Equivariant Pretrained Transformer for Unified Geometric Learning on Multi-Domain 3D Molecules

Rui Jiao^{1,2*}**, Xiangzhe Kong**^{1,2*}**, Ziyang Yu, Wenbing Huang**^{3,4†}**, Yang Liu**^{1,2†}
[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University
[2]Institute for AIR, Tsinghua University
[3]Gaoling School of Artificial Intelligence, Renmin University of China
[4]Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China

Abstract

Pretraining on a large number of unlabeled 3D molecules has showcased superiority in various scientific applications. However, prior efforts typically focus on pretraining models on a specific domain, either proteins or small molecules, missing the opportunity to leverage the cross-domain knowledge. To mitigate this gap, we introduce **E**quivariant **P**retrained **T**ransformer (EPT), a novel pretraining framework designed to harmonize the geometric learning of small molecules and proteins. To be specific, EPT unifies the geometric modeling of multi-domain molecules via the block-enhanced representation that can attend a broader context of each atom. Upon transformer framework, EPT is further enhanced with E(3) equivariance to facilitate the accurate representation of 3D structures. Another key innovation of EPT is its block-level pretraining task, which allows for joint pretraining on datasets comprising both small molecules and proteins. Experimental evaluations on a diverse group of benchmarks, including ligand binding affinity prediction, molecular property prediction, and protein property prediction, show that EPT significantly outperforms previous SOTA methods for affinity prediction, and achieves the best or comparable performance with existing domain-specific pretraining models for other tasks.

1 Introduction

Representing and understanding the 3D geometric structure of molecular systems is of crucial importance across a multitude of scientific domains, including life science (Eslami et al., 2022), drug discovery (Blanco-Gonzalez et al., 2023), and material design (Pyzer-Knapp et al., 2022). This is largely owing to the fact that 3D structures mostly determine molecular properties and effects of various downstream tasks which are hardly captured by 1D representations such as SMILES of chemical molecules and amino acid sequences of proteins.

In recent years, geometric graph neural networks (Schütt et al., 2018; Satorras et al., 2021; Liao & Smidt, 2022) that perfectly model the physical interactions between atoms and conform to the E(3) symmetry of physics, have emerged as powerful tools for characterizing the geometric structure of small molecules and proteins, leading to remarkable performance in molecule property prediction (Zhou et al., 2023), protein generation (Watson et al., 2023), antibody design (Xu et al., 2022; Kong et al., 2023a), etc.

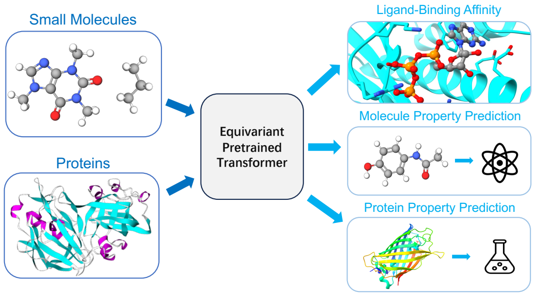
Small Molecules
Proteins
Equivariant Pretrained Transformer
Ligand-Binding Affinity
Molecule Property Prediction
Protein Property Prediction

Figure 1: EPT aims at pretraining one model for multiple domains.

[*] Equal Contribution.

[†] Wenbing Huang and Yang Liu are corresponding authors.

Adopting deep learning for scientific purposes faces a central challenge: the shortage of labeled data. To address this issue, recent researchers have incorporated the concept of self-supervised pretraining techniques from Natural Language Processing (NLP), exemplified by models like BERT (Kenton & Toutanova, 2019) and GPT (Radford et al., 2018). Their experiments confirm that the models pretrained from large-scale unsupervised 3D molecules do exhibit improved performance in downstream tasks after fine-tuning with labeled data. In general, existing pretraining methods on 3D molecules mainly focus on *one pretrained model for one domain*. These include small-molecule-specific models (Luo et al., 2022; Jiao et al., 2023), protein-specific models (Zhang et al., 2022), and dual-tower architectures which can cope with two domains of small molecules and proteins but still leverage one tower for one domain separately (Zhou et al., 2023; Feng et al., 2023a).

In contrast to previous methods, this paper proposes *one pretrained model for multiple domains* (see Figure 1), to enable unified geometric learning on 3D molecules. Undoubtedly, developing a molecular foundation model is a formidable task. First, cross-domain data are constructed in different formats. For example, small molecules are usually of single-level representation with atoms as the basic building unit. However, proteins are of two-level representation, consisting of amino acids each of which consists of a certain number of atoms. It is necessary to derive a consistent representation method to unify this domain difference. Second, molecular systems are always driven by the essential physical rules in the atom space. The model we propose should appropriately capture the physical interaction between atoms and comply with E(3) symmetry. Last, existing self-supervised pretraining objectives are almost designed for specific domains or even specific tasks. It is demanded to develop multi-domain pretraining objectives that are capable of balancing domain-specific learning and cross-domain transferring.

In response to these challenges, we introduce Equivariant Pretrained Transformer (EPT), which consists of three components: unified molecular modeling, equivariant full-atom transformer, and block-level denoising pretraining, each of which is described below:

- The unified molecular modeling enhances each atom representation by incorporating block-level features that attend a broader context of each atom, such as the atom-level surroundings for small molecules and the residue-level belongings for proteins.
- The equivariant full-atom transformer is thoroughly designed upon generic transformer. It derives the embedding layer with one-layer equivariant GNN to reflect the graph geometry, and then update the atom-level scalar and vector features via the equivariant self-attention and feed-forward mechanisms in each layer.
- We propose a block-level denoised pretraining task, which requires the model to recognize the translation and rotation perturbations applied on each block, thereby enhancing the model's ability to model the complex hierarchical geometry of molecules.

We conduct experimental evaluations on a diverse group of benchmarks, including Ligand Binding Affinity (LBA) prediction, molecular property prediction on QM9, and protein property prediction on EC and MSP. The results show that EPT significantly outperforms previous SOTA methods on LBA, and achieves the best or comparable performance with existing domain-specific pretraining methods on other datasets, which desirably affirm the benefits of our work.

2 Equivariant Pretrained Transformer

The overview of our method is shown in Figure 2. In this section, we first introduce our hierarchical modelling technique to unify the molecular representation across various domains in §2.1, and then present the design of the transformer-based backbone model in §2.2. Finally, we propose the block-level denoising task for multiple domain pretraining in §2.3.

2.1 Unified Molecular Modelling

In computational chemistry and molecular modeling, it is common to represent a molecule with N atoms as a graph $\mathcal{G}$, where atoms are depicted as nodes and their interactions are depicted as edges denoted as $\mathcal{E}$. Each atom, indexed by i, is characterized by a set of features $(a_i, \vec{z}_i)$, with $a_i \in \mathcal{A}$ specifying the atom type and $\vec{z}_i \in \mathbb{R}^3$ representing the atom's 3D coordinate. To capture the high-level structure within molecules, atoms are additionally grouped into M predefined blocks to

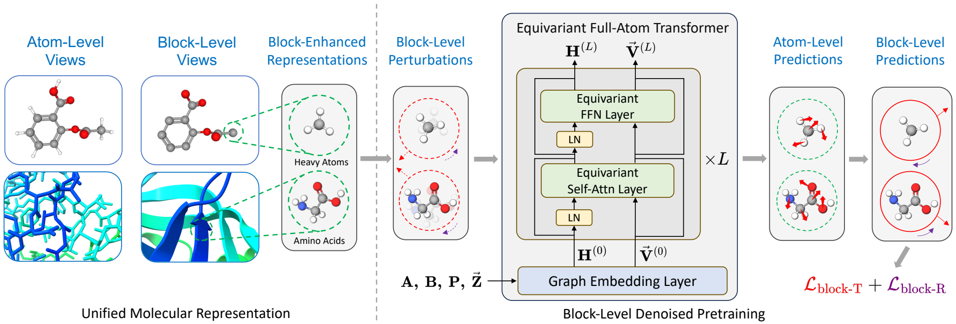
Atom-Level Views
Block-Level Views
Block-Enhanced Representations
Heavy Atoms
Amino Acids
Block-Level Perturbations
Equivariant Full-Atom Transformer
$\mathbf{H}^{(L)}$
$\vec{\mathbf{V}}^{(L)}$
Equivariant FFN Layer
LN
Equivariant Self-Attn Layer
LN
$\times L$
$\mathbf{H}^{(0)}$
$\vec{\mathbf{V}}^{(0)}$
$\mathbf{A}, \mathbf{B}, \mathbf{P}, \vec{\mathbf{Z}}$
Graph Embedding Layer
Atom-Level Predictions
Block-Level Predictions
$\mathcal{L}_{\text{block-T}} + \mathcal{L}_{\text{block-R}}$
Unified Molecular Representation
Block-Level Denoised Pretraining

Figure 2: Overview of our proposed EPT. Our framework is designed to harness the rich information in multi-domain 3D molecular datasets. To achieve this goal, we adopt the concept of blocks to unify the molecular representations (§2.1), design an Equivariant Full-Atom Transformer as the backbone model (§2.2), and proposed the block-level denoising technique for pretraining (§2.3).

enrich the node features (Kong et al., 2023b). Specifically, in the case of small molecules, the blocks are composed of non-hydrogen atoms (or so-called heavy atoms) and their directly bonded hydrogen neighbors. For proteins, the blocks correspond to the amino acid residues. Let m_i denote the index of the block containing atom i, the feature set for an atom is extended to $(a_i, b_{m_i}, p_i, \vec{\boldsymbol{z}}_i)$, where b_{m_i} indicates the block type of m_i, and p_i denotes the atom's predefined sequential number within its block. We provide more details of the vocabularies for atom types, block types and atom positions in Appendix C. Interactions between atoms are categorized into three distinct edge types to reflect both intra-block and inter-block relationships. Mathematically,

$$e_{ij} = \begin{cases} 0, & m_i = m_j, \\ 1, & m_i \neq m_j, d(m_i, m_j) \leq \delta_{\text{topo}}, \\ 2, & m_i \neq m_j, \delta_{\text{topo}} < d(m_i, m_j) \leq \delta_{\text{max}}, \end{cases} \tag{1}$$

where δ_{topo} and δ_{max} are predefined thresholds that represent topological and maximum allowable distances, respectively. The function $d(m_i, m_j) = \min_{m_p=m_i, m_q=m_j} \|\vec{z}_p - \vec{z}_q\|_2$ calculates the minimum Euclidean distance between any two atoms belonging to blocks m_i and m_j. This framework allows for a nuanced representation of molecular structures, accommodating the complex nature of atomic interactions within a molecule.

2.2 Equivariant Full-Atom Transformer

In the pursuit of efficiently capturing the nuanced interactions of atoms within molecules, we present the Equivariant Full-Atom Transformer. It utilizes the Transformer-based backbone (Vaswani et al., 2017) to model the complex interactions among atoms, while keeps updating the scalar and vector features to capture the rich geometric information inherent in molecular structures.

Specifically, our model first acquires the input features from the Graph Embedding layer, and iteratively updates the features at each layer l. Let $\boldsymbol{H}^{(l)} = [\boldsymbol{h}_1^{(l)}, \boldsymbol{h}_2^{(l)}, \cdots, \boldsymbol{h}_N^{(l)}] \in \mathbb{R}^{h \times N}$ denote the scalar, and $\vec{\boldsymbol{V}}^{(l)} = [\vec{\boldsymbol{v}}_1^{(l)}, \vec{\boldsymbol{v}}_2^{(l)}, \cdots, \vec{\boldsymbol{v}}_N^{(l)}] \in \mathbb{R}^{3 \times h \times N}$ the vector. The model is constructed in this way:

$$[\boldsymbol{H}^{(0)}, \vec{\boldsymbol{V}}^{(0)}] = \text{Embedding}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}, \vec{\boldsymbol{Z}}), \tag{2}$$

$$[\boldsymbol{H}^{(l-0.5)}, \vec{\boldsymbol{V}}^{(l-0.5)}] = [\boldsymbol{H}^{(l-1)}, \vec{\boldsymbol{V}}^{(l-1)}] + \text{Self-Attn}(\text{LN}(\boldsymbol{H}^{(l-1)}), \vec{\boldsymbol{V}}^{(l-1)}), \tag{3}$$

$$[\boldsymbol{H}^{(l)}, \vec{\boldsymbol{V}}^{(l)}] = [\boldsymbol{H}^{(l-0.5)}, \vec{\boldsymbol{V}}^{(l-0.5)}] + \text{FFN}(\text{LN}(\boldsymbol{H}^{(l-0.5)}), \vec{\boldsymbol{V}}^{(l-0.5)}). \tag{4}$$

After the Embedding layer, Self-attention (Self-Attn) and Feed-Forward Networks (FFN) are applied alternately, with pre-layer normalization (LN) and residual connections preceding each operation. We modify the Self-Attn and FFN layers to be E(3)-equivariant, preserving the geometrical symmetry of molecular structures. These layers are detailed as follows. For conciseness, we omit the layer subscribe l unless otherwise specified.

Graph Embedding Layer. The input features are obtained:

$$\boldsymbol{f}_i = f_b(b_{m_i}) + f_a(a_i) + f_p(p_i), \tag{5}$$

$$\boldsymbol{e}'_{ij} = [\boldsymbol{f}_i, \boldsymbol{f}_j, \boldsymbol{e}_{ij}, \text{RBF}(\|\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_j\|_2)], \tag{6}$$

$$\boldsymbol{h}_i^{(0)} = \varphi_h(\boldsymbol{f}_i, \sum_{j \in \mathcal{N}(i)} \varphi_s(\boldsymbol{e}'_{ij}) \cdot \boldsymbol{f}_j), \tag{7}$$

$$\boldsymbol{v}_i^{(0)} = \sum_{j \in \mathcal{N}(i)} \varphi_v(\boldsymbol{e}'_{ij}) \cdot (\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_j), \tag{8}$$

where f_b, f_a, f_p separately embed the block types, atom types and atom orders, RBF$(\cdot)$ denote the radial basis functions, and $\varphi_h, \varphi_s, \varphi_v$ are MLPs to aggregate neighbor information to enrich the 0-th layer features. It is necessary to initialize $\vec{\boldsymbol{V}}^{(0)}$ with SE(3)-equivariant non-zero values via Eq. (8), otherwise the vector features will remain zeros in subsequent layers.

Equivariant Self-Attention Layer. The self-attention layer plays a crucial role in modeling interatomic interactions. For each layer, query $\boldsymbol{Q}_s$, key $\boldsymbol{K}_s$, and value $\boldsymbol{V}_s$ matrices of the s-th head are computed as

$$\boldsymbol{Q}_s = \boldsymbol{H}\boldsymbol{W}_s^Q, \boldsymbol{K}_s = \boldsymbol{H}\boldsymbol{W}_s^K, \boldsymbol{V}_s = [\boldsymbol{H}\boldsymbol{W}_s^{Vh}, \vec{\boldsymbol{V}}\boldsymbol{W}_s^{Vv}], \tag{9}$$

where h_s is the dimension of each head, $\boldsymbol{W}_s^Q, \boldsymbol{W}_s^K \in \mathbb{R}^{h \times 4h_s}$, $\boldsymbol{W}_s^{Vh}, \boldsymbol{W}_s^{Vv} \in \mathbb{R}^{h \times h_s}$ are trainable parameters that map the features to the appropriate query, key, and value spaces. And the attention mechanism is given by

$$\boldsymbol{H}_s, \vec{\boldsymbol{V}}_s = \text{Softmax}\Big(\frac{\mathbf{Q_s^\top K_s}}{2\sqrt{h_s}} - \boldsymbol{D} + \boldsymbol{R}\Big)\boldsymbol{V}_s, \tag{10}$$

where $\boldsymbol{D} = \{d_{ij}\}_{i,j=1}^N = \{\|\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_j\|_2\}_{i,j=1}^N$ is the distance matrix, and $\boldsymbol{R} = \{r_{ij}\}_{i,j=1}^N$ encodes the edge interactions and geometric relations as

$$r_{ij} = \begin{cases} \varphi_r(\boldsymbol{e}_{ij}, \text{RBF}(\|\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_j\|_2)), & (i,j) \in \mathcal{E}, \\ 0, & (i,j) \notin \mathcal{E}. \end{cases} \tag{11}$$

Here φ_r is an MLP. We explicitly include distances $\boldsymbol{D}$ and edge features $\boldsymbol{R}$ in Eq. (10) to enhance the modeling of interatomic interactions, and we explore the efficacy of this design in Appendix F and H.

The outputs of the self-attention layer combines the contributions of all heads:

$$\Delta \boldsymbol{H} = \sum_s \boldsymbol{H}_s \boldsymbol{W}_s^{Oh}, \Delta \vec{\boldsymbol{V}} = \sum_s \vec{\boldsymbol{V}}_s \boldsymbol{W}_s^{Ov}, \tag{12}$$

where $\boldsymbol{W}_s^{Oh}, \boldsymbol{W}_s^{Ov} \in \mathbb{R}^{h_s \times h}$ are head-specific trainable parameters.

Equivariant Feed-Forward Layer. Building upon the Geometric Vector Perceptron (GVP, Jing et al. (2021)) concept, the equivariant feed-forward layer is where the scalar and vector features are fused and updated simultaneously:

$$\vec{\boldsymbol{V}}_1, \vec{\boldsymbol{V}}_2 = \vec{\boldsymbol{V}}\boldsymbol{W}_1, \vec{\boldsymbol{V}}\boldsymbol{W}_2, \tag{13}$$

$$\Delta \boldsymbol{H}, \boldsymbol{U} = \varphi_{\text{FFN}}(\boldsymbol{H}, \|\vec{\boldsymbol{V}}_1\|_2), \tag{14}$$

$$\Delta \vec{\boldsymbol{V}} = \text{LN}(\boldsymbol{U}) \odot \vec{\boldsymbol{V}}_2, \tag{15}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{h \times h}$ are learnable linear projectors, φ_{FFN} is an MLP that integrates the scalar features with the magnitude of the vector features, and $\odot$ denotes element-wise multiplication. The intermediate matrix $\boldsymbol{U}$ is layer-normalized to conserve the scale of the updated vectors.

2.3 Block-Level Denoised Pretraining

In this section, we describe the block-level denoised pretraining approach designed to incorporate the hierarchical information into our backbone model (denoted as φ hereinafter). The algorithm leverages the concept of Denoising Score Matching (DSM, Song & Ermon (2019)) to enable the model to learn from perturbed data representations. We provide more discussions about denoising strategies in Appendix B, and detail our training target as follows.

Complete Block-level Denoising. We design an additional rotation denoising task from the perspective of Euler's rotation equation (Soper, 2008) above the translation-only denoising target, which is previously proved effective on complex binding tasks (Jin et al., 2023).

To begin with, the torque on each block is aggregated as

$$\vec{\boldsymbol{M}}'_b[:,m_i] = \sum_{m_j=m_i} (\vec{\boldsymbol{z}}_j - \vec{\boldsymbol{Z}}_b[:,m_i]) \times \vec{\boldsymbol{f}}'_j, \tag{16}$$

According to Euler's rotation equation, the time derivative of the angular momentum of each block is given by $d\vec{\boldsymbol{L}}_b/dt = \vec{\boldsymbol{M}}'_b = \boldsymbol{I}_b\vec{\boldsymbol{\alpha}}_b$, where $\boldsymbol{I}_b \in \mathbb{R}^{3\times3\times M}$ represents the inertia matrix defined as

$$\boldsymbol{I}_b[:,:,m_i] = \sum_{m_j=m_i} \Big(\|\vec{\boldsymbol{u}}_j\|^2\boldsymbol{I} - \vec{\boldsymbol{u}}_j\vec{\boldsymbol{u}}_j^\top\Big), \tag{17}$$

where $\vec{\boldsymbol{u}}_j = \vec{\boldsymbol{z}}_j - \vec{\boldsymbol{Z}}_b[:,m_i]$, and the angular acceleration $\vec{\boldsymbol{\alpha}}_b \in \mathbb{R}^{3M}$ can be calculated as $\vec{\boldsymbol{\alpha}}_b = \boldsymbol{I}_b^{-1}\vec{\boldsymbol{M}}'_b$.

To design an objective on $\vec{\boldsymbol{\alpha}}_b$, we additionally perturb blocks by random rotations $\boldsymbol{\omega}_b$ sampled from the isotropic Gaussian distribution $\mathcal{IG}_{SO(3)}(\sigma_r)$ (Leach et al., 2022). Specifically, each rotation $\boldsymbol{\omega}_b[:,m_i] \in \mathfrak{so}(3)$ is constructed as $\boldsymbol{\omega} = \theta\hat{\boldsymbol{\omega}}$, where $\hat{\boldsymbol{\omega}}$ is a uniformly sampled unit vector, and $\theta \in [0,\pi]$ is a rotation angle with density

$$f(\theta) = \frac{1-\cos\theta}{\pi}\sum_{l=0}^{\infty}(2l+1)e^{-l(l+1)\sigma_r^2}\frac{\sin\big((l+1/2)\theta\big)}{\sin(\theta/2)}. \tag{18}$$

And the corresponding rotation matrix $\boldsymbol{Q}(\boldsymbol{\omega}) \in SO(3)$ is acquired by the exponential mapping on $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^\top$:

$$\boldsymbol{Q}(\boldsymbol{\omega}) = \exp\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{19}$$

Overall, the perturbation scheme combining block-level translation and rotation is designed as

$$\vec{\boldsymbol{Z}}_r = \vec{\boldsymbol{Z}} - g_b(\vec{\boldsymbol{Z}}_b), \tag{20}$$

$$\vec{\boldsymbol{Z}}' = C\Big(g_b(\vec{\boldsymbol{Z}}_b + \sigma_t\boldsymbol{\epsilon}_{\boldsymbol{Z}_b}) + \boldsymbol{Q}_b\vec{\boldsymbol{Z}}_r\Big). \tag{21}$$

The translation and rotation loss are defined as

$$\mathcal{L}_{\text{block-T}} = \mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{Z}_b}\sim\mathcal{N}(0,\boldsymbol{I}_{3M})}\Big[\|\mu_b(\vec{\boldsymbol{F}}') - \frac{\mu_b(\vec{\boldsymbol{Z}}') - \vec{\boldsymbol{Z}}_b}{\sigma^2}\|_2^2\Big]. \tag{22}$$

$$\mathcal{L}_{\text{block-R}} = \mathbb{E}_{\boldsymbol{\omega}\sim\mathcal{IG}_{SO(3)}(\sigma_r)}\Big[\|\vec{\boldsymbol{\alpha}}_b - \nabla_{\boldsymbol{\omega}}p(\boldsymbol{\omega})\|_2^2\Big]. \tag{23}$$

The complete block-level training objective is added as

$$\mathcal{L}_{\text{block-C}} = \mathcal{L}_{\text{block-T}} + \mathcal{L}_{\text{block-R}}. \tag{24}$$

3 Experiments

We detail the construction of the multi-domain pretraining dataset in Appendix **??** and present the results on the ligand binding affinity (LBA) prediction task in the following. More results on single domains and ablation studies are provided in Appendix E and F.

Setup The task of Ligand Binding Affinity (LBA) aims at predicting the binding affinity value between a protein pocket and the corresponding ligand. We follow the setting in Atom3D (Townshend et al.,

Table 1: The mean and standard deviations of 3 runs on the LBA dataset. The best results are in **bold** and the second best are underlined.

Model	Sequence Identity 30%			Sequence Identity 60%		
	RMSE↓	Pearson↑	Spearman↑	RMSE↓	Pearson↑	Spearman↑
DeepDTA	1.866 ± 0.080	0.472 ± 0.022	0.471 ± 0.024	1.762 ± 0.261	0.666 ± 0.012	0.663 ± 0.015
B & B	1.985 ± 0.006	0.165 ± 0.006	0.152 ± 0.024	1.891 ± 0.004	0.249 ± 0.006	0.275 ± 0.008
TAPE	1.890 ± 0.035	0.338 ± 0.044	0.286 ± 0.124	1.633 ± 0.016	0.568 ± 0.033	0.571 ± 0.021
ProtTrans	1.544 ± 0.015	0.438 ± 0.053	0.434 ± 0.058	1.641 ± 0.016	0.595 ± 0.014	0.588 ± 0.009
MaSIF	1.484 ± 0.018	0.467 ± 0.020	0.455 ± 0.014	1.426 ± 0.017	0.709 ± 0.008	0.701 ± 0.001
IEConv	1.554 ± 0.016	0.414 ± 0.053	0.428 ± 0.032	1.473 ± 0.024	0.667 ± 0.011	0.675 ± 0.019
Holoprot-Full Surface	1.464 ± 0.006	0.509 ± 0.002	0.500 ± 0.005	1.365 ± 0.038	0.749 ± 0.014	0.742 ± 0.011
Holoprot-Superpixel	1.491 ± 0.004	0.491 ± 0.014	0.482 ± 0.032	1.416 ± 0.022	0.724 ± 0.011	0.715 ± 0.006
ProtNet-Amino Acid	1.455 ± 0.009	0.536 ± 0.012	0.526 ± 0.012	1.397 ± 0.018	0.741 ± 0.008	0.734 ± 0.009
ProtNet-Backbone	1.458 ± 0.003	0.546 ± 0.007	0.550 ± 0.008	1.349 ± 0.019	0.764 ± 0.006	0.759 ± 0.001
ProtNet-All-Atom	1.463 ± 0.001	0.551 ± 0.005	0.551 ± 0.008	1.343 ± 0.025	0.765 ± 0.009	0.761 ± 0.003
Atom3D-3DCNN	1.416 ± 0.021	0.550 ± 0.021	0.553 ± 0.009	1.621 ± 0.025	0.608 ± 0.020	0.615 ± 0.028
Atom3D-ENN	1.568 ± 0.012	0.389 ± 0.024	0.408 ± 0.021	1.620 ± 0.049	0.623 ± 0.015	0.633 ± 0.021
Atom3D-GNN	1.601 ± 0.048	0.545 ± 0.027	0.533 ± 0.033	1.408 ± 0.069	0.743 ± 0.022	0.743 ± 0.027
DeepAffnity	1.893 ± 0.650	0.415	0.426	–	–	–
GeoSSL	1.451 ± 0.030	0.577 ± 0.020	0.572 ± 0.010	–	–	–
EGNN-PLM	1.403 ± 0.010	0.565 ± 0.020	0.544 ± 0.010	1.559 ± 0.020	0.644 ± 0.020	0.646 ± 0.020
Uni-Mol	1.520 ± 0.030	0.558 ± 0.000	0.540 ± 0.000	1.619 ± 0.040	0.645 ± 0.020	0.653 ± 0.020
ProFSA	1.377 ± 0.010	0.628 ± 0.010	0.620 ± 0.010	1.377 ± 0.010	0.764 ± 0.000	0.762 ± 0.010
EPT-Scratch	1.356 ± 0.041	0.604 ± 0.022	0.591 ± 0.025	1.303 ± 0.015	0.777 ± 0.001	0.776 ± 0.003
EPT-Molecule	1.325 ± 0.007	0.627 ± 0.006	0.618 ± 0.004	1.263 ± 0.022	0.791 ± 0.006	0.783 ± 0.006
EPT-Protein	1.326 ± 0.035	0.628 ± 0.014	0.611 ± 0.019	1.223 ± 0.014	0.805 ± 0.002	0.803 ± 0.004
EPT-MultiDomain	**1.318 ± 0.020**	**0.643 ± 0.005**	**0.630 ± 0.005**	**1.165 ± 0.007**	**0.822 ± 0.002**	**0.819 ± 0.002**

2020), where each sample is provided as a protein-ligand complex along with its binding affinity. The dataset includes two different splits based on a threshold of protein sequence similarity: one where the sequence identity is capped at 30% and another at 60%. Each split contains 3507, 466, and 490 complexes as the training, validation, and testing sets. We utilize the Root Mean Square Error (RMSE), Pearson correlation coefficient, and Spearman correlation coefficient as the evaluation metrics. To validate the consistency of our evaluation, we conduct experiments on three random seeds and present the mean and standard deviations of the mentioned statistical measures.

Baselines We compare our method with three lines of prior works: sequence-based methods including DeepDTA (Öztürk et al., 2018), B&B (Bepler & Berger, 2019), TAPE (Rao et al., 2019) and ProtTrans (Elnaggar et al., 2021a); structure-based models such as MaSIF (Gainza et al., 2020), IEConv (Hermosilla et al., 2020), Holoprot (Somnath et al., 2021), ProtNet (Wang et al., 2023) and three backbone models proposed by Atom3D (Townshend et al., 2020); and recent pretrain-based methods containing DeepAffnity (Karimi et al., 2019), GeoSSL (Liu et al., 2023), EGNN-PLM (Wu et al., 2022), Uni-Mol (Zhou et al., 2023) and ProFSA (Gao et al., 2023).

Results Results in Table 1 evaluates our EPT model trained under four different training conditions: from scratch, pretrained only on the small molecule or protein subset, and pretrained on the entire multi-domain dataset. We have the following observations: **1.** Structure-based models generally surpass sequence-based counterparts, underscoring the significance of 3D geometry in capturing interactive information. **2.** Pretraining on each individual subset is capable to enhance performance. Remarkably, EPT-Molecule, which is pretrained without exposure to protein or complex structures, still outshines the scratch-trained model, suggesting the presence of cross-domain transferable knowledge. **3.** EPT-MultiDomain, benefited from the entire dataset containing diverse domains, outperforms previous methods and achieve state-of-the-art performance on both of the splits. This implies that the breadth of pretraining data correlates positively with the model's performance, and enables a more generalizable understanding of biological interactions.

4 Conclusion

In this work, we propose EPT, an equivariant transformer-based model pretrained on multi-domain 3D molecule structures. We unify the representation of molecules from different domains, design an equivariant Transformer efficient for large-scale systems, and implement a block-level denoising strategy specifically for pretraining with diverse multi-domain datasets. The effectiveness of EPT is demonstrated through its superior performance on LBA, QM9, EC and MSP, highlighting the model's ability to generalize across different molecular domains.

References

Bepler, T. and Berger, B. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

Blanco-Gonzalez, A., Cabezon, A., Seco-Gonzalez, A., Conde-Torres, D., Antelo-Riveiro, P., Pineiro, A., and Garcia-Fandino, R. The role of ai in drug discovery: challenges, opportunities, and strategies. *Pharmaceuticals*, 16(6):891, 2023.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10): 7112–7127, 2021a.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Yu, W., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021b. doi: 10.1109/TPAMI.2021.3095381.

Eslami, M., Adler, A., Caceres, R. S., Dunn, J. G., Kelley-Loughnane, N., Varaljay, V. A., and Martin, H. G. Artificial intelligence for synthetic biology. *Communications of the ACM*, 65(5):88–97, 2022.

Feng, S., Li, M., Jia, Y., Ma, W., and Lan, Y. Protein-ligand binding representation learning from fine-grained interactions. *arXiv preprint arXiv:2311.16160*, 2023a.

Feng, S., Ni, Y., Lan, Y., Ma, Z.-M., and Ma, W.-Y. Fractional denoising for 3d molecular pre-training. In *International Conference on Machine Learning*, pp. 9938–9961. PMLR, 2023b.

Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M., and Correia, B. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

Gao, B., Jia, Y., Mo, Y., Ni, Y., Ma, W., Ma, Z., and Lan, Y. Self-supervised pocket pretraining via protein fragment-surroundings alignment. *arXiv preprint arXiv:2310.07229*, 2023.

Gasteiger, J., Giri, S., Margraf, J. T., and Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

Godwin, J., Schaarschmidt, M., Gaunt, A. L., Sanchez-Gonzalez, A., Rubanova, Y., Veličković, P., Kirkpatrick, J., and Battaglia, P. Simple gnn regularisation for 3d molecular property prediction and beyond. In *International Conference on Learning Representations*, 2021.

Hermosilla, P., Schäfer, M., Lang, M., Fackelmann, G., Vázquez, P.-P., Kozlikova, B., Krone, M., Ritschel, T., and Ropinski, T. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. In *International Conference on Learning Representations*, 2020.

Jiao, R., Han, J., Huang, W., Rong, Y., and Liu, Y. Energy-motivated equivariant pretraining for 3d molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8096–8104, 2023.

Jin, W., Sarkizova, S., Chen, X., Hacohen, N., and Uhler, C. Unsupervised protein-ligand binding energy prediction via neural euler's rotation equation. *arXiv preprint arXiv:2301.10814*, 2023.

Jing, B., Eismann, S., Soni, P. N., and Dror, R. O. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021.

Karimi, M., Wu, D., Wang, Z., and Shen, Y. Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35 (18):3329–3338, 2019.

Kenton, J. D. M.-W. C. and Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Kong, X., Huang, W., and Liu, Y. End-to-end full-atom antibody design. *arXiv preprint arXiv:2302.00203*, 2023a.

Kong, X., Huang, W., and Liu, Y. Generalist equivariant transformer towards 3d molecular interaction learning. *arXiv preprint arXiv:2306.01474*, 2023b.

Leach, A., Schmon, S. M., Degiacomi, M. T., and Willcocks, C. G. Denoising diffusion probabilistic models on SO(3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022. URL `https://openreview.net/forum?id=BY88eBbkpe5`.

Lefaudeux, B., Massa, F., Liskovich, D., Xiong, W., Caggiano, V., Naren, S., Xu, M., Hu, J., Tintore, M., Zhang, S., Labatut, P., and Haziza, D. xformers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`, 2022.

Liao, Y.-L. and Smidt, T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*, 2022.

Liu, S., Wang, H., Liu, W., Lasenby, J., Guo, H., and Tang, J. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2021.

Liu, S., Guo, H., and Tang, J. Molecular geometry pretraining with se (3)-invariant denoising distance matching. *arXiv preprint arXiv:2206.13602*, 2022.

Liu, S., Du, W., Ma, Z.-M., Guo, H., and Tang, J. A group symmetric stochastic differential equation model for molecule multi-modal pretraining. In *International Conference on Machine Learning*, pp. 21497–21526. PMLR, 2023.

Luo, S., Chen, T., Xu, Y., Zheng, S., Liu, T.-Y., Wang, L., and He, D. One transformer can understand both 2d & 3d molecular data. *arXiv preprint arXiv:2210.01765*, 2022.

Öztürk, H., Özgür, A., and Ozkirimli, E. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Puny, O., Atzmon, M., Smith, E. J., Misra, I., Grover, A., Ben-Hamu, H., and Lipman, Y. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2021.

Pyzer-Knapp, E. O., Pitera, J. W., Staar, P. W., Takeda, S., Laino, T., Sanders, D. P., Sexton, J., Smith, J. R., and Curioni, A. Accelerating materials discovery using artificial intelligence, high performance computing and robotics. *npj Computational Materials*, 8(1):84, 2022.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Schütt, K., Unke, O., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pp. 9377–9388. PMLR, 2021.

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

Somnath, V. R., Bunne, C., and Krause, A. Multi-scale representation learning on proteins. *Advances in Neural Information Processing Systems*, 34:25244–25255, 2021.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Soper, D. E. *Classical field theory*. Courier Dover Publications, 2008.

Stärk, H., Beaini, D., Corso, G., Tossou, P., Dallago, C., Günnemann, S., and Liò, P. 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*, pp. 20479–20502. PMLR, 2022.

Thölke, P. and De Fabritiis, G. Torchmd-net: equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*, 2022.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Townshend, R. J., Vögele, M., Suriana, P., Derry, A., Powers, A., Laloudakis, Y., Balachandar, S., Jing, B., Anderson, B., Eismann, S., et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, L., Liu, H., Liu, Y., Kurtin, J., and Ji, S. Learning hierarchical protein representations via complete 3d graph networks. In *International Conference on Learning Representations (ICLR)*, 2023.

Wang, Z., Combs, S. A., Brand, R., Calvo, M. R., Xu, P., Price, G., Golovach, N., Salawu, E. O., Wise, C. J., Ponnapalli, S. P., et al. Lm-gvp: an extensible sequence and structure informed deep learning framework for protein property prediction. *Scientific reports*, 12(1):6832, 2022.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Wu, F., Li, S., Wu, L., Li, S. Z., Radev, D., and Zhang, Q. Discovering the representation bottleneck of graph neural networks from multi-order interactions. *arXiv preprint arXiv:2205.07266*, 2022.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Yu, Q., Zhang, Y., Ni, Y., Feng, S., Lan, Y., Zhou, H., and Liu, J. Unified molecular modeling via modality blending. *arXiv preprint arXiv:2307.06235*, 2023.

Zaidi, S., Schaarschmidt, M., Martens, J., Kim, H., Teh, Y. W., Sanchez-Gonzalez, A., Battaglia, P., Pascanu, R., and Godwin, J. Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*, 2022.

Zhang, Z., Xu, M., Jamasb, A., Chenthamarakshan, V., Lozano, A., Das, P., and Tang, J. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

Zhang, Z., Xu, M., Lozano, A., Chenthamarakshan, V., Das, P., and Tang, J. Pre-training protein encoder via siamese sequence-structure diffusion trajectory prediction. In *Annual Conference on Neural Information Processing Systems*, 2023.

Zhou, G., Gao, Z., Ding, Q., Zheng, H., Xu, H., Wei, Z., Zhang, L., and Ke, G. Uni-mol: a universal 3d molecular representation learning framework. 2023.

A Related Works

Geometric Graph Neural Networks. Geometric graph neural networks have emerged as a powerful paradigm for learning on graph-structured data while respecting the inherent symmetries present in many physical and biological systems. These networks employ geometric graphs that assign 3D coordinates to each node, ensuring that the graph's scalar attributes and dynamic processes remain invariant and equivariant under E(3) or SE(3) transformations in 3D space. To preserve these symmetries, prior research has employed a range of strategies, including irreducible representations Thomas et al. (2018), frame averaging Puny et al. (2021), and scalarization mechanism Schütt et al. (2018); Satorras et al. (2021). More recently, Transformer-based models have demonstrated their superior performance on 3D tasks Thölke & De Fabritiis (2022); Liao & Smidt (2022); Zhou et al. (2023). In this work, we leverage vector features into the Transformer-based backbone for effective geometric modelling, while faithful to the standard Transformer architecture to enable the memory efficient techniques Lefaudeux et al. (2022) to handle the molecular structures across various scales.

Pretraining on Domain-Specific Datasets. The scarcity and high cost of labeled molecular data necessitate the use of label-free pretraining methods for molecular representation models. In the domain of small molecules, GraphMVP Liu et al. (2021) and 3D Infomax Stärk et al. (2022) apply contrastive learning on 2D-3D pairs, while MoleBlend Yu et al. (2023) introduces a multimodal pretraining framework to align 2D and 3D features. For protein domains, GearNet Zhang et al. (2022) applies contrastive learning on sequencial and structural views of the proteins. Moreover, several works also focus on modelling the interactions across different domains. For instance, Uni-Mol Zhou et al. (2023) pretrains two separate models for small molecules and protein pockets, and then finetune the combined model on a binding dataset. Inspired from score-based generative models, denoising has emerged as a powerful pretraining method to construct a learned force field Zaidi et al. (2022); Liu et al. (2022); Jiao et al. (2023); Feng et al. (2023b). NERE Jin et al. (2023) designs translation and rotation denoising task on binding complexes. While NERE conducts perturbations on the entire ligand, our approach introduces noise at the block-level, enhancing the model's capability to capture the hierarchical interactions.

B Details of More Denoising Strategies

Predict Forces from EPT To applying the denoising task on the EPT model, one additional requirement is to predict forces from the backbone for pretraining. In practice, we apply an additional FFN-like layer over layer-L to fuse the output scalars and vectors as

$$\boldsymbol{H}_{\text{out}}, \vec{\boldsymbol{V}}_{\text{out}} = \boldsymbol{H}^{(L)}, \vec{\boldsymbol{V}}^{(L)}/\|\vec{\boldsymbol{V}}^{(L)}\|_2, \tag{25}$$

$$\vec{\boldsymbol{F}}' = \varphi_{\text{out}}(\boldsymbol{H}_{\text{out}}, \|\vec{\boldsymbol{V}}_{\text{out}}\boldsymbol{W}_1'\|_2) \odot \vec{\boldsymbol{V}}_{\text{out}}\boldsymbol{W}_2'. \tag{26}$$

Generally, the overview of DSM is outlined as follows. The training process begins by sampling perturbed coordinates $\vec{\boldsymbol{Z}}'$ from a predefined noise distribution parameterized by σ. The atom-level pseudo forces $\vec{\boldsymbol{F}}'$ are then predicted by the model φ to recover $\vec{\boldsymbol{Z}}$ from $\vec{\boldsymbol{Z}}'$, and finally used to compute the denoising loss $\mathcal{L}$. The key points of DSM lie in the design of the perturbation mechanism and the corresponding loss function to align the predicted forces $\vec{\boldsymbol{F}}'$ with the Denoising Force Field (DFF). In the following, we first introduce the simple atom-level denoising method, then extend the denoising targets from atoms to blocks, and finally apply additional rotations on blocks to better depict the geometric landscape.

Atom-level Denoising. The atom-level denoising process Zaidi et al. (2022); Jiao et al. (2023) independently introduces Gaussian noise to each atom as $\boldsymbol{\epsilon}_{\boldsymbol{Z}} \sim \mathcal{N}(0, \boldsymbol{I}_{3N})$ rescaled by σ_t:

$$\vec{\boldsymbol{Z}}' = C(\vec{\boldsymbol{Z}} + \sigma_t \boldsymbol{\epsilon}_{\boldsymbol{Z}}), \tag{27}$$

where the operation $C(\vec{\boldsymbol{Z}}) = \vec{\boldsymbol{Z}} - \sum \vec{\boldsymbol{Z}}/N$ projects the noised sample to the mean-centered subspace to neutralize the translation introduced by the noises Xu et al. (2022).

The training objective is to match the predicted $\vec{\boldsymbol{F}}'$ with the denoising force field yielded by $\nabla_{\vec{\boldsymbol{Z}}'} p_{\sigma_t}(\vec{\boldsymbol{Z}}'|\vec{\boldsymbol{Z}})$ as

$$\mathcal{L}_{\text{atom}} = \mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{Z}} \sim \mathcal{N}(0, \boldsymbol{I}_{3N})} \Big[\| \vec{\boldsymbol{F}}' - \frac{\vec{\boldsymbol{Z}}' - \vec{\boldsymbol{Z}}}{\sigma^2} \|_2^2 \Big]. \tag{28}$$

Translation-only Block-level Denoising. To conserve the intra-block geometry, we extend the atom-level denoising task into block-level by considering blocks as rigid bodies, and all the atoms within the same block are applied by the same noise. For simplification, we first define the operators $\mu_b : \mathbb{R}^{3\times N} \to \mathbb{R}^{3\times M}, g_b : \mathbb{R}^{3\times M} \to \mathbb{R}^{3\times N}$ denote the atom-to-block averaging and the block-to-atom duplication. In particular, we have

$$\begin{cases} \mu_b(\vec{\boldsymbol{Z}})[:, m_i] = \frac{\sum_{m_j = m_i} \vec{\boldsymbol{z}}_j}{\sum_{m_j = m_i} 1}, \\ g_b(\vec{\boldsymbol{Z}}_b)[:, i] = \vec{\boldsymbol{Z}}_b[:, m_i]. \end{cases} \tag{29}$$

Slightly different from the atom-level setting, we apply noises on the center of each block as $\boldsymbol{\epsilon}_{\boldsymbol{Z}_b} \sim \mathcal{N}(0, \boldsymbol{I}_{3M})$. For noise scale σ_t, the perturbation and mean-centered projection are sequentially calculated as

$$\vec{\boldsymbol{Z}}' = C\big(\vec{\boldsymbol{Z}} + \sigma_t g_b(\boldsymbol{\epsilon}_{\boldsymbol{Z}_b})\big). \tag{30}$$

The training objective adapts Eq. (28) into block-level as

$$\mathcal{L}_{\text{block-T}} = \mathbb{E}_{\boldsymbol{\epsilon}_{\boldsymbol{Z}_b} \sim \mathcal{N}(0, \boldsymbol{I}_{3M})} \Big[\| \mu_b(\vec{\boldsymbol{F}}') - \frac{\mu_b(\vec{\boldsymbol{Z}}') - \vec{\boldsymbol{Z}}_b}{\sigma^2} \|_2^2 \Big]. \tag{31}$$

C Vocabulary Construction

The vocabulary of atom types, block types and atom positions are detailed in Table 2.

Table 2: Construction of the vocabulary of atom types, block types and atom positions.

Vocabulary	Index	Descriptions
Atom	0∼2	<pad>, <mask>, <global>
	3∼120	118 elements
Block	0∼3	<pad>, <mask>, <unk>, <global>
	4∼23	20 amino acids
	24∼141	118 elements (H is included for completeness)
Position	0∼2	<pad>, <mask>, <global>
	3∼12	position codes for atoms in protein, i.e. α, β, etc.
	13	<sml>for atoms in small molecules

D Multi-Domain Pretraining

D.1 Dataset Collection

We collection the 3D molecule datasets from small molecules and protein domain, as detailed in Table 3. Moreover, for each time loading data from PDB, we further randomly extract a local scope with three sequencially continuous residues as a training sample. This random segmentation approach is applied as the data augmentation for proteins.

D.2 Hyperparameters

We pretrain EPT on 8 NVIDIA Tesla A800 with hyperparameters in Table 4.

Table 3: Statistics of the structural datasets for pretraining.

Domain	Source	# of entries
Small Molecule	GEOM-QM9	430,201
	GEOM-Drugs	1,465,181
	PCQM4Mv2	3,378,606
Protein	PDB	599,699
	PDBBind-PP	2,852
	PDBBind-refined-set	5,316
	PDBBind-v2020-other-PL	14,127

Table 4: Hyperparameters for constructing and training EPT.

Name	h_{hidden}	h_{ffn}	h_{edge}	h_{rbf}	L	H	δ_{max}	δ_{topo}
Value	512	512	64	64	6	8	10.0	1.6
Name	epoch	scheduler	lr	min_lr	σ_t	σ_r	max_n_vertex	max_vertex_per_gpu
Value	50	cosine	1.0×10^{-4}	1.0×10^{-5}	0.04	0.1	5,000	10,000

E More Results

E.1 Molecule Property Prediction

Setup We select QM9 Ramakrishnan et al. (2014) to evaluate the performance of EPT on small molecules. In detail, QM9 serves as a quantum chemistry benchmark that offers 12 chemical properties for each 3D molecule composed of C, H, O, N, and F elements. Following Thölke & De Fabritiis (2022), we randomly select 10,000 and 10,831 structures for validation and testing, and the remaining 110,000 structures are used to finetune the model. We use Mean Absolute Error (MAE) on each property to evaluate the model performance.

Baselines EPT is benchmarked against a range of 3D geometric models and pretraining approaches tailored to small molecules. For geometric GNNs, we include SchNet Schütt et al. (2018), E(n)-GNN Satorras et al. (2021), DimeNet++ Gasteiger et al. (2020), PaiNN Schütt et al. (2021), and the Transformer-based architectures TorchMD-Net Thölke & De Fabritiis (2022) and Equiformer Liao & Smidt (2022), which incorporate vector or higher-degree features. Pretraining comparisons are drawn from the work of Feng et al. (2023b), featuring GeoSSL Liu et al. (2022) and 3D-EMGP Jiao et al. (2023)—methods that apply denoising techniques to PaiNN and E(n)-GNN—as well as Transformer-M Luo et al. (2022), DP-TorchMD-Net Zaidi et al. (2022), and Frad (Feng et al., 2023b), which implement various denoising strategies on Transformer-based models.

Results We pretrain our model on the multi-domain dataset and present the results in Table 5. Following Zhang et al. (2023), we compute the average rank for each method over the 12 tasks to summarize the results succinctly. Our approach demonstrates superior or comparable performance to existing denoising-based methods, validating the effectiveness of block-level denoising. Additionally, we experiment with an augmented model, **EPT-10**, which comprises 10 layers against the original 6-layer setting. The enhanced results, as depicted in the last row of Table 5, affirm the scalability of our proposed pretraining paradigm.

E.2 Protein Property Prediction

Setup We evaluate EPT on protein-related tasks to verify its generalization on macro molecular systems: **Enzyme Commission** number prediction (EC) requires predicting the catalyst properties of proteins characterized by EC numbers, including 538 binary classification tasks; **Mutation Stability Prediction** (MSP) seeks to predict whether a point mutation at the interface of protein complexes leads to better binding affinity, formalized as a binary classification task. For EC, we follow (Zhang et al., 2022) to quantify the performance with F1 max and AUPRC, with size of training, validation, testing set as 15550, 1729, and 1919, respectively. For MSP, we follow Townshend et al. (2020) to

Table 5: MAEs on the QM9 dataset. The best results are in **bold** and the second best are $\underline{\text{underlined}}$. The right-most column provides the averaged rank of each method across 12 tasks.

Model	$\mu\downarrow$ (D)	$\alpha\downarrow$ (a_0^3)	$\epsilon_{\text{HOMO}}\downarrow$ (meV)	$\epsilon_{\text{LUMO}}\downarrow$ (meV)	$\Delta\epsilon\downarrow$ (meV)	$< R^2 >\downarrow$ (a_0^2)	ZPVE$\downarrow$ (meV)	$U_0\downarrow$ (meV)	$U\downarrow$ (meV)	$H\downarrow$ (meV)	$G\downarrow$ (meV)	$C_v\downarrow$ ($\frac{cal}{molK}$)	Avg.$\downarrow$ Rank
SchNet	0.033	0.235	41.0	34.0	63.0	$\underline{0.070}$	1.70	14.00	19.00	14.00	14.00	0.033	11.83
E(n)-GNN	0.029	0.071	29.0	25.0	48.0	0.110	1.55	11.00	12.00	12.00	12.00	0.031	11.17
DimeNet++	0.030	0.043	24.6	19.5	32.6	0.330	1.21	6.32	6.28	6.53	7.56	0.023	7.17
PaiNN	0.012	0.045	27.6	20.4	45.7	$\underline{0.070}$	1.28	5.85	5.83	5.98	7.35	0.024	6.33
TorchMD-Net	0.011	0.059	20.3	18.6	36.1	**0.033**	1.84	6.15	6.38	6.16	7.62	0.026	7.08
Equiformer	0.011	0.046	**15.0**	14.0	30.0	0.251	1.26	6.59	6.74	6.63	7.63	0.023	6.00
Transformer-M	0.037	$\underline{0.041}$	17.5	16.2	**27.4**	0.075	1.18	9.37	9.41	9.39	9.63	0.022	6.92
GeoSSL	0.015	0.046	23.5	19.5	40.2	0.122	1.31	6.92	6.99	7.09	7.65	0.024	8.42
3D-EMGP	0.020	0.057	21.3	18.2	37.1	0.092	1.38	8.60	8.60	8.70	9.30	0.026	8.83
DP-TorchMD-Net	0.012	0.052	17.7	14.3	31.8	0.450	1.71	6.57	6.11	6.45	6.91	**0.020**	6.67
Frad	**0.010**	**0.037**	15.3	$\underline{13.7}$	$\underline{27.8}$	0.342	1.42	**5.33**	$\underline{5.62}$	5.55	**6.19**	**0.020**	$\underline{3.17}$
EPT	0.011	0.045	16.2	14.1	29.6	0.122	$\underline{1.14}$	5.53	5.70	$\underline{5.52}$	6.42	**0.020**	3.33
EPT-10	**0.010**	0.045	$\underline{15.2}$	**13.6**	29.0	0.152	**1.11**	$\underline{5.44}$	**5.54**	**5.42**	$\underline{6.37}$	**0.020**	**2.33**

report AUROC on the split based on 30% sequence identity, where there are 2864, 937, and 347 samples for training, validation and testing.

Baselines We compare our EPT against baselines from Zhang et al. (2022) and Jing et al. (2021). Due to the space limit, we highlight representative models from various categories here: (1) traditional networks such as GCN (Kipf & Welling, 2016) and Atom3D-CNN (Townshend et al., 2020); (2) equivariant geometric GNNs including Atom3D-ENN (Townshend et al., 2020) and GVP (Jing et al., 2021); (3) GNNs talored for protein domain represented by GearNet and GearNet-Edge (Zhang et al., 2022); (4) sequence-based pretraining methods containing LM-GVP (Wang et al., 2022) and ProtBERT-BFD (Elnaggar et al., 2021b); (5) structure-based pretraining methods such as GearNet-Edge-MC (Zhang et al., 2022).

Results Table 6 illustrates that our EPT outperforms all baselines when trained from scratch, indicating its remarkable expressiveness in the protein domain. Notably, when compared to the pretrained models, our EPT pretrained on the multi-domain dataset surpasses the baselines on MSP by a large margin, emphasizing its capability of capturing shared physics across different molecular domains. Further, though pretrained on atomic view, EPT exhibits surprising knowledge transferability to residue-level graphs in the EC task, achieving comparable performance with GearNet-Edge-MC which is pretrained on the residue-level view of proteins.

Table 6: Results on Enzyme Commission number prediction (EC) and mutation stability prediction (MSP). The best scores are marked in **bold** and the second best $\underline{\text{underlined}}$.

	Model	EC		MSP
		F1 Max	AUPRC	AUROC
w/o Pretrain	GCN	0.320	0.319	0.621
	Atom3D-CNN	-	-	0.574
	Atom3D-ENN	-	-	0.574
	GVP	0.489	0.482	$\underline{0.680}$
	GearNet	0.730	0.751	-
	GearNet-Edge	$\underline{0.810}$	$\underline{0.835}$	0.633
	EPT (ours)	**0.823**	**0.844**	**0.700**
w/ Pretrain	LM-GVP	0.664	0.710	-
	ProtBERT-BFD	0.838	0.859	-
	GearNet-Edge	**0.874**	**0.892**	$\underline{0.646}$
	EPT (ours)	$\underline{0.858}$	$\underline{0.871}$	**0.741**

F Ablation Studies

Design of the backbone model. We provide a series of ablation studies on the QM9 dataset to elucidate the contribution of each component to the performance of our backbone model, as detailed in Table 7. Specifically, we explore the following aspects. **1.** We first substitute the input embedding delineated in Eq. (5) with a straightforward atom-level embedding, denoted as $\boldsymbol{f}_i = f_a(a_i)$. The findings suggest that enriching atom features with block-level information slightly improves model performance. **2.** In Eq. (10), we integrate the distance matrix $\mathbf{D}$ and edge features $\mathbf{R}$ into the attention

Table 7: Evaluation of each proposed component on the QM9 dataset. Best results are marked in **bold**.

	Block Emb.	Dist. Attn.	Edge Attn.	FFN	ϵ_{HOMO}	ϵ_{LUMO}
Ours	✓	✓	✓	✓	**19.2**	**17.4**
Block-level Embeddings						
Atom Emb. Only		✓	✓	✓	19.5	17.6
Attention Mechanism						
Standard Attn.	✓			✓	23.2	20.4
w/ Dist. Attn.	✓	✓		✓	22.3	19.4
w/ Edge Attn.	✓		✓	✓	20.4	17.7
Feed Forward Network						
w/o FFN	✓	✓	✓		28.2	26.6

mechanism. Eliminating either or both of these elements leads to a decline in performance, thereby underscoring their collective significance in effectively capturing the varying interatomic relations. **3.** In Eq. (13-15), we employ the FFN layer to amalgamate scalar and vector features. The resultant sharp drop after removing the FFN layer underscores the critical role of feature fusion in our model.

Comparison on pretraining settings. We evaluate the influence of pretraining datasets and denoising strategies on LBA and QM9 in Table 8. As an extension of Table 1, we observe a consistent trend where pretraining on one domain confers benefits to downstream tasks in another domain. Specifically, the model pretrained on small molecules demonstrates enhanced performance on the LBA tasks, while the model pretrained on proteins exhibits improved results on the QM9 benchmark. Furthermore, the model pretrained on the multi-domain dataset shows superior performance across all evaluated downstream tasks.

We further compare the three kinds of denoising strategies introduced in §2.3. The $\mathcal{L}_{\text{atom}}$ strategy focuses on atom-level denoising, showing superior results for small molecular structures on QM9. However, its benefits are less pronounced when applied to large, complex systems such as LBA. On the contrary, $\mathcal{L}_{\text{block-T}}$ adopts a more macroscopic approach by only considering the translations of each block's center of mass. This coarse-grained strategy improves performance for larger systems but tends to struggle in smaller molecules. Finally, our model, which accounts for both block-level rotations and translations, provides a comprehensive supervision for predicted forces, resulting in the optimal performance across all strategies.

Table 8: Comparison on the LBA and QM9 results from different pretraining settings. Best results are marked in **bold**.

Pretraining	PCC↑		MAE↓	
Setting	ID30	ID60	ϵ_{HOMO}	ϵ_{LUMO}
-	0.604	0.777	19.2	17.4
Dataset Domain				
Molecule Only	0.627	0.791	17.5	15.5
Protein Only	0.628	0.805	18.9	16.7
Denoising Strategy				
$\mathcal{L}_{\text{atom}}$	0.613	0.807	17.1	14.6
$\mathcal{L}_{\text{block-T}}$	0.636	0.817	17.5	14.8
Ours	**0.643**	**0.822**	**16.2**	**14.1**

G Implementation Details

G.1 LBA

We utilize the pretrained model as the encoder and additional apply an output head to predict the affinity. Specifically, we consider three types of output heads based on an MLP φ_E as follows:

$$\varphi_{\text{atom}}(\boldsymbol{H}^{(l)}) = \sum_i \varphi_E(\boldsymbol{h}_i^{(l)}), \tag{32}$$

$$\varphi_{\text{block}}(\boldsymbol{H}^{(l)}) = \sum_{m_i} \varphi_E\Big(\sum_{m_j=m_i} \boldsymbol{h}_j^{(l)})\Big), \tag{33}$$

$$\varphi_{\text{graph}}(\boldsymbol{H}^{(l)}) = \varphi_E(\sum_i \boldsymbol{h}_i^{(l)}). \tag{34}$$

Based on these heads, the hyperparameters for finetuning on LBA are provided in Table 9.

Table 9: Hyperparameters for finetuning on LBA.

Name	lr	batch_size	output_head	label_norm	epoch	save_topk	factor	patience	min_lr	omit_sml_pos
Sequence Identity 30										
EPT-Scratch	1.0×10^{-4}	16	graph	std	15	5	0.8	5	1.0×10^{-7}	True
EPT-Molecule	1.0×10^{-5}	16	block	mad	15	5	0.8	5	1.0×10^{-7}	True
EPT-Protein	1.0×10^{-5}	8	atom	std	20	5	0.8	5	1.0×10^{-7}	False
EPT-MultiDomain	1.0×10^{-4}	16	graph	mad	20	5	0.8	5	1.0×10^{-7}	False
Sequence Identity 60										
EPT-Scratch	1.0×10^{-4}	16	block	mad	30	5	0.8	5	1.0×10^{-7}	True
EPT-Molecule	5.0×10^{-5}	16	block	none	30	5	0.8	5	1.0×10^{-7}	True
EPT-Protein	5.0×10^{-5}	8	atom	none	30	5	0.8	5	1.0×10^{-7}	False
EPT-MultiDomain	5.0×10^{-5}	16	atom	none	30	5	0.8	5	1.0×10^{-7}	False

G.2 QM9

Following previous studies Zaidi et al. (2022); Feng et al. (2023b), we utilize the noisy node technique Godwin et al. (2021) by adding $\mathcal{L}_{\text{block-C}}$ as an auxiliary training objective, and the entire loss for finetuning on QM9 can be formulated as $\mathcal{L} = \mathcal{L}_{\text{MAE}} + \lambda\mathcal{L}_{\text{block-C}}$, where λ balances the weight of each term. We utilize the same hyperparameters for all 12 tasks, which are detailed in Table 10.

Table 10: Hyperparameters for finetuning on QM9.

Name	lr	batch_size	σ_t	σ_r	λ	epoch	save_topk	factor	patience	min_lr	omit_sml_pos
EPT	5.0×10^{-5}	64	0.04	0.1	0.1	1,000	1	0.8	15	1.0×10^{-7}	True
EPT-10	5.0×10^{-5}	64	0.04	0.05	0.1	1,000	1	0.8	15	1.0×10^{-7}	True

G.3 EC

Following Zhang et al. (2022), we integrate IEConv (Hermosilla et al., 2020) for better expressiveness on residue-level protein graph. Specifically, we insert an adapter layer after the attention module to update the invariant features $\boldsymbol{h}$ as follows:

$$\boldsymbol{h}_i' = \boldsymbol{h}_i + \sum_{j \in \mathcal{N}(i)} \phi_m(\phi_{e_1}(\boldsymbol{e}_{ij}) + \phi_{e_2}(\boldsymbol{e}_{ij}) \circ \phi_h(\boldsymbol{h}_i)), \tag{35}$$

where $\boldsymbol{e}_{ij}$ is the intrinsic-extrinsic edge features in (Hermosilla et al., 2020), $\mathcal{N}(i)$ denotes the neighborhood of node i, $\circ$ denotes element-wise multiplication, and $\phi_m, \phi_{e_1}, \phi_{e_2}, \phi_h$ are 2-layer MLPs. To fully utilize the adapted model, we further post-pretrain the model with pretrained original parameters and randomly initialized adapters. The model is post-pretrained on an integrated dataset comprising PDB Berman et al. (2000) and AlphaFoldDB Varadi et al. (2022). During this phase, we employ block-level denoising and masked prediction as the post-pretraining tasks. We provide the hyperparamters for post-pretraining in Table 11, and for finetuning on EC in Table 12.

Table 11: Hyperparameters for post-pretraining.

Name	lr	epoch	scheduler	min_lr	σ_t	σ_r	mask_ratio	max_vertex_per_gpu
Value	1.0×10^{-4}	30	cosine	1.0×10^{-5}	0.1	0.1	0.15	10000

Table 12: Hyperparameters for finetuning on EC.

Name	lr	batch_size	epoch	save_topk	factor	patience	min_lr
EPT (w/o pretrain)	5.0×10^{-5}	8	200	1	0.6	5	5.0×10^{-6}
EPT (w/ pretrain)	5.0×10^{-5}	8	200	1	0.6	5	5.0×10^{-6}

G.4 MSP

We use the split by sequence identity over 30% provided by Atom3D (Townshend et al., 2020), and extract all residues within 6Å distance to the mutation point as the local view for input, where the distance between two residues is measured by the minimum distance between atom pairs. We further present the mean and standard deviation across three rounds in Table 13. The hyperparameters for finetuning MSP is listed in Table 14.

Table 13: Mean and standard deviation across 3 runs on mutation stability prediction (MSP). The best scores are marked in **bold** and the second best underlined.

	Model	AUROC
w/o Pretrain	GCN	0.621 ± 0.009
	Atom3D-CNN	0.574 ± 0.005
	Atom3D-ENN	0.574 ± 0.040
	GVP	0.680 ± 0.015
	GearNet-Edge	0.633 ± 0.067
	EPT (ours)	$\mathbf{0.700 \pm 0.017}$
w/ Pretrain	GearNet-Edge	0.646 ± 0.006
	EPT (ours)	$\mathbf{0.741 \pm 0.007}$

Table 14: Hyperparameters for finetuning on MSP.

Name	lr	batch_size	epoch	save_topk	factor	patience	min_lr
EPT (w/o Pretrain)	1.0×10^{-5}	16	10	5	0.6	5	5.0×10^{-6}
EPT (w/ Pretrain)	1.0×10^{-5}	4	10	5	0.6	5	5.0×10^{-6}

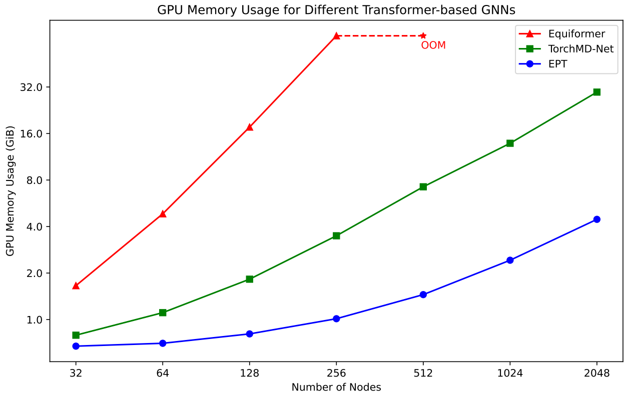

Figure 3: GPU memory usage comparison of three Transformer-based backbones as the number of nodes increases. EPT demonstrates a more memory-efficient scaling behavior compared to TorchMD and EquiFormer. EquiFormer encounters an Out-of-Memory (OOM) error at 512 nodes on a NVIDIA Tesla A800 with 80G GPU memory.

H Memory Efficiency

H.1 Efficient Attention Mechanism

Consider the attention mechanism introduced in Eq. (9-12), the query, key matrix $\boldsymbol{Q}_s, \boldsymbol{K}_s$ and the concatenated value matrix $\boldsymbol{V}_s$ share the shape of $\mathbb{R}^{B \times N_{\max} \times S \times 4h_s}$, where $B, N_{\max}, S, h_s$ denote the batch size, the maximum number of atoms, the number of heads and the size of each head's hidden state. Such shape consistency enables the usage of previous memory efficient techniques Lefaudeux et al. (2022) that provide the interface of attention biases for $\boldsymbol{D}$ and $\boldsymbol{R}$. We provide the pseudo codes in PyTorch Paszke et al. (2019) style as follows.

```
from xformers.ops import memory_efficient_attention

def equivariant_memory_efficient_self_attention(H_in, V_in, D, R, mask):

    ```
    Params:
        H_in: B * N_max * h
        V_in: B * N_max * 3 * h
        D: B * N_max * N_max
        R: B * N_max * N_max
        mask: B * N_max
    Returns:
        H_out: B * N_max * h
        V_out: B * N_max * 3 * h
    ```
    # Eq. (6)
    Query_s = linear_scalar_Q(H_in).view(B, N_max, S, h_s * 4)
    Key_s = linear_scalar_K(H_in).view(B, N_max, S, h_s * 4)
    Value_s_scalar = linear_scalar_K(H_in).view(B, N_max, S, h_s)
    Value_s_vector = linear_scalar_K(V_in).view(B, N_max, 3, S, h_s)
    # B * N_max * S * 3h_s
    Value_s_vector = Value_s_vector.transpose(2,3).flatten(start_dim=-2)
    # B * N_max * S * 4h_s
    Value_s = cat([Value_s_scalar, Value_s_vector], dim=-1)
    # Eq. (7)
    bias = R - D
    bias = bias.masked_fill(mask.unsqueeze(1).unsqueeze(2) == 0, float("
    inf"))
    bias = bias.expand(-1, S, -1, -1)
    HV = memory_efficient_attention(
        query = Query_s,
        key = Key_s,
        value = Value_s,
        attn_bias = bias
    )
    H_s = HV[..., :h_s]
    V_s = HV[..., h_s:].view(B, N_max, S, 3, h_s).transpose(2,3)
    # Eq. (9)
    H_out = linear_scalar_O(H_s.view(B, N_max, h))
    V_out = linear_vector_O(V_s.view(B, N_max, 3, h))
    return H_out, V_out
```

H.2 Comparison on Transformer-based Backbones

We further compared the GPU memory consumption of our model, a 6-layer, 512-hidden EPT, with two previous Transformer-based backbones: the 6-layer, 512-hidden TorchMD-Net (Thölke & De Fabritiis, 2022) and the 6-layer, 128-hidden, 3-degree Equiformer (Liao & Smidt, 2022), which have 31M, 19M, and 18M parameters, respectively. Our tests measured memory usage of one forward step on point clouds with 32 to 2048 nodes, sampled uniformly within a sphere of radius $\sqrt[3]{N}$, and connected by edges within a 4.0 cutoff distance. Figure 3 illustrates that EPT is consistently more memory-efficient across various node counts. This efficiency enables our model to effectively process large-scale point clouds, facilitating the study of expansive molecular systems.