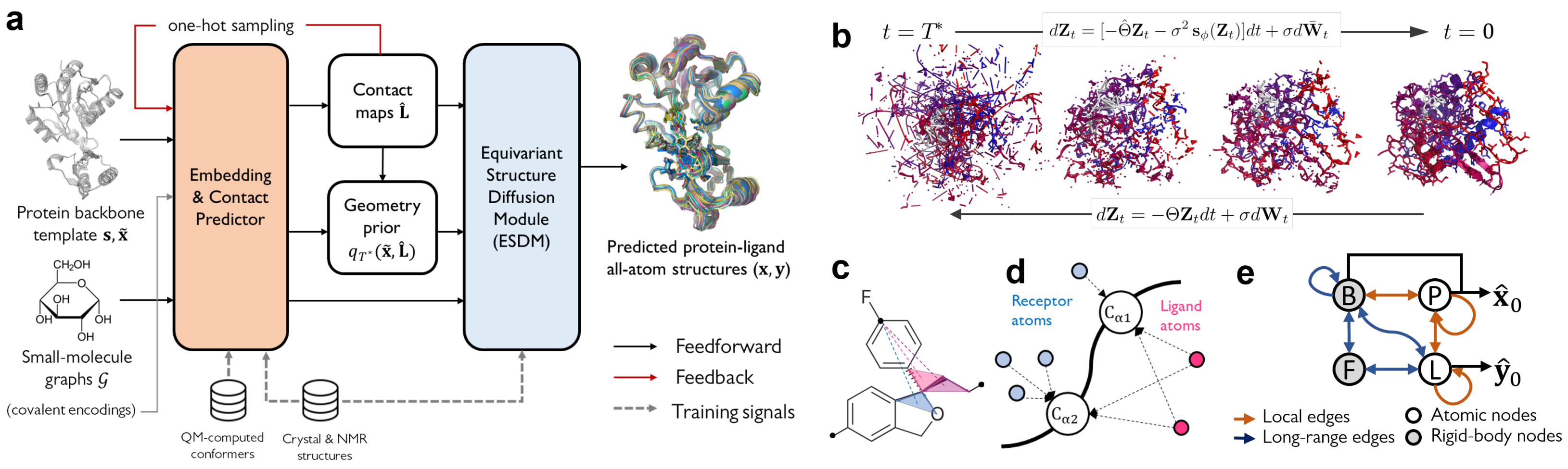
a
one-hot sampling
Protein backbone template $\mathbf{s}, \tilde{\mathbf{x}}$
Small-molecule graphs $\mathcal{G}$
(covalent encodings)
Embedding & Contact Predictor
Contact maps $\hat{\mathbf{L}}$
Geometry prior $q_{T^*}(\tilde{\mathbf{x}}, \hat{\mathbf{L}})$
Equivariant Structure Diffusion Module (ESDM)
Predicted protein-ligand all-atom structures $(\mathbf{x}, \mathbf{y})$
QM-computed conformers
Crystal & NMR structures
Feedforward
Feedback
Training signals
b
$t = T^*$
$d\mathbf{Z}_t = [-\hat{\Theta}\mathbf{Z}_t - \sigma^2 \mathbf{s}_\phi(\mathbf{Z}_t)]dt + \sigma d\bar{\mathbf{W}}_t$
$t = 0$
$d\mathbf{Z}_t = -\Theta\mathbf{Z}_t dt + \sigma d\mathbf{W}_t$
c
F
d
Receptor atoms
Ligand atoms
$C_{\alpha 1}$
$C_{\alpha 2}$
e
B
P
F
L
$\hat{\mathbf{x}}_0$
$\hat{\mathbf{y}}_0$
Local edges
Long-range edges
Atomic nodes
Rigid-body nodes

Figure 4.1: NeuralPLexer enables protein-ligand complex structure prediction with full receptor flexibility. (a) Method overview. (b) Sampling from NeuralPLexer. The protein (colored as red-blue from N- to C-terminus) and ligand (colored as grey) 3D structures are jointly generated from a learned SDE, with a partially-diffused initial state q_{T^*} approximated by the protein backbone template and predicted interface contact maps. (c-e) Key elements of the NeuralPLexer technical design. (c) Ligand molecules and monomeric entities are encoded as the collection of atoms, local coordinate frames (depicted as semi-transparent triangles), and stereospecific pairwise embeddings (depicted as dashed lines) representing their interactions. (d) The forward-time SDE introduces relative drift terms among protein Cα atoms, non-Cα atoms and ligand atoms, such that the SDE erases local-scale details at $t = T^*$ to enable resampling from a noise distribution. (e) Information flow in the equivariant structure diffusion module (ESDM). ESDM operates on a heterogeneous graph formed by protein atoms (P), ligand atoms (L), protein backbone frames (B) and ligand local frames (F) to predict clean atomic coordinates $\hat{\mathbf{x}}_0, \hat{\mathbf{y}}_0$ using the coordinates at a finite diffusion time $t > 0$.

biomolecular structure generator, we first consider a general class of linear SDEs known as the multivariate Ornstein–Uhlenbeck (OU) process [209] for point cloud $\mathbf{Z} \in \mathbb{R}^{N\times 3}$:

$$d\mathbf{Z}_t = -\Theta\mathbf{Z}_t dt + \sigma d\mathbf{W}_t \tag{4.1}$$

where $\Theta \in \mathbb{R}^{N\times N}$ is an invertible matrix of affine drift coefficients and $\mathbf{W}_t$ is a standard $3N$-dimensional Wiener process. The forward noising SDEs used in standard diffusion models [210, 211] can be recovered by setting $\Theta = \theta\mathbf{I}$, converging to an isotropic Gaussian prior distribution at the $t \longrightarrow \infty$ (often expressed as $t \longrightarrow 1$ with reparameterized t [212]) limit. In contrast, we design a multivariate SDE with data-dependent drift matrix $\Theta(\mathbf{Z}_0)$ and truncate the SDE at $t = T^* < \infty$ such that the final state of forward noising process is a partially-diffused, structured distribution q_{T*} that can be well approximated by a coarse-scale model. We propose a set of SDEs depicted by Figure4.1d and detailed in Table 4.1, with separated lengthscale parameters σ_1, σ_2 such that the forward diffusion process erases residue-scale local details but retains global information about protein domain packing and ligand binding interfaces, yielding the following time-dependent transition kernels:

$$q_t\big(\mathbf{x}_{\mathrm{C}\alpha}(t)|\mathbf{x}(0), \mathbf{y}(0)\big) = \mathcal{N}\big(\mathbf{x}_{\mathrm{C}\alpha}(0); \sigma_1^2\tilde{\tau}\mathbf{I}\big) \tag{4.2}$$

$$q_t\big(\mathbf{x}_{\mathrm{nonC}\alpha}(t) - \mathbf{x}_{\mathrm{C}\alpha}(t)|\mathbf{x}(0), \mathbf{y}(0)\big) = \mathcal{N}\big(e^{-\tilde{\tau}}\big(\mathbf{x}_{\mathrm{nonC}\alpha}(0) - \mathbf{x}_{\mathrm{C}\alpha}(0)\big); 2\sigma_1^2(1 - e^{-2\tilde{\tau}})\mathbf{I}\big) \tag{4.3}$$

$$q_t\big(\mathbf{y}(t) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(t)|\mathbf{x}(0), \mathbf{y}(0)\big) = \mathcal{N}\big(e^{-\tilde{\tau}}\big(\mathbf{y}(0) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(0)\big); \sigma_1^2(1 - e^{-2\tilde{\tau}})(\mathbf{I} + \mathbf{c}^{\mathrm{T}}\mathbf{c})\big) \tag{4.4}$$

where we use an exponential schedule $\tilde{\tau} = (\sigma_{\mathrm{min}}^2/\sigma_1^2)e^t$ with truncation $T^* = 2\log(\sigma_2/\sigma_{\mathrm{min}})$. $\mathbf{c}$ is a softmax-transformed *contact map* as detailed in Sec. 4.2, which attracts the diffused ligand coordinates $\mathbf{y}(t)$ towards binding interface Cα atoms while preserving SE(3)-equivariance. We choose $\sigma_1 = 2.0$ Å to match the average radius of standard amino acids with task-specific $\sigma_2 > \sigma_1$ such that at $t = T^*$: (a) the terms involving $\mathbf{x}_{\mathrm{nonC}\alpha}(0)$ and $\mathbf{y}(0)$ approximately vanishes thus are set to zeros to initialize the reverse-time SDE, and (b) the Cα-atom coordinate marginal $q_{T^*}\big(\mathbf{x}_{\mathrm{C}\alpha}(t)|\mathbf{x}(0)\big)$ is sufficiently close to which approximated by the backbone template $q_{T^*}\big(\mathbf{x}_{\mathrm{C}\alpha}(t)|\tilde{\mathbf{x}}\big)$, guided by the theoretical result proposed in [213]. Proofs regarding SE(3)-equivariance are stated in the Appendix 4.5.

Contact map prediction and sampling from the truncated reverse-time SDE

Given protein-ligand coordinates $(\mathbf{x}, \mathbf{y})$, we define the contact map $\mathbf{L} \in \mathbb{R}^{n_{\text{res}} \times m}$ with matrix elements $L_{Ai} = \log(\frac{\sum_{j\in\{A\}} e^{-2\alpha\|\mathbf{x}_j - \mathbf{y}_i\|^2}}{\sum_{j\in\{A\}} e^{-\alpha\|\mathbf{x}_j - \mathbf{y}_i\|^2}})$ where j runs over all protein atoms in amino acid residue A and $\alpha = 0.2\,\text{Å}^{-1}$. The term $\mathbf{c}$ in (4.4) is then defined as $c_{Ai}(\mathbf{L}) = \frac{\exp(L_{Ai})}{\sum_A \exp(L_{Ai})}$. To sample from the reverse-time SDE, we use the contact predictor to generate inferred contact maps $\hat{\mathbf{L}}$ and parameterize the geometry prior $q_{T*}(\cdot|\tilde{\mathbf{x}}, \hat{\mathbf{L}})$ — the initial condition of reverse-time SDE — by replacing $\mathbf{x}(0)$ in q_{T^*} with the backbone template $\tilde{\mathbf{x}}$ and the ligand-Cα relative drift coefficient $\mathbf{c}$ with the predicted $\mathbf{c}(\hat{\mathbf{L}})$. Note that in the general multivariate OU formulation, this corresponds to replacing the clean-data-dependent drift coefficients $\Theta(\mathbf{Z}_0)$ by a model estimation $\hat{\Theta}$. To account for the multimodal nature of protein-ligand contact distributions, the contact predictor models $\mathbf{L}$ as the logits of a categorical posterior distribution over a sequence of one-hot observations $\{\mathbf{l}\}_{k=1}^{K}$ sampled for individual molecules in $\{\mathcal{G}\}$. The forward pass of contact predictor ψ takes an iterative form:

$$\hat{\mathbf{L}}_k = \psi(\sum_{r=1}^{k} \mathbf{l}_r; \mathbf{s}, \tilde{\mathbf{x}}, \{\mathcal{G}\});\ \mathbf{l}_k = \text{OneHot}(A_k, i_k);\ (A_k, i_k) \sim \text{Categorical}_{n_{\text{res}} \times m}(\hat{\mathbf{L}}_{k-1}),\ i_k \in \mathcal{G}_k \tag{4.5}$$

where $k \in \{1, \cdots, K\}$ and we set $\hat{\mathbf{L}} := \hat{\mathbf{L}}_K$. All results reported in this study are obtained with $K = 1$ due to the curation scheme of standard annotated protein-ligand datasets, but we note that the model can be readily trained on more diverse structural databases with multi-ligand samples.

Architecture overview

Here we outline the key neural network design ideas and defer the featurization, architecture, and training details to the Appendix. To enable stereospecific molecular geometry generation and explicit reasoning about long-range geometrical correlations, NeuralPLexer hybridizes two types of elementary molecular representations (Figure4.1c): (a) atomic nodes and (b) rigid-body nodes representing coordinate frames formed by two adjacent chemical bonds. For small-molecule ligand encoding, we introduce a graph transformer with learnable chirality-aware pairwise embeddings that are constructed through graph-diffusion-kernel-like transformations [214]; such pairwise embeddings are pretrained to align with the intra-molecular 3D coordinate distributions from experimental and computed molecular conformers. The protein backbone template encoding module and the contact predictor are built upon a sparsified version of invariant point attention (IPA) adapted from AlphaFold2 [5] and are combined with standard graph attention layers [187, 215] and edge update blocks.

The architecture of ESDM (Figure4.1e) is inspired by prior works on 3D graph and attentional neural networks for point clouds [216, 217], rigid-body simulations [218] and biopolymer representation learning [5, 219–221]. In ESDM, each node is associated with a stack of standard scalar features $\mathbf{f}_\mathrm{s} \in \mathbb{R}^c$ and cartesian vector features $\mathbf{f}_\mathrm{v} \in \mathbb{R}^{3\times c}$ representing the displacements of a virtual point set relative to the node's Euclidean coordinate $\mathbf{t} \in \mathbb{R}^3$. A rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ is additionally attached to each rigid-body node. Geometry-aware messages are synchronously propagated among all nodes by encoding the pairwise distances among virtual point sets into graph transformer blocks. Explicit non-linear transformation on vector features $\mathbf{f}_\mathrm{v}$ is solely performed on rigid-body nodes through a coordinate-frame-inversion mechanism, such that the node update blocks are sufficiently expressive without sacrificing equivariance or computational efficiency. On the contrary, 3D coordinates are solely updated for atomic nodes while the rigid-body frames $(\mathbf{t}, \mathbf{R})$ are passively reconstructed according to the updated atomic coordinates, circumventing numerical issues regarding fitting quaterion or axis-angle variables when manipulating rigid-body objects. The nontrivial actions of a parity inversion operation on rigid-body nodes ensure that ESDM can capture the correct chiral-symmetry-breaking behavior that adheres to the molecular stereochemistry constraints.

4.3 Results

Fixed-backbone protein-ligand docking. In this setting the ground-truth receptor protein backbone is given as input $\tilde{\mathbf{x}}$, and both ligand coordinates and protein sidechain coordinates are predicted using the input protein backbone and ligand graphs. Results are compared to a recent learning-based method EquiBind [222]; for reference, we also include results from a physics-based blind docking method CB-Dock [223] obtained with ground-truth all-atom receptor inputs and using a computing budget similar to learning-based methods. Models are trained and tested on the PDBBind-2020 [224] dataset split used in [222], with additional test dataset processing to ensure a reasonable comparison to docking-based methods (see Appendix 4.5). As shown in Figure 4.2a-c, NeuralPLexer achieves both improved geometrical accuracy (reported as the ligand heavy atom root-mean-squre-deviation (RMSD)) and lower steric clash rate (the fraction of ligand heavy atoms with a Lennard-Jones energy > 100 kcal/mol, using UFF [225] parameters). We found that good ligand structure quality and geometrical accuracy can be achieved using as few as 10 integrator steps

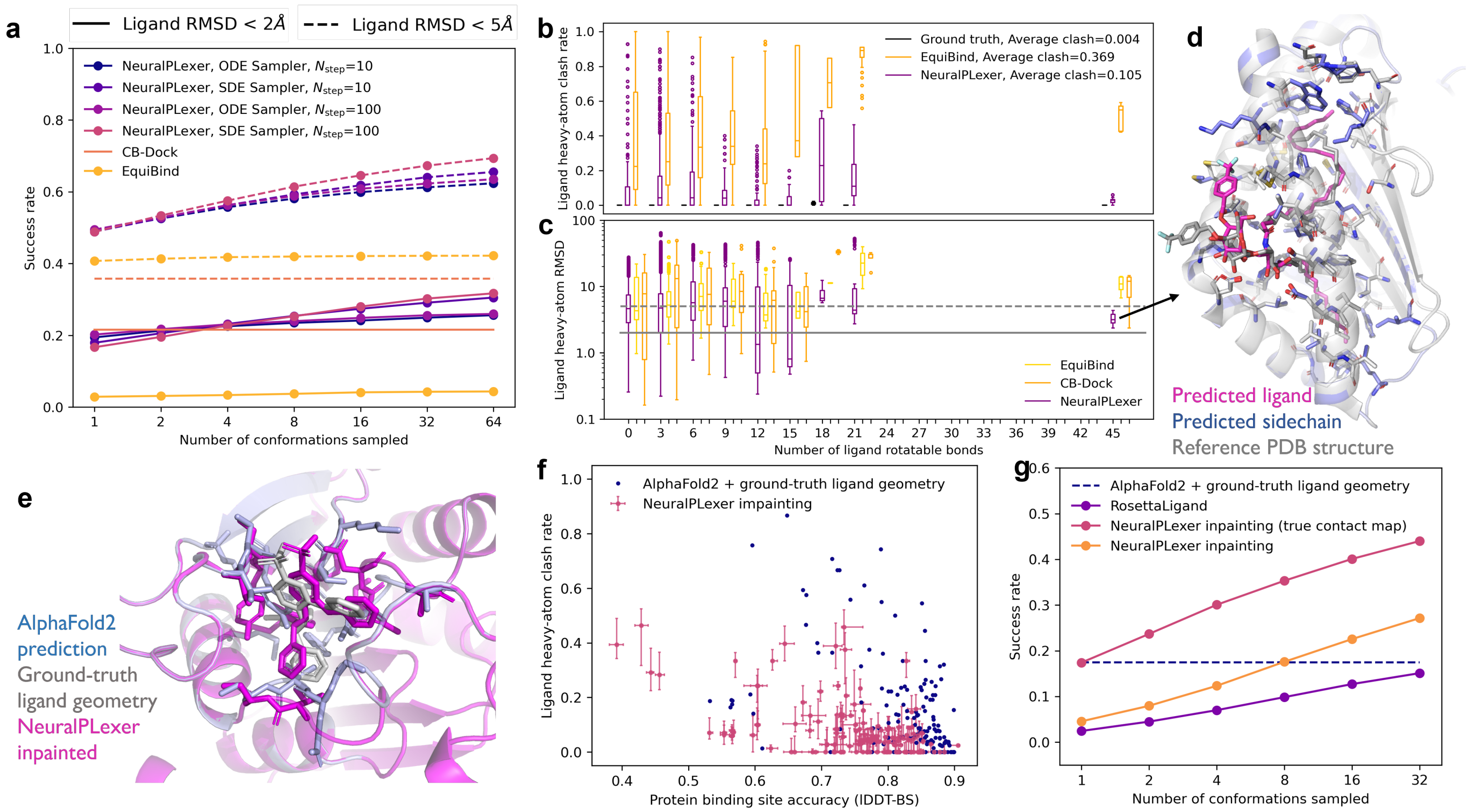

Figure 4.2: Model performance on benchmarking problems. (a-d) Fixed-backbone blind protein-ligand docking. (a) Success rates over the test dataset are plotted against the number of conformations sampled per protein-ligand pair; a success is defined as the ligand RMSD being lower than given threshold for at least one of the sampled conformations. Distributions of (b) the physical plausibility of sampled conformations as measured by the ligand heavy-atom steric clash rate with receptor atoms and (c) the geometrical accuracy as measured by the ligand RMSD are plotted against the number of ligand rotatable bonds, an indicatior of molecular flexibility. (d) Overlay of NeuralPLexer-predicted ligand and side-chain structures on the ground-truth for a challenging example (PDB: 6MJQ). (e-g) Ligand-coupled binding site repacking via diffusion-based inpainting. (e) A selected example (PDB:6TEL) where NeuralPLexer accurately inpaints the binding site protein-ligand structure, while directly aligning AlphaFold2 prediction to the ground-truth complex resulted in steric clashes between the ligand and binding site residues. (f) Summary of binding site accuracy (measured by the all-atom lDDT-BS score) and ligand clash rate over the test dataset. 32 conformations are sampled for each protein-ligand pair; dots indicates the median value and errorbars indicates 25% and 75% percentiles. (g) Success rates compared to baseline methods. A success is defined as: lDDT-BS > 0.7, ligand RMSD < 2.0 Å, and clash rate = 0.0. The pink "true contact map" curves are obtained by initializing the geometry prior q_{T^*} using the true protein-ligand contact map, while the gold curves are obtained by generating both protein and ligand conformations end-to-end.

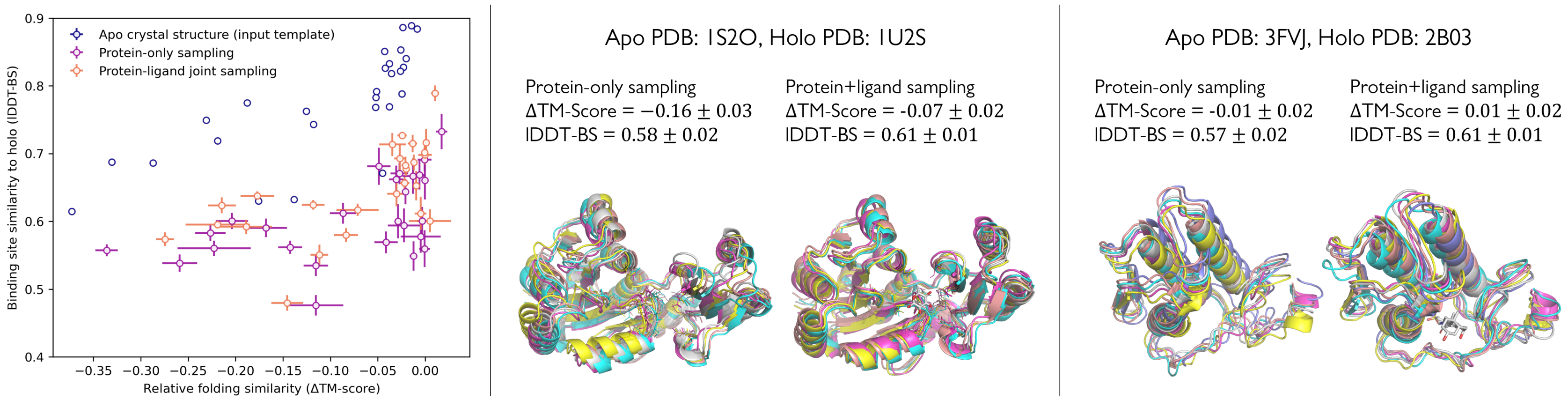

Figure 4.3: Assessments on systems with large binding-induced protein conformational transitions. Apo protein structures are used as the input backbone template. (a) Summary statistics of the relative protein folding similarity with respect to apo and holo PDB (measure by ΔTM-Score, the difference between TM-Scores computed against holo and apo structures) and binding site similarity with respect to holo (measured by lDDT-BS) for sampled structures. Purple dots are obtained with protein-only inputs and gold dots are obtained using protein+ligand inputs. Ligand-conditioning increases average ΔTM-Score from -9.0% to -7.7% (p=0.03), and average lDDT-BS from 0.59 to 0.63 (p<0.001). (b-c) Two examples for which neither their holo nor apo reference structures were observed during training. A marginal improvement in ΔTM-Score or lDDT-BS may indicate substantial protein conformational differences, while NeuralPLexer can qualitatively capture the correct protein state transitions.

(0.2 second per conformation on a single V100 GPU).

Ligand-coupled binding site repacking. Here we apply a diffusion-based inpainting strategy to jointly sample ligand and protein structure for a cropped region within 6.0 Å of the ligand conditioning on the uncropped parts of the protein. Protein binding site accuracy is measured by the lDDT-BS metric [226] with cutoff parameters consistent with CAMEO [227]. Input backbones are obtained using template-free AlphaFold2 (AF2) predictions of 154 selected chains whose TM-score [228]>0.8 and lDDT-BS<0.9 out of the abovementioned PDBBind test set, a subset representing cases where AF2 correctly predicts the global protein folding but unable to reproduce the exact bound-state binding site structure. We found 82% of structures contain steric clash with the ligand when directly aligned to reference complex structure in PDB, while NeuralPLexer is able to rescue 44% of these AF2 binding sites with joint protein-ligand inpainting (Figure 4.2e-g). Comparing to an energy-based flexible ligand-receptor modeling method RosettaLigand [202], NeuralPLexer increases success rate by up to 60% on the combined metric for ligand accuracy, binding site accuracy and physical plausibility.

Cryptic pockets and binding-induced protein conformation transitions. Lastly, we assessed NeuralPLexer-sampled structures for 31 systems from the PocketMiner dataset [229] which represents proteins with substantial ligand-binding-induced conformation changes. As a preliminary examination, we use the ligand-unbound (apo) crystal structure from PDB as the input backbone template and fix the ligand conformation to ground-truth coordinates along sampling. We found NeuralPLexer shifts the sampled ensemble toward bound-state (holo) structures when performing joint protein-ligand generation, compared to unconditioned protein-only sampling results (Figure 3a). Human evaluations reveal that NeuralPLexer correctly predicts biologically-relevant motions as illustrated by examples in Figure 3b-c, but a more systematic examination is currently hampered by the sensitivity of TM-Score and lDDT-BS to binding-irrelevant fluctuations. We note that native contact analysis algorithms [230] may provide improved metrics for interpreting protein generative models and consider that a future direction.

4.4 Discussion and outlooks

We have presented a learning-based method for dynamic-backbone protein-ligand structure prediction, establishing an accuracy and sampling efficiency advantage

Table 4.1: Summary of the forward-time SDEs with a constant effective diffusion coefficient ($\sigma(\tau) = \sigma$).

Atom type	SDE Expression	Approximate marginal at $t = T^*$
Receptor Cα	$d\mathbf{x}_{\text{C}\alpha} = \sigma d\mathbf{w}_1$	$q_{T^*}\big(\mathbf{x}_{\text{C}\alpha} \vert \mathbf{x}(0), \mathbf{y}(0)\big) = \mathcal{N}(\mathbf{x}_{\text{C}\alpha}(0); \sigma_2^2\mathbf{I})$
Receptor non-Cα	$d\mathbf{x}_{\text{nonC}\alpha} = \theta(\mathbf{x}_{\text{C}\alpha} - \mathbf{x}_{\text{nonC}\alpha})d\tau + \sigma d\mathbf{w}_2$	$q_{T^*}\big(\mathbf{x}_{\text{nonC}\alpha} - \mathbf{x}_{\text{C}\alpha} \vert \mathbf{x}(0), \mathbf{y}(0)\big) = \mathcal{N}(\mathbf{0}; 2\sigma_1^2\mathbf{I})$
Ligand atoms	$d\mathbf{y} = \theta(\mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha} - \mathbf{y})d\tau + \sigma d\mathbf{w}_3$	$q_{T^*}\big(\mathbf{y} - \mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha} \vert \mathbf{x}(0), \mathbf{y}(0)\big) = \mathcal{N}(\mathbf{0}; \sigma_1^2(\mathbf{I} + \mathbf{c}^{\text{T}}\mathbf{c}))$

relative to baseline approaches. We anticipate the incorporation of state-of-the-art protein representation learning techniques such as the use of sequence evolutionary signals, pretrained language models, or higher-level attention mechanisms [5, 203, 204] and training on large-scale structure datasets to further improve the methodology and facilitate applications in various downstream molecular design problems.

4.5 Appendix

The forward-time and reverse-time SDEs

The forward-time SDEs in NeuralPLexer are summarized in Table 4.1. For generality, we introduce an effective time stamp τ such that the drift and diffusion coefficients are constant $\theta(t) = \theta, \sigma(\tau) = \sigma$. The symbolic conventions are as following:

- $\mathbf{x}_{\text{C}\alpha} \in \mathbb{R}^{n_{\text{res}}\times 3}$ denotes the collection of alpha-carbon coordinates in the protein, following the standard nomenclature for amino acid atom types:
- $\mathbf{x}_{\text{nonC}\alpha} \in \mathbb{R}^{(n-n_{\text{res}})\times 3}$ denotes the set of coordinates for all non-alpha-carbon protein atoms (backbone N, C, O, and all side-chain heavy atoms);
- $\mathbf{y} \in \mathbb{R}^{m\times 3}$ denotes all ligand heavy atom coordinates. Note that $m := \sum_{k=1}^{K} m_k$ with m_k being the number of heavy atoms in each ligand molecule $\mathcal{G}_k$.

Transition kernel densities and sampling

Following the general result for Ornstein–Uhlenbeck processes [231]

$$q_{0:t}(\mathbf{x}_t) = \mathcal{N}(\exp(-\Theta t)\mathbf{x}_0; \int_0^t e^{\Theta(s-t)}\boldsymbol{\sigma}\boldsymbol{\sigma}^{\text{T}}e^{\Theta^{\text{T}}(s-t)}ds) \tag{4.6}$$

given the effective time-homogeneous diffusion process described in Table 4.1, for internal coordinates $\mathbf{x}_{\text{nonC}\alpha} - \mathbf{x}_{\text{C}\alpha}$:

$$d(\mathbf{x}_{\text{nonC}\alpha} - \mathbf{x}_{\text{C}\alpha}) = -\theta(\mathbf{x}_{\text{nonC}\alpha} - \mathbf{x}_{\text{C}\alpha})d\tau + \sigma d\mathbf{w}_2 - \sigma d\mathbf{w}_1 \tag{4.7}$$

since the Brownian motions $\mathbf{w}_1, \mathbf{w}_2$ are independent, we obtain the transition kernel for the finite time interval s:

$$\begin{aligned} & q(\mathbf{x}_{\text{nonC}\alpha}(\tau + s) - \mathbf{x}_{\text{C}\alpha}(\tau + s) | \mathbf{x}_{\text{nonC}\alpha}(t) - \mathbf{x}_{\text{C}\alpha}(\tau)) \\ & = \mathcal{N}\big(e^{-\theta s}(\mathbf{x}_{\text{nonC}\alpha}(\tau) - \mathbf{x}_{\text{C}\alpha}(\tau)); (1 - e^{-2\theta s})\frac{\sigma^2}{\theta^2}\mathbf{I}\big) \end{aligned} \tag{4.8}$$

Similarly, for the ligand degrees of freedom

$$d(\mathbf{y} - \mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha}) = -\theta(\mathbf{y} - \mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha})dt + \sigma d\mathbf{w}_3 - \sigma\mathbf{c}^{\text{T}} d\mathbf{w}_1 \tag{4.9}$$

the transition kernel is

$$\begin{aligned} & q(\mathbf{y}(\tau + s) - \mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha}(\tau + s) | \mathbf{y}(\tau) - \mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha}(\tau)) \\ & = \mathcal{N}\big(e^{-\theta s}(\mathbf{y}(\tau) - \mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha}(\tau)); (1 - e^{-2\theta s})\frac{\sigma^2}{2\theta^2}(\mathbf{I} + \mathbf{c}^{\text{T}}\mathbf{c})\big) \end{aligned} \tag{4.10}$$

The transition kernel for alpha-carbon atoms is a standard Gaussian

$$q(\mathbf{x}_{\text{C}\alpha}(\tau + s) | \mathbf{x}_{\text{C}\alpha}(\tau)) = \mathcal{N}\big(\mathbf{x}_{\text{C}\alpha}(\tau); \sigma^2 s\mathbf{I}\big). \tag{4.11}$$

Defining $\sigma_1^2 = \frac{\sigma^2}{2\theta}$, $\sigma_2^2 = \sigma^2 \cdot \tau(T^*)$, and $\tilde{\tau} = 2\theta\tau$, we recover (2-4). For model training in practice, we use an exponential noise schedule defined by $\tau = \tau_0 e^t$ and $\tau_0 = \frac{\sigma_{\text{min}}^2}{\sigma^2}$ with σ_{min} being a minimum perturbation scale as commonly adopted in variance-exploding (VE) [201] SDEs. For completeness, the SDEs defined in the transformed time horizon $t \in [0, T^*]$ is given by replacing the drift coefficient θ and the diffusion coefficient σ with the following time-dependent counterparts:

$$\theta(t) = \theta \cdot \frac{d\tau}{dt} = \frac{\sigma_{\text{min}}^2}{2\sigma_1^2} e^t \tag{4.12}$$

and

$$\sigma(t) = \sqrt{\sigma^2 \cdot \frac{d\tau}{dt}} = \sigma_{\text{min}} e^{\frac{1}{2}t}. \tag{4.13}$$

To sample from the marginal distribution $q_t := p_{\text{data}} * q_{0:t}$ derived from the forward SDEs:

$$\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \sim \mathcal{N}(0; \mathbf{I}) \tag{4.14a}$$

$$(\mathbf{x}, \mathbf{y}) \sim p_{\text{data}} \tag{4.14b}$$

$$\mathbf{x}_{\text{C}\alpha}(t) = \mathbf{x}_{\text{C}\alpha} + \sigma\sqrt{\tau(t)}\mathbf{z}_1 \tag{4.14c}$$

$$\mathbf{x}_{\text{nonC}\alpha}(t) = \mathbf{x}_{\text{C}\alpha}(t) + \sqrt{\alpha(t)}(\mathbf{x}_{\text{nonC}\alpha} - \mathbf{x}_{\text{C}\alpha}) + \sqrt{1 - \alpha(t)}\sigma_1(\mathbf{z}_2 - \mathbf{z}_1) \tag{4.14d}$$

$$\mathbf{y}(t) = \mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha}(t) + \sqrt{\alpha(t)}(\mathbf{y} - \mathbf{c}^{\text{T}}\mathbf{x}_{\text{C}\alpha}) + \sqrt{1 - \alpha(t)}\sigma_1(\mathbf{z}_3 - \mathbf{c}^{\text{T}}\mathbf{z}_1) \tag{4.14e}$$

where $\alpha(t) = e^{-2\theta\tau(t)}$.

For the reverse-time SDE

$$d\mathbf{Z}_t = [-\Theta(t)\mathbf{Z}_t - \sigma^2(t)\nabla_{\mathbf{Z}_t} \log q_t(\mathbf{Z}_t)]dt + \sigma(t)d\mathbf{W}_t \tag{4.15}$$

the ESDM ϕ predicts the denoised observations $\hat{\mathbf{x}}(0), \hat{\mathbf{y}}(0)$ using $\hat{\mathbf{x}}(t), \hat{\mathbf{y}}(t)$ which is formally equivalent to estimating the score function $\nabla_{\mathbf{Z}} \log q_t(\mathbf{Z})$ [232]. Given a time discretization schedule with interval s, we obtain the expression for the predicted observation mean $\bar{\mathbf{Z}}(\phi, t-s)$ in one denoising step $\mathbf{Z}(t) \mapsto \mathbf{Z}(t-s)$:

$$\bar{\mathbf{x}}_{\mathrm{C}\alpha}(\phi, t-s) = -(\mathbf{x}_{\mathrm{C}\alpha}(t) - \hat{\mathbf{x}}_{\mathrm{C}\alpha}(0))\frac{\sigma(t-s)}{\sigma(t)} + \mathbf{x}_{\mathrm{C}\alpha}(t) \tag{4.16a}$$

$$\begin{aligned}\bar{\mathbf{x}}_{\mathrm{nonC}\alpha}(\phi, t-s) = &-\frac{(\mathbf{x}_{\mathrm{nonC}\alpha}(t) - \mathbf{x}_{\mathrm{C}\alpha}(t)) \cdot \sqrt{\alpha(t)} - (\hat{\mathbf{x}}_{\mathrm{nonC}\alpha}(0) - \hat{\mathbf{x}}_{\mathrm{C}\alpha}(0))}{\sqrt{1-\alpha(t)}}\sqrt{1-\alpha(t-s)} \\ &+ \bar{\mathbf{x}}_{\mathrm{C}\alpha}(t-s) + \sqrt{\alpha(t-s)}(\hat{\mathbf{x}}_{\mathrm{nonC}\alpha}(0) - \hat{\mathbf{x}}_{\mathrm{C}\alpha}(0))\end{aligned} \tag{4.16b}$$

$$\begin{aligned}\bar{\mathbf{y}}(\phi, t-s) = &-\frac{(\mathbf{y}(t) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(t)) \cdot \sqrt{\alpha(t)} - (\hat{\mathbf{y}}(0) - \mathbf{c}^{\mathrm{T}}\hat{\mathbf{x}}_{\mathrm{C}\alpha}(0))}{\sqrt{1-\alpha(t)}}\sqrt{1-\alpha(t-s)} \\ &+ \mathbf{c}^{\mathrm{T}}\bar{\mathbf{x}}_{\mathrm{C}\alpha}(t-s) + \sqrt{\alpha(t-s)}(\hat{\mathbf{y}}(0) - \mathbf{c}^{\mathrm{T}}\hat{\mathbf{x}}_{\mathrm{C}\alpha}(0))\end{aligned} \tag{4.16c}$$

standard ODE-based or SDE-based integrators can then be adapted to sample from (4.15).

Euclidean equivariance

Given group G, a function $f : X \to Y$ is said to be equivariant if for all $g \in G$ and $x \in X$, $f(\varphi_X(g) \cdot x) = \varphi_Y(g) \cdot f(x)$. Specifically f is said to be invariant if $f(\varphi_X(g) \cdot x) = f(x)$. We are interested in the special Euclidean group $G = \mathrm{SE}(3)$ consists of all global rigid translation and rotation operations $g \cdot \mathbf{Z} := \mathbf{t} + \mathbf{Z} \cdot \mathbf{R}$ where $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{R} \in \mathrm{SO}(3)$. To adhere to the physical constraint that p_{data} is always SE(3)-invariant, the transition kernels of forward-time SDE should satisfy SE(3)-equivariance $q(\mathbf{Z}_{t+s}|\mathbf{Z}_t) = q(g \cdot \mathbf{Z}_{t+s}|g \cdot \mathbf{Z}_t)$ such that the marginals are invariant $q_t(\mathbf{Z}_t) = q_t(g \cdot \mathbf{Z}_t)$ for any time t. The proofs are straightforward:

For receptor Cα degrees of freedom

$$
\begin{aligned}
&q(\mathbf{t} + \mathbf{x}_{\mathrm{C}\alpha}(\tau + s) \cdot \mathbf{R} | \mathbf{t} + \mathbf{x}_{\mathrm{C}\alpha}(\tau) \cdot \mathbf{R}) \\
&= \mathcal{N}\big(\mathbf{t} + \mathbf{x}_{\mathrm{C}\alpha}(\tau + s) \cdot \mathbf{R}; \mathbf{t} + \mathbf{x}_{\mathrm{C}\alpha}(\tau) \cdot \mathbf{R}, \sigma^2 s \mathbf{I}\big) \\
&= \mathcal{N}\big((\mathbf{x}_{\mathrm{C}\alpha}(\tau + s) - \mathbf{x}_{\mathrm{C}\alpha}(\tau)) \cdot \mathbf{R}\mathbf{R}^{\mathrm{T}}; 0, \sigma^2 s \mathbf{R} \cdot \mathbf{I} \cdot \mathbf{R}^{\mathrm{T}}\big) \\
&= \mathcal{N}\big((\mathbf{x}_{\mathrm{C}\alpha}(\tau + s) - \mathbf{x}_{\mathrm{C}\alpha}(\tau)); 0, \sigma^2 s \mathbf{I}\big) \\
&= q(\mathbf{x}_{\mathrm{C}\alpha}(\tau + s) | \mathbf{x}_{\mathrm{C}\alpha}(\tau)).
\end{aligned}
$$

For receptor non-Cα degrees of freedom

$$
\begin{aligned}
&q((\mathbf{t} + \mathbf{x}_{\mathrm{nonC}\alpha}(\tau + s) \cdot \mathbf{R} - \mathbf{t} - \mathbf{x}_{\mathrm{C}\alpha}(\tau + s) \cdot \mathbf{R}) | (\mathbf{t} + \mathbf{x}_{\mathrm{nonC}\alpha}(\tau) \cdot \mathbf{R} - \mathbf{t} - \mathbf{x}_{\mathrm{C}\alpha}(\tau) \cdot \mathbf{R})) \\
&= \mathcal{N}\big((\mathbf{x}_{\mathrm{nonC}\alpha}(\tau + s) \cdot \mathbf{R} - \mathbf{x}_{\mathrm{C}\alpha}(\tau + s) \cdot \mathbf{R}); e^{-\theta s}(\mathbf{x}_{\mathrm{nonC}\alpha}(\tau) \cdot \mathbf{R} - \mathbf{x}_{\mathrm{C}\alpha}(\tau) \cdot \mathbf{R}), (1 - e^{-2\theta s})\frac{\sigma^2}{\theta^2}\mathbf{I}\big) \\
&= \mathcal{N}\big((\mathbf{x}_{\mathrm{nonC}\alpha}(\tau + s) - \mathbf{x}_{\mathrm{C}\alpha}(\tau + s)); e^{-\theta s}(\mathbf{x}_{\mathrm{nonC}\alpha}(\tau) - \mathbf{x}_{\mathrm{C}\alpha}(\tau)), (1 - e^{-2\theta s})\frac{\sigma^2}{\theta^2}\mathbf{R} \cdot \mathbf{I} \cdot \mathbf{R}^{\mathrm{T}}\big) \\
&= q((\mathbf{x}_{\mathrm{nonC}\alpha}(\tau + s) - \mathbf{x}_{\mathrm{C}\alpha}(\tau + s) | (\mathbf{x}_{\mathrm{nonC}\alpha}(\tau) - \mathbf{x}_{\mathrm{C}\alpha}(\tau))).
\end{aligned}
$$

For ligand degrees of freedom

$$
\begin{aligned}
&q(\mathbf{t} + \mathbf{y}(\tau + s) \cdot \mathbf{R} - \mathbf{c}^{\mathrm{T}}(\mathbf{t} + \mathbf{x}_{\mathrm{C}\alpha}(\tau + s) \cdot \mathbf{R}) | \mathbf{t} + \mathbf{y}(\tau) \cdot \mathbf{R} - \mathbf{c}^{\mathrm{T}}(\mathbf{t} + \mathbf{x}_{\mathrm{C}\alpha}(\tau) \cdot \mathbf{R})) \\
&= q(\mathbf{t} + \mathbf{y}(\tau + s) \cdot \mathbf{R} - \mathbf{c}^{\mathrm{T}}\mathbf{t} - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau + s) \cdot \mathbf{R} | \mathbf{t} + \mathbf{y}(\tau) \cdot \mathbf{R} - \mathbf{c}^{\mathrm{T}}\mathbf{t} - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau) \cdot \mathbf{R}) \\
&= q(\mathbf{y}(\tau + s) \cdot \mathbf{R} - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau + s) \cdot \mathbf{R} | \mathbf{y}(\tau) \cdot \mathbf{R} - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau) \cdot \mathbf{R}) \\
&= \mathcal{N}\big(e^{-\theta s}(\mathbf{y}(\tau) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau)); (1 - e^{-2\theta s})\frac{\sigma^2}{2\theta^2}\mathbf{R} \cdot (\mathbf{I} + \mathbf{c}^{\mathrm{T}}\mathbf{c}) \cdot \mathbf{R}^{\mathrm{T}}\big) \\
&= q(\mathbf{y}(\tau + s) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau + s) | \mathbf{y}(\tau) - \mathbf{c}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}\alpha}(\tau))
\end{aligned}
$$

where we have used $\mathbf{c}^{\mathrm{T}}\mathbf{t} = \mathbf{t}$ up to a column-wise broadcasting operation based on the row-wise normalization property of the softmax-transformed contact map $\mathbf{c}$. Since all transition kernels are SE(3)-equivariant, it then follows that the score $\nabla_{\mathbf{Z}} \log q_t(\mathbf{Z})$ is also SE(3)-equivariant: $\nabla_{\mathbf{Z}'} \log q_t(\mathbf{Z}') = \nabla_{\mathbf{Z}} \log q_t(\mathbf{Z}) \cdot \mathbf{R}$ where $\mathbf{Z}' = \mathbf{t} + \mathbf{Z} \cdot \mathbf{R}$ and thus the reverse-time SDE is equivariant. While the forward SDE is also E(3)-equivariant as the noising process satisfies $q(-\mathbf{Z}(\tau + s)| - \mathbf{Z}(\tau)) = q(\mathbf{Z}(\tau + s)|\mathbf{Z}(\tau))$, it is worth noting that the reverse SDE is only SE(3)-equivariant as parity-inversion transformations $i : \mathbf{Z} \mapsto -\mathbf{Z}$ on the data distribution p_{data} is physically forbidden and thus the score $\nabla_{\mathbf{Z}} \log q_t(\mathbf{Z})$ is of broken chiral symmetry in general: $\exists \mathbf{Z}$ such that $\nabla_{-\mathbf{Z}} \log q_t(-\mathbf{Z}) \neq -\nabla_{\mathbf{Z}} \log q_t(\mathbf{Z})$.

Small-molecule featurization and encoding

We consider two types of nodes to construct a graph-based molecular representation: (a) heavy-atoms $i \in \{1, 2, \cdots, N_{\mathrm{atom}}\}$ and (b) local coordinate frames

$u \in \{1, 2, \cdots, N_{\text{frame}}\}, u := u(ijk)$ formed by atom triplets (i, j, k) that are connected by bonds (ij) and (jk). We introduce Path-integral Graph Transformer (PiFormer), an attentional neural network with edge-level operations inspired by the path-integral formulation of quantum mechanics, to infer the long-range interatomic geometrical correlations for small molecules based on their graph-topological properties. PiFormer operates on the collection of following classes of embeddings:

- Atom representations $\mathbf{H} \in \mathbb{R}^{N_{\text{atom}}} \times c$. The input atom representations is a concatenation of one-hot encodings of element group index and period index for the given atom, which is embedded by a linear projection layer $\mathbb{R}^{18+7} \longrightarrow \mathbb{R}^{c}$;

- Frame representations $\mathbf{F} \in \mathbb{R}^{N_{\text{frame}}} \times c$. For a given frame u, $\mathbf{F}_u$ is initialized by a 2-layer MLP $\mathbb{R}^{4*2+18+7} \longrightarrow \mathbb{R}^{c}$ that embed the bond type encodings (defined as $[\text{is\_single}, \text{is\_double}, \text{is\_triple}, \text{is\_aromatic}]$) of the "incoming" bond $(i(u), j(u))$, "outgoing" bond $(j(u), k(u))$, and the atom type encoding of the center atom $j(u)$;

- Stereochemistry encodings $\mathbf{S} \in \mathbb{R}^{N_{\text{frame}} \times N_{\text{frame}} \times c_{\text{s}}}$. $\mathbf{S}$ is a sparse tensor where an element $\mathbf{S}_{uv}$ is nonzero only if the pair of frames (u, v) is adjacent, i.e., u and v sharing a common incoming or outgoing bond;

- Pair representations $\mathbf{G} \in \mathbb{R}^{N_{\text{frame}} \times N_{\text{atom}} \times c_{\text{p}}}$. $\mathbf{G}$ is initialized by an outer sum of $\mathbf{H}$ and $\mathbf{F}$ which is added to linear-projected $\mathbf{S}$ and passed to a 2-layer MLP.

Elements of the stereochemistry encoding tensor $\mathbf{S}$ are defined as

$$\mathbf{S}_{uv,0} := (\mathrm{common\_bond(u,v)} = \mathrm{incoming\_bond(u)}) \tag{4.17a}$$

$$\mathbf{S}_{uv,1} := (\mathrm{common\_bond(u,v)} = \mathrm{incoming\_bond(v)}) \tag{4.17b}$$

$$\mathbf{S}_{uv,2} := (\mathrm{common\_bond(u,v)} = \mathrm{outgoing\_bond(u)}) \tag{4.17c}$$

$$\mathbf{S}_{uv,3} := (\mathrm{common\_bond(u,v)} = \mathrm{outgoing\_bond(v)}) \tag{4.17d}$$

$$\mathbf{S}_{uv,4} := \mathrm{i(v)} \in \{\mathrm{i(u), j(u), k(u)}\} \tag{4.17e}$$

$$\mathbf{S}_{uv,5} := \mathrm{j(v)} \in \{\mathrm{i(u), j(u), k(u)}\} \tag{4.17f}$$

$$\mathbf{S}_{uv,6} := \mathrm{k(v)} \in \{\mathrm{i(u), j(u), k(u)}\} \tag{4.17g}$$

$$\mathbf{S}_{uv,7} := (\mathrm{j(u)} = \mathrm{j(v)}) \wedge \mathrm{is\_above\_plane(u,v)} \tag{4.17h}$$

$$\mathbf{S}_{uv,8} := (\mathrm{j(u)} = \mathrm{j(v)}) \wedge \mathrm{is\_below\_plane(u,v)} \tag{4.17i}$$

$$\mathbf{S}_{uv,9} := \mathrm{is\_double\_or\_aromatic(common\_bond(u,v))} \vee \mathrm{is\_same\_side(u,v)} \tag{4.17j}$$

$$\mathbf{S}_{uv,10} := \mathrm{is\_double\_or\_aromatic(common\_bond(u,v))} \vee \mathrm{not\_same\_side(u,v)} \tag{4.17k}$$

is_above_plane(u, v) is defined as one of the three atoms in frame v is above the plane formed by frame u with normal vector $\mathbf{v}_u = \frac{(\mathbf{r}_{j(u)}-\mathbf{r}_{i(u)})\times(\mathbf{r}_{k(u)}-\mathbf{r}_{j(u)})}{\|\mathbf{r}_{j(u)}-\mathbf{r}_{i(u)}\|\|\mathbf{r}_{k(u)}-\mathbf{r}_{j(u)}\|}$; is_same_side$(u, v)$ is defined as the two bonds not shared between u, v being on the same side of the common bond, equivalent to $\mathbf{v}_u \cdot \mathbf{v}_v > 0$, vice versa. Our current technical implementations for is_above_plane and is_same_side are based on computing the normal vectors and dot-products using the coordinates from an auxiliary conformer, but we note that in principle all stereochemistry encodings can be generated based on cheminformatic rules without explicit coordinate generations. We additionally denote $\mathrm{MASK_s}$ as a $N_{\mathrm{frame}} \times N_{\mathrm{frame}}$ logical matrix defined as the adjacency matrix of frame pairs (u, v).

The notion of "frames" in a coordinate-free topological molecular graph is justified by the inductive bias that most bending and stretching modes in molecular vibrations are of high frequency, i.e., most bond lengths and bond angles fall into a small range as predicted by valence bond theory, such that the local frames forms a consistent molecular representation without prior knowledge on 3D coordinates. PiFormer operates solely on the molecular representation defined by the input graph, and the frame coordinates $(\mathbf{t}, \mathbf{R})$ are initialized right before the ESDM blocks.

Table 4.2: Composition of the dataset used for pretraining the small-molecule encoder.

Data source	Num. samples collected	Sampling weight	$\mathcal{L}_{3D}$	$\mathcal{L}_{CC}$	$\mathcal{L}_{MLM}$
BioLip [233] ligands (deposited date<2019.1.1)	160k	2.0	+	-	+
GEOM [234]	450k * 5	0.4	+	-	+
DES370k [235]	370k	1.0	+	-	+
PEPCONF [236]	3775	5.0	+	-	+
PCQM4Mv2 [237, 238]	3.4M	0.1	+	-	+
Chemical Checker [239]	800k	1.0	-	+	+

The forward pass of single PiFormer block is expressed as:

$$\mathbf{U}_l = \text{Softmax}_{\text{row-wise}}\Big(\frac{(\mathbf{F}\cdot\mathbf{W}_{\text{K},l})\cdot(\mathbf{F}\cdot\mathbf{W}_{\text{Q},l})^{\text{T}} + \mathbf{S}\cdot\mathbf{W}_{\text{S},l}}{\sqrt{c_{\text{P}}}} + \text{Inf}\cdot\text{MASK}_{\text{s}}\Big) \tag{4.18a}$$

$$\mathbf{G}_{\text{out}} = (\mathbf{1} + \frac{1}{K}\mathbf{U}_l)^K \cdot (\mathbf{G}_l \cdot \mathbf{W}_{\text{G},l}), \quad \mathbf{G}_{l+1} = \text{MLP}([\mathbf{G}_{\text{out}}||(\mathbf{F}_l)^{\text{T}} \cdot \mathbf{H}_l||\mathbf{G}_l]) + \mathbf{G}_l \tag{4.18b}$$

$$\mathbf{F}_{\text{out}} = \text{MHAwithEdgeBias}(\mathbf{F}_l, \mathbf{H}_l, (\mathbf{G}_{l+1})^{\text{T}}), \quad \mathbf{F}_{l+1} = \text{MLP}(\mathbf{F}_{\text{out}} + \mathbf{F}_l) + \mathbf{F}_l \tag{4.18c}$$

$$\mathbf{H}_{\text{out}} = \text{MHAwithEdgeBias}(\mathbf{H}_l, \mathbf{F}_{l+1}, \mathbf{G}_{l+1}), \quad \mathbf{H}_{l+1} = \text{MLP}(\mathbf{H}_{\text{out}} + \mathbf{H}_l) + \mathbf{H}_l \tag{4.18d}$$

where K denotes the propagation length truncation for the learnable graph kernel $\exp(\mathbf{U}_l) \approx (\mathbf{1} + \frac{1}{K}\mathbf{U}_l)^K$ in a single PiFormer block, MLP denotes a 3-layer multilayer perceptron combined with layer normalization [158]. $\mathbf{W}_{\text{K}}, \mathbf{W}_{\text{Q}}, \mathbf{W}_{\text{S}}, \mathbf{W}_{\text{G}}$ are trainable linear weight matrices. $\text{MHAwithEdgeBias}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_{\text{edge}})$ denotes a multi-head cross-attention layer between source node embeddings $\mathbf{X}_1$ and target node embeddings $\mathbf{X}_1$, with edge embeddings $\mathbf{X}_{\text{edge}}$ entering attention computation as a relative positional encoding term as in the relation-aware transformer introduced in [187]. For all models descibed in this study, we set $l_{\max} = 6$ and $K = 8$.

PiFormer model pretraining

In Table 4.2 we summarize the small-molecule datasets used for training the PiFormer encoder used in the reported NeuralPLexer model. The loss function used in PiFormer pretraining is the following:

$$\mathcal{L}_{\text{lig-pretraining}} = \mathcal{L}_{\text{3D-marginal}} + \mathcal{L}_{\text{3D-DSM}} + \mathcal{L}_{\text{CC-regression}} + 0.01\cdot\mathcal{L}_{\text{CC-ismask}} + 0.1\cdot\mathcal{L}_{\text{MLM}} \tag{4.19}$$

We use a mixture density network head to encourage alignment between the learned last-layer pair representations $\mathbf{G}$ and the intra-molecular 3D coordinate marginals. For a single training sample with 3D coordinate observation $\mathbf{y}$:

$$\mathcal{L}_{\text{3D}-\text{marginal}} = \sum_{u}^{N_{\text{frame}}} \sum_{i}^{N_{\text{atom}}} \log \Big[\sum_{l}^{N_{\text{modes}}} \frac{\exp(w_{iul}) \cdot q_{\text{3D}}(T_u^{-1} \circ \mathbf{y}_i | \mathbf{m}_{iul})}{\sum_{l}^{N_{\text{modes}}} \exp(w_{iul})} \Big] \tag{4.20}$$

where $T_u := (\mathbf{R}_u, \mathbf{t}_u)$, $T_u^{-1} \circ \mathbf{y}_i := (\mathbf{y}_i - \mathbf{t}_u) \cdot \mathbf{R}_u^{\text{T}}$. $\mathbf{t}_u \in \mathbb{R}^3$ and $\mathbf{R}_u \in \text{SO}(3)$ are given by

$$(\mathbf{R}_u, \mathbf{t}_u) = \text{rigidFrom3Points}(\mathbf{y}_{i(u)}, \mathbf{y}_{j(u)}, \mathbf{y}_{k(u)}) \tag{4.21}$$

where rigidFrom3Points is the Gram–Schmidt-based frame construction operation described in Ref. [5], Alg. 21; we additionally add a numerical stability factor of 0.01 Å to the vector-norm calculations to handle edge cases when computing the rotation matrices from perturbed coordinates. Each component the 3D distance-angle distribution q^{3D} is parameterized by

$$q_{\text{3D}}(\mathbf{t}|\mu, \sigma, \mathbf{v}) = \text{Gaussian}(\|\mathbf{t}\|_2|\mu, \sigma) \times \text{PowerSpherical}(\frac{\mathbf{t}}{\|\mathbf{t}\|_2}|\mathbf{v}, d = 3) \tag{4.22}$$

where PowerSpherical is a power spherical distribution introduced in [240]; $\mathbf{m}_{iul} := (\mu, \sigma, \mathbf{v})_{iul}$, and

$$[\mathbf{w}_{iu}, \mathbf{m}_{iu}] = \text{3DMixtureDensityHead}\big(\mathbf{G}_{l_{\max}}\big)_{iu}. \tag{4.23}$$

whre 3DMixtureDensityHead is a 3-layer MLP.

Using an equivariant graph transformer similar to ESDM (see Sec. 4.5) but with all receptor nodes dropped, we construct a geometry prediction head to perform global molecular 3D structure denoising. We sample noised coordinates $\mathbf{y}(t)$ from a VPSDE [201] and introduce a SE(3)-invariant denoising score matching loss based on the Frame Aligned Point Error (FAPE) [5]:

$$\mathcal{L}_{\text{3D}-\text{DSM}} = \mathbb{E}_{t\sim(0,1],\mathbf{y}_t\sim q_{0:t}(\cdot|\mathbf{y})} \big[\text{mean}_{u,i} \min(\|T_u^{-1} \circ \mathbf{y}_i - \hat{T}_u^{-1} \circ \hat{\mathbf{y}}_i\|_2, 10\,\text{Å}) \cdot \sqrt{\alpha_t}\big] \tag{4.24}$$

where

$$\hat{\mathbf{y}} = \text{GeometryPredictionHead}(\mathbf{y}_t; \mathbf{H}_{l_{\max}}, \mathbf{F}_{l_{\max}}, \mathbf{G}_{l_{\max}}) \tag{4.25}$$

$\mathcal{L}_{\text{CC}-\text{regression}}$ is a mean squared loss for fitting the "level 1" chemical checker (CC) [239] embeddings which represent harmonized and integrated bioactivity data, and $\mathcal{L}_{\text{CC}-\text{ismask}}$ is an auxiliary binary cross entropy loss for classifying whether a specific CC entry is available for any molecule in the chemical checker dataset. Model

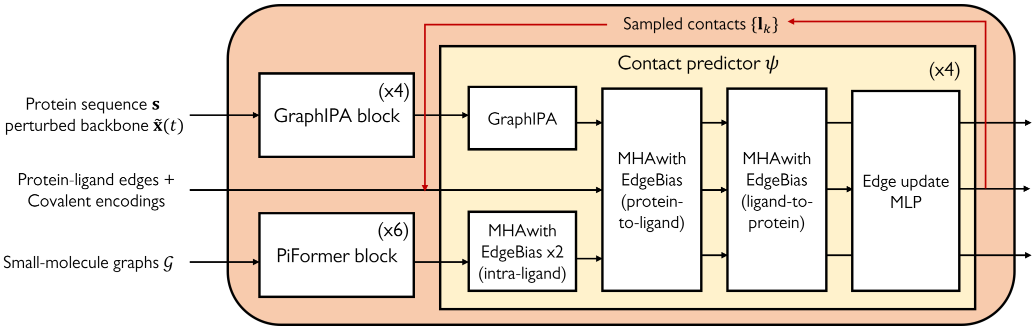
Sampled contacts $\{\mathbf{l}_k\}$
Contact predictor ψ
(x4)
Protein sequence $\mathbf{s}$
perturbed backbone $\tilde{\mathbf{x}}(t)$
(x4)
GraphIPA block
GraphIPA
MHAwith EdgeBias (protein-to-ligand)
MHAwith EdgeBias (ligand-to-protein)
Edge update MLP
Protein-ligand edges + Covalent encodings
(x6)
PiFormer block
MHAwith EdgeBias x2 (intra-ligand)
Small-molecule graphs $\mathcal{G}$

Figure 4.4: Network architecture schematics for the encoders and contact prediction modules.

is trained for 20 epochs with 15% masking ratio for atom and bond encodings, 40% masking ratio for stereochemistry encodings, and dropout=0.1; $\mathcal{L}_{\text{MLM}}$ is a standard cross-entropy loss for predicting the masked tokens which is added to encourage learning on molecular graph topology distributions, but empirically we found $\mathcal{L}_{\text{MLM}}$ converged within the first two epochs and did not find it to influence the learning dynamics of other tasks.

Protein sequence and backbone encoding

The inputs to the protein encoder are (i) the one-hot amino-acid type (20 standard residues + 1 "unknown" token) encoding of the 1D sequence s, (ii) the backbone (N, Cα, C) coordinates of a perturbed protein structure $\mathbf{x}(t)$ sampled from the forward SDEs described in Table 4.1, and (iii) a random Fourier encoding of the diffusion time step t. To reduce memory cost, the protein backbone is represented as a sparse graph with each node mapped to each amino acid residue and randomized edges according to the inclusion probability $p(\text{add\_edge}(i, j)) = \exp(-\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|/10.0\,\text{Å})$ for all residue pairs (i, j). The edge representations are initialized as a random Fourier encoding of the signed sequence distance between two residues (i, j) if i and j are located on the same chain, and are initialized as zeros if (i, j) are located on different chains.

The protein encoder is composed of 4 stacks of invariant point attention (IPA) [5] blocks with two technical modifications:

- The attention scores are computed on the sparsified protein graph, instead of the densely-connected graph as in standard self-attention layers;
- Each node i is associated with n_{head} replicas of coordinate frames $\{T\}_i$, instead of a single frame as in a static structure representation. $\{T\}_i$

is initialized as n_{head} copies of the backbone frames constructed by rigidFrom3Points$(\mathbf{x}_{\text{N},i}, \mathbf{x}_{\text{C}\alpha,i}, \mathbf{x}_{\text{C},i})$. The layer output is $n_{\text{head}} \times 7$ scalars representing the translation vector and the quaternion variable to update the frame associated with each attention head.

the multi-replica design is found to moderately improve model convergence at a fixed network size. For conciseness, we refer to the modified invariant point attention as GraphIPA.

Contact predictor

As illustrated in Figure 4.4, the embeddings from the protein and small-molecule ligand graph encoders are passed to the contact predictor to estimated the contact maps $\mathbf{L}$. A protein-ligand graph is created before the contact predictor forward pass, with pairwise intermolecular edges connecting all protein residues and ligand atoms. The contact predictor is composed of 4 modules each comprises of an intra-protein GraphIPA block, a bidirectional intra-ligand-graph self-attention layer, a bidirectional self-attention layer on the protein-ligand intermolecular edges, and a MLP to update protein-ligand edge representations using the attention maps and previous-layer edge representations. The final edge representations are used to predict $\mathbf{L}$ as described by Equation 4.5. The contact predictor weights are shared across all one-hot contact matrix sampling iterations.

All-atom graph featurization

All protein heavy-atoms nodes (features and 3D coordinates) and the ligand 3D coordinates sampled from the geometry prior q_{T^*} are added to the network inputs right before the ESDM block forward pass. Each protein atom representation is initialized as the concatenation of:

- The residue-wise representation from the protein backbone encoder;
- An one-hot encoding of its atom type as defined by the 37 standard amino acid heavy atom symbols in the PDB format [241];
- A random Fourier encoding of the diffusion time step t.

A random Fourier encoding of the diffusion time step t is also concatenated to the ligand atom representations from the ligand graph encoder and are transformed by a 2-layer MLP.

Given the noised all-atom protein coordinates at diffusion time t, the following edges are added to the protein-ligand graph:

- Edges connecting a protein atom node and the residue node that the protein atom belongs to;
- Edges connecting two protein atom nodes that are within the same residue;
- Edges connecting two protein atom nodes that are within 6.0 Å distance;
- Edges connecting a protein atom node and a ligand atom node that are within 8.0 Å distance;

The protein-atom-involving edges are initialized as a concatenation of the following features:

- A boolean code indicating whether the source node and target node belong to the same residue or the same ligand molecule;
- A boolean code indicating whether there is a covalent bond between the source and target nodes. The covalent bonding information for protein-ligand edges are resolved based on the reference protein-ligand complex structure, where an atom pair (i, j) is considered as a covalent bond if the distance satisfies $d_{ij} < 1.2\sigma_{ij}$ where $\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j)$ is the average Van der Waals (VdW) radius for the atom pair.

To focus the learning problem on binding-site parts of the protein-ligand complex structure, the following *native contact encoding* features are added to the protein sub-graph that do not involve residues that are within 6.0 Å of any ligand heavy atom; given two amino acid residues, we define the native contact encoding as the concatenation of clean-protein-structure $\mathrm{N}-\mathrm{N}$, $\mathrm{C}\alpha-\mathrm{C}\alpha$, and $\mathrm{C}-\mathrm{C}$ distances discretized into [2.0 Å, 4.0 Å, 6.0 Å, 8.0 Å] bins. Such features are embedded by a 2-layer MLP and added to the residue-residue edge representations. Note that at training time the native contact encodings are computed from the protein structure in the ground-truth protein-ligand complex, while at sampling time they are computed from the input backbone template.

The ESDM architecture

The neural network architecture of the proposed equivariant structure diffusion module (ESDM) is summarized in Figure 4.5. The forward pass expression of the trainable

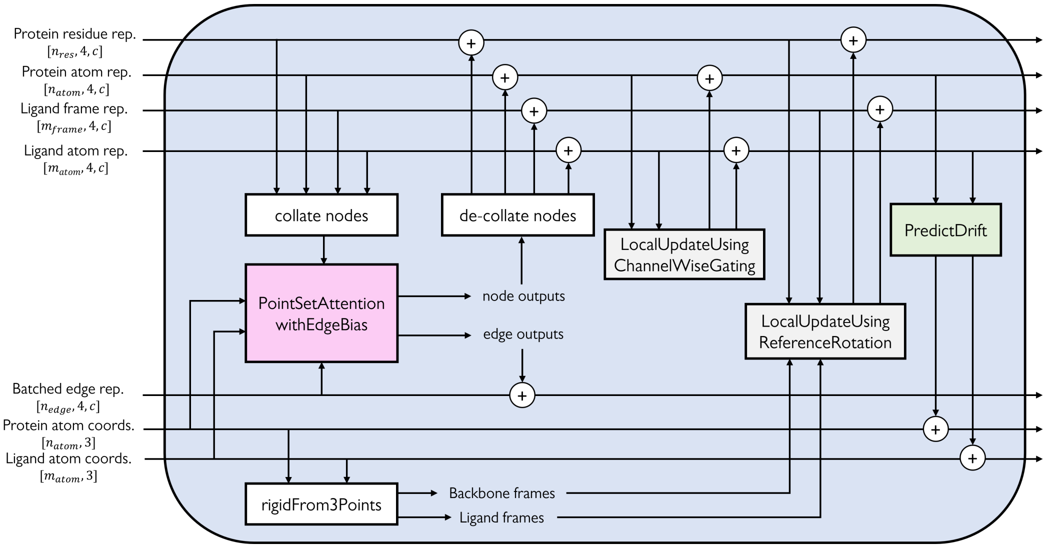

Figure 4.5: Network architecture of a single block in the equivariant structure diffusion module (ESDM). Arrows indicate information flow directions, and "+" indicates an element-wise tensor summation.

modules PointSetAttentionwithEdgeBias, LocalUpdateUsingChannelWiseGating, LocalUpdateUsingReferenceRotation, PredictDrift are defined as:

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}'_{\mathrm{v}}, \mathbf{e}' = \mathrm{PointSetAttentionwithEdgeBias}(\mathbf{f}_{\mathrm{s}}, \mathbf{f}_{\mathrm{v}}, \mathbf{e}, \mathbf{t}) \quad \text{where} \tag{4.26a}$$

$$\mathbf{f}_{\mathrm{Q}}, \mathbf{f}_{\mathrm{K}}, \mathbf{f}_{\mathrm{V}} = \mathbf{W}_{\mathrm{s}} \cdot \mathbf{f}_{\mathrm{s}}, \quad \mathbf{t}_{\mathrm{Q}}, \mathbf{t}_{\mathrm{K}}, \mathbf{t}_{\mathrm{V}} = (\mathbf{t}/10\,\text{Å} + \mathbf{f}_{\mathrm{v}} \cdot \mathbf{W}_{\mathrm{v}}) \tag{4.26b}$$

$$\mathbf{z}_{ij} = \frac{1}{\sqrt{c_{\text{head}}}}(\mathbf{f}_{\mathrm{Q,i}}^{\mathrm{T}} \cdot \mathbf{f}_{\mathrm{K,j}}) + \mathbf{W}_{\mathrm{e}} \cdot \mathbf{e}_{ij} - \frac{\mathbf{w}_{ij}}{\sqrt{18 c_{\text{head}}}} \|\mathbf{t}_{\mathrm{Q}} - \mathbf{t}_{\mathrm{K}}\|_2^2 \tag{4.26c}$$

$$\boldsymbol{\alpha}_{ij} = \mathrm{Softmax}_{j \in \{i\}}(\mathbf{z}_{ij}), \quad \mathbf{e}' = \mathrm{MLP}(\mathbf{z}_{ij}) \tag{4.26d}$$

$$\mathbf{f}'_{\mathrm{s}} = \sum_{j \in \{i\}} \boldsymbol{\alpha}_{ij} \odot \mathbf{f}_{\mathrm{V}}, \quad \mathbf{f}'_{\mathrm{v}} = (\sum_{j \in \{i\}} \boldsymbol{\alpha}_{ij} \odot \mathbf{t}_{\mathrm{V}}) - \mathbf{t}/10\,\text{Å} \tag{4.26e}$$

where $\mathbf{f}_{\mathrm{s}} \in \mathbb{R}^{N_{\text{nodes}} \times c}, \mathbf{f}_{\mathrm{v}} \in \mathbb{R}^{N_{\text{nodes}} \times 3 \times c}, \mathbf{e} \in \mathbb{R}^{N_{\text{edges}} \times c}, \mathbf{t} \in \mathbb{R}^{N_{\text{nodes}} \times 3}$. Note that the expression for computing attention weights $\mathbf{z}$ is directly adapted from IPA.

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}'_{\mathrm{v}} = \mathrm{LocalUpdateUsingChannelWiseGating}(\mathbf{f}_{\mathrm{s}}, \mathbf{f}_{\mathrm{v}}) \quad \text{where} \tag{4.27a}$$

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}_{\text{gate}} = \mathrm{MLP}(\mathbf{f}_{\mathrm{s}} \oplus \|\mathbf{f}_{\mathrm{v}}\|_2) \tag{4.27b}$$

$$\mathbf{f}'_{\mathrm{v}} = (\mathbf{f}_{\mathrm{v}} \cdot \mathbf{W}_{\mathrm{v}}) \odot \mathbf{f}_{\text{gate}} \tag{4.27c}$$

As only linear layers and vector scaling operations are used to update the vector representations $\mathbf{f}_{\mathrm{v}}$, LocalUpdateUsingChannelWiseGating is E(3)-equivariant.

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}'_{\mathrm{v}} = \mathrm{LocalUpdateUsingReferenceRotation}(\mathbf{f}_{\mathrm{s}}, \mathbf{f}_{\mathrm{v}}, \mathbf{R} \in \mathrm{SO}(3)) \quad \text{where} \tag{4.28a}$$

$$\mathbf{f}'_{\mathrm{s}}, \mathbf{f}_{\mathrm{vloc}} = \mathrm{MLP}(\mathbf{f}_{\mathrm{s}} \oplus \mathbf{R}^{\mathrm{T}} \cdot \mathbf{f}_{\mathrm{v}} \oplus \|\mathbf{f}_{\mathrm{v}}\|_2) \tag{4.28b}$$

$$\mathbf{f}'_{\mathrm{v}} = \mathbf{R} \cdot \mathbf{f}_{\mathrm{vloc}} \tag{4.28c}$$

Since the third row of $\mathbf{R}$ is a pseudovector as described in rigidFrom3Points, the determinant of the rotation matrix $\mathbf{R}$ is unchanged under parity inversion transformations $i : \mathbf{x} \mapsto -\mathbf{x}$ and thus the intermediate quantity $\mathbf{f}_{\mathrm{vloc}}$ is SE(3)-invariant but in general **not** invariant under parity inversion i. This property ensures that ESDM can learn the correct chiral symmetry breaking behaviors in molecular 3D conformation distributions.

$$\Delta\mathbf{t} = \mathrm{PredictDrift}(\mathbf{f}_{\mathrm{s}}, \mathbf{f}_{\mathrm{v}}) \quad \text{where} \tag{4.29a}$$

$$\mathbf{o}_{\mathrm{scale}} = \mathrm{Softplus}(\mathrm{MLP}(\mathbf{f}_{\mathrm{s}})) \tag{4.29b}$$

$$\Delta\mathbf{t} = (\mathbf{f}_{\mathrm{v}} \cdot \mathbf{W}_{\mathrm{drift}}) \odot \mathbf{o}_{\mathrm{scale}}. \tag{4.29c}$$

The predicted drift vectors $\Delta\mathbf{t}$ are added to the input node coordinates; the final coordinate outputs are taken as the predicted denoised observations $\hat{\mathbf{x}}(0), \hat{\mathbf{y}}(0)$.

Model training and hyperparameters

The loss function for NeuralPLexer training is:

$$\mathcal{L}_{\mathrm{training}} = \mathbb{E}_{t\sim(0,1]}\left[\mathcal{L}_{\mathrm{contact}}(t) + \mathcal{L}_{\mathrm{gp-mean}}(t) + \mathcal{L}_{\mathrm{DSM-prot}}(t) + \mathcal{L}_{\mathrm{DSM-ligand}}(t) + \mathcal{L}_{\mathrm{DSM-site}}(t)\right] \tag{4.30}$$

We train the contact predictor ψ to match the posterior distribution defined by the observed contact map $q_{\mathbf{L}} := \mathrm{Categorical}_{n_{\mathrm{res}}\times m}(\mathbf{L})$ where $\mathbf{L} := \bigoplus_k \mathbf{L}_k$ with intermediate ligand-wise one-hot matrices $\mathbf{l}_k$ sampled from $q_{\mathbf{L}_k}$:

$$\mathcal{L}_{\mathrm{contact}}(t) = \mathrm{KL}(q_{\mathbf{L}} \| q_{\psi}(\cdot|\mathbf{0}, \mathbf{s}, \tilde{\mathbf{x}}(t), \mathcal{G})) + \sum_{k=1}^{K} \mathbb{E}_{\mathbf{l}_k \sim q_{\mathbf{L}_k}}\left[\mathrm{JS}(q_{\mathbf{L}_k} \| q_{\psi,k}(\cdot| \sum_{r=1}^{k} \mathbf{l}_r, \mathbf{s}, \tilde{\mathbf{x}}(t), \mathcal{G}))\right] \tag{4.31}$$

where KL denotes a Kullback–Leibler divergence and JS denotes a Jensen–Shannon divergence. An auxiliary loss is added to the mean term in the predicted geometry prior:

$$\mathcal{L}_{\mathrm{gp-mean}}(t) = \mathbb{E}_{\mathbf{l}_k \sim q_{\mathbf{L}_k}}\left[\|\mathbf{c}^{\mathrm{T}}_{\psi,k}(\sum_{r=1}^{k} \mathbf{l}_r, \mathbf{s}, \tilde{\mathbf{x}}(t), \mathcal{G}) \cdot \tilde{\mathbf{x}}(t) - \mathbf{c} \cdot \tilde{\mathbf{x}}(t)\|\right] \tag{4.32}$$

The denoising score matching (DSM) loss expressions are given by

$$\mathcal{L}_{\text{DSM-prot}} = \mathbb{E}_{\mathbf{x}(t),\mathbf{y}(t)\sim q_{0:t}(\cdot|\mathbf{x}(0),\mathbf{y}(0))}\big[\frac{1}{n}\sum_i \|\mathbf{x}_i(0) - \hat{\mathbf{x}}_i(0)\|_2/\sigma(t)\big] \tag{4.33}$$

$\mathcal{L}_{\text{DSM-site}}$ is defined analogously but averaged for residues that are within 6.0 Å of the ligand in the ground-truth structure. Lastly

$$\mathcal{L}_{\text{DSM-ligand}} = \mathbb{E}_{\mathbf{x}(t),\mathbf{y}(t)\sim q_{0:t}(\cdot|\mathbf{x}(0),\mathbf{y}(0))}\big[\frac{1}{m}\sum_i \|\mathbf{y}_i(0) - \hat{\mathbf{y}}_i(0)\|_2/\sigma(t)\big]. \tag{4.34}$$

For the ligand graph encoder, we use 6 PiFormer blocks with an embedding dimension of 512 for atom representation and frame representations, and a dimension of 128 for pair representations. For the protein encoder, we use 4 GraphIPA blocks with a node embedding dimension of 256 and edge embedding dimension of 64. For the contact predictor we use 4 blocks with the same embeddings sizes (256, 64) as in the protein encoder; linear layers are added to project the ligand representations to the length of protein representations before they are passed to the contact predictor. For ESDM, we use a stack of 4 blocks with a embedding dimension of 64 for both node and edge representations, that is, each node i is associated with scalar representations $\mathbf{f}_{\text{s},i}$ of size 64 and vector representations $\mathbf{f}_{\text{v},i}$ of size [3, 64].

The pretrained small-molecule encoder weights are frozen during training. Model is trained with batch size of 8 for 40 epochs, using dropout=0.05, an inital learning rate of 3E-4 with 1000 warmup steps followed by a cosine annealing learning rate decay schedule. On the PDBBind 2020 training set (170k samples), the training run took 20 hours a single NVIDIA-Tesla-V100-SXM2-32GB GPU.

Task-specific fine-tuning

The model used for fixed-backbone protein-ligand docking is fine-tuned on the original PDBBind training dataset, while all backbone atoms $(\text{N}, \text{C}\alpha, C, O)$ and Cβ atoms are set to the ground-truth coordinates. Fine-tuning is performed for 20 epochs with a batch size of 8 without teacher forcing for the geometry prior (i.e., sampling the one-hot matrix $\mathbf{l}$ from the observed contact map $q_{\mathbf{L}} = \text{Categorical}_{n_{\text{res}}\times m}(\mathbf{L})$, using the predicted contact map $\psi(\mathbf{l}, \mathbf{s}, \tilde{\mathbf{x}}, \mathcal{G})$ to parameterize the finite-time transition kernels $q_t(\mathbf{Z}(t)|\mathbf{Z}(0))$ during model forward pass, and then backpropagating the model end-to-end) using a cosine annealing schedule with an initial learning rate of $1E-4$.

The model used for binding-site inpainting is fine-tuned on all split-chain samples from the original PDBBind training dataset. A protein-chain/ligand pair is included in the fine-tuning dataset if any heavy atom of the ligand is within 10 Å of any heavy atom of the protein chain. All receptor residues that are not within 6.0 Å of the ligand are set to the ground-truth coordinates with the residue-wise and protein-atom-wise time-step encoding set to zeros. Fine-tuning is performed for 40 epochs with a batch size of 10 without teacher forcing for the geometry prior using a cosine annealing schedule with a initial learning rate of $1E-4$.

Computational details

Test datasets and post-processing

While the time-split-based PDBBind 2020 dataset has been used in previous works for studying model generalization to novel protein-ligand pairs, we noticed that the 363-sample test set curated by [222] contains samples with improperly removed alternative ligand conformation ground truths or deleted adjacent chains that strongly interact with the ligand molecule in the full structure (e.g., binding sites near protein-protein interfaces). To ensure a reasonable comparison to docking-based methods, for the test dataset used fixed-backbone ligand conformation prediction experiments we keep all protein chains that are within 10 Å of the ligand from the original PDB file instead of using the receptor PDB files curated by PDBBind; we further removed all covalent ligands and pipetide binders from the test set as such cases are usually tackled by specialized algorithms [242, 243], resulting in 275 test samples in total to produce the results presented in Figure 4.2a-d.

The AlphaFold2 structures used in the ligand-coupled binding site repacking task are predicted using ColabFold [244] with default MSA, recycling, and AMBER relaxation settings, and without using templates in order to best reflect the prediction fidelity of AlphaFold2 on novel targets (since all PDBBind test set samples are deposited before year 2021). The input sequences for all protein chains are obtained from `https://www.ebi.ac.uk/pdbe/api/pdb/entry/molecules/` to avoid issues related to unresolved residues and to represent a realistic testing scenario where the protein backbone models are obtained from the full sequence.

Baseline method configurations

We run CB-Dock [223] with a heuristic low-sampling-intensity configuration (exhaustiveness=1, number of clustered binding sites to start local docking = 1) such that the execution time (43 seconds per ligand on average on single core of an Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz CPU) is comparable to deep-learning-based methods that were proposed to perform docking at a low computing budget. The top-scored ligand conformations collected for each protein-ligand pair as ranked by Autodock Vina [245] are used to obtain the success rate results in Figure 2a. EquiBind [222] are launched with the default configuration file, and for each protein-ligand pair 64 ligand conformations are generated using different random RDKit [246] input conformers. We note that the incorporation of side-chain flexibility as provided by AutoDock Vina and the systematic tuning of sampling intensity in docking-based methods may offer a more accurate comparision regarding the accuracy/computational time relationship among physics-based and learning-based methods.

RosettaLigand [202] runs are launched with a configuration modified from the standard protocol. We set the receptor Calpha constraint parameter to 100.0 to enable a fully flexible receptor; the ligand coordinates are initialized using the aligned-ground-truth conformation as obtained by TM-Align [228], with randomized torsion angles using the BCL [247] library as described in the standard protocol. We set the docking box width to 4.0 Å and remove the ligand center perturbation step to ensure the ligand search space during the low-resolution docking stage is constrained to the binding site location. While high-fideltiy physics-based methods such as IFD-MD [197] have been proposed for flexible-receptor ligand docking, such algorithms often incur orders-of-magnitude higher computational cost, and thus are not included within the scope of this study.

Evaluation metric details

All protein structure alignments and TM-Score calculations are performed using TMAlign [228]. All reported TM-Scores are normalized by the chain length of the reference PDB structure. The per-residue all-atom lDDT score is computed using OpenStructure [248]; the lDDT-BS score is then computed by averaging the per-residue scores for ligand binding site residues with a cutoff of 4.0 Å as used in CAMEO [227]. The symmetry-corrected heavy-atom RMSD for ligand structure comparison is computed using the obrms function in OpenBabel [249]. A standard 6-12 Lennard-Jones energy functional form is used for computing the clash rate statistics; the L-J energy and VdW radius parameters are obtained from the UFF parameter file retrieved from `https://github.com/kbsezginel/lammps-data-file/blob/master/uff-parameters.csv`.

BIBLIOGRAPHY

[1] Or Sharir et al. “Deep autoregressive models for the efficient variational simulation of many-body quantum systems”. In: *Physical review letters* 124.2 (2020), p. 020503.

[2] Jan Hermann, Zeno Schätzle, and Frank Noé. “Deep-neural-network solution of the electronic Schrödinger equation”. In: *Nature Chemistry* 12.10 (2020), pp. 891–897.

[3] James Kirkpatrick et al. “Pushing the frontiers of density functionals by solving the fractional electron problem”. In: *Science* 374.6573 (2021), pp. 1385–1389.

[4] Bingqing Cheng et al. “Evidence for supercritical behaviour of high-pressure liquid hydrogen”. In: *Nature* 585.7824 (2020), pp. 217–220.

[5] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589.

[6] Raphael JL Townshend et al. “Geometric deep learning of RNA structure”. In: *Science* 373.6558 (2021), pp. 1047–1051.

[7] Michael M Bronstein et al. “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges”. In: *arXiv preprint arXiv:2104.13478* (2021).

[8] Robin Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695.

[9] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry*. Mineola: Dover, 1996, pp. 231–239. ISBN: 0486691861.

[10] John P Perdew and Karla Schmidt. “Jacob’s ladder of density functional approximations for the exchange-correlation energy”. In: *AIP Conference Proceedings*. Vol. 577. 1. American Institute of Physics. 2001, pp. 1–20.

[11] Oliver Lampret et al. “The roles of long-range proton-coupled electron transfer in the directionality and efficiency of [FeFe]-hydrogenases”. In: *Proceedings of the National Academy of Sciences* 117.34 (2020), pp. 20520–20529.

[12] Alexander Urban, Dong-Hwa Seo, and Gerbrand Ceder. “Computational understanding of Li-ion batteries”. In: *npj Computational Materials* 2.1 (2016), pp. 1–13.

[13] Fernand Spiegelman et al. “Density-functional tight-binding: basic concepts and applications to molecules and clusters”. In: *Advances in physics: X* 5.1 (2020), p. 1710252.

[14] Christoph Bannwarth et al. “Extended tight-binding quantum chemistry methods”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 11.2 (2021), e1493.

[15] Jun John Sakurai and Eugene D Commins. *Modern quantum mechanics, revised edition*. 1995.

[16] Taco Cohen and Max Welling. “Group equivariant convolutional networks”. In: *International conference on machine learning*. PMLR. 2016, pp. 2990–2999.

[17] Tamar Schlick. *Molecular modeling and simulation: an interdisciplinary guide*. Vol. 2. Springer, 2010.

[18] Brian DO Anderson. “Reverse-time diffusion equation models”. In: *Stochastic Processes and their Applications* 12.3 (1982), pp. 313–326.

[19] James Glimm and Arthur Jaffe. *Quantum physics: a functional integral point of view*. Springer Science & Business Media, 2012.

[20] Albert P Bartók et al. “Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons”. In: *Physical review letters* 104.13 (2010), p. 136403.

[21] Matthias Rupp et al. “Fast and accurate modeling of molecular atomization energies with machine learning”. In: *Phys. Rev. Lett.* 108.5 (2012), p. 58301.

[22] Anders S Christensen et al. “FCHL revisited: Faster and more accurate quantum machine learning”. In: *J. Chem. Phys.* 152.4 (2020), p. 044107.

[23] Anders S Christensen and O Anatole von Lilienfeld. “Operator quantum machine learning: Navigating the chemical space of response properties”. In: *CHIMIA* 73.12 (2019), pp. 1028–1031.

[24] Raghunathan Ramakrishnan et al. “Big data meets quantum chemistry approximations: the Δ-machine learning approach”. In: *J. Chem. Theory Comput.* 11.5 (2015), p. 2087.

[25] Thuong T. Nguyen et al. “Comparison of permutationally invariant polynomials, neural networks, and Gaussian approximation potentials in representing water interactions through many-body expansions”. In: *J. Chem. Phys.* 148.24 (2018), p. 241725.

[26] So Fujikake et al. “Gaussian approximation potential modeling of lithium intercalation in carbon nanostructures”. In: *J. Chem. Phys.* 148.24 (June 2018), p. 241714. ISSN: 0021-9606. DOI: 10.1063/1.5016317. URL: http://aip.scitation.org/doi/10.1063/1.5016317.

[27] Andrea Grisafi et al. “Transferable Machine-Learning Model of the Electron Density”. In: *ACS Cent. Sci.* 5.1 (2019), pp. 57–64.

[28] Yaoguang Zhai et al. “Active learning of many-body configuration space: Application to the Cs^+–water MB-nrg potential energy function as a case study”. In: *The Journal of Chemical Physics* 152.14 (2020), p. 144103.

[29] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. “ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost”. In: *Chemical science* 8.4 (2017), pp. 3192–3203.

[30] Felix Brockherde et al. “Bypassing the Kohn-Sham equations with machine learning”. In: *Nat. Commun.* 8.1 (2017), p. 872.

[31] Zhenqin Wu et al. “MoleculeNet: a benchmark for molecular machine learning”. In: *Chem. Sci.* 9.2 (2018), p. 513.

[32] Kun Yao et al. “The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics”. In: *Chem. Sci.* 9.8 (2018), pp. 2261–2269. ISSN: 20416539. DOI: 10.1039/c7sc04934j.

[33] Haichen Li et al. “A Density Functional Tight Binding Layer for Deep Learning of Chemical Hamiltonians”. In: *J. Chem. Theory Comput.* 14.11 (2018), pp. 5764–5776. DOI: 10.1021/acs.jctc.8b00873.

[34] Linfeng Zhang et al. “Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics”. In: *Phys. Rev. Lett.* 120 (14 Apr. 2018), p. 143001. DOI: 10.1103/PhysRevLett.120.143001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.120.143001.

[35] Justin S Smith et al. “Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning”. In: *Nat. Commun.* 10.1 (2019), pp. 1–8.

[36] Nicholas Lubbers, Justin S. Smith, and Kipton Barros. “Hierarchical modeling of molecular energies using a deep neural network”. In: *J. Chem. Phys.* 148.24 (2018), p. 241715. DOI: 10.1063/1.5011181.

[37] Grégoire Montavon et al. “Machine learning of molecular electronic properties in chemical compound space”. In: *New J. Phys.* 15.9 (2013), p. 95003.

[38] Katja Hansen et al. “Assessment and validation of machine learning methods for predicting molecular atomization energies”. In: *J. Chem. Theory Comput.* 9.8 (2013), p. 3404.

[39] Piero Gasparotto and Michele Ceriotti. “Recognizing molecular patterns by machine learning: an agnostic structural definition of the hydrogen bond”. In: *J. Chem. Phys.* 141.17 (2014), p. 174110.

[40] Jörg Behler. “Perspective: Machine learning potentials for atomistic simulations”. In: *J. Chem. Phys.* 145.17 (Nov. 2016), p. 170901. ISSN: 0021-9606. DOI: 10.1063/1.4966192. URL: http://aip.scitation.org/doi/10.1063/1.4966192.

[41] Steven Kearnes et al. "Molecular graph convolutions: moving beyond fingerprints". In: *J. Comput. Aided Mol. Des.* 30.8 (2016), p. 595.

[42] Kristof T Schütt et al. "Quantum-chemical insights from deep tensor neural networks". In: *Nat. Commun.* 8 (2017), p. 13890.

[43] Matthew Welborn, Lixue Cheng, and Thomas F Miller III. "Transferability in machine learning for electronic structure via the molecular orbital basis". In: *J. Chem. Theory Comput.* 14.9 (2018), pp. 4772–4779.

[44] Lixue Cheng et al. "A universal density matrix functional from molecular orbital-based machine learning: Transferability across organic molecules". In: *J. Chem. Phys.* 150.13 (2019), p. 131103.

[45] Lixue Cheng et al. "Regression Clustering for Improved Accuracy and Training Costs with Molecular-Orbital-Based Machine Learning". In: *J. Chem. Theory Comput.* 15.12 (2019), pp. 6668–6677. DOI: 10.1021/acs.jctc.9b00884.

[46] Sebastian Dick and Marivi Fernandez-Serra. "Machine learning accurate exchange and correlation functionals of the electronic density". In: *Nat. Commun.* 11.1 (July 2020), p. 3509. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17265-7. URL: https://doi.org/10.1038/s41467-020-17265-7.

[47] Yixiao Chen et al. "Ground State Energy Functional with Hartree–Fock Efficiency and Chemical Accuracy". In: *The Journal of Physical Chemistry A* 124.35 (2020), pp. 7155–7165.

[48] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations* (ICLR). Palais des Congrès Neptune, Toulon, France, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

[49] Petar Veličković et al. "Graph Attention Networks". In: *International Conference on Learning Representations*. 2018.

[50] Kevin Yang et al. "Analyzing learned molecular representations for property prediction". In: *J. Chem. Inf. Model.* 59.8 (2019), pp. 3370–3388.

[51] Kristof Schütt et al. "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions". In: *Advances in neural information processing systems*. 2017, pp. 991–1001.

[52] Oliver T Unke and Markus Meuwly. "PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges". In: *J. Chem. Theory Comput.* 15.6 (2019), pp. 3678–3693.

[53] Johannes Klicpera, Janek Groß, and Stephan Günnemann. "Directional Message Passing for Molecular Graphs". In: *International Conference on Learning Representations (ICLR)*. 2020.

[54] Ziteng Liu et al. “Transferable multi-level attention neural network for accurate prediction of quantum chemistry properties via multi-task learning”. In: *ChemRxiv* 12588170 (2020), p. v1.

[55] Stefan Grimme. “A simplified Tamm-Dancoff density functional approach for the electronic excitation spectra of very large molecules”. In: *J. Chem. Phys.* 138.24 (2013), p. 244104.

[56] Stefan Grimme and Christoph Bannwarth. “Ultra-fast computation of electronic spectra for large systems by tight-binding based simplified Tamm-Dancoff approximation (sTDA-xTB)”. In: *J. Chem. Phys.* 145.5 (2016), p. 054103.

[57] Tobias Risthaus, Andreas Hansen, and Stefan Grimme. “Excited states using the simplified Tamm–Dancoff-Approach for range-separated hybrid density functionals: development and application”. In: *Phys. Chem. Chem. Phys.* 16.28 (2014), pp. 14408–14419.

[58] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[59] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[60] Stefan Grimme, Christoph Bannwarth, and Philip Shushkov. “A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z=1–86)”. In: *J. Chem. Theory Comput.* 13.5 (2017), pp. 1989–2009.

[61] Philipp Pracht et al. “A Robust Non-Self-Consistent Tight-Binding Quantum Chemistry Method for large Molecules”. In: *ChemRxiv preprint ChemRxiv:10.26434/chemrxiv.8326202.v1* (June 2019).

[62] Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. “GFN2-xTB — An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions”. In: *J. Chem. Theory Comput.* 15.3 (2019), pp. 1652–1671.

[63] Hongyan Jiang et al. “Nuclear Quantum Effects in Scattering of H and D from Graphene”. In: *arXiv preprint arXiv:2007.03372* (2020).

[64] Dakota Folmsbee and Geoffrey Hutchison. “Assessing conformer energies using electronic structure and machine learning methods”. In: *Int. J. Quantum Chem.* (2020), e26381. doi: `10.1002/qua.26381`.

[65] Raghunathan Ramakrishnan et al. “Quantum chemistry structures and properties of 134 kilo molecules”. In: *Sci. Data* 1.1 (2014), pp. 1–7.

[66] *Semiempirical Extended Tight-Binding Program Package*. https://github.com/grimme-lab/xtb. 2020, accessed July 14, 2020.

[67] Sebastian JR Lee et al. "Analytical gradients for projection-based wavefunction-in-DFT embedding". In: *The Journal of Chemical Physics* 151.6 (2019), p. 064112.

[68] Martin Schütz et al. "Analytical energy gradients for local second-order Møller–Plesset perturbation theory using density fitting approximations". In: *The Journal of chemical physics* 121.2 (2004), pp. 737–750.

[69] Adam Paszke et al. "Automatic Differentiation in PyTorch". In: *NIPS 2017 Workshop on Autodiff*. Long Beach, California, USA, 2017. URL: https://openreview.net/forum?id=BJJsrmfCZ.

[70] Tobias Risthaus, Marc Steinmetz, and Stefan Grimme. "Implementation of nuclear gradients of range-separated hybrid density functionals and benchmarking on rotational constants for organic molecules". In: *Journal of Computational Chemistry* 35.20 (2014), pp. 1509–1516.

[71] Uma R Fogueri et al. "The melatonin conformer space: Benchmark and assessment of wave function and DFT methods for a paradigmatic biological and pharmacological molecule". In: *The Journal of Physical Chemistry A* 117.10 (2013), pp. 2269–2277.

[72] Zhao Chen et al. "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 794–803.

[73] Weihua Hu et al. "Strategies for Pre-training Graph Neural Networks". In: *International Conference on Learning Representations*. 2019.

[74] Garrett B Goh et al. "Using rule-based labels for weak supervised learning: a ChemNet for transferable chemical property prediction". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 302–310.

[75] Zhuoran Qiao, Feizhi Ding, Matthew Welborn, Peter J. Bygrave, Daniel G. A. Smith, Animashree Anandkumar, Frederick R. Manby, and Thomas F. Miller. "Multi-task learning for electronic structure to predict and explore molecular potential energy surfaces". In: *arXiv preprint arXiv:2011.02680* (2020). Appeared at *Machine Learning for Molecules workshop at NeurIPS 2020* as a Contributed Talk. DOI: 10.48550/ARXIV.2011.02680.

[76] Zhuoran Qiao, Matthew Welborn, Animashree Anandkumar, Frederick R Manby, and Thomas F Miller III. "OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features". In: *The Journal of Chemical Physics* 153.12 (2020), p. 124111. DOI: 10.1063/5.0021955.

[77] Guoqing Zhou et al. "GPU-Accelerated Semi-Empirical Born Oppenheimer Molecular Dynamics using PyTorch". In: *J. Chem. Theory Comput.* (2020).

[78] Juho Lee et al. “Set transformer: A framework for attention-based permutation-invariant neural networks”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3744–3753.

[79] Lixue Cheng et al. *Thermalized (350K) QM7b, GDB-13, water, and short alkane quantum chemistry dataset including MOB-ML features*. https://data.caltech.edu/records/1177. 2019. DOI: 10.22002/D1.1177.

[80] L. C. Blum and J.-L. Reymond. “970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13”. In: *J. Am. Chem. Soc.* 131 (2009), p. 8732.

[81] Vivian Law et al. “DrugBank 4.0: shedding new light on drug metabolism”. In: *Nucleic Acids Res.* 42.D1 (2014), pp. D1091–D1097.

[82] S H Vosko, L Wilk, and M Nusair. “Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis”. In: *Can. J. Phys.* 58.8 (1980), p. 1200. DOI: 10.1139/p80-159.

[83] Chengteh Lee, Weitao Yang, and Robert G Parr. “Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density”. In: *Phys. Rev. B* 37.2 (1988), p. 785. DOI: 10.1103/PhysRevB.37.785.

[84] Axel D Becke. “Density-functional thermochemistry. III. The role of exact exchange”. In: *J. Chem. Phys.* 98.7 (1993), p. 5648. DOI: 10.1063/1.464913.

[85] P J Stephens et al. “Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields”. In: *J. Phys. Chem.* 98.45 (Nov. 1994), p. 11623. DOI: 10.1021/j100096a001.

[86] P C Hariharan and J A Pople. “The influence of polarization functions on molecular orbital hydrogenation energies”. In: *Theor. Chim. Acta* 28.3 (1973), p. 213. DOI: 10.1007/BF00533485.

[87] Giovanni Bussi and Michele Parrinello. “Accurate sampling using Langevin dynamics”. In: *Phys. Rev. E* 75.5 (2007), p. 056707.

[88] You-Sheng Lin et al. “Long-range corrected hybrid density functionals with improved dispersion corrections”. In: *J. Chem. Theory Comput.* 9.1 (2013), pp. 263–272.

[89] Florian Weigend and Reinhart Ahlrichs. “Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy”. In: *Phys. Chem. Chem. Phys.* 7 (2005).

[90] Robert Polly et al. “Fast Hartree-Fock theory using local density fitting approximations”. In: *Mol. Phys.* 102.21-22 (Nov. 2004), pp. 2311–2321. ISSN: 0026-8976. DOI: 10.1080/0026897042000274801. URL: http://www.tandfonline.com/doi/abs/10.1080/0026897042000274801.

[91] Florian Weigend. “Hartree-Fock exchange fitting basis sets for H to Rn”. In: *J. Comput. Chem.* 29 (2008), pp. 167–175.

[92] Daniel G. A. Smith et al. “PSI4 1.4: Open-source software for high-throughput quantum chemistry”. In: *J. Chem. Phys.* 152.18 (2020), p. 184108.

[93] Frederick Manby et al. “entos: A Quantum Molecular Simulation Package”. In: *ChemRxiv preprint 10.26434/chemrxiv.7762646.v2* (2019). URL: `https://chemrxiv.org/articles/entos_A_Quantum_Molecular_Simulation_Package/7762646`.

[94] You-Sheng Lin et al. “Long-range corrected hybrid density functionals with improved dispersion corrections”. In: *Journal of Chemical Theory and Computation* 9.1 (2013), pp. 263–272.

[95] Lee-Ping Wang and Chenchen Song. “Geometry optimization made simple with translation and rotation coordinates”. In: *The Journal of chemical physics* 144.21 (2016), p. 214108.

[96] Jan Gerit Brandenburg et al. “B97-3c: A revised low-cost variant of the B97-D density functional method”. In: *The Journal of chemical physics* 148.6 (2018), p. 064104.

[97] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[98] Leslie N Smith and Nicholay Topin. “Super-convergence: Very fast training of neural networks using large learning rates”. In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Vol. 11006. International Society for Optics and Photonics. 2019, p. 1100612.

[99] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *International Conference on Machine Learning*. 2015, pp. 448–456.

[100] Priya Goyal et al. “Accurate, large minibatch sgd: Training imagenet in 1 hour”. In: *arXiv preprint arXiv:1706.02677* (2017).

[101] Chen Ling. “A review of the recent progress in battery informatics”. In: *npj Computational Materials* 8.1 (2022), pp. 1–22.

[102] Jessica Vamathevan et al. “Applications of machine learning in drug discovery and development”. In: *Nature reviews Drug discovery* 18.6 (2019), pp. 463–477.

[103] Mu-Hyun Baik and Richard A Friesner. “Computing redox potentials in solution: Density functional theory as a tool for rational design of redox agents”. In: *The Journal of Physical Chemistry A* 106.32 (2002), pp. 7407–7412.

[104] Brian K Shoichet. “Virtual screening of chemical libraries”. In: *Nature* 432.7019 (2004), pp. 862–865.

[105] Manuel Cordova et al. "Data-driven advancement of homogeneous nickel catalyst activity for aryl ether cleavage". In: *Acs Catalysis* 10.13 (2020), pp. 7021–7031.

[106] Walter Kohn and Lu Jeu Sham. "Self-consistent equations including exchange and correlation effects". In: *Physical review* 140.4A (1965), A1133.

[107] Alexander D MacKerell Jr. "Empirical force fields for biological macromolecules: overview and issues". In: *Journal of computational chemistry* 25.13 (2004), pp. 1584–1604.

[108] Yi Liu et al. "Spherical Message Passing for 3D Graph Networks". In: *arXiv preprint arXiv:2102.05013* (2021).

[109] Kristof T Schütt, Oliver T Unke, and Michael Gastegger. "Equivariant message passing for the prediction of tensorial properties and molecular spectra". In: *arXiv preprint arXiv:2102.03150* (2021).

[110] Jörg Behler and Michele Parrinello. "Generalized neural-network representation of high-dimensional potential-energy surfaces". In: *Phys. Rev. Lett.* 98.14 (Apr. 2007), p. 146401. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.98.146401. URL: https://link.aps.org/doi/10.1103/PhysRevLett.98.146401.

[111] Linfeng Zhang et al. "Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics". In: *Physical review letters* 120.14 (2018), p. 143001.

[112] David Rosenberger, Justin S Smith, and Angel E Garcia. "Modeling of peptides with classical and novel machine learning force fields: A comparison". In: *The Journal of Physical Chemistry B* 125.14 (2021), pp. 3598–3612.

[113] Li Li et al. "Kohn-Sham equations as regularizer: Building prior knowledge into machine-learned physics". In: *Physical review letters* 126.3 (2021), p. 036401.

[114] Ryo Nagai, Ryosuke Akashi, and Osamu Sugino. "Completing density functional theory by machine learning hidden messages from molecules". In: *npj Computational Materials* 6.1 (2020), pp. 1–8.

[115] Kaycee Low, Michelle L Coote, and Ekaterina I Izgorodina. "Inclusion of More Physics Leads to Less Data: Learning the Interaction Energy as a Function of Electron Deformation Density with Limited Training Data". In: *Journal of Chemical Theory and Computation* (2022).

[116] Konstantin Karandashev and O Anatole von Lilienfeld. "An orbital-based representation for accurate Quantum Machine Learning". In: *arXiv preprint arXiv:2112.12877* (2021).

[117] Maurice Weiler et al. "3d steerable cnns: Learning rotationally equivariant features in volumetric data". In: *arXiv preprint arXiv:1807.02547* (2018).

[118] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. "Clebsch–gordan nets: a fully fourier space spherical convolutional neural network". In: *Advances in Neural Information Processing Systems* 31 (2018).

[119] Brandon Anderson, Truong Son Hy, and Risi Kondor. "Cormorant: Covariant Molecular Neural Networks". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 14537–14546. URL: `http://papers.nips.cc/paper/9596-cormorant-covariant-molecular-neural-networks.pdf`.

[120] Nathaniel Thomas et al. "Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds". In: *arXiv preprint arXiv:1802.08219* (2018).

[121] Fabian B Fuchs et al. "SE (3)-transformers: 3D roto-translation equivariant attention networks". In: *arXiv preprint arXiv:2006.10503* (2020).

[122] Leanne D Chen et al. "Embedded Mean-Field Theory for Solution-Phase Transition-Metal Polyolefin Catalysis". In: *Journal of Chemical Theory and Computation* 16.7 (2020), pp. 4226–4237.

[123] Jon Paul Janet et al. "Accurate multiobjective design in a space of millions of transition metal complexes with neural-network-driven efficient global optimization". In: *ACS central science* 6.4 (2020), pp. 513–524.

[124] Abigail Dommer et al. "# COVIDisAirborne: AI-Enabled Multiscale Computational Microscopy of Delta SARS-CoV-2 in a Respiratory Aerosol". In: *bioRxiv* ().

[125] Felix A Faber et al. "Alchemical and structural distribution based representation for universal quantum machine learning". In: *J. Chem. Phys.* 148.24 (2018), p. 241717. DOI: `10.1063/1.5020710`.

[126] Anders S. Christensen, Felix A. Faber, and O. Anatole von Lilienfeld. "Operators in quantum machine learning: Response properties in chemical space". In: *J. Chem. Phys.* 150.6 (2019), p. 064105. DOI: `10.1063/1.5053562`.

[127] Bing Huang and O Anatole von Lilienfeld. "Efficient accurate scalable and transferable quantum machine learning with am-ons". In: *arXiv preprint arXiv:1707.04146* (2017).

[128] Albert P Bartók et al. "Machine learning unifies the modeling of materials and molecules". In: *Science advances* 3.12 (2017), e1701816.

[129] Anders S Christensen, Felix A Faber, and O Anatole von Lilienfeld. "Operators in quantum machine learning: Response properties in chemical space". In: *The Journal of chemical physics* 150.6 (2019), p. 064105.

[130] Max Veit et al. "Predicting molecular dipole moments by combining atomic partial charges and atomic dipoles". In: *The Journal of Chemical Physics* 153.2 (2020), p. 024113.

[131] Alberto Fabrizio et al. “Electron density learning of non-covalent systems”. In: *Chemical science* 10.41 (2019), pp. 9424–9432.

[132] Johannes Klicpera et al. “Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules”. In: *arXiv preprint arXiv:2011.14115* (2020).

[133] Bing Huang and O. Anatole von Lilienfeld. “Quantum machine learning using atom-in-molecule-based fragments selected on the fly”. In: *Nature Chemistry* 12.10 (Sept. 2020), pp. 945–951. doi: 10.1038/s41557-020-0527-z. url: https://doi.org/10.1038/s41557-020-0527-z.

[134] Weitao Yang and Wilfried J Mortier. “The use of global and local molecular parameters for the analysis of the gas-phase basicity of amines”. In: *Journal of the American Chemical Society* 108.19 (1986), pp. 5708–5711.

[135] Gloria L Silva et al. “Experimental and computational investigation of unsymmetrical cyanine dyes: understanding torsionally responsive fluorogenic dyes”. In: *Journal of the American Chemical Society* 129.17 (2007), pp. 5710–5718.

[136] Kenneth Atz et al. “Δ-Quantum machine learning for medicinal chemistry”. In: *ChemRxiv* (2021). doi: 10.26434/chemrxiv-2021-fz6v7-v2.

[137] Anders S Christensen and O Anatole von Lilienfeld. “On the role of gradients for machine learning of molecular energies and forces”. In: *Machine Learning: Science and Technology* 1.4 (2020), p. 045018.

[138] Simon Batzner et al. “SE (3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials”. In: *arXiv preprint arXiv:2101.03164* (2021).

[139] Stefan Chmiela et al. “Machine learning of accurate energy-conserving molecular force fields”. In: *Science advances* 3.5 (2017), e1603015.

[140] Lori A. Burns et al. “The BioFragment Database (BFDb): An open-data platform for computational chemistry analysis of noncovalent interactions”. In: *The Journal of Chemical Physics* 147.16 (Oct. 2017), p. 161727. doi: 10.1063/1.5001028. url: https://doi.org/10.1063/1.5001028.

[141] Peter Bjørn Jørgensen and Arghya Bhowmik. “DeepDFT: Neural Message Passing Network for Accurate Charge Density Prediction”. In: *arXiv preprint arXiv:2011.03346* (2020).

[142] Susan Marqusee and Robert L Baldwin. “Helix stabilization by Glu-... Lys+ salt bridges in short peptides of de novo design”. In: *Proceedings of the National Academy of Sciences* 84.24 (1987), pp. 8898–8902.

[143] Justin S Smith et al. “Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning”. In: *Nature communications* 10.1 (2019), pp. 1–8.

[144] Peikun Zheng et al. “Artificial intelligence-enhanced quantum chemical method with broad applicability”. In: *Nature communications* 12.1 (2021), pp. 1–13.

[145] Stefan Grimme et al. “r2SCAN-3c: A “Swiss army knife” composite electronic-structure method”. In: *The Journal of Chemical Physics* 154.6 (2021), p. 064103.

[146] Brajesh Rai et al. “TorsionNet: A Deep Neural Network to Rapidly Predict Small Molecule Torsion Energy Profiles with the Accuracy of Quantum Mechanics”. In: (2020).

[147] Daniel GA Smith et al. “Revised damping parameters for the D3 dispersion correction to density functional theory”. In: *The journal of physical chemistry letters* 7.12 (2016), pp. 2197–2203.

[148] Larry A Curtiss et al. “Gaussian-2 theory for molecular energies of first-and second-row compounds”. In: *The Journal of chemical physics* 94.11 (1991), pp. 7221–7230.

[149] Armido Studer and Dennis P Curran. “The electron is a catalyst”. In: *Nature Chemistry* 6.9 (2014), pp. 765–773.

[150] Lars Goerigk et al. “A look at the density functional theory zoo with the advanced GMTKN55 database for general main group thermochemistry, kinetics and noncovalent interactions”. In: *Physical Chemistry Chemical Physics* 19.48 (2017), pp. 32184–32215.

[151] Christian Devereux et al. “Extending the Applicability of the ANI Deep Learning Molecular Potential to Sulfur and Halogens”. In: *Journal of Chemical Theory and Computation* 16.7 (2020), pp. 4192–4202.

[152] Ka Un Lao et al. “Accurate description of intermolecular interactions involving ions using symmetry-adapted perturbation theory”. In: *Journal of Chemical Theory and Computation* 11.6 (2015), pp. 2473–2486.

[153] Amir Karton, Shauli Daon, and Jan ML Martin. “W4-11: A high-confidence benchmark dataset for computational thermochemistry derived from first-principles W4 data”. In: *Chemical Physics Letters* 510.4-6 (2011), pp. 165–178.

[154] Bo Li et al. “Dispersion and Steric Effects on Enantio-/Diastereoselectivities in Synergistic Dual Transition-Metal Catalysis”. In: *Journal of the American Chemical Society* ().

[155] Wuyang Chen et al. “Automated synthetic-to-real generalization”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1746–1756.

[156] Justin Gilmer et al. “Neural message passing for Quantum chemistry”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017, pp. 1263–1272.

[157] Thom H Dunning and P Jeffrey Hay. “Gaussian basis sets for molecular calculations”. In: *Methods of electronic structure theory*. Springer, 1977, pp. 1–27.

[158] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[159] *CHEMBL database release 27*. May 2020. DOI: `10.6019/chembl.database.27`. URL: `https://doi.org/10.6019/chembl.database.27`.

[160] Justin S. Smith, Olexandr Isayev, and Adrian E. Roitberg. “ANI-1, A data set of 20 million calculated off-equilibrium conformations for organic molecules”. In: *Scientific Data* 4.1 (Dec. 2017). DOI: `10.1038/sdata.2017.193`. URL: `https://doi.org/10.1038/sdata.2017.193`.

[161] Patrick J. Ropp et al. “Dimorphite-DL: an open-source program for enumerating the ionization states of drug-like small molecules”. In: *Journal of Cheminformatics* 11.1 (Feb. 2019). DOI: `10.1186/s13321-019-0336-9`. URL: `https://doi.org/10.1186/s13321-019-0336-9`.

[162] A. Patrícia Bento et al. “An open source chemical structure curation pipeline using RDKit”. In: *Journal of Cheminformatics* 12.1 (Sept. 2020). DOI: `10.1186/s13321-020-00456-1`. URL: `https://doi.org/10.1186/s13321-020-00456-1`.

[163] Petr Jurečka et al. “Benchmark database of accurate (MP2 and CCSD(T) complete basis set limit) interaction energies of small model complexes, DNA base pairs, and amino acid pairs”. In: *Phys. Chem. Chem. Phys.* 8.17 (2006), pp. 1985–1993. DOI: `10.1039/b600027d`. URL: `https://doi.org/10.1039/b600027d`.

[164] Florian Weigend and Reinhart Ahlrichs. “Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy”. In: *Physical Chemistry Chemical Physics* 7.18 (2005), p. 3297. DOI: `10.1039/b508541a`. URL: `https://doi.org/10.1039/b508541a`.

[165] Florian Weigend. “Hartree–Fock exchange fitting basis sets for H to Rn”. In: *Journal of computational chemistry* 29.2 (2008), pp. 167–175.

[166] Alexander I Molev. “Gelfand-Tsetlin bases for classical Lie algebras”. In: *arXiv preprint math/0211289* (2002).

[167] Brian C Hall. *Quantum theory for mathematicians*. Vol. 267. Springer, 2013.

[168] WH Klink and T Ton-That. “Multiplicity, invariants, and tensor product decompositions of compact groups”. In: *Journal of Mathematical Physics* 37.12 (1996), pp. 6468–6485.

[169] William Fulton and Joe Harris. *Representation theory: a first course*. Vol. 129. Springer Science & Business Media, 2013.

[170] Arne Alex et al. “A numerical algorithm for the explicit calculation of SU (N) and SL (N, C) Clebsch–Gordan coefficients”. In: *Journal of Mathematical Physics* 52.2 (2011), p. 023507.

[171] S Gliske, W Klink, and T Ton-That. “Algorithms for computing U (N) Clebsch Gordan coefficients”. In: *Acta Applicandae Mathematicae* 95.1 (2007), pp. 51–72.

[172] Trygve Helgaker, Poul Jorgensen, and Jeppe Olsen. *Molecular electronic-structure theory*. John Wiley & Sons, 2014.

[173] Miguel A Blanco, Manuel Flórez, and Margarita Bermejo. “Evaluation of the rotation matrices in the basis of real spherical harmonics”. In: *Journal of Molecular Structure: THEOCHEM* 419.1-3 (1997), pp. 19–27.

[174] Pavel Izmailov et al. “Averaging weights leads to wider optima and better generalization”. In: *arXiv preprint arXiv:1803.05407* (2018).

[175] Stefan Chmiela et al. “Towards exact molecular dynamics simulations with machine-learned force fields”. In: *Nature communications* 9.1 (2018), pp. 1–10.

[176] William J Allen and Robert C Rizzo. “Implementation of the Hungarian algorithm to account for ligand symmetry and similarity in structure-based design”. In: *Journal of chemical information and modeling* 54.2 (2014), pp. 518–529.

[177] Xiang Gao et al. “TorchANI: A free and open source PyTorch-based deep learning implementation of the ANI neural network potentials”. In: *Journal of chemical information and modeling* 60.7 (2020), pp. 3408–3415.

[178] Prajit Ramachandran, Barret Zoph, and Quoc V Le. “Searching for activation functions”. In: *arXiv preprint arXiv:1710.05941* (2017).

[179] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[180] Ilya Loshchilov and Frank Hutter. “Sgdr: Stochastic gradient descent with warm restarts”. In: *arXiv preprint arXiv:1608.03983* (2016).

[181] Katherine Henzler-Wildman and Dorothee Kern. “Dynamic personalities of proteins”. In: *Nature* 450.7172 (2007), pp. 964–972.

[182] Nataliya Popovych et al. “Dynamically driven protein allostery”. In: *Nature structural & molecular biology* 13.9 (2006), pp. 831–838.

[183] Ruth Nussinov and Chung-Jung Tsai. “Allostery in disease and in drug discovery”. In: *Cell* 153.2 (2013), pp. 293–305.

[184] Alastair DG Lawson. “Antibody-enabled small-molecule drug discovery”. In: *Nature Reviews Drug Discovery* 11.7 (2012), pp. 519–525.

[185] Amanda R Moore et al. “RAS-targeted therapies: is the undruggable drugged?” In: *Nature Reviews Drug Discovery* 19.8 (2020), pp. 533–552.

[186] Christopher J Draper-Joyce et al. “Positive allosteric mechanisms of adenosine A1 receptor-mediated analgesia”. In: *Nature* 597.7877 (2021), pp. 571–576.

[187] David E Shaw et al. “Atomic-level characterization of the structural dynamics of proteins”. In: *Science* 330.6002 (2010), pp. 341–346.

[188] Yibing Shan et al. “How does a small molecule bind at a cryptic binding site?” In: *PLoS computational biology* 18.3 (2022), e1009817.

[189] Minkyung Baek et al. “Accurate prediction of protein structures and interactions using a three-track neural network”. In: *Science* 373.6557 (2021), pp. 871–876.

[190] Marcus Fischer et al. “Incorporation of protein flexibility and conformational energy penalties in docking screens to improve ligand discovery”. In: *Nature chemistry* 6.7 (2014), pp. 575–583.

[191] Jue Wang et al. “Scaffolding protein functional sites using deep learning”. In: *Science* 377.6604 (2022), pp. 387–394.

[192] William Sinko, Steffen Lindert, and J Andrew McCammon. “Accounting for receptor flexibility and enhanced sampling methods in computer-aided drug design”. In: *Chemical biology & drug design* 81.1 (2013), pp. 41–49.

[193] Noah Ollikainen, René M de Jong, and Tanja Kortemme. “Coupling protein side-chain and backbone flexibility improves the re-design of protein-ligand specificity”. In: *PLoS computational biology* 11.9 (2015), e1004335.

[194] Lim Heo and Michael Feig. “Multi-State Modeling of G-protein Coupled Receptors at Experimental Accuracy”. In: *Proteins: Structure, Function, and Bioinformatics* (2022).

[195] Yuqi Zhang et al. “Benchmarking Refined and Unrefined AlphaFold2 Structures for Hit Discovery”. In: (2022).

[196] Marta Amaral et al. “Protein conformational flexibility modulates kinetics and thermodynamics of drug binding”. In: *Nature communications* 8.1 (2017), pp. 1–14.

[197] Qianqian Zhao et al. “Enhanced Sampling Approach to the Induced-Fit Docking Problem in Protein–Ligand Binding: The Case of Mono-ADP-Ribosylation Hydrolase Inhibitors”. In: *Journal of chemical theory and computation* 17.12 (2021), pp. 7899–7911.

[198] Richard A Stein and Hassane S Mchaourab. “Modeling alternate conformations with alphafold2 via modification of the multiple sequence alignment”. In: *bioRxiv* (2021).

[199] Lucas SP Rudden, Mahdi Hijazi, and Patrick Barth. “Deep learning approaches for conformational flexibility and switching properties in protein design”. In: *Frontiers in Molecular Biosciences* (2022), p. 840.

[200] Jascha Sohl-Dickstein et al. “Deep unsupervised learning using nonequilibrium thermodynamics”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265.

[201] Yang Song et al. “Score-based generative modeling through stochastic differential equations”. In: *arXiv preprint arXiv:2011.13456* (2020).

[202] Ian W Davis and David Baker. “RosettaLigand docking with full ligand and receptor flexibility”. In: *Journal of molecular biology* 385.2 (2009), pp. 381–392.

[203] Tristan Bepler and Bonnie Berger. “Learning the protein language: Evolution, structure, and function”. In: *Cell systems* 12.6 (2021), pp. 654–669.

[204] Alexander Rives et al. “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118.

[205] Ahmed Elnaggar et al. “ProtTrans: towards cracking the language of lifes code through self-supervised deep learning and high performance computing”. In: *IEEE transactions on pattern analysis and machine intelligence* (2021).

[206] Chengxi Zang and Fei Wang. “MoFlow: an invertible flow model for generating molecular graphs”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 617–626.

[207] Tianfan Fu et al. “Differentiable scaffolding tree for molecular optimization”. In: *arXiv preprint arXiv:2109.10469* (2021).

[208] Ernest L Eliel and Samuel H Wilen. *Stereochemistry of organic compounds*. John Wiley & Sons, 1994.

[209] Attilio Meucci. “Review of statistical arbitrage, cointegration, and multivariate Ornstein-Uhlenbeck”. In: *Cointegration, and Multivariate Ornstein-Uhlenbeck (May 14, 2009)* (2009).

[210] Yang Song and Stefano Ermon. “Generative modeling by estimating gradients of the data distribution”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[211] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[212] Tero Karras et al. “Elucidating the Design Space of Diffusion-Based Generative Models”. In: *arXiv preprint arXiv:2206.00364* (2022).

[213] Weili Nie et al. “Diffusion Models for Adversarial Purification”. In: *arXiv preprint arXiv:2205.07460* (2022).

[214] Risi Imre Kondor and John Lafferty. “Diffusion kernels on graphs and other discrete structures”. In: *Proceedings of the 19th international conference on machine learning*. Vol. 2002. 2002, pp. 315–322.

[215] Petar Veličković et al. “Graph attention networks”. In: *arXiv preprint arXiv:1710.10903* (2017).

[216] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. “E (n) equivariant graph neural networks”. In: *arXiv preprint arXiv:2102.09844* (2021).

[217] Johannes Brandstetter et al. “Geometric and physical quantities improve e (3) equivariant message passing”. In: *arXiv preprint arXiv:2110.02905* (2021).

[218] Yunzhu Li et al. “Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids”. In: *arXiv preprint arXiv:1810.01566* (2018).

[219] Bowen Jing et al. “Learning from protein structure with geometric vector perceptrons”. In: *arXiv preprint arXiv:2009.01411* (2020).

[220] Tao Shen et al. “E2Efold-3D: End-to-End Deep Learning Method for accurate de novo RNA 3D Structure Prediction”. In: *arXiv preprint arXiv:2207.01586* (2022).

[221] Namrata Anand and Tudor Achim. “Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models”. In: *arXiv preprint arXiv:2205.15019* (2022).

[222] Hannes Stärk et al. “Equibind: Geometric deep learning for drug binding structure prediction”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 20503–20521.

[223] Yang Liu et al. “CB-Dock: a web server for cavity detection-guided protein–ligand blind docking”. In: *Acta Pharmacologica Sinica* 41.1 (2020), pp. 138–144.

[224] Renxiao Wang et al. “The PDBbind database: methodologies and updates”. In: *Journal of medicinal chemistry* 48.12 (2005), pp. 4111–4119.

[225] Anthony K Rappé et al. “UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations”. In: *Journal of the American chemical society* 114.25 (1992), pp. 10024–10035.

[226] Valerio Mariani et al. “lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests”. In: *Bioinformatics* 29.21 (2013), pp. 2722–2728.

[227] Xavier Robin et al. “Continuous Automated Model EvaluatiOn (CAMEO)—Perspectives on the future of fully automated evaluation of structure prediction methods”. In: *Proteins: Structure, Function, and Bioinformatics* 89.12 (2021), pp. 1977–1986.

[228] Yang Zhang and Jeffrey Skolnick. “TM-align: a protein structure alignment algorithm based on the TM-score”. In: *Nucleic acids research* 33.7 (2005), pp. 2302–2309.

[229] Artur Meller et al. “Predicting the locations of cryptic pockets from single protein structures using the PocketMiner graph neural network”. In: *bioRxiv* (2022).

[230] Robert B Best, Gerhard Hummer, and William A Eaton. “Native contacts determine protein folding mechanisms in atomistic simulations”. In: *Proceedings of the National Academy of Sciences* 110.44 (2013), pp. 17874–17879.

[231] Pat Vatiwutipong and Nattakorn Phewchean. “Alternative way to derive the distribution of the multivariate Ornstein–Uhlenbeck process”. In: *Advances in Difference Equations* 2019.1 (2019), pp. 1–7.

[232] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising diffusion implicit models”. In: *arXiv preprint arXiv:2010.02502* (2020).

[233] Jianyi Yang, Ambrish Roy, and Yang Zhang. “BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions”. In: *Nucleic acids research* 41.D1 (2012), pp. D1096–D1103.

[234] Simon Axelrod and Rafael Gomez-Bombarelli. “GEOM, energy-annotated molecular conformations for property prediction and molecular generation”. In: *Scientific Data* 9.1 (2022), pp. 1–14.

[235] Alexander G Donchev et al. “Quantum chemical benchmark databases of gold-standard dimer interaction energies”. In: *Scientific data* 8.1 (2021), pp. 1–9.

[236] Viki Kumar Prasad, Alberto Otero-de-La-Roza, and Gino A DiLabio. “PEPCONF, a diverse data set of peptide conformational energies”. In: *Scientific data* 6.1 (2019), pp. 1–9.

[237] Maho Nakata and Tomomi Shimazaki. “PubChemQC project: a large-scale first-principles electronic structure database for data-driven chemistry”. In: *Journal of chemical information and modeling* 57.6 (2017), pp. 1300–1308.

[238] Weihua Hu et al. “OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs”. In: *arXiv preprint arXiv:2103.09430* (2021).

[239] Miquel Duran-Frigola et al. “Extending the small-molecule similarity principle to all levels of biology with the Chemical Checker”. In: *Nature Biotechnology* 38.9 (2020), pp. 1087–1096.

[240] Nicola De Cao and Wilker Aziz. “The power spherical distribution”. In: *arXiv preprint arXiv:2006.04437* (2020).

[241] Helen M Berman et al. “The protein data bank”. In: *Nucleic acids research* 28.1 (2000), pp. 235–242.

[242] Andrea Scarpino, György G Ferenczy, and György M Keserű. “Comparative evaluation of covalent docking tools”. In: *Journal of Chemical Information and Modeling* 58.7 (2018), pp. 1441–1458.

[243] Gaoqi Weng et al. “Comprehensive evaluation of fourteen docking programs on protein–peptide complexes”. In: *Journal of chemical theory and computation* 16.6 (2020), pp. 3959–3969.

[244] Milot Mirdita et al. “ColabFold: making protein folding accessible to all”. In: *Nature Methods* (2022), pp. 1–4.

[245] Oleg Trott and Arthur J Olson. “AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading”. In: *Journal of computational chemistry* 31.2 (2010), pp. 455–461.

[246] Greg Landrum et al. “RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling”. In: *Greg Landrum* (2013).

[247] Benjamin Brown et al. “Introduction to the BioChemical Library (BCL): An application-based open-source toolkit for integrated cheminformatics and machine learning in computer-aided drug discovery”. In: *Frontiers in pharmacology* (2022), p. 341.

[248] Marco Biasini et al. “OpenStructure: an integrated software framework for computational structural biology”. In: *Acta Crystallographica Section D: Biological Crystallography* 69.5 (2013), pp. 701–709.

[249] Noel M O’Boyle et al. “Open Babel: An open chemical toolbox”. In: *Journal of cheminformatics* 3.1 (2011), pp. 1–14.