Generalized Implicit Neural Representations for Dynamic Molecular Surface Modeling

Fang Wu[1], **Bozhen Hu**[2], **Stan Z. Li**†[2]

[1] Computer Science Department, Stanford University
[2] School of Engineering, Westlake University
fangwu97@stanford.edu, {hubozhen,stan.zq.li}@westlake.edu.cn

Abstract

Molecular dynamics (MD) has long been the *de facto* choice for simulating intricate physical systems from first principles. Recent efforts utilize the implicit neural representation (INR) to directly learn surface point clouds' signed distance function (SDF) with promising outcomes. However, INR's temporal generalization to unexplored molecular systems remains limited, which poses a significant barrier to applying INR to a broader range of real-world scenarios. This study introduces MoE-DSR, an enhanced version of dynamic surface representations (DSR) that effectively integrates the mixture-of-experts (MoE) strategy. Specifically, the router employs a novel geometric surface cloud network to extract the structural information from the initial static protein conformation as the prior knowledge. Meanwhile, experts compromising a team of equivariant implicit neural networks (E-INNs), each responsible for distinct protein families, ensure precise SDF estimation across varied protein data landscapes. We showcase the ability of MoE-DSR to model dynamic protein surface shapes using ensembles from ATLAS, the largest available protein MD simulations database. Extensive experiments validate its effectiveness in analyzing complex molecular systems across continuous space and time domains.

Introduction

Proteins are essential macromolecules in living organisms, composed of chains of amino acids (Tang et al. 2024b). They assume specific 3D structures that enable them to perform critical biological functions, including catalyzing enzymatic reactions, facilitating molecular transport, and providing structural support to cells and tissues. In biological systems, proteins exhibit dynamic behavior, with amino acids continuously vibrating, rotating, and shifting in response to environmental changes (Zheng et al. 2023). This dynamic nature is crucial to proteins' biological roles, and molecular dynamics (MD) simulations are often employed to study these movements and their effects on function.

MD is a robust technique for simulating trajectories of complex atomic systems based on fundamental physical principles (Hollingsworth and Dror 2018). It adopts the density functional theory and calculates forces from electronic states by solving the Schrödinger equation (Car and Parrinello 1985). Unlike experimental methods such as X-ray crystallography or cryo-electron microscopy, offering average structural data, MD can capture the sequential behavior of molecules with atomic-level detail and high temporal resolution, enabling measuring various molecular regions' movement and structural fluctuations at equilibrium. In drug discovery, MD is pivotal for assessing the mobility and flexibility of crucial areas, facilitating observations of key functional processes such as ligand binding (Shan et al. 2011; Wu et al. 2023e, 2024d,b; Wu 2024; Wu et al. 2023c; Cui et al. 2023), conformational changes induced by voltage, protein folding (Lindorff-Larsen et al. 2011; Wu et al. 2024a; Lai et al. 2024; Tang et al. 2024a), protein folding (Lindorff-Larsen et al. 2011), and membrane transport (Suomivuori et al. 2017) However, its computational cost typically scales cubically with the number of electronic degrees of freedom, presenting challenges for simulating large systems or processes that occur on timescales longer than classical all-atom MD can achieve. Enhanced sampling (Bernardi et al. 2015) has been developed to capture longer-timescale events but introduces trade-offs between accuracy and efficiency. Besides, MD requires small integration time steps to accurately capture fast motions, such as hydrogen bond vibrations within a few femtoseconds. This necessity restricts MD's ability to achieve higher speeds. To address them, deep learning (DL) is used to approximate high-dimensional functions and integrate physical principles such as symmetry constraints (Chmiela et al. 2017; Wu et al. 2022b, 2024c, 2023a). It can optimize force fields (FF) or potential energy surfaces, significantly expediting FF-based methods compared to *ab initio* ones with extensive quantum mechanical calculations. Despite successes, they face limitations in scalability, accuracy, and reliability (Wang et al. 2020).

Instead of describing dynamic proteins as discrete particles, a recent line (Sun et al. 2024) explores modeling the dynamic shapes of proteins by implicit neural representations (INR) on molecular surfaces, which holds a more direct relevance to biomolecular interactions than buried structures (Leem et al. 2022; Lee et al. 2023). INR can encode complex geometric shapes via a continuous function without relying on explicit discrete representations (De Luigi et al. 2023), particularly effective for modeling dynamic protein shapes as they exhibit intricate and diverse conformations. Although great progress is made in pushing the frontier of INR targeting MD simulations of substantial proteins, cur-

rent approaches fall short in generalizing to unfamiliar protein families, posing a primary challenge in extending their application to real-world scenarios. Furthermore, they pay insufficient attention to equivariance, a crucial aspect of Euclidean geometry (Liao et al. 2022), rendering their outputs susceptible to any transformation of coordinate systems.

To overcome these drawbacks, we introduce MoE-DSR, an innovative approach that seamlessly integrates the mixture-of-experts (MoE) technology into the dynamic surface representation (DSR). MoE is a well-established architectural pattern in creating powerful large-scale language models (LLMs), enabling dramatic capacity improvements without parameter explosion (Shazeer et al. 2017; Fedus, Zoph, and Shazeer 2022). In MoE-DSR, we propose a novel geometric surface cloud network that acts as the router, making gating decisions based on the structural characteristics of the initial static protein conformation. Meanwhile, a team of experts composed of equivariant implicit neural networks (INNs) assesses dynamic changes in signed distance function (SDF) across continuous space and time domains. By adopting a divide-and-conquer strategy in MoE-DSR, each expert focuses on specific protein data subsets, enhancing the overall framework's adaptability to unseen scenarios of dynamic molecular systems. Additionally, we conquer the equivariance issue by leveraging irreducible representations founded on spherical harmonics. We verify the effectiveness of our MoE-DSR on ATLAS (Vander Meersche et al. 2024), the largest existing MD simulation database of proteins. Comprehensive results demonstrate that incorporating the MoE architecture and geometric symmetries significantly boosts INR's capability to comprehend protein dynamic changes and handle diverse protein distributions. Empirical analysis reveals that different experts specialize in learning SDFs of varied protein family classes.

Related Works

Molecular Surface Representations. DL, inspired by advances in language and vision, has recently leveraged the evolutionary information contained in protein sequences and structures (Wu et al. 2022c; Lin et al. 2022; Wu et al. 2023e, 2022a). Concurrently, interest grows in incorporating surfaces to improve protein representation learning, as the characteristics of protein surfaces influence the types and strengths of their interactions with other molecules. The molecular surface is defined based on van der Waals (vdW) radii and is often represented as meshes derived from SDFs. These surface attributes are pivotal in determining proteins' functional properties, particularly in interactions like ligand binding, enzymatic catalysis, and inter-molecular signal transduction. MaSIF (Leem et al. 2022) pioneered mesh-based DL to abstract internal protein folds and analyze protein interactions. Later studies (Sverrisson et al. 2021) reduced high pre-computation costs associated with featurization and showed competitiveness by modeling surfaces as point clouds. Other seminal works attempted to depict the protein surfaces by INRs for self-supervised pretraining (Lee et al. 2023; Wu and Li 2024) and dynamic structure modeling (Sun et al. 2024). Nonetheless, representations of dynamic molecular surfaces remain underexplored.

Deep Learning for Molecular Dynamics. DL is increasingly applied in molecular simulation, and approximation of molecular potential energy surfaces has become a well-established subfield (Jackson et al. 2023), such as DTNN (Schutt et al. 2017) and SchNet (Schutt et al. 2018). These approaches are primarily effective with small molecules but face challenges with large proteins due to their complexity and numerous atoms. Consequently, DL's influence has expanded beyond quantum chemistry and atomistic MD to encompass multiscale and coarse-grained (CG) modeling, including CGnet (Wang et al. 2020), DeePCG (Zhang et al. 2018), and ARCG (Durumeric et al. 2019). As rigorous CG simulation faces challenges in practicality and the demand for greater accuracy, some leverages normalizing flows (Kohler et al. 2023) and diffusion models (Arts et al. 2023) to create CG-FFs for dynamic simulation of small proteins. Another line such as DiffMD (Wu et al. 2023b) aims to predict simulation trajectories without relying on energy or forces. A recent study (Sun et al. 2024) uses surface representations for dynamic modeling of large proteins, but this method is sensitive to noise in protein dynamic trajectories and has very limited generalization.

Preliminary and Background

Proteins. A protein comprises a set of atoms $\mathcal{V}^a = \{v_i^a\}_{i=1}^{N_a}$ at the fine-grained level and residues $\mathcal{V}^R = \{v_i^R\}_{i=1}^{N_R}$ at the coarse-grained level. Each atom v_i^a has a chemical type $\mathbf{t}_i^a$ encoded as one-hot vectors and corresponding 3D coordinates $\mathbf{x}_i^a \in \mathbb{R}^3$. Each residue v_i^R has an amino acid type $\mathbf{t}_i^R$ and 3D coordinates $\mathbf{x}_i^R \in \mathbb{R}^3$ based on its α-Carbon position. $\{\mathbf{x}_{i,t}^a\}_{i=1}^{N_a}$ and $\{\mathbf{x}_{i,t}^R\}_{i=1}^{N_R}$ are denoted as $\mathcal{X}^a$ and $\mathcal{X}^R$, respectively . Then we rely on software like Pymol (DeLano et al. 2002) or MSMS to obtain its surface point cloud $\mathcal{X}^s = \{\mathbf{x}_i^s\}_{i=1}^M$ with unit normal vector data $\mathcal{N}^s = \{\mathbf{n}_i^s\}_{i=1}^M$. We denote the entire structural information of this protein as $\mathcal{P} = \left(\mathcal{V}^a, \mathcal{V}^R, \mathcal{X}^s, \mathcal{N}^s\right)$.

MD Trajectories. The trajectories of a molecular system are collected as $\{\mathcal{P}_t\}_{t=1}^T$, which spans T timesteps and effectively characterizes the equilibrium distribution of the system. For different durations, biologically inherent features $\{\mathbf{t}_i^a\}_{i=1}^{N_a}$ and $\{\mathbf{t}_i^R\}_{i=1}^{N_R}$ are invariant, while other geometric features $\mathcal{X}_t^a, \mathcal{X}_t^R, \mathcal{X}_t^s$ and $\mathcal{N}_t^s$ vary due to interatomic potentials and molecular mechanical force fields. Our objective is to forecast the future dynamic surface trajectories $\{\mathcal{X}_t^s\}_{t=1}^T$ given its initial static protein structure $\mathcal{P}_0$.

SDF Representations. Let $\Omega = [-1, 1]^3$ and $\tau = [-1, 1]$ be a normalized spatial domain and a normalized temporal domain, separately. We represent a 3D manifold embedded in Ω at time $t \in \tau$ as $\mathcal{M}_t$. Then, for any point $\mathbf{x} \in \Omega$ at time t, a SDF of this manifold $f_{\mathcal{M}_t} : \Omega \to \mathbb{R}$ is define as

$$f_{\mathcal{M}_t}(\mathbf{x}) = \begin{cases} \mathrm{d}\left(\mathbf{x}, \mathcal{M}_t\right) & \text{if } \mathbf{x} \text{ outside } \mathcal{M}_t \\ 0 & \text{if } \mathbf{x} \text{ belonging to } \mathcal{M}_t \\ -\mathrm{d}\left(\mathbf{x}, \mathcal{M}_t\right) & \text{if } \mathbf{x} \text{ inside } \mathcal{M}_t \end{cases},$$

where $\mathrm{d}\left(\mathbf{x}, \mathcal{M}_t\right) := \inf_{\mathbf{y} \in \mathcal{M}_t} \mathrm{d}(\mathbf{x}, \mathbf{y})$ computes the point-to-manifold distance and $\mathrm{d}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ takes the Eu-

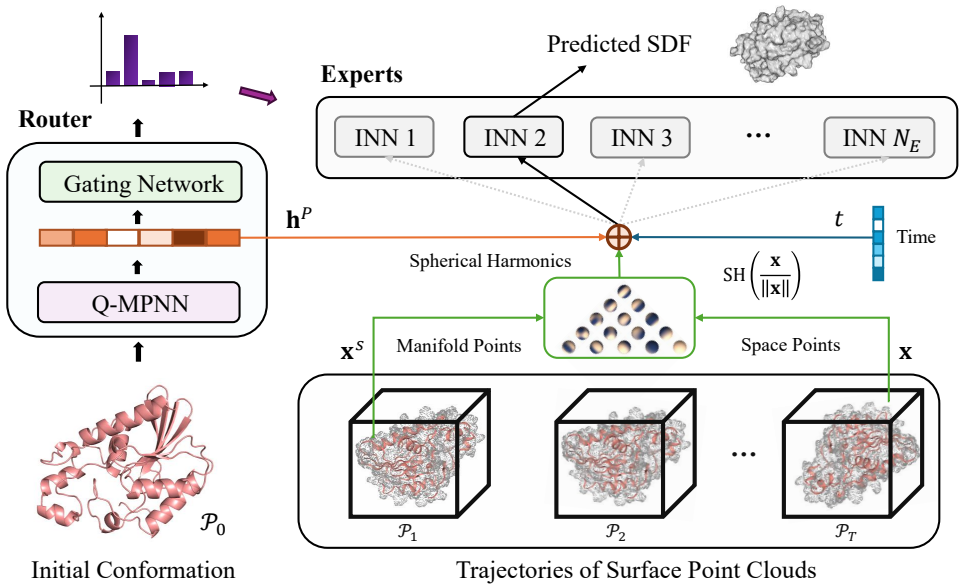

Figure 1: **The overall illustration of our MoE-DSR**. The initial static conformation is first forwarded into the quasi-geodesic message-passing neural network (Q-MPNN) to extract the structural information. Then the gating network determines which expert this molecular system should be sent to. Each expert is an equivariant implicit neural network (INN), which forecasts the SDF value of any point across the continuous space and time domains. Its input consists of irreducible representations of 3D coordinates, the timestamp, and the conformational feature from Q-MPNN.

lidean norm. The zero-level set at time t is a collection of points that satisfy $f_{\mathcal{M}_t}(\cdot) = 0$. If we define the manifold of a protein surface as $\mathcal{M}_t$, then its zero-level set is $\mathcal{S}_t = \{\mathbf{x} \mid f_{\mathcal{M}_t}(\mathbf{x}) = 0, \mathbf{x} \in \Omega\}$. Accordingly, $\mathcal{X}_t^s$ is a proper subset of $\mathcal{S}_t$ and can be used to estimate $\mathcal{M}_t$.

Irreducible Representations. A group representation defines transformation matrics $\mathbf{D}_{\mathcal{X}}(g)$ of group elements g that act on a vector space $\mathcal{X}$. Any group representation of $SO(3)$ can be decomposed to a concatenation of provably smallest transformation matrics called irreducible representations (irreps). For $g \in SO(3)$, there exist $(2l+1)$-by-$(2l+1)$ irreps matrices $\mathbf{D}^{(l)}(g)$ called Wigner-D matrices acting on $(2l+1)$-dimensional vector spaces, where degree $l \in \mathbb{Z}^+$. Vectors transformed by $\mathbf{D}^{(l)}(g)$ are type-l vectors, with scalars and Euclidean vectors being type-0 and type-1 vectors. Then, we concatenate multiple type-l vectors to form the $SE(3)$-equivariant irreps feature $\mathbf{h}^{\text{irreps}}$ with N_l type-l vectors, where $0 \leq l \leq l_{\max}$ and N_l is the number of channels for type-l vectors. Irreps features $\mathbf{h}^{\text{irreps}}$ is indexed by channel c, degree l, and order m as $\mathbf{h}_{c,m}^{(l)}$.

Spherical Harmonics and Tensor Product. Spherical harmonics (SH) $Y_m^{(l)}$ project any $\vec{r}$ from $\mathbb{R}^3$ to any type-l vectors $\mathbf{h}^{(l)}$ as $\mathbf{h}^{(l)} = Y^{(l)}\left(\frac{\vec{r}}{\|\vec{r}\|}\right)$. SH is $E(3)$-equivariant with $\mathbf{D}^{(l)}(g)\mathbf{h}^{(l)} = Y^{(l)}\left(\frac{\mathbf{D}^{(l)}(g)\vec{r}}{\|\mathbf{D}^{(l)}(g)\vec{r}\|}\right)$. Then equivariant operations such as tensor product (TP) can propagate SH to other irreps features. TP denoted as $\otimes$ uses Clebsch-Gordan coefficients to combine type-l_1 vector $\mathbf{h}^{(l_1)}$ and type-l_2 vector $\mathbf{h}^{(l_2)}$ and produces type-l_3 vector $\mathbf{h}^{(l_3)}$:

$$\mathbf{h}_{m_3}^{(l_3)} = \left(\mathbf{h}^{(l_1)} \otimes \mathbf{h}^{(l_2)}\right)_{m_3} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C_{(l_1,m_1)(l_2,m_2)}^{(l_3,m_3)} \mathbf{h}_{m_1}^{(l_1)} \mathbf{h}_{m_2}^{(l_2)}, \tag{1}$$

where m_1 denotes order and refers to the m_1-th element of $\mathbf{h}^{(l_1)}$. Clebsch-Gordan coefficients $C_{(l_1,m_1)(l_2,m_2)}^{(l_3,m_3)} \neq 0$ only when $|l_1 - l_2| \leq l_3 \leq | l_1 + l_2 |$ and thus restrict output vectors to be of certain types. For efficiency, vectors with $l > l_{\max}$ are discarded, where $l_{\max}$ is a hyper-parameter. Each distinct non-trivial combination of $l_1 \otimes l_2 \rightarrow l_3$ is called a path and is independently equivariant. Learnable weights can be assigned to each path in TP, similar to linear layers.

Method

Model Overview

The Mixture-of-Experts (MoE) layer constitutes a group of N_E expert networks $\{E_i(\cdot)\}_{i=1}^{N_E}$ and a router $G(\cdot)$ that selects which learners to perform computations (see Fig. 1). On the one hand, each expert $E_i(\cdot)$ is a neural network that can accurately estimate the SDF to a surface $\mathcal{M}_t$ described by the point cloud $\mathcal{X}_t^s$ and normals $\mathcal{N}_t^s$ at time t. On the other hand, the router $G(\cdot)$ makes the gating decision based on the raw protein structure $\mathcal{P}_0$, which is the only signal to identify the input molecular system. The output of the entire MoE module is therefore written as $\sum_{i=1}^{N_E} G(\mathcal{P}_0)_i E_i(\cdot)$.

Router Architecture

Proteins are intricate macromolecules with multi-faceted characteristics and have several modalities, including but not limited to amino acid sequences, 3D structures, and molecular surfaces (Wu et al. 2022d, 2023d). To align with different paradigms, the router $G(\cdot)$ is designed as follows.

Geometric Surface Cloud Network. First, we borrow ideas from dMaSIF (Strokach et al. 2020) and employ a simple geometric aggregation network (GeoAN) to attain the chemical features of each point $\mathbf{h}_i^s \in \mathbb{R}^{\phi_h}$ in an end-to-end manner. Particularly, for each surface point $\mathbf{x}_i^s$, we find the ζ nearest residues and their related residue types $\left\{\mathbf{x}_i^R, \mathbf{t}_i^R\right\}_{i=1}^{\zeta}$. Then we use an embedding layer to map $\mathbf{t}_i^R$ to its embedding vector $\mathbf{e}_i^R \in \mathbb{R}^{\phi_R}$ and apply a multi-layer perceptron (MLP) to the concatenated one $\mathbf{e}_i^R \oplus 1/||\mathbf{x}_i^s - \mathbf{x}_i^R||^2 \in \mathbb{R}^{\phi_R+1}$. Lastly, we perform an average pooling over these transformed vectors (*i.e.*, $i = 1, ..., \zeta$) and append a second MLP to compute $\mathbf{h}_i^s$.

Subsequently, we propose a novel equivariant quasi-geodesic message-passing neural network (Q-MPNN) to extract effective surface representations. Given target node i and source node j, we begin with the approximation of the geodesic distance between these two surface points $\mathbf{x}_i^s$ and $\mathbf{x}_j^s$ with their unit normals as:

$$\mathrm{d}_{ij} = \left\|\mathbf{x}_i^s - \mathbf{x}_j^s\right\| \cdot \left(2 - \left\langle \mathbf{n}_i^s, \mathbf{n}_j^s \right\rangle\right). \tag{2}$$

Here, the surface point normals $\mathcal{N}^s$ are smoothed by a Gaussian kernel with $\sigma_{\mathbf{n}}$ (Strokach et al. 2020). Mathematically, $\mathbf{n}_i^s \leftarrow \text{Normalize}\left(\sum_{j=1}^{M} \exp\left(-\left\|\mathbf{x}_i^s - \mathbf{x}_j^s\right\|^2 / 2\sigma_{\mathbf{n}}^2\right) \mathbf{n}_j^s\right)$. Then at the ν-th layer, we acquire the initial message $\mathbf{h}_{ij}^{\nu}$ by immediately combining their features as $\mathbf{h}_{ij}^{\nu} = \text{MLP}\left(\mathbf{h}_i^{s,\nu}\right) + \text{MLP}\left(\mathbf{h}_j^{s,\nu}\right)$. $\mathbf{h}_{ij}^{\nu}$ is further passed to a depth-wise TP (DTP) layer and a linear layer to consider geometric information including relative position and geodesic distance in different type-l vectors in irreps features:

$$\mathbf{h}_{ij}^{\nu'} = \mathbf{h}_{ij}^{\nu} \otimes_{w\left(\mathrm{d}_{ij}\right)}^{\text{DTP}} \text{SH}\left(\mathbf{x}_i^s - \mathbf{x}_j^s\right), \mathbf{e}_{ij}^{\nu} = \text{Linear}\left(\mathbf{h}_{ij}^{\nu'}\right), \tag{3}$$

where $\mathbf{h}_{ij}^{\nu'}$ is the DTP's result of $\mathbf{h}_{ij}^{\nu}$ and SH embeddings of relative direction $\mathbf{x}_i^s - \mathbf{x}_j^s$ with weights parameterized by the geodesic distance d_{ij} (Liao et al. 2022). Specifically, TP in Equ. 1 can be generalized to irreps features and include multiple channels of vectors of different types through iterating over all paths, and weights $w(\cdot)$ are indexed by $(c_1, l_1, c_2, l_2, c_3, l_3)$, where (c_1, l_1, c_2, l_2) and (c_3, l_3) correspond to input and output irreps features, respectively. DTP improves this vanilla TP's efficiency by forcing one type-l vector in output irreps features to only depend on one type-l' vector in input irreps features. This one-to-one channel dependency significantly reduces the weight numbers and thus memory complexity. The subscript $w(\cdot)$ in DTP is a learnable weight conditioned on d_{ij}. To reflect subtle changes in d_{ij}, we represent geodesic distances with the Gaussian radial basis as $\mathbf{r}_{ij,k} = \exp\left(-\gamma\left(\|\mathrm{d}_{ij} - \mu_k\|^2\right)\right)$, which is located at different centers $0\text{Å} \leq \mu_k \leq 30\text{Å}$ every $\frac{30}{\phi_{\mathrm{P}}}\text{Å}$ with $\gamma = 10\text{Å}$ (Schutt et al. 2018). Then these non-parametric radial basis $\mathbf{r}_{ij,k}$ are transformed to a learnable radial function with a two-layer MLP to generate weights for our DTP layers. In addition, Linear$(\cdot)$ is a generalized linear layer on irreps features, where separate linear operations with removed bias terms are applied to each group of type-l vectors for non-scalar features with $l > 0$ for not breaking equivariance (Liao et al. 2022).

In a later stage, a vertex update function $U(\cdot)$ (Gilmer et al. 2017) incorporates messages from neighboring points and updates the vector signal for the local reference surface point i as:

$$\mathbf{h}_i^{s,\nu+1} = U\left(\mathbf{h}_i^{s,\nu}, \sum_{j \in \mathcal{N}(i)} \mathbf{e}_{ij}^{\nu}\right), \tag{4}$$

where $\mathcal{N}(i)$ is the i's geodesic neighborhood set determined by the filter window size $\sigma_{\mathrm{d}} \in \left[9\text{Å}, 12\text{Å}\right]$. We stack L_1 layers of this Q-MPNN as the surface feature extractor. At its last layer, we perform a max-pooling operation as the readout function to aggregate features of $\mathcal{P}_0$ as $\mathbf{h}^{\mathrm{P}} = \text{MLP}\left(\text{Pool}\left(\left\{\mathbf{h}_i^{s,L_1}\right\}_{i=1}^{M}\right)\right) \in \mathbb{R}^{\phi_{\mathrm{P}}}$. Notably, the number of surface points M is orders of magnitude larger than that of atoms in small molecules or proteins, which limits the use of more complicated but computationally expensive architectures for our surface cloud network such as the attention mechanism (Vaswani et al. 2017).

Gating Network. We employ a sparsely-gated MoE scheme (Shazeer et al. 2017) to acquire routing decision. The gating network with sparsity and tunable Gaussian noise is defined as:

$$\begin{aligned} G(\mathcal{P}_0) &= \text{Softmax}(\text{TopK}(\mathbf{W}_g \mathbf{h}^{\mathrm{P}} + \sigma_{\text{noise}}, N_K)), \\ \sigma_{\text{noise}} &= \text{StandardNormal}(\cdot) \cdot \text{Softplus}(\mathbf{W}_{\text{noise}} \mathbf{h}^{\mathrm{P}}), \end{aligned} \tag{5}$$

where $\mathbf{W}_g \in \mathbb{R}^{N_E \times \phi_{\mathrm{P}}}$ is the first trainable weight. TopK$(\cdot)$ is a function that keeps the top-N_K entries of a vector the same, but sets all other entries to $-\infty$. which serves to save computation. The noise term σ_{noise} helps with load balancing, and the amount of noise per component is controlled by a second trainable weight matrix $\mathbf{W}_{\text{noise}} \in \mathbb{R}^{N_E \times \phi_{\mathrm{P}}}$. Here, the choice of N_K is a hyperparameter whose value is chosen according to application, and typically, $N_K = 1, 2$ (Fedus, Zoph, and Shazeer 2022).

Experts for Dynamic Molecular Surface Modeling

SDF Learners. The representation of the protein surface takes the form of a 3D shape. To leverage the implicit function form of SDF, a common approach involves partitioning the space into a grid and then calculating the SDF value for each grid point. However, for irregular objects, accurately computing SDF values can be challenging, leading to the use of approximation algorithms like the 8SSEDT algorithm (Ye 1988), but this method has two notable limitations. Firstly, protein surfaces are intricate and rugged, making estimation errors more pronounced during modeling. Secondly, SDF calculation for dynamic models is time-linear and resolution-cubically dependent, resulting in high computational costs for pre-computed SDFs.

Motivated by (Sun et al. 2024), we propose an alternative approach where we directly learn SDF from raw point

clouds $(\mathcal{X}^s, \mathcal{N}^s)$ instead of relying on pre-computed SDFs for supervised learning. This approach eliminates the need for pre-computing SDF, substantially reducing computational expenses and enhancing efficiency. In this framework, each expert network is a multi-layer perceptron (MLP) with parameter θ that forecasts the SDF value of any point $\mathbf{x}$ in the spatial domain Ω, denoted as $E(\mathbf{x}, t, \mathbf{h}^{\mathrm{P}}, \mathbf{z}; \theta) : \mathbb{R}^{4+\phi_P+\phi_z} \to \mathbb{R}$. It takes input elements of spatial coordinate $\mathbf{x} \in \Omega$, temporal indicator $t \in \tau$, structural information of the initial conformation $\mathbf{h}^{\mathrm{P}}$, and latent code $\mathbf{z} \in \mathbb{R}^{\phi_z}$, which is added to specify different protein types.

Equivariant Implicit Neural Representations. Notably, each learner $E_i(\cdot)$ is a naive MLP that receives and maps the 3-dimension $\mathbf{x}$ directly to the succeeding neurons, ignoring the symmetry property of 3D space. To address this issue, we rely on SH to achieve irreps representations and reformulate the learner input as $E\left(\|\mathbf{x}\| \cdot \mathrm{SH}\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right), t, \mathbf{h}^{\mathrm{P}}, \mathbf{z}; \theta\right) : \mathbb{R}^{2+2N_{\mathrm{SH}}+\phi_P+\phi_z} \to \mathbb{R}$ with a predefined maximum degree N_{SH}. It is worth noting that $\mathrm{SH}(\cdot)$ are $E(3)$-equivariant, and $\|\mathbf{x}\|$ is invariant to rotation and inversion. As a consequence, our INR is $O(3)$-equivariant.

Training Loss

The overall training loss consists of two parts. First is the SDF sampling loss $\mathcal{L}_{\mathrm{SDF}}$, which constrains experts $\{E_i\}_{i=1}^{N_E}$ to accurately estimate SDF to any given molecular surface. Second is the learner load balancing constraint $\mathcal{L}_{\mathrm{MoE}}$.

SDF Sampling Loss. Consider a protein surface point cloud $(\mathcal{X}_t^s, \mathcal{N}_t^s)$ representing the surface manifold $\mathcal{M}t$ at time t. We aim to determine the optimal parameters θ for each learner $E(\cdot)$ to effectively predict the signed distance function (SDF) to this manifold $\mathcal{M}_t$. To achieve this objective, the expression for the loss function is (Gropp et al. 2020; Sun et al. 2024):

$$\begin{aligned}\mathcal{L}_{\mathrm{SDF}} &= \mathcal{L}_{\mathrm{manifold}} \\ &+ \lambda_{\mathrm{SDF}} \mathbb{E}_{\mathbf{x}\in\Omega} \left(\|\nabla_{\mathbf{x}} E(\Psi(\mathbf{x}), t, \mathbf{h}^{\mathrm{P}}, \mathbf{z}; \theta)\| - 1\right)^2 + \lambda_{\mathrm{Z}} \|\mathbf{z}\|,\end{aligned} \tag{6}$$

$$\begin{aligned}\mathcal{L}_{\mathrm{manifold}} =& \mathbb{E}_{(\mathbf{x},\mathbf{n})\in\left(\mathcal{X}_t^s, \mathcal{N}_t^s\right)} \big(|E(\Psi(\mathbf{x}), t, \mathbf{h}^{\mathrm{P}}, \mathbf{z}; \theta)| \\ &+ \tau \left\|\nabla_{\mathbf{x}} E(\Psi(\mathbf{x}), t, \mathbf{h}^{\mathrm{P}}, \mathbf{z}; \theta) - \mathbf{n}\right\| \big),\end{aligned} \tag{7}$$

where λ_{SDF}, λ_{Z}, and τ are hyperparameters to control the importance of different loss terms. $\Psi(\boldsymbol{x}) = \|\mathbf{x}\| \cdot \mathrm{SH}\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right)$. $\mathcal{L}_{\mathrm{manifold}}$ promotes the vanishing of $E(\cdot)$ on $\mathcal{X}_t^s$ and drives $\nabla_{\mathbf{x}} E(\cdot)$ to the given normals $\mathcal{N}_t^s$. The first term in the summation part of $\mathcal{L}_{\mathrm{manifold}}$ indicates that the SDF value of those surface points $\mathcal{X}_t^s$ should be as small as possible. The next term in $\mathcal{L}_{\mathrm{manifold}}$ represents the loss between the predicted point normal vector and the ground truth. The second item in $\mathcal{L}_{\mathrm{SDF}}$ is called the Eikonal term. It promotes the gradients $\nabla_{\mathbf{x}} E(\cdot)$ to possess a 2-norm of unity, *i.e.*, $\|\nabla_{\mathbf{x}} E(\cdot)\| = 1$, which is the gradient property of SDFs. In particular, this characteristic also accords to the fact that surface normals $\mathcal{N}_t^s$ are unit vectors, *i.e.*, $\|\mathbf{n}_i^s\| = 1$. The last term in $\mathcal{L}_{\mathrm{SDF}}$ imposes a restraint on the learnable latent code $\mathbf{z}$, which illustrates different protein types.

Remarkably, the global minimum of $\mathcal{L}_{\mathrm{SDF}}$ will be the solution of the Eikonal partial differential equation, *i.e.*, $\|\nabla_{\mathbf{x}} E(\cdot)\| = 1$. This solution is also an SDF, where $E(\cdot)$ approaches 0 on $\mathcal{X}_t^s$ with gradients $\mathcal{N}_t^s$. Moreover, it is infeasible to iterate all space points in Ω to calculate the expectation $\mathbb{E}_{\mathbf{x}\in\Omega}(\cdot)$. As a remedy, we follow (Gropp et al. 2020; Sun et al. 2024) adopt a substitute probability distribution $\mathcal{D}$ in $\mathbb{R}^3$. To be specific, we determine $\mathcal{D}$ by taking the average of a uniform distribution and a sum of Gaussians that are centered at $\mathcal{X}_t^s$ with a standard deviation equal to the distance to the $N_{\mathcal{D}}$-th nearest neighbor, where $N_{\mathcal{D}} = 50$.

Load Balancing Loss. Besides, the vanilla MoE has issues with load balancing: some experts are consulted often, while others rarely or not at all. To encourage the gate to select each expert with equal frequency (proper load balancing) within each batch, we utilize an auxiliary loss (Fedus, Zoph, and Shazeer 2022) to simplify the separate load-balancing and importance-weighting losses (Shazeer et al. 2017). Concretely, for a given batch of queries $\{\mathcal{P}_{0,i}\}_{i=1}^{N_B}$, where N_B is the batch size. The auxiliary loss is designed as the scaled dot-product between vectors Γ and η:

$$\begin{aligned}\mathcal{L}_{\mathrm{MoE}} &= N_E \sum_{i=1}^{N_E} \Gamma_i \cdot \eta_i, \quad \eta_i = \frac{1}{N_B} \sum_{j=1}^{N_B} G\left(\mathcal{P}_{0,j}\right)_i, \\ \Gamma_i &= \frac{1}{N_B} \sum_{j=1}^{N_B} \mathbb{1}\{\arg\max G\left(\mathcal{P}_{0,j}\right) = i\},\end{aligned} \tag{8}$$

where Γ_i is the fraction of molecular systems dispatched to learner i, and η_i is the fraction of the router probability allocated for learner i. This equation encourages uniform routing since it is minimized under a uniform distribution. Remarkably, this objective is differentiable as η is differentiable in spite of Γ. The final loss of our MoE-DSR is thus a weighted sum of $\mathcal{L}_{\mathrm{SDF}}$ and $\mathcal{L}_{\mathrm{MoE}}$ with different multiplicative coefficients λ_1 and $\lambda_2 = 1e-2$, respectively.

Experiments

Setup

Dataset. To comprehensively demonstrate and assess the ability of our method, we train MoE-DSR on ATLAS (Vander Meersche et al. 2024), the largest up-to-date dataset of all-atom MD simulations for single-chain proteins. ATLAS consists of 1,390 proteins chosen for structural diversity by ECOD domain classification (Schaeffer et al. 2017). To formulate a more challenging circumstance for generalization evaluation, the **test split** includes proteins having no more than 40% sequence identity (SI) with other proteins in ATLAS. For data preprocessing, each ensemble has 30,000 timesteps and we save the entire trajectory by every 300 timesteps, which leads to 100 snapshots for each data point. We use MMseqs2 to compute the sequence identity of sequences in ATLAS and split the train/val/test datasets. The **training split** contains monomers not involved during the curation of the test split. Then selected test data points are divided randomly into the validation and final test sets with a ratio of 1:1. Using this cutoff, we obtain train/val/test splits of 1,290/50/50 ensembles.

	DSR			MoE-DSR		
	IoU↑	CHAMFER_DIST↓	NC↑	IoU↑	CHAMFER_DIST↓	NC↑
6irx_A	0.4891±0.0783	0.0115±0.0092	0.5401±0.0422	0.7289±0.0361	0.0033±0.0026	0.7182±0.0247
1tg7_A	0.4523±0.0828	0.0139±0.0095	0.5085±0.0689	0.6812±0.0420	0.0042±0.0038	0.6352±0.0339
4ayg_B	0.5169±0.0783	0.0146±0.0091	0.5233±0.0548	0.7237±0.0454	0.0041±0.0020	0.6958±0.0372
5jbg_A	0.6071±0.0484	0.0054±0.0038	0.6798±0.0188	0.8309±0.0162	0.0001±0.0001	0.8759±0.0045
7p41_D	0.4631±0.1259	0.0092±0.0076	0.4989±0.0901	0.6722±0.0661	0.0045±0.0017	0.6407±0.0432
1gte_D	0.4883±0.1350	0.0132±0.0099	0.5073±0.0485	0.7121±0.0336	0.0035±0.0016	0.7115±0.0173
2wsi_A	0.5502±0.1067	0.0089±0.0065	0.5436±0.0647	0.7122±0.0304	0.0035±0.0011	0.7114±0.0142
2inc_A	0.5388±0.1067	0.0132±0.0089	0.5988±0.0703	0.7237±0.0428	0.0041±0.0013	0.6958±0.0198
1bkp_A	0.5804±0.0427	0.0085±0.0062	0.5608±0.0266	0.7024±0.0279	0.0039±0.0010	0.6677±0.0256
2v4b_B	0.5274±0.0853	0.0078±0.0080	0.5553±0.0408	0.7053± 0.0605	0.0046±0.0015	0.6853±0.0208
All	0.5396±0.0725	0.0099±0.0068	0.5788±0.0532	**0.7028±0.0390**	**0.0039±0.0019**	**0.6723±0.0185**

Table 1: Evaluation on the surface reconstruction across time.

Symbol	Description	Range
Router		
ϕ_s	The dimension of input chemical point features.	[16, 32, 128]
ϕ_R	The dimension of residue embedding vectors.	[32]
ϕ_{P}	The dimension of protein-level features.	[32]
ζ	The number of nearest residues in GeoAN.	[8, 16]
$\sigma_{\mathbf{n}}$	The Gaussian kernel size.	[9Å, 12Å]
γ	Weight in the Gaussian radial basis.	[10Å]
l_{max}	The largest type of SH vectors $Y_{(\cdot)}$ in the DTP operation.	[1, 2]
N_K	The number of selected experts.	[2]
Experts		
N_E	The number of expert networks	[4, 8, 16]
ϕ_{SH}	The dimension of input SH vectors.	[1, 3, 9, 16]
ϕ_{z}	The dimension of the latent code vector.	[16, 64, 192]
Training		
N_B	The input batch size.	[4, 8, 16, 32]
λ_{SDF}	Weight of the SDF gradient property loss.	[0.1, 1.0]
λ_{Z}	Weight of the latent code loss.	[0.001, 0.1]
λ_1	Weight of $\mathcal{L}_{\text{SDF}}$.	[1.0, 10.0]
α_2	Weight of $\mathcal{L}_{\text{MoE}}$.	[0.01]

Table 2: The hyperparameter choices.

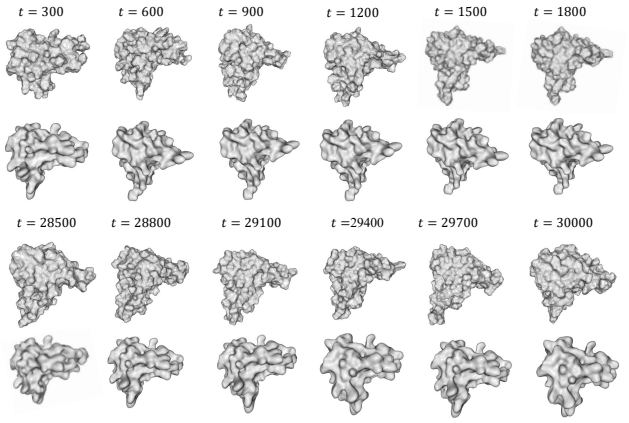

Figure 2: The first and second rows are the ground truth and the predicted surface trajectories of *1bgf_A*, respectively.

Implementation Details. All experiments are implemented in a data-parallel mode on 4 A100 GPUs, each with a memory of 80GB. Following (Sun et al. 2024), we adopt the Softplus (*i.e.*, $\Upsilon(x) = \frac{1}{\beta}\ln 1 + \exp\beta x$) as the activation function for the experts with $\beta = 100$. The gradient of leaners $\nabla_{\mathbf{x}}E(\cdot)$ is calculated by PyTorch Autograd. We repeat three trials and report their mean and standard deviation. we utilize MLPs as the experts in our MoE-DSR. Each MLP has the same architecture with 8 layers and 512 hidden units as well as a single skip connection from the input to the middle layer. The initial latent code vector $\mathbf{z}$ is sampled from a normal distribution $\mathcal{N}(0, 1)$. For baseline implementation, DSR (Sun et al. 2024) was reproduced using its official GitHub website at https://github.com/Sundw-818/DSR.

Hyperparameters. We ran a random hyperparameter search on the choices listed in Tab. 2 and selected the best combinations of hyperparameters for different datasets based on their corresponding validation performance.

Level set extraction. We extract the zero (or any other) level set of a trained model $f(x, t; \theta)$ using the Marching Cubes algorithm (Lorensen and Cline 1998) implemented in the Python scikit-image package (Van der Walt et al. 2014), which can use large-size grids to achieve any high resolution

Evaluation Metrics

Following (Sun et al. 2024), we use three evaluation metrics commonly used in 3D modeling to evaluate the similarity between two 3D shapes from three aspects: volume, distance, and normal vectors, which are Volumetric Intersection over Union (IoU), Chamfer distance (CD), and Normal Consistency (NC). These three metrics are all normalized to a range of $0 - 1$, providing a comprehensive evaluation of the model's performance from different perspectives. **IoU** compares the reconstructed volume with the ground truth shape (higher is better). For two arbitrary shapes $A, B \subseteq \mathbb{S} \in \mathbb{R}^n$ is attained by $\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$. **CD** is a standard metric to evaluate the distance between two point sets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathbb{R}^n$ (lower is better) as $\text{d}_{\text{C}}(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{2}\left(\text{d}_{\overrightarrow{\text{C}}}(\mathcal{X}_1, \mathcal{X}_2) + \text{d}_{\text{C}}(\mathcal{X}_2, \mathcal{X}_1)\right)$, where $\text{d}_{\overrightarrow{\text{C}}}(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{|\mathcal{X}_1|}\sum_{\boldsymbol{x}_1 \in \mathcal{X}_1} \min_{\boldsymbol{x}_2 \in \mathcal{X}_2} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$. **NC** evaluates estimated surface normals (higher is better). Normal consistency between two normalized unit vectors n_i and n_j is defined as the dot product between the two vectors. For evaluating the surface normals, given the object surface points and normal vectors: $X_{\text{pred}} = \{(\boldsymbol{x}_i, \overrightarrow{n_i})\}$, and the ground truth surface points and normal vectors: $X_{gt} = \left\{\left(\boldsymbol{y}_j, \overrightarrow{m_j}\right)\right\}$, the surface normal consis-

Expert ID	Protein Syetsms	SI
1	[1k5n_A, 1fx4_A, 4byz_A, 3it4_B, ...]	42.05%
2	[3vor_A, 1r6w_A, 1vaj_A, 6rwt_A, ...]	41.30%
3	[2o7o_A, 2xol_B, 3rlk_A, 3st1_A, ...]	43.75%
4	[3frr_A, 3ezu_A, 3k5j_A, 4lpq_A, 2h1t_B, ...]	43.45%

Table 3: Notable examples of specialization in INN experts.

tency between X_{pred} and X_{gt}, denoted as Γ, is defined as: $\Gamma\left(X_{gt}, X_{\text{pred}}\right) = \frac{1}{|X_{gt}|}\sum_{j\in|X_{gt}|}\left|\overrightarrow{n_j}\cdot\vec{m}_{\theta\left(\boldsymbol{y}_j, X_{\text{pred}}\right)}\right|$, where $\theta\left(\boldsymbol{y}_j, X_{\text{pred}} := \{(\boldsymbol{x}_i, \overrightarrow{n_i})\}\right) = \underset{i\in|X_{\text{pred}}|}{\arg\min}\left\|\boldsymbol{y}_j - \boldsymbol{x}_i\right\|_2^2$.

Quantitative Results

We quantitatively evaluate our model's generalization capacities to forecast the moving surface trajectories of single-chain protein systems. The results are put in Tab. 1 with a portion of test samples and the overall performance. It can be observed that MoE-DSR outperforms the vanilla DSR by a large margin. Specifically, it brings significant improvements of 30.24%, 39.40%, and 16.15% in IoU, CD, and NC, respectively. Moreover, we also compute the performance of naive extension, where the initial conformation $\mathcal{P}_0$ is used as substitute predictions for all following timesteps, resulting in an IoU, CD, and NC of 0.3375, 0.0298, and 0.3694, separately. These statistics demonstrate that MoE-DSR is not acting as a trivial task, and verify that our incorporation of the MoE framework and the geometric symmetry successfully enhances INR's generalization to predict dynamic surfaces of unfamiliar protein systems. In addition, we also visualize the zero-level set of the SDF value predicted by MoE-DSR for a randomly selected protein *1bgf_A* in the test set in Fig. 2, where the time ranges from 300 to 30,000.

Analysis of Experts' Knowledge

Prior studies (Zoph et al. 2022) have observed that learners specialize in a group of tokens or shallow concepts in natural language. This intrigues us to investigate what each expert in our MoE-DSR learns. Tab. 3 presents the distribution of molecular systems assigned to each expert. It is evident that the experts primarily handle protein systems with high SI greater than 40%, indicating that they specialize in similar types of proteins. This aligns with our approach of directing each expert's focus towards particular subsets of protein data, enabling MoE-DSR to better adapt to new and changing scenarios in dynamic molecular systems.

Ablation Studies

We compete our Q-MPNN with other popular 3D point cloud algorithms to examine the superiority of different surface cloud algorithms. Specificly, we replace Q-MPNN with PointNet++ (Qi et al. 2017), PointCNN (Hua, Tran, and Yeung 2018), Point Transformer (Zhao et al. 2021), PointMLP (Ma et al. 2022), and dMaSIF (Sverrisson et al. 2021) as the key backbone of the router, and document the results in Tab. 4. Notably, PointNet++, PointCNN, Point Transformer, and PointMLP are all designed for common point cloud understanding without considering the role of

	IoU↑	CHAMFER_DIST↓	NC↑
PointNet++	0.6656	0.0055	0.6188
PointCNN	0.6680	0.0053	0.6213
PointTrans.	0.6749	0.0047	0.6453
PointMLP	0.6857	0.0042	0.6519
dMaSIF	0.6923	0.0040	0.6560
MoE-DSR	**0.7028**	**0.0039**	**0.6723**

Table 4: Performance of different backbone architectures in the surface cloud network in MoE-DSR.

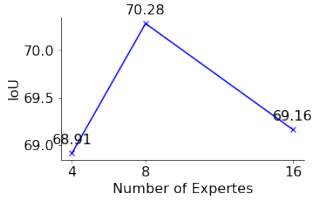

Figure 3: The performance of different numbers of experts.

normals. Experiments show that Q-MPNN achieves the best performance in extracting structural information from the initial conformation's surface. Moreover, we replace the irreps representations $\|\mathbf{x}\|\cdot\mathrm{SH}(\cdot)$ with the ordinary coordinate input $\mathbf{x}$ and observe a significant IoU drop to 0.6884 ± 0.04, demonstrating the necessity to incorporate equivariance. We also study the sensitivity of MoE-DSR to several key hyperparameters and find that simply increasing the number of experts ($N_E \geq 16$) will not cause a positive impact.

Hperparameter Sensitivity. We report our model's sensitivity to N_E. Fig. 3 shows that simply increasing N_E would not lead to better performance, which contradicts MoE-based large language models (LLMs). Notably, some studies (Dai, Deng et al. 2024) find that LLMs have better results with more experts although more experts also increase the parameter count. However, those LLMs are trained on massive natural language databases, but our MoE-DSR only has thousands of protein systems for training. This significantly diminishes the benefits of increasing the number of experts.

Conclusion and Future Work

Modeling long-term MD is a persistent goal for computational biologists, facilitating the study of molecules' dynamic behaviors. This work introduces MoE-DSR and expands the boundaries of INRs on protein surfaces to expedite the conventional MD simulations. However, there is room for future exploration. For instance, as self-supervised learning is prevailing to improve models' generalization, a further enhancement would be expected for our MoE-DSR by pretraining it on the abundant static molecular surfaces. Last but not least, we merely consider the moving trajectories of monomers due to the limitation of ATLAS. However, interactions are crucial to many biological processes and it is more challenging to forecast complexes' dynamic trajectories. We anticipate more succeeding efforts in building a larger MD simulation dataset with complex structures.

References

Arts, M.; et al. 2023. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18): 6151–6159.

Bernardi, R. C.; et al. 2015. Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1850(5): 872–877.

Car, R.; and Parrinello, M. 1985. Unified approach for molecular dynamics and density-functional theory. *Physical review letters*, 55(22): 2471.

Chmiela, S.; et al. 2017. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5): e1603015.

Cui, J.; et al. 2023. Direct prediction of gas adsorption via spatial atom interaction learning. *Nature Communications*, 14(1): 7043.

Dai, D.; Deng, C.; et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

De Luigi, L.; et al. 2023. Deep learning on implicit neural representations of shapes. *arXiv preprint arXiv:2302.05438*.

DeLano, W. L.; et al. 2002. Pymol: An open-source molecular graphics tool. *CCP4 Newsl. Protein Crystallogr*, 40(1): 82–92.

Durumeric, A. E.; et al. 2019. Adversarial-residual-coarse-graining: Applying machine learning theory to systematic molecular coarse-graining. *The Journal of chemical physics*, 151(12).

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120): 1–39.

Gilmer, J.; et al. 2017. Neural message passing for quantum chemistry. In *ICML*, 1263–1272. PMLR.

Gropp, A.; et al. 2020. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*.

Hollingsworth, S. A.; and Dror, R. O. 2018. Molecular dynamics simulation for all. *Neuron*, 99(6): 1129–1143.

Hua, B.-S.; Tran, M.-K.; and Yeung, S.-K. 2018. Pointwise convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 984–993.

Jackson, N. E.; Savoie, B. M.; Statt, A.; and Webb, M. A. 2023. Introduction to Machine Learning for Molecular Simulation.

Kohler, J.; et al. 2023. Flow-matching: Efficient coarse-graining of molecular dynamics without forces. *Journal of Chemical Theory and Computation*, 19(3): 942–952.

Lai, H.; et al. 2024. Interformer: an interaction-aware model for protein-ligand docking and affinity prediction. *Nature Communications*, 15(1): 10223.

Lee, Y.; Yu, H.; Lee, J.; and Kim, J. 2023. Pre-training Sequence, Structure, and Surface Features for Comprehensive Protein Representation Learning. In *The Twelfth ICLR*.

Leem, J.; Mitchell, L. S.; Farmery, J. H.; Barton, J.; and Galson, J. D. 2022. Deciphering the language of antibodies using self-supervised learning. *Patterns*, 100513.

Liao, Y.-L.; et al. 2022. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*.

Lin, Z.; et al. 2022. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*.

Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; and Shaw, D. E. 2011. How fast-folding proteins fold. *Science*, 334(6055): 517–520.

Lorensen, W. E.; and Cline, H. E. 1998. Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, 347–353.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123*.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 30.

Schaeffer, R. D.; Liao, Y.; Cheng, H.; and Grishin, N. V. 2017. ECOD: new developments in the evolutionary classification of domains. *Nucleic acids research*, 45(D1): D296–D302.

Schutt, K. T.; Arbabzadah, F.; Chmiela, S.; Muller, K. R.; and Tkatchenko, A. 2017. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8(1): 13890.

Schutt, K. T.; et al. 2018. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24).

Shan, Y.; et al. 2011. How does a drug molecule find its target binding site? *Journal of the American Chemical Society*, 133(24): 9181–9183.

Shazeer, N.; et al. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Strokach, A.; et al. 2020. Fast and flexible protein design using deep graph neural networks. *Cell systems*, 11(4): 402–411.

Sun, D.; et al. 2024. DSR: Dynamical Surface Representation as Implicit Neural Networks for Protein. *NeurIPS*, 36.

Suomivuori, C.-M.; et al. 2017. Energetics and dynamics of a light-driven sodium-pumping rhodopsin. *Proceedings of the National Academy of Sciences*, 114(27): 7043–7048.

Sverrisson, F.; et al. 2021. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15272–15281.

Tang, X.; et al. 2024a. BC-Design: A Biochemistry-Aware Framework for Highly Accurate Inverse Protein Folding. *bioRxiv*, 2024–10.

Tang, X.; et al. 2024b. A survey of generative AI for de novo drug design: new frontiers in molecule and protein generation. *Briefings in Bioinformatics*, 25(4): bbae338.

Van der Walt, S.; et al. 2014. scikit-image: image processing in Python. *PeerJ*, 2: e453.

Vander Meersche, Y.; et al. 2024. ATLAS: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic Acids Research*, 52(D1): D384–D392.

Vaswani, A.; et al. 2017. Attention is all you need. *NeurIPS*, 30.

Wang, Y.; et al. 2020. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Current opinion in structural biology*, 61: 139–145.

Wu, F. 2024. A Semi-supervised Molecular Learning Framework for Activity Cliff Estimation. In *33rd International Joint Conference on Artificial Intelligence, IJCAI 2024*, 6080–6088. International Joint Conferences on Artificial Intelligence.

Wu, F.; Li, S.; Jin, X.; Jiang, Y.; Radev, D.; Niu, Z.; and Li, S. Z. 2023a. Rethinking explaining graph neural networks via non-parametric subgraph matching. In *International Conference on Machine Learning*, 37511–37523. PMLR.

Wu, F.; and Li, S. Z. 2024. Surface-vqmae: Vector-quantized masked auto-encoders on molecular surfaces. In *International Conference on Machine Learning*, 53619–53634. PMLR.

Wu, F.; Xu, T.; Jin, S.; Tang, X.; Xu, Z.; Zou, J.; and Hie, B. 2024a. D-Flow: Multi-modality Flow Matching for D-peptide Design. *arXiv preprint arXiv:2411.10618*.

Wu, F.; et al. 2022a. Discovering and explaining the representation bottleneck of graph neural networks from multi-order interactions. *arXiv preprint arXiv:2205.07266*.

Wu, F.; et al. 2022b. Discovering the Representation Bottleneck of Graph Neural Networks from Multi-order Interactions. *arXiv preprint arXiv:2205.07266*.

Wu, F.; et al. 2022c. Pre-Training of Equivariant Graph Matching Networks with Conformation Flexibility for Drug Binding. *Advanced Science*, 9(33): 2203796.

Wu, F.; et al. 2022d. When Geometric Deep Learning Meets Pretrained Protein Language Models. *arXiv preprint arXiv:2212.03447*.

Wu, F.; et al. 2023b. DIFFMD: A Geometric Diffusion Model for Molecular Dynamics Simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5321–5329.

Wu, F.; et al. 2023c. Improving molecular representation learning with metric learning-enhanced optimal transport. *Patterns*, 4(4).

Wu, F.; et al. 2023d. Integration of pre-trained protein language models into geometric deep learning networks. *Communications Biology*, 6(1): 876.

Wu, F.; et al. 2023e. Molformer: Motif-based transformer on 3d heterogeneous molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5312–5320.

Wu, F.; et al. 2024b. Discovering the Representation Bottleneck of Graph Neural Networks. *IEEE Transactions on Knowledge and Data Engineering*.

Wu, F.; et al. 2024c. A hierarchical training paradigm for antibody structure-sequence co-design. *Advances in Neural Information Processing Systems*, 36.

Wu, F.; et al. 2024d. Instructor-inspired Machine Learning for Robust Molecular Property Prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ye, Q.-Z. 1988. The signed Euclidean distance transform and its applications. In *9th International conference on pattern recognition*, 495–496. IEEE Computer Society.

Zhang, L.; et al. 2018. DeePCG: Constructing coarse-grained models via deep neural networks. *The Journal of Chemical Physics*, 149(3).

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16259–16268.

Zheng, L.-E.; et al. 2023. Machine Learning Generation of Dynamic Protein Conformational Ensembles. *Molecules*, 28(10): 4047.

Zoph, B.; et al. 2022. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.