Pi-SAGE: Permutation-invariant surface-aware graph encoder for binding affinity prediction

Sharmi Banerjee [1] **Mostafa Karimi** [1] **Melih Yilmaz** [1] **Tommi Jaakkola** [2] **Bella Dubrov** [1] **Shang Shang** [1] **Ron Benson** [1]

Abstract

Protein surface fingerprint encodes chemical and geometric features that govern protein–protein interactions and can be used to predict changes in binding affinity between two protein complexes. Current state-of-the-art models for predicting binding affinity change, such as GearBind, are all-atom based geometric models derived from protein structures. Although surface properties can be implicitly learned from the protein structure, we hypothesize that explicit knowledge of protein surfaces can improve a structure based model's ability to predict changes in binding affinity. To this end, we introduce Pi-SAGE, a novel Permutation-Invariant Surface-Aware Graph Encoder. We first train Pi-SAGE to create a protein surface codebook directly from the structure and assign a token for each surface exposed residue. Next, we augmented the node features of the GearBind model with surface features from domain adapted Pi-SAGE to predict binding affinity change on the SKEMPI dataset. We show that explicitly incorporating local, context-aware chemical properties of residues enhances the predictive power of all-atom graph neural networks in modeling binding affinity changes between wild-type and mutant proteins.

1. Introduction

A protein's surface encodes critical chemical and geometric fingerprints such as charge, shape, and hydrogen bond interactions that enables tasks like identifying active binding sites, designing proteins with specific properties, predicting ligand–protein binding affinity Gainza et al. (2020); Song et al. (2024); Lee & Kim (2023); Somnath et al. (2021) and so on. The seminal MaSIF paper Gainza et al. (2020) demonstrated that these fingerprints can be extracted from protein structures and used efficiently in downstream tasks such as binding site prediction, protein–ligand interaction modeling, and binding site search in protein complexes. Subsequently studies have reinforced the importance of explicitly modeling surface-level chemical and geometric features to perform tasks such as protein function prediction Somnath et al. (2021) and surface property guided protein design Lee & Kim (2023); Song et al. (2024).

In parallel, recent studies have proposed structure-aware protein language models (PLMs) by incorporating 3D structural information and showed that explicit structure information combined with sequence information improves performance across various predictive tasks. Su et al. (2023); Li et al. (2024). Most notably, GearBind Cai et al. (2024), an all-atom geometric neural network model outperformed SOTA models on the binding affinity prediction problem.

Motivated by these advances, we hypothesize that while protein surface information may be implicitly learned from structural models, explicitly modeling the surface provides additional inductive bias—especially for tasks such as predicting binding affinity changes at the protein–protein interface. To test our hypothesis, we propose Pi-SAGE: a Permutation-invariant Surface-Aware Graph Encoder that explicitly creates a vocabulary of protein surface from local geometric and chemical features of residues. Pi-SAGE is pre-trained in two stages: first on the 200k RCSB PDB database Burley et al. (2023) to capture general surface representations, and then on the SKEMPI dataset Jankauskaitė et al. (2019), which contains 6k binding affinity data for wild-type and mutated protein complexes. The model learns from residue graphs, where each node encodes a local surface "snippet" constructed from neighboring atoms' geometric and chemical properties. We train GearBind Cai et al. (2024), all-atom model by augmenting the one-hot residue features with these surface-aware features and demonstrate improved accuracy in predicting $\Delta\Delta G$ (binding affinity changes). Pi-SAGE outperforms both large-scale sequence-based models that attempt to learn structure implicitly as

[1]Amazon, Seattle, WA, USA [2]Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Sharmi Banerjee <sharmiba@amazon.com>.

Proceedings of the Workshop on Generative AI for Biology at the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

well as existing structure-aware models. In summary, our main contributions are as follows.

- We introduce a novel surface-aware vocabulary that builds a protein surface codebook from local geometric and chemical residue-level features.
- We pre-train Pi-SAGE in two stages—on the RCSB PDB and SKEMPI databases—to capture both general and task-specific surface information.
- We empirically demonstrate that augmenting the GearBind model with explicit surface features improves its ability to predict binding affinity changes ($\Delta\Delta G$), outperforming both sequence-only and existing structure-based approaches.

2. Related Work

In recent years, general protein models trained on millions of sequences have rapidly advanced, with most adopting the transformer architecture (Vaswani et al., 2017; Rao et al., 2020; Elnaggar et al., 2021; Madani et al., 2020). These models, trained on masked language modeling (MLM), have shown that protein structure can be implicitly learned and applied to downstream tasks like contact map and secondary structure prediction, as well as solubility and cellular localization (Rao et al., 2020; Elnaggar et al., 2021). Model capacity and pretraining strategies have expanded, including span masking in ProtT5 (3B/11B) (Elnaggar et al., 2021), blank-filling in xTrimoPGLM (100B) (Chen et al., 2024), and multi-task learning in AminoBERT (Bouatta, 2022).

The release of 200M predicted structures by AlphaFold DB (Varadi et al., 2022) spurred development of structure-aware models such as ProtT5-XL-UniRef50-Structure (Heinzinger et al., 2023) and SaProt (Su et al., 2023), which combine sequence and structure inputs and outperform sequence-only models on tasks like contact prediction, thermostability, and protein–protein interaction (PPI) prediction (Meier et al., 2021; Xu et al., 2022).

In parallel, surface-based representations have gained traction due to their ability to capture functionally relevant chemical and geometric fingerprints. MaSIF (Gainza et al., 2020) pioneered extracting five such features from protein surfaces and used geometric CNNs for binding site and ligand pocket prediction. Follow-up work proposed continuous surface representations (SurfPro) (Song et al., 2024), multi-view integration of sequence, structure, and surface properties (HoloProt) (Somnath et al., 2021), and surface-based masked autoencoders (Surface-VQMAE) (Wu & Li, 2024).

Building on these insights, we propose learning a quantized surface-aware vocabulary that encodes local chemical and geometric fingerprints of surface residues—analogous to the structure-aware vocabulary in SaProt. We integrate these surface features into the GearBind framework (Cai et al., 2024) and show that explicit surface context improves performance on tasks such as binding affinity prediction.

3. Methods

3.1. Featurization based on local context for surface residues

Given a protein structure, our approach represents each residue based on its local neighborhood in structural space, explicitly computing its chemical and geometric properties. We created local, context-aware surface features for each residue, which are then used to encode the residue graph and train a model to learn a codebook for protein surface. Initially, we processed each protein structure using the MaSIF Gainza et al. (2020). MaSIF decomposes a surface into overlapping radial patches (which consists of three vertices) with a fixed geodesic radius, where each vertex is assigned five features: electrostatic potential (charge), hydrophobicity, hydrogen bond interaction propensity, shape index, and distance-dependent curvature. To enhance this representation, we introduced two additional geometric features: (1) the distance from the patch centroid to the residue's C_α atom $C_\alpha - > centroid$ and (2) the angle between the vectors from the patch centroid to the C_α atom and from the C_α atom to the C_β atom $C_\alpha - > C_\beta$. For residues lacking a side chain, we generated a virtual C_β atom following the method described in FoldSeek van Kempen et al. (2022). These two new measurements constitute the geometric features in our model. Finally, we averaged the chemical features across three vertices within a patch, resulting in a 5-dimensional chemical feature vector. Combined with the two geometric features, this yields a 7-dimensional feature vector per patch.

We modified the MaSIF processing code to output a vertex-to-residue mapping, enabling accurate feature computation for each surface-exposed residue. For a given residue A, we first selected all patches that are within 3Å from any of its atoms. As not all of the patches will be mapped to the residue (as per MaSIF calculation) we then categorized the patches into three groups: (1) *Core* — patches where all three vertices map to atoms from residue A, (2) *Border* — patches where at least one vertex maps to an atom from another residue, and (3) *Borrowed* — patches where no vertices map to atoms from residue A. Borrowed patches are filtered out for residue A. We assigned a label to each patch: (1) *High* — if its closest atoms include $\{C, O, N, S\}$ or all heavy atoms from its assigned residue, (2) *Medium* — if its closest atoms include $\{C, O, N, S\}$ or all heavy atoms from neighboring residue , and (3) *Low* — for patches whose closest atoms are all hydrogen atoms. We

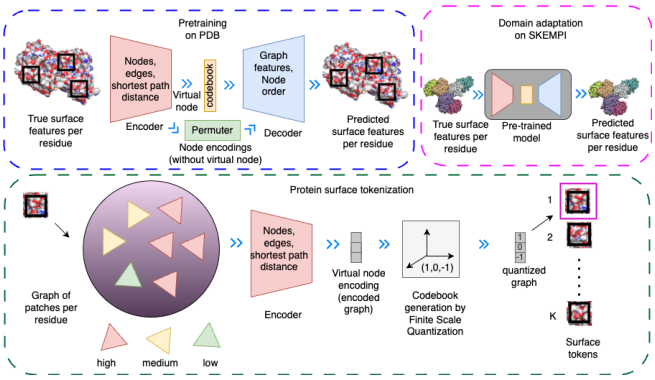

Figure 1. Pi-SAGE is trained in two stages, 1) on the RSCB database with 200K protein structures (top left), 2) on SKEMPI database with 6k wild type and mutated complexes from 340 unique protein complexes (top right). Each residue is represented by a graph of high, medium and low patches that are first fed to the encoder, followed by the quantizer that creates a codebook for the residue (bottom).

represented each residue as a graph $G = (V, E)$, where the nodes $V = \{n_i\}_{i=1:N}$ correspond to N randomly sampled patches following 70% from high, 20% from medium and 10% from low patches. We chose N as 32 following MaSIF Gainza et al. (2020) in their tasks and chose majority of the patches that are most closely tied to the residue, followed by those that contain side-chain atoms of the neighboring residues and finally the last fraction consisting of only hydrogen atoms from its own or neighboring residue. When reconstructing the node features we weighed the nodes according to this classification. The above-mentioned 7-dimensional features are used as node features. The patch types are illustrated in Figure 4.

The edges $E = \{e_{ij}|j \in \mathcal{N}_i\}_{i=1:N}$ are defined where $\mathcal{N}_i = \{j|\text{dist}(n_i, n_j) < 3\text{Å}\}$ is a set of neighbors of a node n_i and $\text{dist}(.,.)$ is defined as the distance between the patch centroids of nodes n_i and n_j. In addition, a virtual node has been added to the graph that is connected to every other node in the graph through its special virtual edge. This virtual node will serve a similar purpose as the [CLS] token in transformers. Since the [CLS] token has been commonly utilized to provide sentence embedding, we used the virtual node to calculate the final graph embedding and its tokenized representation. Edges are featurized according to the centroid distances: (1) *Short* — where their distance is less than 1Å , (2) *Medium* — where their distance is between 1Å and 2Å , (3) *Long* — where their distance is between 2Å and 3Å (4) *Virtual* — edge between virtual node and any other node, (5) *Self* — edge connecting each node to itself, and (6) *No* — nodes that are not connected to each other. Therefore, there are six categories of edges. Inspired by (Park et al., 2022), we have used the topological relationship $\psi(i, j)$ between nodes n_i and n_j based on their shortest path distance with the maximum cutoff *max-hop*. Formally $\psi(i, j)$ is featurized as following: (1) *Unreachable* — No connection between two nodes, (2) *shortest path distance* s — Shortest path distance value $s \in \{0, \cdots, \textit{max-hop}\}$ between nodes n_i and n_j, (3) *Far distance* — If the shortest path distance is greater than *max-hop*, and (4) *Virtual* — edge between virtual node and any other node. Therefore, there are *max-hop* + 4 categories of topological relation.

3.2. Surface Aware Graph Encoding

We developed two approaches for surface-aware graph encoding (1) *naive* approach named SAGE (Surface Aware Graph Encoding) where we only reconstructed node features provided by the adjacency matrix at both encoder and decoder modules, (2) Pi-SAGE where we reconstructed both nodes and edges simultaneously. We introduced permutation-invariance property in our modeling in order to properly align reconstructed node features and adjacency matrix. Both models have similar (1) graph encoder g_{enc}, (2) finite scale quantized protein surface tokenizer (FSQ) while their decoder g_{dec} is different. In addition, Pi-SAGE has an additional module named *permuter* p_{perm} to learn the alignment between the input graph and the reconstructed one. We described each of these components in the following subsections.

3.2.1. Graph Encoder

We used the graph transformer with learnable relative positional encoding developed by (Park et al., 2022) that use (1) dot-product attention commonly used in transformers, (2) learnable topological relationship $\mathcal{P}_{\psi(i,j)} \in \mathbb{R}^{d_z}$ between nodes n_i and n_j, and (3) learnable edge relationship $\mathcal{E}_{(i,j)} \in \mathbb{R}^{d_z}$ between nodes n_i and n_j. Let us assume $x_i \in \mathbb{R}^{d_x}$ denotes the input feature of the node n_i with d_x as its dimension, and $z_i \in \mathbb{R}^{d_z}$ denotes the final output feature of transformer's layer with d_z. First, self-attention module computes query q_i, key k_i, and value v_i with independent linear transformations $W^{\text{query}} \in \mathbb{R}^{d_x \times d_z}$, $W^{\text{key}} \in \mathbb{R}^{d_x \times d_z}$ and $W^{\text{value}} \in \mathbb{R}^{d_x \times d_z}$.

$$q_i = W^{\text{query}} x_i, k_i = W^{\text{key}} x_i \; v_i = W^{\text{value}} x_i \tag{1}$$

Second, topological relationship between nodes n_i and n_j is calculated as:

$$a^{\text{topology}}_{(i,j)} = q_i \mathcal{P}^{\text{query}}_{\psi(i,j)} + k_i \mathcal{P}^{\text{key}}_{\psi(i,j)} \tag{2}$$

Next, edge relationship between nodes n_i and n_j is calculated as:

$$a^{\text{edge}}_{(i,j)} = q_i \mathcal{E}^{\text{query}}_{(i,j)} + k_i \mathcal{E}^{\text{key}}_{(i,j)} \tag{3}$$

Finally, the overall attention map is computed summing these three terms. Attention here denotes full pairwise attention between the nodes adjusted by the graph features from the two additional matrices.

$$\begin{aligned} a_{(i,j)} &= \frac{q_i.k_j + a^{\text{topology}}_{(i,j)} + a^{\text{edge}}_{(i,j)}}{\sqrt{d_z}}, \\ \hat{a}_{(i,j)} &= \frac{\exp(a_{(i,j)})}{\sum_{k=1}^{N} \exp(a_{(i,k)})} \end{aligned} \tag{4}$$

The overall attention module outputs the next hidden feature by applying weighted summation on the values

$$z_i = \sum_{j=1}^{N} \hat{a}_{(i,j)} v_j \tag{5}$$

The utilized learnable relative positional encoding can be seen as an alterative to linearizing graphs, thus enabling richer node-topology and node-edge interactions since it preserves structural graph information.

3.2.2. Surface tokenizer

We adopted Finite Scale Quantization Mentzer et al. (2023) to create protein surface codebook. FSQ creates a simple, fixed grid partition in a lower-dimensional space. Let us assume the FSQ's internal dimension is represented as d_{FSQ} and the i^{th} dimension can have L_i different integers or *levels*. Therefore, overall *implicit* codebook size for FSQ with $\{\text{L}_1, \cdots, \text{L}_{d_{\text{FSQ}}}\}$ can be $|\mathcal{C}| = \prod_{i=1}^{d_{\text{FSQ}}} \text{L}_i$. FSQ module takes in the virtual node of a residue graph from the encoder $z_{\text{graph}} \in \mathbb{R}^{d_z}$, down-project the graph representation down to d_{FSQ} dimension through $z_{\text{latent}} = \text{MLP}(z_{\text{graph}}) \in \mathbb{R}^{d_{\text{FSQ}}}$. Then, non-differentiable online quantization step occurs for each dimension i through $z_{\text{FSQ},i} = \text{round}(\lfloor \text{L}_i/2 \rfloor \tanh(z_{\text{latent},i}))$. The quantization step will bound the encoder output to L values, which is the number of dimensions of the quantizer, and then rounding to integers, leading to quantized codebook. Since $\text{round}(.)$ function is a non-differential operation, straight-through estimator (STE) (Bengio et al., 2013) can be used to propagate gradient through $\text{round\_ste}(x) = x + \text{stop\_gradient}(\text{round}(x) - x)$.

3.2.3. Permutation invariance

The nodes in the original residue graph do not have any positional encoding and their order is arbitrary but fixed during training. Inferring this order during training allows the decoder to align the nodes which would help in the feature loss calculation. Inspired by Winter et al. (2021), we added a permuter module to reconstruct the residue graph with node features and adjacency matrix in Pi-SAGE. The permuter module learns to align input and output graph through *soft* alignment. Note that in Pi-SAGE, patches from residues form un-directed graphs with the adjacency matrix $\mathbf{A}_\pi \in \{0,1\}^{n \times n}$ for $n \in N$ in the node order $\pi \in \Pi$ with Π is the set of all permutations over V. We defined a permutation matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$ that reorders nodes from order π to order π' as $\mathbf{P}_{\pi \to \pi'} = (p_{ij}) \in \{0,1\}^{n \times n}$ with $p_{ij} = 1$ if $\pi_i = \pi'_j$ and $p_{ij} = 0$ otherwise.

Input to the permuter are the node encodings $N \times \mathbb{R}^{d_z}$ obtained from the output the encoder module. We discard the virtual node at this step and do not try to reconstruct it. The permuter module has to learn how the ordering of nodes in the graph generated by the decoder model will differ from a specific node order present in the input graph. During the learning process, the decoder will learn its own canonical ordering so that, given a latent code z_{latent}, it will always reconstruct a graph in that order. The permuter learns to transform/permute this canonical order to a given input node order. For each node i of the input graph, the permuter predicts a score s_i corresponding to its probability of having a low node index in the decoded graph. By sorting the input nodes indices by their assigned scores, we inferred the output node order and constructed the corresponding permutation matrix $\mathbf{P}_{\pi \to \pi'} = (p_{ij}) \in \{0,1\}^{n \times n}$ with

$$p_{ij} = \begin{cases} 1, & \text{if } j = \text{argsort}(s)_i \\ 0, & \text{else} \end{cases} \tag{6}$$

to align input and output node order. The argsort operation being non-differentiable, the continuous relaxation of the argsort operator proposed in Prillo & Eisenschlos (2020); Grover et al. (2019) has been used as follows

$$\mathbf{P} \approx \hat{\mathbf{P}} = \text{softmax}(\frac{-d(\text{sort}(s)\mathbb{1}^\top, \mathbb{1}s^\top)}{\tau}) \tag{7}$$

where the softmax operator is applied row-wise, $d(x,y)$ is the L_1-norm and $\tau \in \mathbb{R}_+$ a temperature parameter.

3.2.4. GRAPH DECODER

Since the quantized graph encoding from the FSQ module is in d_{FSQ}, we used a simple linear layer to project it back to the d_z embedding that serves the purpose of node features for the decoder denoted z_{dec}. Inspired by Winter et al. (2021), we defined sinusoidal positional embedding $\text{PE} \in \mathbb{R}^{N \times d_z}$ with the i-th node's embedding for k-th dimension as follows:

$$\text{PE}(i)_k = \begin{cases} \sin(i/10000^{2k/d_z}), & \text{if k is even} \\ \cos(i/10000^{2k/d_z}), & \text{k is odd} \end{cases} \tag{8}$$

Then we used the learned permutation matrix $\hat{\mathbf{P}}$ to reorder the positional embedding by multiplication $\text{PE}_{\text{update}} = \hat{\mathbf{P}} \times \text{PE}$. Finally, we concatenated the node features of the decoder with the updated positional encoding and passed them to the graph decoder. The graph decoder exactly follows the graph encoder with minor differences at the final project layers:

$$\begin{aligned} z_o &= g_{\text{dec}}([z_{\text{dec}}||(PE)_{\text{update}}]) \\ \hat{m}_{\text{node}} &= W_{\text{node}} z_o + b_{\text{node}} \\ \hat{m}_{\text{edge}} &= W_{\text{edge}} z_o + b_{\text{edge}} \end{aligned} \tag{9}$$

where $\hat{m}_{\text{node}}$ is used to reconstruct the initial node features m_{node} and $\hat{m}_{\text{edge}}$ is used to reconstruct the un-directed adjacency matrix $\mathbf{A}_\pi$.

3.2.5. LOSSES

Following (Yang et al., 2024) we defined node and edge reconstruction as

$$\begin{aligned} \mathcal{L}_{\text{rec}} = \frac{1}{N} \sum_{i=1}^{N} (1 - \frac{m_{\text{node}}^T \hat{m}_{\text{node}}}{||m_{\text{node}}||.||\hat{m}_{\text{node}}||} \\ + ||\mathbf{A}_\pi - \sigma(\hat{m}_{\text{edge}}.\hat{m}_{\text{edge}}^T)||^2) \end{aligned} \tag{10}$$

where $\sigma(.)$ is the sigmoid function. In order to push the *soft* permutation matrix towards a real permutation matrix (i.e. contains one 1 in every row and column), an additional penalty term was introduced to minimize the Shannon entropy both row-wise and column-wise:

$$\text{C}(\hat{\mathbf{P}}) = \sum_i \text{H}(\bar{\mathbf{p}}_i) + \sum_j \text{H}(\bar{\mathbf{p}}_j) \tag{11}$$

with Shannon entropy $\text{H}(x) = -\sum_i x_i \log(x_i)$ and normalized probabilities $\bar{\mathbf{p}}_i, = \frac{\hat{\mathbf{P}}_i}{\sum_j \hat{\mathbf{P}}_{i,j}}$.

The final loss would be:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda \text{C}(\hat{\mathbf{P}}) \tag{12}$$

where λ hyper-parameter would balance between main reconstruction loss and the additional penalty.

3.3. Graph-based binding affinity prediction

We tested the hypothesis that chemical and geometric fingerprints obtained from protein surface contain information complementary to structure to improve protein binding affinity prediction. To test it, we adopted the GearBind Cai et al. (2024) architecture based on multi-level geometric message passing network, augmented the one-hot residue features from the residue graph (obtained by pooling the atom level features of the graph after the attention step in GearBind) with the residue graph embeddings z_{graph} from the fine-tuned surface tokenizer and trained the augmented feature GearBind model on the SKEMPI dataset Jankauskaitė et al. (2019). Since the SKEMPI dataset contains both single and multiple mutations to the wild type protein complexes, we domain adapted the pre-trained surface tokenizer on these 6k proteins. This step ensured that the surface tokenizer model had seen the distribution of the mutational dataset with the protein complexes. In addition to surface tokenizer we trained 3 ESM2 Verkuil et al. (2022) models: ESM2-150M, ESM2-650M, ESM2-3B, th 3B ProstT5 model Elnaggar et al. (2021), the 3B ProstT5 Heinzinger et al. (2024) model and the SaProt model Su et al. (2023).

4. Experiments

We performed three stage training to predict binding affinity change on SKEMPI dataset using four different sizes of surface tokenizer models with five different vocabulary sizes. In the pre-training stage, we trained on the $200K$ experimentally validated protein structures in Mentzer et al. (2023) RSCB dataset. We removed the SKEMPI protein complexes from the database and randomly split the proteins into 90% train and 10% validation splits. We used a learning rate of $2e-04$, Adam optimizer with a per GPU batch size of 32 on a single P5 NVIDIA H200 Tensor Core GPU instances with a global batch size of 256 for 20 epochs. We adopted the implementation of the graph encoder module (SAGE and Pi-SAGE) from the GRPE GitHub Park et al. (2022), the FSQ quantizer part from Lucidrains GitHub Wang (2024) and the permutation invariant part (permuter module and the graph decoder for Pi-SAGE) from Winter et al. (2021) GitHub. For SAGE the decoder module has the same architecture as the encoder module. We followed the seminal FSQ paper Mentzer et al. (2023) to select different vocabulary sizes and hidden dimensions. For collating graphs in batches we used the DGL library Wang et al. (2019) and followed the examples from their GitHub. We trained SAGE and Pi-SAGE separately using the same batch size, learning rate and the number of epochs.

In the second stage, we domain adapted the pre-trained

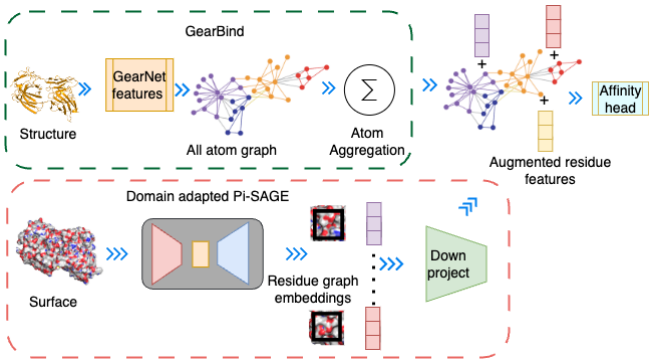

Figure 2. Training flowchart of GearBind model with domain adapted Pi-SAGE. Once the all-atom graph is constructed and reduced to the residue level graph, the one-hot residue features are augmented with surface residue embeddings to predict $\Delta\Delta G$.

models by further pre-training the tokenizers on the 6k wild type and mutated SKEMPI protein complexes from 340 unique protein complexes. Finally, we performed supervised fine tuning on the SKEMPI dataset with the GearBind model to predict binding affinity change or $\Delta\Delta G$ between wild type and mutated protein complexes. We trained GearBind using the same architecture as mentioned in the GitHub repository with 4 geometric graph convolution layers with 128 hidden dimension and using residual attention at the final layer. We used Rosetta Rohl et al. (2004) to generate the mutated protein structures from the wild-type protein complexes. We obtained 3-fold cross-validation splits from the RDE paper Luo et al. (2023). The dataset is split into three folds by structure, each containing unique protein complexes that do not appear in other folds.

We reported the average metrics across three splits. We employed five metrics to assess the accuracy of binding affinity change predictions: Pearson and Spearman correlation coefficients, root mean square error (RMSE), mean absolute error (MAE), and area Under the receiver operating characteristic curve (AUROC). For per-structure metrics, we followed the approach of Luo et al. (2023) by organizing mutations according to their associated structures. Groups with fewer than ten mutation data points are excluded from this analysis. Correlation calculations are done independently for each structure, with two additional metrics: the average per-structure Pearson and Spearman correlation coefficients. Calculating AUROC involves classifying mutations according to the direction of their $\Delta\Delta$G values. For each of the baselines (ESM, ProstT5), we augmented the node features with residue embeddings after down projection.

5. Results

5.1. Pre-training

We pre-trained four different model sizes on five vocabulary sizes for both SAGE and Pi-SAGE (see Table 1. We performed hyper-parameter optimization experiments on the CATH 4.3 dataset Sillitoe et al. (2015) and used the learning rate, learning scheduler and weight decay from these experiments in pre-training on the RSCB database. As illustrated in Figure 6c and 6d, the total loss for Pi-SAGE begins higher than that of SAGE in the early stages of training but drops below SAGE's loss as training progresses. This is because the permuter loss in training mode is quite high at the beginning and then becomes 10^{-4} (Figure 6e and 6f) leading to the lower loss for Pi-SAGE version. We hypothesize that by requiring Pi-SAGE to figure out the whole residue graph (by reconstructing both node features and the adjacency matrix), it learns to better reconstruct the node features that is demonstrated with lower feature reconstruction loss (See figure 6a and 6b). The feature reconstruction loss also decreases with increases in both model and vocabulary sizes (See Figure 3 left panel). This suggests that, for a fixed model size, increasing the vocabulary yields finer-grained surface representations, while scaling both the encoder and decoder enhances the model's overall capacity.

Table 1. Different model sizes of Pi-SAGE

Model	#layers	#heads	hdim	#params
Small	2	2	512	13M
Medium	4	4	768	44M
Large	8	8	1024	134M
XLarge	16	16	1280	378M

5.2. Binding affinity prediction

The residue graph in GearBind is formed by pooling the atom features per-residue which contains protein structure information. We show in Table 2 that augmenting the one-hot node features of the residue graph with residue embeddings from large sequence, structure or surface aware models improves its ability to predict change in binding affinity. This suggests that each model contains comple-

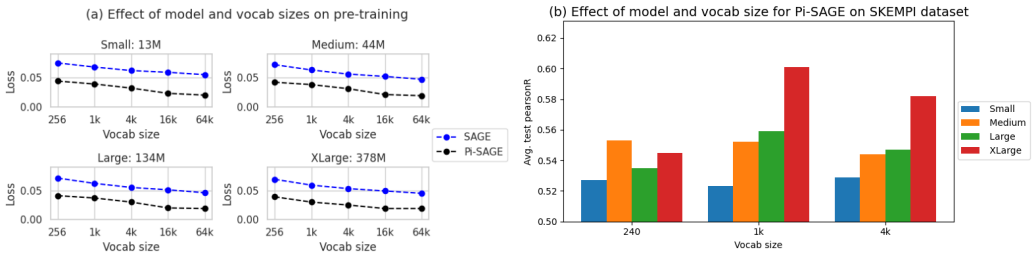

Figure 3. Effect of model and vocab sizes on pre-training and downstream task. Left: Loss curves for different model sizes and vocab sizes from pre-training the SAGE and Pi-SAGE on the RSCB database shows that with increase in model and vocab size total loss decreases. Right: Average pearson-R across three folds for different model and vocab sizes of Pi-SAGE

mentary information that improves the already powerful GearBind model's ability to predict $\Delta\Delta G$. For example, the large protein models like the ESM 3B model Rives et al. (2019)) that learn about protein structures implicitly improved the overall pearson R from 0.525 to 0.567 (Table 2 row 4). Structure aware models trained explicitly on protein structures like ProstT5 Heinzinger et al. (2023)) improved the overall pearson R from 0.525 to 0.545 (Table 2 row 7). Even within the same multi-modal protein model, one modality might outperform the other. For example, ProstT5 structure embeddings outperform ProstT5 sequence embeddings (Table 2 rows 6 and 7). Finally, as most mutations occur at the interface of two proteins for binding affinity changes, a smaller model like Pi-SAGE, trained to explicitly encode context-aware surface features of residues, consistently outperforms the larger sequence-only and sequence-structure models by improving the GearBind's prediction 0.525 to 0.6 on average across the test splits. We expect that Pi-SAGE performance would improve with increase in both data and model sizes.

6. Ablation studies

6.1. Effect of permutation-invariance

The difference between SAGE and Pi-SAGE is that in the former both the encoder and decoder modules are provided the shortest path distance matrix and the edge type matrix along with the node features for the encoder. While the decoder module reconstructs the node features of the residue graph, it does not need to explicitly learn the residue graph with the node connections. On the other hand, in Pi-SAGE the encoder receives the same three matrices but the decoder needs to reconstruct both the node features in a specific order (learned by the permuter module) and the adjacency matrix containing the node connections. We hypothesize that by explicitly learning the node connections through the adjacency matrix the feature reconstruction ability of the decoder in Pi-SAGE increases. This approach allows the tokenizer to capture complex spatial relationships and biochemical properties that are crucial to understanding protein function, while ensuring that the encoded representation remains consistent regardless of arbitrary node orderings in the input graph. As we show in Table 2 both overall average metrics and per structure metrics for Pi-SAGE is better than SAGE for the same model and vocab size.

6.2. Effect of scaling model and vocabulary sizes

Similar to our observation in pre-training, we noticed that increasing the model size improves performance on $\Delta\Delta G$e prediction task (Figure 3, right panel). However, we noticed that the model with 1K tokens in its vocabulary had a higher average Pearson r than both 240 and 4K tokens. We hypothesize that due to the small size of SKEMPI dataset, which has only 340 unique protein complexes (and 6K wild type and mutated complexes), larger vocabulary might not be adding more information for the model to improve the binding affinity change (the XLarge with 1k vocab size has the higest average pearson R of 0.6 shown in Figure 3, right panel compared to the XLarge model with 4k vocab size with an average pearson R of 0.58). But increasing the model size might still help capture the nuances of the surface properties of the interface and improve the prediction power.

6.3. Effect of further pre-training on in-distribution data

Pre-training the surface tokenizer on the RSCB database enables it to encode residue graphs of experimentally validated single chain and multi-chain proteins but it does not learn about single or multiple mutations for a protein complex. We tested whether further domain adapting the surface tokenizer on the mutated protein complexes improves the downstream task of predicting the binding affinity change. We trained GearBind with surface features from a pre-trained Pi-SAGE on RSCB dataset and showed that domain adaptation helps the model understand mutated protein complexes better than only pre-training with closed complexes (Table 3 row 1).

Table 2. Performance on SKEMPI dataset

Model	Per structure		Overall				
	Pearson ↑	Spear. ↑	Pearson ↑	Spear. ↑	RMSE ↓	MAE ↓	AUROC ↑
Gearbind	0.365 +/- 0.082	0.299 +/- 0.053	0.525 +/- 0.106	0.372 +/- 0.035	1.921 +/- 0.277	1.403 +/- 0.208	0.650 +/- 0.006
+ ESM150M	0.378 +/- 0.050	0.326 +/- 0.047	0.563 +/- 0.088	0.400 +/- 0.014	1.866 +/- 0.259	1.359 +/- 0.209	0.655 +/- 0.028
+ ESM650M	0.381 +/- 0.063	0.316 +/- 0.052	0.539 +/- 0.096	0.377 +/- 0.047	1.852 +/- 0.226	1.349 +/- 0.170	0.652 +/- 0.032
+ ESM3B	0.418 +/- 0.088	0.338 +/- 0.067	0.567 +/- 0.057	0.425 +/- 0.039	1.834 +/- 0.144	1.331 +/- 0.114	0.671 +/- 0.026
+ ProtT5	0.376 +/-0.112	0.325 +/- 0.080	0.551 +/- 0.088	0.400 +/- 0.056	1.873 +/- 0.179	1.375 +/- 0.135	0.665 +/- 0.019
+ ProstT5 (seq)	0.372 +/- 0.094	0.316 +/- 0.087	0.540 +/- 0.085	0.390 +/- 0.070	1.90 +/- 0.173	1.401 +/- 0.146	0.660 +/- 0.046
+ ProstT5 (struct)	0.400 +/- 0.076	**0.347 +/- 0.049**	0.545 +/- 0.092	0.408 +/- 0.032	1.953 +/- 0.190	1.436 +/- 0.137	0.662 +/- 0.020
+ SaProt	0.332 +/- 0.092	0.268 +/- 0.071	0.527 +/- 0.065	0.362 +/- 0.014	1.948 +/- 0.234	1.439 +/- 0.183	0.659 +/- 0.009
+ SAGE	0.386 +/- 0.082	0.314 +/- 0.068	0.546 +/- 0.114	0.383 +/- 0.039	1.864 +/- 0.246	1.350 +/- 0.176	0.660 +/- 0.013
+ Pi-SAGE	**0.423 +/- 0.091**	0.345 +/- 0.077	**0.600 +/- 0.084**	**0.428 +/- 0.038**	**1.817 +/- 0.241**	**1.306 +/- 0.200**	**0.691 +/- 0.026**

Table 3. Pi-SAGE ablation on SKEMPI dataset

Pi-SAGE	Per structure		Overall				
	Pearson ↑	Spear. ↑	Pearson ↑	Spear. ↑	RMSE ↓	MAE ↓	AUCROC ↑
- Finetune	0.386 +/- 0.071	0.321 +/- 0.052	0.549 +/- 0.101	0.400 +/- 0.048	1.883 +/- 0.191	1.355 +/- 0.134	0.67 +/- 0.025
+ Finetune	0.423 +/- 0.091	0.345 +/- 0.077	0.600 +/- 0.084	0.428 +/- 0.038	1.817 +/- 0.241	1.306 +/- 0.200	0.691 +/- 0.026
+ VQ	0.359 +/- 0.078	0.281 +/- 0.053	0.512 +/- 0.105	0.353 +/- 0.013	1.998 +/- 0.277	1.477 +/- 0.232	0.634 +/- 0.007

6.4. Effect of VQ vs. FSQ

For a given model size, the training time for FSQ remains the same whereas for VQ it increases with increase in vocab size. For example, with VQ, time to train a 44M model for one epoch increased from 9hrs for 4k vocab to 1 day for 16k vocab on a P5 instance with 8 GPUs. Consequently, we did not train any surface tokenizer with VQ beyond a vocabulary of 4K tokens (Table 3 row 3).

Conclusion

Current state-of-the-art models such as GearBind is an all-atom based geometric neural network for predicting binding affinity changes between wild type and mutated protein structures. We hypothesized that explicit knowledge of surface features will improve a structure based model's ability to predict binding affinity change. We proposed Pi-SAGE, a novel approach of creating a codebook for surface exposed residues from protein structure. At its core Pi-SAGE has a graph based encoder module to encode residue graphs, a Finite Scale Quantizer to create codebook, a permuter module to learn node order of the residue graph and a decoder module to reconstruct node features and adjacency matrix. We evaluated Pi-SAGE by augmenting the residue features of GearBind to predict $\Delta\Delta G$ on SKEMPI dataset and showed that explicit knowledge of surface features improved GearBind's prediction from 0.525 to 0.6 on average on the test set. These results prove our hypothesis that the surface residue features from Pi-SAGE contain information above and beyond what structure can provide and boost the affinity change prediction.

Impact Statement

We propose Pi-SAGE, a novel surface-aware, permutation-invariant graph encoder that explicitly captures protein surface features to enhance protein binding affinity prediction. By integrating Pi-SAGE into the state-of-the-art GearBind model, we demonstrate improved accuracy in predicting $\Delta\Delta G$ between wild-type and mutant proteins. This work highlights the value of incorporating explicit surface representations in geometric deep learning models, with implications for advancing protein design, and the broader field of computational biology. We acknowledge the complexity of the method and the need for further sensitivity analysis. We plan to perform ablation studies to test the robustness of the model.

References

Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Bouatta, N. Single-sequence protein structure prediction using language models and deep learning. In *APS March Meeting Abstracts*, volume 2022, pp. Z18–004, 2022.

Burley, S. K., Bhikadiya, C., Bi, C., Bittrich, S., Chao,

H., Chen, L., Craig, P. A., Crichlow, G. V., Dalenberg, K., Duarte, J. M., et al. Rcsb protein data bank (rcsb. org): delivery of experimentally-determined pdb structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic acids research*, 51(D1):D488–D508, 2023.

Cai, H., Zhang, Z., Wang, M., Zhong, B., Li, Q., Zhong, Y., Wu, Y., Ying, T., and Tang, J. Pretrainable geometric graph neural network for antibody affinity maturation. *Nature communications*, 15(1):7785, 2024.

Chen, B., Cheng, X., Li, P., Geng, Y.-a., Gong, J., Li, S., Bei, Z., Tan, X., Wang, B., Zeng, X., et al. xtrimopglm: unified 100b-scale pre-trained transformer for deciphering the language of protein. *arXiv preprint arXiv:2401.06199*, 2024.

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, 2021.

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

Grover, A., Wang, E., Zweig, A., and Ermon, S. Stochastic optimization of sorting networks via continuous relaxations. *arXiv preprint arXiv:1903.08850*, 2019.

Heinzinger, M., Weissenow, K., Sanchez, J. G., Henkel, A., Mirdita, M., Steinegger, M., and Rost, B. Bilingual language model for protein sequence and structure. *bioRxiv*, pp. 2023–07, 2023.

Heinzinger, M., Weissenow, K., Sanchez, J. G., Henkel, A., Mirdita, M., Steinegger, M., and Rost, B. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6(4):lqae150, 2024.

Jankauskaitė, J., Jiménez-García, B., Dapkūnas, J., Fernández-Recio, J., and Moal, I. H. Skempi 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics*, 35(3):462–469, 2019.

Lee, Y. and Kim, J. Shapeprot: Top-down protein design with 3d protein shape generative model. *bioRxiv*, pp. 2023–12, 2023.

Li, M., Tan, Y., Ma, X., Zhong, B., Yu, H., Zhou, Z., Ouyang, W., Zhou, B., Hong, L., and Tan, P. Prosst: Protein language modeling with quantized structure and disentangled attention. *bioRxiv*, pp. 2024–04, 2024.

Luo, S., Su, Y., Wu, Z., Su, C., Peng, J., and Ma, J. Rotamer density estimator is an unsupervised learner of the effect of mutations on protein-protein interaction. *bioRxiv*, pp. 2023–02, 2023.

Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S., and Socher, R. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

Mentzer, F., Minnen, D., Agustsson, E., and Tschannen, M. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

Park, W., Chang, W., Lee, D., Kim, J., and Hwang, S.-w. Grpe: Relative positional encoding for graph transformer. *arXiv preprint arXiv:2201.12787*, 2022.

Prillo, S. and Eisenschlos, J. Softsort: A continuous relaxation for the argsort operator. In *International Conference on Machine Learning*, pp. 7793–7802. PMLR, 2020.

Rao, R., Meier, J., Sercu, T., Ovchinnikov, S., and Rives, A. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pp. 2020–12, 2020.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

Rohl, C. A., Strauss, C. E., Misura, K. M., and Baker, D. Protein structure prediction using rosetta. In *Methods in enzymology*, volume 383, pp. 66–93. Elsevier, 2004.

Sillitoe, I., Lewis, T. E., Cuff, A., Das, S., Ashford, P., Dawson, N. L., Furnham, N., Laskowski, R. A., Lee, D., Lees, J. G., et al. Cath: comprehensive structural and functional annotations for genome sequences. *Nucleic acids research*, 43(D1):D376–D381, 2015.

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, Z., Huang, T., Li, L., and Jin, W. Surfpro: Functional protein design based on continuous surface. *arXiv preprint arXiv:2405.06693*, 2024.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pp. 2023–10, 2023.

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Gilchrist, C. L., Söding, J., and Steinegger, M. Foldseek: fast and accurate protein structure search. *Biorxiv*, pp. 2022–02, 2022.

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.

Verkuil, R., Kabeli, O., Du, Y., Wicky, B. I., Milles, L. F., Dauparas, J., Baker, D., Ovchinnikov, S., Sercu, T., and Rives, A. Language models generalize beyond natural proteins. *BioRxiv*, pp. 2022–12, 2022.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., et al. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Wang, P. Lucidrains github repository. https://github.com/lucidrains, 2024.

Winter, R., Noé, F., and Clevert, D.-A. Permutation-invariant variational autoencoder for graph-level representation learning. *Advances in Neural Information Processing Systems*, 34:9559–9573, 2021.

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

Xu, M., Zhang, Z., Lu, J., Zhu, Z., Zhang, Y., Chang, M., Liu, R., and Tang, J. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. *Advances in Neural Information Processing Systems*, 35: 35156–35173, 2022.

Yang, L., Tian, Y., Xu, M., Liu, Z., Hong, S., Qu, W., Zhang, W., Bin, C., Zhang, M., and Leskovec, J. Vqgraph: Rethinking graph representation space for bridging gnns and mlps. In *The Twelfth International Conference on Learning Representations*, 2024.

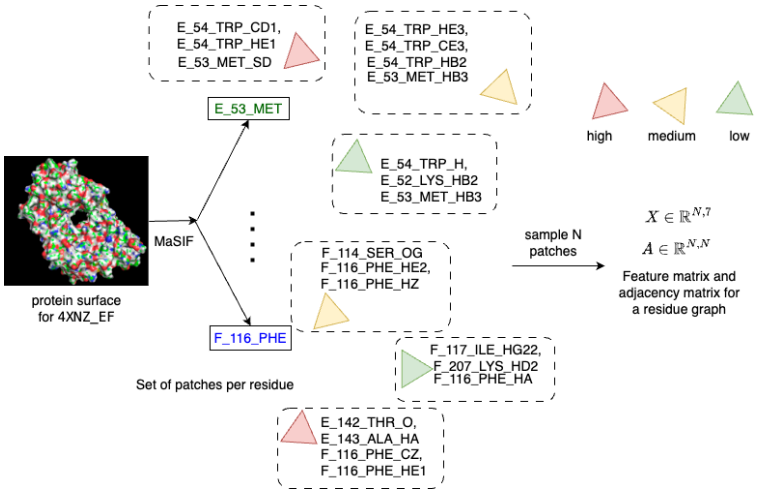

Figure 4. Creating local geometric and chemical features per residue: Given a protein structure, we first run MaSIF and get 5 features mapped to each surface exposed residue: charge, hydrophobicity, shape index, distance dependent curvature, hydrogen bond interaction. We compute 2 more features: patch centroid to C-α atom of residue and angle between C-α to patch centroid and C-α to C-β. The patches are classified as high medium or low depending on the type of core or border or borrowed atoms

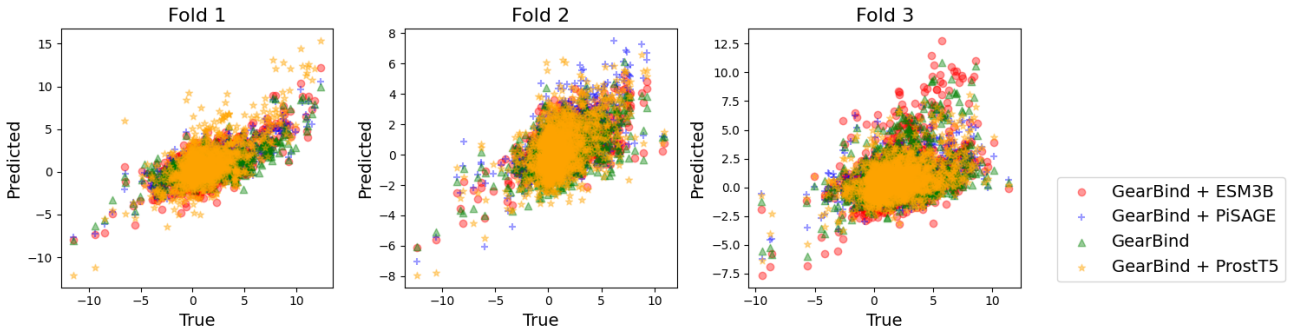

Figure 5. Pearson-R of $\Delta\Delta G$ on three folds by different methods

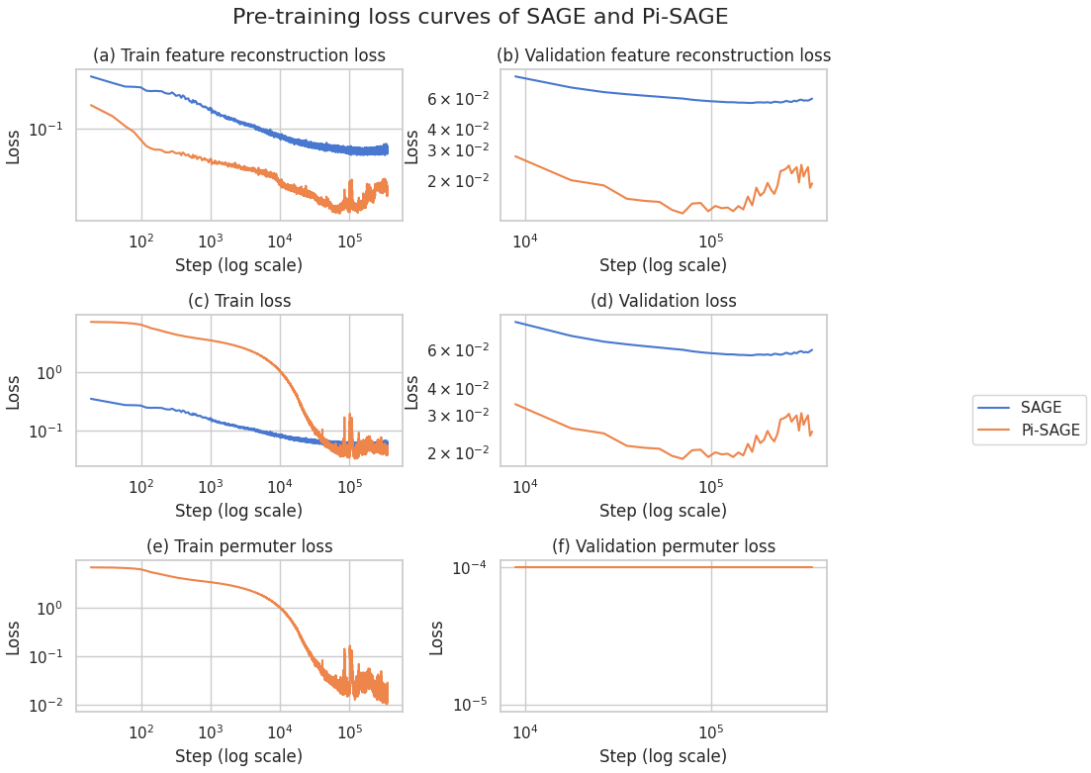

Figure 6. Pre-training loss curves for 44M 4k vocab size of SAGE and Pi-SAGE