Learning to Predict Mutational Effects of Protein-Protein Interactions by Microenvironment-aware Hierarchical Prompt Learning

Lirong Wu [1] **Yijun Tian** [2] **Haitao Lin** [1] **Yufei Huang** [1] **Siyuan Li** [1] **Nitesh V Chawla** [2] **Stan Z. Li**[† 1]

Abstract

Protein-protein bindings play a key role in a variety of fundamental biological processes, and thus predicting the effects of amino acid mutations on protein-protein binding is crucial. To tackle the scarcity of annotated mutation data, pre-training with massive unlabeled data has emerged as a promising solution. However, this process faces a series of challenges: (1) complex higher-order dependencies among multiple (more than paired) structural scales have not yet been fully captured; (2) it is rarely explored how mutations alter the local conformation of the surrounding microenvironment; (3) pre-training is costly, both in data size and computational burden. In this paper, we first construct a hierarchical prompt codebook to record common microenvironmental patterns at different structural scales independently. Then, we develop a novel codebook pre-training task, namely masked microenvironment modeling, to model the joint distribution of each mutation with their residue types, angular statistics, and local conformational changes in the microenvironment. With the constructed prompt codebook, we encode the microenvironment around each mutation into multiple hierarchical prompts and combine them to flexibly provide information to wild-type and mutated protein complexes about their microenvironmental differences. Such a hierarchical prompt learning framework has demonstrated superior performance and training efficiency over state-of-the-art pre-training-based methods in mutation effect prediction and a case study of optimizing human antibodies against SARS-CoV-2.

1. Introduction

Proteins usually interact with other proteins to perform specific biological functions that are essential for all organisms (Hu et al., 2021; Kastritis & Bonvin, 2013; Lu et al., 2020; Gao et al., 2024; Lin et al., 2024; Huang et al., 2024; Wu et al., 2024a). A prime example is antibodies, a family of Y-shape proteins produced by the immune system to recognize, bind, and interact with proteins on the surface of pathogens (Murphy & Weaver, 2016; Tan et al., 2024). Therefore, how to develop methods to modulate protein-protein interactions has become a key issue, and one of the most prevalent strategies is to mutate the amino acids at the interaction interface (Luo et al., 2023; Liu et al., 2023). Considering the enormous combinatorial space of over 20^{30} amino acid mutations and the high variability of mutated structures, it is not feasible to test all potential mutations by experimental assays in the web laboratory, which calls for computational methods to screen for desirable mutations by predicting binding affinity changes of protein complexes upon mutations. This problem, also known as *the change in binding free energy* ($\Delta\Delta G$) prediction, is a core challenge in the protein complex design (Marchand et al., 2022).

The computational methods for $\Delta\Delta G$ prediction have undergone a paradigm shift from biophysics-based and statistics-based techniques (Schymkowitz et al., 2005; Park et al., 2016; Alford et al., 2017) to Deep Learning (DL) techniques (Shan et al., 2022; Luo et al., 2023; Liu et al., 2023). Despite the great progress made by DL-based methods, the scarcity of annotated experimental data and the unavailability of mutated complex structures remain two major challenges for effective supervised learning. Therefore, pre-training with massive unlabeled data is becoming a promising solution. On the one hand, the general knowledge learned from the pre-training data can be transferred for $\Delta\Delta G$ prediction, which solves the problem of data sparsity effectively. On the other hand, some of the pre-training tasks can capture sequence-structure dependencies, enabling the model to be implicitly aware of the mutated complex structures rather than explicitly predicting them. Owing to these two benefits, pre-training is becoming one of the most prevalent strategies for $\Delta\Delta G$ prediction (Luo et al., 2023).

Despite the fruitful progress, existing pre-training-based methods still encounter several key issues. The first is the ignorance of modeling multiple types of structural scales and their dependencies. A protein can focus on different structural scales to implement specific functions, and each

[1]Westlake University [2]University of Notre Dame. Correspondence to: Stan Z. Li <stan.zq.li@westlake.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

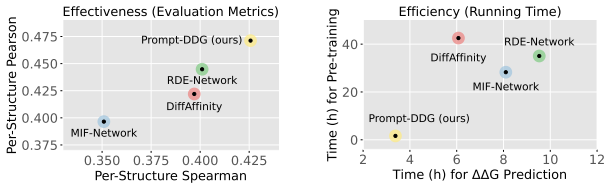

Figure 1. Comparison of our Prompt-DDG with three state-of-the-art methods in effectiveness (per-structure Pearson and Spearman) and training efficiency (for pre-training and $\Delta\Delta G$ prediction). where Prompt-DDG outperforms the other methods a lot in both effectiveness and efficiency, especially the time spent on pre-training.

structural scale has its own merits and cannot replace each other. Besides, the dependencies between different structural scales are diverse, and simply focusing on single or paired structural scales using existing pre-training tasks cannot fully capture their complex higher-order dependencies. The second obstacle is the lack of mutated complex structures. Although Alphafold2 (AF2) (Jumper et al., 2021) and ESMFold (Lin et al., 2023) have made great advances in protein structure prediction, they still struggle to predict the exact conformational changes upon subtle mutations in amino acids. Moreover, it has been found that the performance of a model trained with experimental structures drops significantly when tested on the predicted AF2 structures (Huang et al., 2023). As an alternative, Rotamer Density Estimator (RDE) (Luo et al., 2023) and DiffAffinity (Liu et al., 2023) implicitly model sequence-structure dependence by predicting *global* sidechain conformational changes upon *all mutations*, ignoring how *each mutation* alter the *local* backbone conformation of it surrounding microenvironment. Moreover, the computational cost in existing pre-training tasks caused by the huge amount of data is too expensive and even far beyond the task of $\Delta\Delta G$ prediction itself. For example, there are only 7k labeled mutation data in the SKEMPI v2.0 dataset, but the PDB-REDO dataset used for pre-training by RDE contains more than 143k data.

Present Work. In this paper, we propose a simple yet effective *Microenvironment-aware Hierarchical Prompt Learning* framework for efficient $\Delta\Delta G$ prediction (Prompt-DDG). The core idea of Prompt-DDG is to avoid the computationally heavy pre-training and instead directly generate concise prompts for each mutation in an *lightweight and efficient* manner. These prompts are expected to characterize the microenvironmental differences surrounding the mutation between wild-type and mutated complexes. To enable the generated prompts to fully cover the diversity of structural scales of the microenvironment, we construct a hierarchical prompt codebook to separately record common microenvironmental patterns of different structural scales. A novel codebook pre-training task, namely masked microenvironment modeling, is then proposed to model the joint distribution of each residue mutation and their heterogeneous properties, including residue types, angular statistics, and local conformational changes in the microenvironment. Using the hierarchical prompt codebook, we encode the microenvironment around each mutation into several prompts, which are passed through a lightweight module to flexibly provide wild-type and mutated complexes with multi-scale structural information about their microenvironments. Finally, Prompt-DDG outperforms other leading methods in terms of both effectiveness and efficiency, as shown in Figure. 1,

2. Related Work

2.1. Mutation Effect Prediction For Single Proteins

The prediction of mutation effects for single proteins is mainly aimed at predicting changes in the stability, fluorescence, fitness, or other properties of proteins upon the mutations (Alford et al., 2017; Lei et al., 2023; Meier et al., 2021). The current mainstream is mainly sequence-based methods, which exploit co-evolutionary information mined by Multiple Sequence Alignments (MSAs) (Frazer et al., 2021; Luo et al., 2021) or Protein Language Models (PLMs) (Meier et al., 2021; Notin et al., 2022). However, these methods are difficult to directly extend for the prediction of mutation effects on protein-protein interactions. For one thing, protein complexes involve multiple proteins or chains that may belong to different species and thus lack co-evolutionary information (Luo et al., 2023). Secondly, it is more difficult to predict changes in the binding free energy between proteins upon mutations than changes in the functions of single proteins. Finally, protein-protein interactions are mainly determined by protein structure than sequence. Therefore, mutation effect prediction on PPIs requires more efficient use of protein structures, rather than only protein sequences.

2.2. Mutation Effect Prediction For Protein Complexes

Traditional approaches for predicting the effects of mutations on protein-protein binding ($\Delta\Delta G$) can be mainly divided into two branches: biophysics-based and statistics-based methods. The biophysics-based (Alford et al., 2017; Park et al., 2016; Delgado et al., 2019) approaches sample mutated conformations from the energy function and then simulate inter-atomic interactions for predicting $\Delta\Delta G$, and thus face a trade-off between efficiency and effectiveness. On the other hand, statistics-based approaches (Geng et al., 2019; Li et al., 2016) use geometrical, physical, and evolutionary descriptors of proteins to predict mutational effects, and are thus limited by the choice of descriptors and cannot leverage the growing availability of protein structures.

Recent deep learning-based approaches can be categorized into two classes: end-to-end and pre-training-based methods. The end-to-end approaches (Shan et al., 2022; Luo et al., 2023) train a feature encoder to extract representations of both wild-type and mutated protein complexes, and then combine the two to directly predict $\Delta\Delta G$. The pre-training-based approaches learn sequence-structure dependencies by pre-training on large amounts of unlabeled

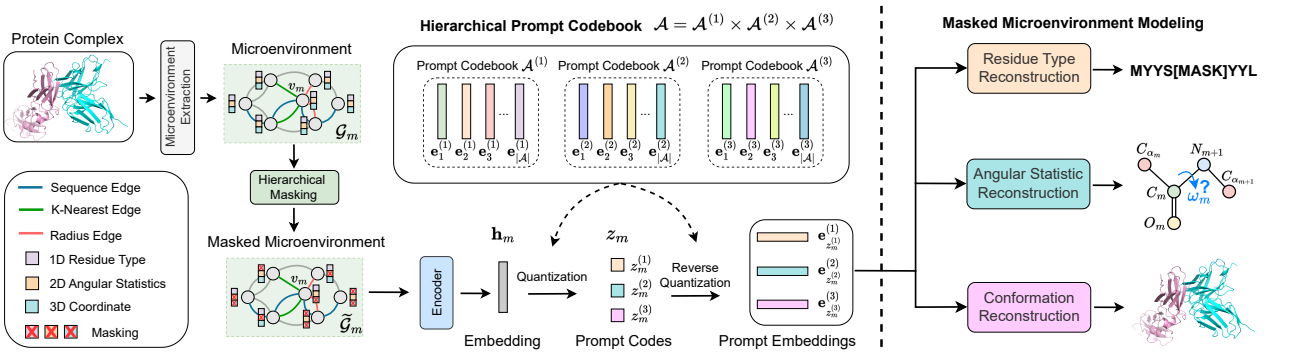

Figure 2. Left: A high-level overview of microenvironment-aware hierarchical prompt learning and adaptation framework for efficient $\Delta\Delta G$ prediction (Prompt-DDG). Right: Illustration of a hierarchical pre-training task by Masked Microenvironment Modeling (MMM).

data and then transfer the learned knowledge for predicting $\Delta\Delta G$. For example, Masked Inverse Folding (MIF) (Yang et al., 2022) treats protein inverse folding as a pretext task to learn deep transformations from structure to sequence. Instead, RDE (Luo et al., 2023) employs normalizing flows to estimate the distribution of protein side-chain conformations and then uses entropy to measure flexibility. Similarly, DiffAffinity (Liu et al., 2023) pre-trains a side-chain diffusion probabilistic model on unlabeled protein structures and leverages the pre-trained representations to predict $\Delta\Delta G$.

3. Preliminary

Graph Construction. We represent each protein-protein complex as a *Heterogeneous Attribute Graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = (\mathcal{V}_L, \mathcal{V}_R)$ is the node (residue) set of the ligand $\mathcal{V}_L$ and the receptor $\mathcal{V}_R$, and $\mathcal{E} = (\mathcal{E}_{\text{in}}, \mathcal{E}_{\text{ex}})$ separately contain internal edges within each component and external edges between components. Each node $v_i = (\mathbf{x}_i, \mathbf{Z}_i) \in \mathcal{V}$ is attributed as a node feature vector $\mathbf{x}_i \in \mathbb{R}^{d_n}$ and a node coordinate matrix $\mathbf{Z}_i \in \mathbb{R}^{3\times 4}$ consisting of 4 backbone atoms $\{N, C_\alpha, C, O\}$. In addition, each edge $e_{i,j} \in \mathcal{E}$ is described by an edge feature vector $\mathbf{E}_{i,j} \in \mathbb{R}^{d_e}$. Therefore, the graph of each protein-protein complex can also be denoted as $\mathcal{G} = (\mathbf{X}, \mathbf{Z}, \mathbf{E})$. For each node $v_i \in \mathcal{V}$, we define its node feature $\mathbf{x}_i$ as E(3)-invariant feature, as follows

$$\mathbf{x}_i = \left\{ E_{\text{type}}(v_i), E_{\text{ang}}(\Omega_i), Q_i^\top \frac{Z_{i,\xi} - Z_{i,C_\alpha}}{\|Z_{i,\xi} - Z_{i,C_\alpha}\|} \mid \Omega_i, \xi \right\}, \quad (1)$$

where $E_{\text{type}}(v_i)$ is trainable type embedding of residue v_i. Ω_i contains three dihedral angles $\alpha_i, \beta_i, \gamma_i$ of the backbone and four torsion angles $\{\chi_i^{(k)}\}_{k=1}^4$ of the side chain, and $E_{\text{ang}}(\cdot)$ denotes the angular encodings (Luo et al., 2023) in Ω_i. The last term in $\mathbf{x}_i$ is direction encodings that correspond to the relative directions of three backbone atoms $\xi \in \{C, N, O\}$ in the local coordinate frame Q_i of residue v_i. The edge feature $\mathbf{E}_{i,j}$ that describes the relationship between two residues v_i and v_j is defined as follows

$$\mathbf{E}_{i,j} = \left\{ E_{\text{pos}}(i,j), E_{\text{dis}}(\mathbf{Z}_i, \mathbf{Z}_j), Q_i^\top \frac{Z_{j,\zeta} - Z_{i,C_\alpha}}{\|Z_{j,\zeta} - Z_{i,C_\alpha}\|} \mid \zeta \right\}, \quad (2)$$

where $E_{\text{pos}}(i,j)$ and $E_{\text{dis}}(\mathbf{Z}_i, \mathbf{Z}_j)$ encode the relative sequential and spatial distances between residue v_i and residue v_j, respectively. $E_{\text{pos}}(i,j)$ is set as 0 for any external edge $e_{i,j} \in \mathcal{E}_{\text{ex}}$. In addition, the last term is the direction encodings of four backbone atoms $\zeta \in \{C_\alpha, C, N, O\}$ of residue v_j in the local coordinate frame Q_i of residue v_i.

Problem Statement. Given a wild-type protein complex $\mathcal{G}^W = (\mathbf{X}, \mathbf{Z}, \mathbf{E})$ and a set of mutations $\mathcal{M}$, the task of mutational effect prediction for protein complexes can be formulated as predicting the change in $\Delta\Delta G$ between the wild-type complex $\mathcal{G}^W$ and mutated complex $\mathcal{G}^M = g(\mathcal{G}^W, \mathcal{M})$, i.e., approximating the mapping $p(\Delta\Delta G \mid \mathcal{G}^W, \mathcal{G}^M)$.

4. Methodology

In this paper, we propose a Prompt-DDG framework with three novel components for $\Delta\Delta G$ prediction. The pipeline is shown in Figure. 2. In particular, the first component constructs a hierarchical prompt codebook that encodes the microenvironment around each mutation as prompts of different structural scales. The second component pre-trains the prompt codebook hierarchically by masked microenvironment modeling. The third component adopts a lightweight prompt adaptation module that combines prompts of different scales to provide the microenvironmental differences.

4.1. Hierarchical Prompt Codebook Construction

4.1.1. Definition of Microenvironment

The microenvironment of a residue describes its surrounding sequence and structure contexts. We follow (Wu et al., 2024b) to define the microenvironment of each mutation $v_m \in \mathcal{M}$ as a v_m-ego subgraph $\mathcal{G}_m \subseteq \mathcal{G}$ of the protein complex graph $\mathcal{G}$, with its node set $V_{\mathcal{G}_m}$ defined as follows

$$V_{\mathcal{G}_m} = \left\{ v_n \mid |m-n| \le d_s, \left\| Z_{m,C_\alpha} - Z_{n,C_\alpha} \right\| \le d_r, v_n \in \mathcal{N}_m^{(K)} \right\},$$

where d_s and d_r are cut-off distances, Z_{m,C_α} and Z_{n,C_α} are the 3D coordinates of carbon-alpha atoms, and $\mathcal{N}_m^{(K)}$ is the K-hop neighborhood of residue v_m in the spatial space.

4.1.2. Hierarchical Codebook Construction

Protein-protein interactions focus on different structural scales to implement their functions, such as 1D for amino

acid sequences, 2D geometric statistics, 3D structural coordinates, etc. Each structural scale has its own merits and cannot replace each other. To fully capture the different structural scales of the microenvironment, we constructed a hierarchical prompt codebook $\mathcal{A}=\mathcal{A}^{(1)}\times\mathcal{A}^{(2)}\times\mathcal{A}^{(3)}$, where $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$, and $\mathcal{A}^{(3)}$ characterize three aspects of the microenvironment, i.e., residue type, angular statistics, and local conformation, respectively. Each sub-codebook $\mathcal{A}^{(k)}$ is parameterized as $\mathcal{A}^{(k)}=\{\mathbf{e}_1^{(k)},\mathbf{e}_2^{(k)},\cdots,\mathbf{e}_{|\mathcal{A}|}^{(k)}\}\in\mathbb{R}^{|\mathcal{A}|\times F}$, where $\{\mathbf{e}_i^{(k)}\}_{i=1}^{|\mathcal{A}|}$ are $|\mathcal{A}|$ learnable prompt embeddings.

To generate microenvironment-aware prompts for different structural scales, we first encode the microenvironment $\mathcal{G}_m$ into a hidden representation $\mathbf{h}_m$ using a self-attention-based graph neural network $f_\theta(\cdot)$ that is invariant to rotation and translation (Jumper et al., 2021). Next, the microenvironments $\{\mathcal{G}_1,\mathcal{G}_2,\cdots,\mathcal{G}_{|\mathcal{M}|}\}$ of $|\mathcal{M}|$ mutations can be tokenized to discrete prompt codes $\{z_1,z_2,\cdots,z_M\}$ by vector quantization (Van Den Oord et al., 2017; Li et al., 2024) that looks up the nearest neighbors in the hierarchical codebook $\mathcal{A}$. For each microenvironment $\mathcal{G}_m$, its prompt codes $z_i=\{z_m^{(1)},z_m^{(2)},z_m^{(3)}\}$ are defined as follows

$$z_m^{(k)}=\operatorname{argmin}_n\left\|\mathbf{h}_m-\mathbf{e}_n^{(k)}\right\|_2,\quad\text{where } 1\le k\le 3. \tag{3}$$

4.2. Masked Microenvironment Modeling (MMM)

To resolve the non-differentiability of the vector quantization, we impose a constraint $\mathcal{L}_{\text{VQ}}$ to bride the codebook $|\mathcal{A}|$ and microenvironment representations by straight-through estimator (Bengio et al., 2013), which is defined as follows

$$\mathcal{L}_{\text{VQ}}=\frac{1}{|\mathcal{V}|}\sum_{v_i\in\mathcal{V}}\sum_{k=1}^{3}\left(\left\|\text{sg}\left[\mathbf{h}_i\right]-\mathbf{e}_{z_i^{(k)}}\right\|_2^2+\eta\left\|\mathbf{h}_i-\text{sg}\left[\mathbf{e}_{z_i^{(k)}}\right]\right\|_2^2\right), \tag{4}$$

where η is a trade-off hyperparameter, and $\text{sg}[\cdot]$ is the stop-gradient operation. The first term in Eq. (4) is a codebook loss, used to update the codebook to make the microenvironment representations $\mathbf{h}_i$ close to the most similar prompt embeddings. The second term in Eq. (4) is a commitment loss, encouraging the encoder outputs to stay close to the chosen prompt embeddings by only training the encoder.

Next, we focus on how to pre-train learnable prompt embeddings in the constructed hierarchical codebook $\mathcal{A}$. To this end, we take three different data reconstruction tasks to simultaneously learn the hierarchical codebook $\mathcal{A}$ and train the microenvironment encoder $f_\theta(\cdot)$. We train each sub-codebook hierarchically with individual reconstruction tasks to make it focus on a specialized structural scale. Furthermore, in order to fully capture the higher-order (more than single and paired) dependencies among various structural scales, we unify the hierarchical training in one pre-training task, i.e., masked microenvironment modeling. Specifically, we independently mask the residue types, geometric angles, and conformation coordinates in the microenvironment $\mathcal{G}_m$ by randomly flipping, zeroing, and Gaussian noising, and then reconstruct the inputs from the masked microenvironment $\widetilde{\mathcal{G}}_m$ by three different reconstruction tasks. The masked residues sets of three structural scales are independent and are denoted as $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$, respectively.

Residue Type Reconstruction. To pre-train the sub-codebook $\mathcal{A}^{(1)}$, we use a symmetric variant of microenvironment encoder $f_\theta(\cdot)$ as the type decoder $\widehat{f}_\theta(\cdot)$, which predicts the residue types $\{\widehat{s}_i\}_{v_i\in\mathcal{C}_1}$ of a masked residue set $\mathcal{C}_1\subseteq\mathcal{V}$ from the prompt embeddings $\{\mathbf{e}_{z_i^{(1)}}\}_{v_i\in\mathcal{V}}$. The reconstruction loss $\mathcal{L}_{\text{seq}}$ is defined by cross-entropy ℓ_{ce}:

$$\mathcal{L}_{\text{seq}}\left(\mathcal{A}^{(1)}\right)=\frac{1}{|\mathcal{C}_1|}\sum_{v_i\in\mathcal{C}_1}\ell_{ce}\big(s_i,\widehat{s}_i\big). \tag{5}$$

Angular Statistic Reconstruction. To pre-train the sub-codebook $\mathcal{A}^{(2)}$, we use another decoder to reconstruct the angular information from the prompt embeddings $\{\mathbf{e}_{z_i^{(2)}}\}_{v_i\in\mathcal{V}}$. Next, we compute the MSE loss between the predicted and the ground-truth ones as the loss $\mathcal{L}_{\text{ang}}$:

$$\mathcal{L}_{\text{ang}}\big(\mathcal{A}^{(2)}\big)=\frac{1}{|\mathcal{C}_2|}\sum_{v_i\in\mathcal{C}_2}\sum_{a\in\Omega_i}\Big\|E_{\text{ang}}(a)-E_{\text{ang}}(\widehat{a})\Big\|_2^2, \tag{6}$$

where $\mathcal{C}_2\subseteq\mathcal{V}$ is the residue set with masked angles, Ω_i contains three dihedral angles $\alpha_i,\beta_i,\gamma_i$ of the backbone and four torsion angles $\{\chi_i^{(k)}\}_{k=1}^4$ of the side chain, and $E_{\text{ang}}(\cdot)$ denotes the angular encodings used by (Luo et al., 2023).

Local Conformation Reconstruction. As mentioned earlier, a key issue for $\Delta\Delta G$ prediction is how to be aware of local conformational changes induced by mutations. Given the absence of mutated structures renders supervised learning infeasible, we take another unsupervised noise estimation task as a pretext task for pre-training the sub-codebook. Specifically, we first add Gaussian noise $\mathbf{O}_i$ to the wild-type structure $\mathbf{Z}_i$ to get noisy structure $\widetilde{\mathbf{Z}}_i=\mathbf{Z}_i+\mathbf{O}_i$, which is encoded by an E(3)-equivariant graph neural network (see **Appendix A** for a detailed description of the architecture) to predict structural noise $\widehat{\mathbf{O}}_i$, e.g., local conformation change, from the noisy structure $\widetilde{\mathbf{Z}}_i$ to the wild-type structure $\mathbf{Z}_i$. Furthermore, we adopt the Huber loss (Huber, 1992) (see **Appendix B** for detailed formulas) other than the common MSE loss as the objective function, defined as follows:

$$\mathcal{L}_{\text{struct}}\left(\mathcal{A}^{(3)}\right)=\frac{1}{|\mathcal{C}_3|}\sum_{v_i\in\mathcal{C}_3}\ell_{\text{huber}}\left(\mathbf{O}_i,\widehat{\mathbf{O}}_i\right), \tag{7}$$

where $\mathcal{C}_3\subseteq\mathcal{V}$ is the residue set with Gaussian noise added.

Summary. While each sub-codebook $\mathcal{A}^{(k)}$ $(1\le k\le 3)$ is hierarchically trained with one individual reconstruction task, they share the same masked inputs and microenvironment encoder, which makes *the reconstruction of individual structural scale dependent not only on itself but also on other scales*. Thus, MMM well models the joint distribution of each mutation with their residue types, angular statistics, and local conformation changes in the microenvironment.

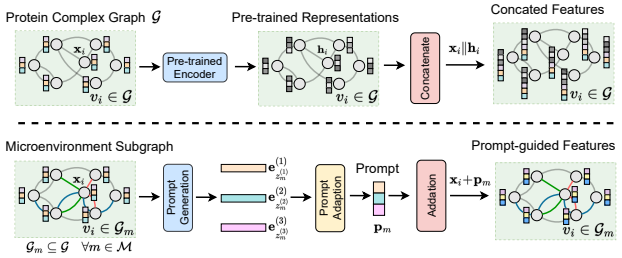

Figure 3. Top (pre-training-based): concat pre-trained representation $\mathbf{h}_i$ of residue $v_i \in \mathcal{V}$ with the original feature $\mathbf{x}_i$. Below (prompt-guided): encode the microenvironment $\mathcal{G}_m$ around mutation $m \in \mathcal{M}$ into a prompt $\mathbf{p}_m$ and add it to each residue $v_i \in \mathcal{G}_m$.

4.3. Prompt Training, Adaptation, and Usage

4.3.1. Lightweight Prompt Adaptation

The final loss function $\mathcal{L}$ for the hierarchical training of the prompt codebook $\mathcal{A} = \mathcal{A}^{(1)} \times \mathcal{A}^{(2)} \times \mathcal{A}^{(3)}$ is defined as:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{VQ}} + \mathcal{L}_{\text{seq}}\left(\mathcal{A}^{(1)}\right) + \mathcal{L}_{\text{ang}}\left(\mathcal{A}^{(2)}\right) + \mathcal{L}_{\text{struct}}\left(\mathcal{A}^{(3)}\right). \quad (8)$$

Using the pre-trained codebook $\mathcal{A}$, we can encode the microenvironment $\mathcal{G}_m$ around each mutation $m \in \mathcal{M}$ into three discrete prompt codes $z_i = \{z_m^{(1)}, z_m^{(2)}, z_m^{(3)}\}$ by Eq. (3). Since different sub-codebooks record different structural scales of the microenvironment, we flexibly combine the acquired prompt embeddings $\{\mathbf{e}_{z_m^{(1)}}, \mathbf{e}_{z_m^{(2)}}, \mathbf{e}_{z_m^{(3)}}\}$ to bridge the gap between pre-trained prompts and downstream tasks. The prompt combination is implemented by a lightweight prompt adaptation module, defined as

$$\begin{aligned} \mathbf{p}_m &= \alpha_1 \cdot \mathbf{e}_{z_m^{(1)}} + \alpha_2 \cdot \mathbf{e}_{z_m^{(2)}} + \alpha_3 \cdot \mathbf{e}_{z_m^{(3)}}, \text{ where} \\ \alpha_k &= \phi_\omega^{(k)}\left(\mathbf{e}_{z_m^{(1)}}, \mathbf{e}_{z_m^{(2)}}, \mathbf{e}_{z_m^{(3)}}\right), \end{aligned} \quad (9)$$

where $\{\phi_\omega^{(k)}(\cdot)\}_{k=1}^3$ are one-layer linear transformation.

4.3.2. Prompt-Guided $\Delta\Delta G$ Prediction

Previous pre-training-based methods learn a pre-trained representation $\mathbf{h}_i$ for each residue $v_i \in \mathcal{V}$ and concat (or add) it to the corresponding residue, i.e., $\widetilde{\mathbf{x}}_i = \mathbf{x}_i \| \mathbf{h}_i$. In contrast, we encode the microenvironment $\mathcal{G}_m$ around each mutation $m \in \mathcal{M}$ as a prompt embedding $\mathbf{p}_m$ and then add it to each residue $v_i \in \mathcal{V}_{\mathcal{G}_m}$ within the microenvironment, i.e., $\widetilde{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{p}_m$, as shown in Figure. 3. Next, we use a network that shares the same architecture as the microenvironment encoder $f_\theta(\cdot)$ to transform prompt-guided inputs $\{\widetilde{\mathbf{x}}_i\}_{v_i \in \mathcal{V}}$ and apply max-pooling to obtain a global structure representation. We then subtract the wild-type representation from the mutant representation, feed it into an MLP to predict $\Delta\Delta\widehat{G}$, and calculate the MSE loss between it and the ground-truth $\Delta\Delta G$ as the final objective function.

4.4. Further Comparison and Discussion

Compared to pre-training, prompt learning on proteins is still a new topic, and the only similar work so far is PromptProtein (Wang et al., 2023), but Prompt-DDG differs from it in four aspects: (1) PromptProtein focuses only on the task of property prediction for *single proteins*, and whether and how it can be extended to *protein complexes* like Prompt-DDG remains unexplored. (2) PromptProtein uses a *learnable* attention mask matrix to model the relationship between prompts and existing residues, but prompts in Prompt-DDG are *explicitly associated* with a specific microenvironment. (3) PromptProtein learns *global task-specific* prompts for all residues, while Prompt-DDG learns a *local mutation-specific* prompt for each residue within the microenvironment. (4) PromptProtein learns *continuous* prompts, but Prompt-DDG constructs a hierarchical prompt codebook to record only those most common microenvironmental patterns in a *discrete* fashion. In addition, another topic related to Prompt-DDG is the protein microenvironment encoding, such as a recent work MAPE-PPI (Wu et al., 2024b), which has been discussed in detail in **Appendix C**. Due to space limitations, the time complexity analysis and pseudo-code of our Prompt-DDG are available in **Appendix D & E**.

5. Experiments

Baselines. We compare Prompt-DDG with four categories of state-of-the-art methods. The first category is to directly extend those sequence-based approaches from single proteins to protein-protein interactions, including ESM-1v (Meier et al., 2021), Position-Specific Scoring Matrix (PSSM), MSA Transformer (Rao et al., 2021), and Tranception (Notin et al., 2022). The second category is the traditional energy-based approaches, including Rosetta (Alford et al., 2017) Cartesian ddG and FoldX (Delgado et al., 2019). The third category is supervised learning approaches, including DDGPred (Shan et al., 2022) and a model that uses a self-attention-based network (Jumper et al., 2021) as the encoder, but uses the MLP to directly predict $\Delta\Delta G$ (End-to-End). The fourth category is pre-training-based approaches, including ESM-1F (Hsu et al., 2022), two variants of MIF (MIF-Δlogits and MIF-Network) (Yang et al., 2020), two variants of RDE (RDE-Linear and RDE-Network) (Luo et al., 2023), DiffAffinity (Liu et al., 2023), and a model that is pre-trained to predict the B-factor of residues and use predicted B-factors to predict $\Delta\Delta G$. A more detailed description of these methods and hyperparameter settings of our Prompt-DDG can be found in **Appendix F & G**.

Datasets. We evaluate the effectiveness of Prompt-DDG for $\Delta\Delta G$ prediction on the SKEMPI v2.0 (Jankauskaitė et al., 2019) dataset, the largest available annotated mutation dataset for protein complexes. The SKEMPI v2.0 dataset contains 7,085 amino acid mutations and corresponding changes in the thermodynamic parameters and kinetic rate constants, but it does not contain any structures of the mutated complexes. To avoid data leakage, we split the dataset into 3 folds by structure, each of which contains unique protein complexes. Then, we follow (Luo et al.,

Table 1. Mean results of 3-fold cross-validation on the SKEMPI v2 dataset, where **bold** and underline denote the best and second metrics.

Category	Method	Per-Structure		Overall				
		Pearson ↑	Spear. ↑	Pearson ↑	Spear. ↑	RMSE ↓	MAE ↓	AUROC ↑
Sequence-based	ESM-1v	0.0073	-0.0118	0.1921	0.1572	1.9609	1.3683	0.5414
	PSSM	0.0826	0.0822	0.0159	0.0666	1.9978	1.3895	0.5260
	MSA Transformer	0.1031	0.0868	0.1173	0.1313	1.9835	1.3816	0.5768
	Tranception	0.1348	0.1236	0.1141	0.1402	2.0382	1.3883	0.5885
Energy Function	Rosetta	0.3284	0.2988	0.3113	0.3468	1.6173	1.1311	0.6562
	FoldX	0.3789	0.3693	0.3120	0.4071	1.9080	1.3089	0.6582
Supervised	DDGPred	0.3750	0.3407	0.6580	0.4687	**1.4998**	1.0821	0.6992
	End-to-End	0.3873	0.3587	0.6373	0.4882	1.6198	1.1761	0.7172
Pre-training-based	B-factor	0.2042	0.1686	0.2390	0.2625	2.0411	1.4402	0.6044
	ESM-1F	0.2241	0.2019	0.3194	0.2806	1.8860	1.2857	0.5899
	MIF-Δlogit	0.1585	0.1166	0.2918	0.2192	1.9092	1.3301	0.5749
	MIF-Network	0.3965	0.3509	0.6523	0.5134	1.5932	1.1469	0.7329
	RDE-Linear	0.2903	0.2632	0.4185	0.3514	1.7832	1.2159	0.6059
	RDE-Network	0.4448	0.4010	0.6447	0.5584	1.5799	1.1123	0.7454
	DiffAffinity	0.4220	0.3970	0.6690	0.5560	1.5350	1.0930	0.7440
Ours	Prompt-DDG	**0.4712**	**0.4257**	**0.6772**	**0.5910**	1.5207	**1.0770**	**0.7568**
	$\Delta_{\text{RDE-Network}}$	+5.94%	+6.16%	+5.04%	+5.84%	+3.74%	+3.17%	+1.53%
	$\Delta_{\text{DiffAffinity}}$	+11.78%	+7.23%	+1.21%	+6.29%	+0.93%	+1.46%	+1.72%

2023) to perform 3-fold cross-validation to ensure that each data in SKEMPI2 is tested once. In terms of pre-training, ESM-1F is pre-trained on millions of predicted AF2 structures (Jumper et al., 2021), and the other six methods are all pre-trained on the PDB-REDO (Joosten et al., 2014) dataset which contains 143k data. In contrast, Prompt-DDG directly learns prompts in a lightweight manner on the SKEMPI v2.0 dataset without requiring any additional pre-training data.

Evaluation Metrics. A total of seven metrics are used to comprehensively evaluate the performance of $\Delta\Delta G$ prediction, including five overall metrics: (1) Pearson correlation coefficient; (2) Spearman correlation coefficient; (3) Root Mean Squared Error (RMSE); (4) Mean Absolute Error (MAE); (5) AUROC. Since the correlation of specific protein complexes is often of greater interest in practice, we group the mutations by structure, calculate the Pearson and Spearman correlation coefficients for each structure separately, and report the average as two additional metrics.

5.1. Comparison with State-of-The-Art Baselines

We report the 7 evaluation metrics for the 15 methods on the SKEMPI v2.0 dataset in Table. 1, as well as the relative improvement of Prompt-GPT over the two leading methods, RDE-Network and DiffAffinity. It can be observed that (1) Prompt-DDG outperforms all baselines in 6 out of 7 evaluation metrics. In addition, it ranks second only in the RMSE metric, close to the state-of-the-art supervised method, DDGPred. (2) Despite not being pre-trained with any additional data, Prompt-DDG exceeds all pre-training-based methods across 7 metrics, which suggests that **specialized** microenvironmental prompts are more effective for $\Delta\Delta G$ prediction than **general** knowledge learned from protein pre-training. (3) Notably, Prompt-DDG achieves the most significant improvement on the two most critical metrics, the per-structure Pearson and Spearman correlations, demonstrating its greater potential for practical applications.

Furthermore, we select five superior methods from Table. 1 based on a comprehensive consideration of the 7 metrics and compare Prompt-DDG with them under single-point, multi-point, and all-point mutations. The results reported in Table. 2 show that Prompt-DDG achieves the best overall performance under the single-point mutation setting, ranking first in 4 out of 7 metrics. In practice, it is a common case to mutate multiple amino acids to reach the desired binding affinity, making the effect prediction of multi-point mutations very important. In particular, Prompt-DDG outperforms all other baselines, including RDE-Network and DiffAffinity, by a large margin in the multi-point mutation setting. The superiority of Prompt-GNN for multi-point mutation is twofold: (1) it generates prompts for the microenvironment around each mutation separately, which captures more fine-grained local (rather than global) differences; and (2) the conformation of a complex with multiple mutations is more variable than that with a single mutation, and Prompt-DDG is good at modeling the effects of each mutation on its local microenvironmental conformation.

5.2. Visualization for Correlation Analysis

We present in Figure. 4 the scatter plots of experimental and predicted $\Delta\Delta G$ for four representative methods, MIF-

Table 2. Performance comparison under single-, multi- and all-point mutation, where **bold** denotes the best metric under each setting.

Method	Pre-training Dataset (Szie)	Mutations	Per-Structure		Overall				
			Pearson ↑	Spear. ↑	Pearson ↑	Spear. ↑	RMSE ↓	MAE ↓	AUROC ↑
DDGPred	✗	all	0.3750	0.3407	0.6580	0.4687	**1.4998**	1.0821	0.6992
		single	0.3711	0.3427	0.6515	0.4390	1.3285	0.9618	0.6858
		multiple	0.3912	0.3896	0.5938	0.5150	2.1813	1.6699	0.7590
End-to-End	✗	all	0.3873	0.3587	0.6373	0.4882	1.6198	1.1761	0.7172
		single	0.3818	0.3426	0.6605	0.4594	1.3148	0.9569	0.7019
		multiple	0.4178	0.4034	0.5858	0.4942	2.1971	1.7087	0.7532
MIF-Network	PDB-REDO (143k)	all	0.3965	0.3509	0.6523	0.5134	1.5932	1.1469	0.7329
		single	0.3952	0.3479	0.6667	0.4802	1.3052	0.9411	0.7175
		multiple	0.3968	0.3789	0.6139	0.5370	2.1399	1.6422	0.7735
RDE-Network	PDB-REDO (143k)	all	0.4448	0.4010	0.6447	0.5584	1.5799	1.1123	0.7454
		single	0.4687	0.4333	0.6421	0.5271	1.3333	0.9392	**0.7367**
		multiple	0.4233	0.3926	0.6288	0.5900	2.0980	1.5747	0.7749
DiffAffinity	PDB-REDO (143k)	all	0.4220	0.3970	0.6690	0.5560	1.5350	1.0930	0.7440
		single	0.4290	0.4090	**0.6720**	0.5230	**1.2880**	0.9230	0.7330
		multiple	0.4140	0.3870	0.6500	0.6020	2.0510	1.5400	0.7840
Prompt-DDG	SKEMPI v2.0 (7k)	all	**0.4712**	**0.4257**	**0.6772**	**0.5910**	1.5207	**1.0770**	**0.7568**
		single	**0.4736**	**0.4392**	0.6596	**0.5450**	1.3072	**0.9191**	0.7355
		multiple	**0.4448**	**0.3961**	**0.6780**	**0.6433**	**1.9831**	**1.4837**	**0.8187**

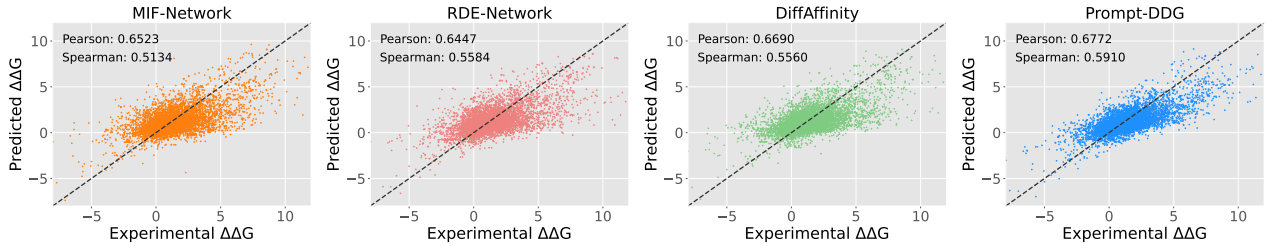

Figure 4. A comparison of correlations between experimental $\Delta\Delta G$ and $\Delta\Delta G$ predicted by four representative methods.

Network, RDE-Network, DiffAffinity, and Prompt-DDG, as well as their overall Pearson and Spearman correlation scores. It can be seen that Prompt-DDG performs better than the other three methods, both for qualitative visualization and quantitative metrics. Moreover, we provide the distribution of per-structure Pearson and Spearman correlation scores in Figure. 5, as well as the average results across all structures. We find that Prompt-DDG not only has the best average performance, but also that its distribution is mostly centered on high correlations and has fewer low-correlation structures. Due to space limitations, a comparison of Prompt-DDG's visualizations for single-point, multi-point, and all-point mutations is available in **Appendix H**.

5.3. Ablation Study and Hyperparametric Sensitivity

Ablation Study. We conduct a comprehensive evaluation on the necessity of prompts, the importance of different (structural scales) prompts, and the prompt combination schemes. From the results reported in Table. 3, three important observations can be made: (1) Three kinds of prompts characterize different structural scales of the microenvironment, each playing a different role and helpful in improving performance compared to the ones without any prompts. (2) Compared to the residue types and angular statistics of the microenvironment, how the mutation affects the local conformation is more important, and thus the corresponding prompt brings a huger performance gain than the other two prompts. (3) Combining all three kinds of prompts, either by simple averaging or weighted adaptation, outperforms any single kind of prompt. Besides, the weighted prompt adaptation module proposed in this paper, albeit lightweight, outperforms the average combination by a wide margin.

Hyperparametric Sensitivity. We studied the sensitivity of Prompt-DDG to two hyperparameters, codebook size $|\mathcal{A}|$ and mask ratio r, in Table. 4 and 5. A small codebook size, e.g., $|\mathcal{A}| = 64$, leads to sub-optimal performance due to the incapacity to cover the diversity of the microenvironment. Conversely, setting $|\mathcal{A}|$ too large may result in redundant codebooks and high computational cost. In addition, we find that the removal of the microenvironment masking, i.e., setting the mask ratio to 0.0, leads to a sharp performance drop. In practice, a mask ratio of 0.1 or 0.2 usually yields good performance, since a small mask ratio, e.g. 0.05,

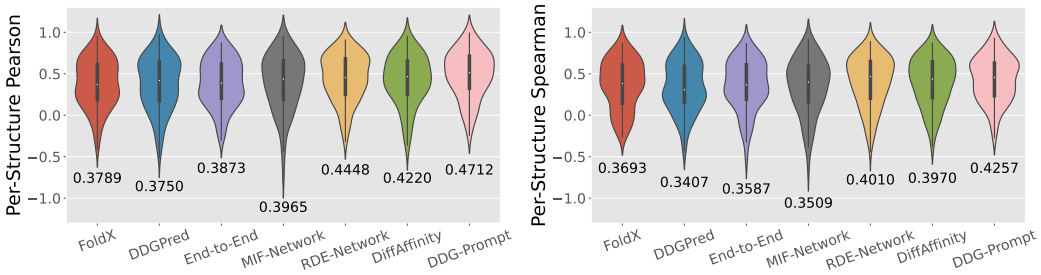

Figure 5. Distributions of per-structure Pearson correlation scores and Spearman correlation scores for seven representative methods.

Table 3. Ablation study on different types of microenvironmental prompts, where **bold** and underline denote the best and second metrics.

Method	Prompt			Per-Structure		Overall				
	1D Type	2D Angle	3D Coor.	Pearson ↑	Spear. ↑	Pearson ↑	Spear. ↑	RMSE ↓	MAE ↓	AUROC ↑
w/o Prompt	✗	✗	✗	0.4114	0.3685	0.6494	0.5417	1.5716	1.1263	0.7309
Single Prompt	✓	✗	✗	0.4263	0.3784	0.6436	0.5397	1.5518	1.1214	0.7352
	✗	✓	✗	0.4462	0.4013	0.6642	0.5674	1.5450	1.1150	0.7456
	✗	✗	✓	0.4583	0.4129	0.6696	0.5745	1.5350	1.1054	0.7532
Average	✓	✓	✓	0.4652	0.4148	0.6673	0.5791	1.5394	1.0874	**0.7587**
Weighted	✓	✓	✓	**0.4712**	**0.4257**	**0.6772**	**0.5910**	**1.5207**	**1.0770**	0.7568

Table 4. Hyperparameter sensitivity analysis on codebook size $|\mathcal{A}|$.

Codebook Size $\|\mathcal{A}\|$	Per-Structure		Overall	
	Pearson ↑	Spear. ↑	Pearson ↑	Spear. ↑
64	0.4219	0.3887	0.6453	0.5512
128	0.4575	0.4132	0.6735	**0.5971**
256	**0.4712**	**0.4257**	**0.6772**	0.5910
512	0.4457	0.4077	0.6587	0.5772
1024	0.4289	0.3916	0.6557	0.5691

Table 5. Hyperparameter sensitivity analysis on mask ratio r.

Mask Ratio	Per-Structure		Overall	
	Pearson ↑	Spear. ↑	Pearson ↑	Spear. ↑
0.00	0.4462	0.4018	0.6467	0.5610
0.05	0.4652	0.4186	0.6661	0.5795
0.10	0.4712	0.4257	**0.6772**	**0.5910**
0.20	**0.4738**	**0.4358**	0.6743	0.5833
0.30	0.4584	0.4136	0.6593	0.5683

weakens the contribution of masked modeling, while a large mask ratio, e.g. 0.3, hinders the prompt codebook learning.

5.4. Antibody Optimization against SARS-CoV-2

An important usage scenario for $\Delta\Delta G$ prediction is to identify those desirable mutations, usually with high binding affinity or neutralization, from a pool of potential mutations. In this subsection, we take the optimization of human antibodies against SARS-CoV-2 as a case study. We predict $\Delta\Delta G$s for 494 possible single-point mutations in the 26 sites within the CDR region of the antibody heavy chain, and rank them in ascending order (lowest $\Delta\Delta G$ in the top). Then, we report in Table. 6 the ranking of five favorable mutations that have been previously shown to help enhance neutralization (Shan et al., 2022). The results in Table. 6 show that only Prompt-DDG can successfully identify the four important mutations with rankings smaller than 10% (in **bold**). Moreover, Prompt-DDG achieves the highest average ranking, 7.57% and 13.80% higher than RDE-Network and DiffAffinity, respectively. More importantly, only Prompt-DDG ranks all five favorable mutations in the top 40%, suggesting good generalizability to different antibodies.

Table 6. Rankings of the five favorable mutations on the antibody against SARS-CoV-2 by various $\Delta\Delta G$ prediction methods.

Method	TH31W	AH53F	NH57L	RH103M	LH104F	Average
Rosetta	10.73%	76.72%	93.93%	11.34%	27.94%	44.13%
FoldX	13.56%	**6.88%**	**5.67%**	16.60%	66.19%	21.78%
DDGPred	68.22%	**2.63%**	12.35%	**8.30%**	**8.50%**	20.00%
End-to-End	29.96%	**2.02%**	14.17%	52.43%	17.21%	23.16%
MIF-Net.	24.49%	**4.05%**	**6.48%**	80.36%	36.23%	30.32%
RDE-Net.	**1.62%**	**2.02%**	20.65%	61.54%	**5.47%**	18.26%
DiffAffinity	**7.28%**	**3.64%**	18.82%	81.78%	10.93%	24.49%
Prompt-DDG	**2.02%**	**6.88%**	**3.24%**	34.81%	**6.48%**	10.69%

6. Conclusion

In this paper, we propose a novel Prompt-DDG framework for efficient and effective $\Delta\Delta G$ prediction. Specifically, a hierarchical prompt codebook is constructed and pre-trained by masked microenvironment modeling to cover the different structural scales of the microenvironment around each mutation. The microenvironment-aware prompts generated for each mutation flexibly provide wild-type and mutated complexes with multi-scale structural information about their microenvironmental differences. Extensive experiments have shown that Prompt-DDG achieves superior performance and efficiency over existing methods in terms of both mutation effect prediction and antibody optimization.

Acknowledgments

This work was supported by the Science & Technology Innovation 2030 Major Program Project No. 2021ZD0150100, National Natural Science Foundation of China Project No. U21A20427, Project No. WU2022A009 from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University, and Project No. WU2023C019 from the Westlake University Industries of the Future Research. Finally, we thank the Westlake University HPC Center for providing part of the computational resources.

Code Resources

Codes of this work are publicly available at: https://github.com/LirongWu/Prompt-DDG.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning for protein modeling. Prompt-DDG is expected to play significant roles in many potential biological applications. Firstly, Prompt-DDG, as a general DDG predictor trained on lots of antigen-antibody complexes with different sequences and structures, can be used as a computationally validated means for virtual screening of candidate antibodies, which helps to increase the success rate of antibody design and reduce the costs. Secondly, Prompt-DDG can be used as an explainer for discovering key functional sites on antibodies. This is because Prompt-DDG works by comparing the microenvironmental differences around each mutation, from which it locates salient mutations that are critical for binding energy. These salient mutations may be potential functional sites. Finally, Prompt-DDG can be used as a data augmentor to address data scarcity in antibody design/optimization by generating high-quality antibody data. Despite the great successes, limitations still exist. As a manuscript submitted to the ML venue, we did not spend much space discussing biology-related details, especially those about wet experiments. Moreover, Prompt-DDG mainly focuses on the interaction between two proteins and is limited in the prediction of mutational effects for multimeric proteins or substrate-related enzymes.

References

Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K., et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13 (6):3031–3048, 2017.

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

Delgado, J., Radusky, L. G., Cianferoni, D., and Serrano, L. Foldx 5.0: working with rna, small molecules and a new graphical interface. *Bioinformatics*, 35(20):4168–4169, 2019.

Elfwing, S., Uchibe, E., and Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

Frazer, J., Notin, P., Dias, M., Gomez, A., Min, J. K., Brock, K., Gal, Y., and Marks, D. S. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

Gao, Z., Tan, C., Zhang, Y., Chen, X., Wu, L., and Li, S. Z. Proteininvbench: Benchmarking protein inverse folding on diverse tasks, models, and metrics. *Advances in Neural Information Processing Systems*, 36, 2024.

Geng, C., Vangone, A., Folkers, G. E., Xue, L. C., and Bonvin, A. M. isee: Interface structure, evolution, and energy-based machine learning predictor of binding affinity changes upon mutations. *Proteins: Structure, Function, and Bioinformatics*, 87(2):110–119, 2019.

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. Learning inverse folding from millions of predicted structures. In *International Conference on Machine Learning*, pp. 8946–8970. PMLR, 2022.

Hu, L., Wang, X., Huang, Y.-A., Hu, P., and You, Z.-H. A survey on computational models for predicting protein–protein interactions. *Briefings in bioinformatics*, 22(5): bbab036, 2021.

Huang, Y., Wu, L., Lin, H., Zheng, J., Wang, G., and Li, S. Z. Data-efficient protein 3d geometric pretraining via refinement of diffused protein structure decoy. *arXiv preprint arXiv:2302.10888*, 2023.

Huang, Y., Li, S., Wu, L., Su, J., Lin, H., Zhang, O., Liu, Z., Gao, Z., Zheng, J., and Li, S. Z. Protein 3d graph structure learning for robust structure-based protein property prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12662–12670, 2024.

Huber, P. J. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

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.

Joosten, R. P., Long, F., Murshudov, G. N., and Perrakis, A. The pdb_redo server for macromolecular structure model optimization. *IUCrJ*, 1(4):213–220, 2014.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kastritis, P. L. and Bonvin, A. M. On the binding affinity of macromolecular interactions: daring to ask why proteins interact. *Journal of The Royal Society Interface*, 10(79): 20120835, 2013.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kong, X., Huang, W., and Liu, Y. Conditional antibody design as 3d equivariant graph translation. *arXiv preprint arXiv:2208.06073*, 2022.

Lei, R., Garcia, A. H., Tan, T. J., Teo, Q. W., Wang, Y., Zhang, X., Luo, S., Nair, S. K., Peng, J., and Wu, N. C. Mutational fitness landscape of human influenza h3n2 neuraminidase. *Cell reports*, 42(1), 2023.

Li, M., Simonetti, F. L., Goncearenco, A., and Panchenko, A. R. Mutabind estimates and interprets the effects of sequence variants on protein–protein interactions. *Nucleic acids research*, 44(W1):W494–W501, 2016.

Li, S., Wang, Z., Liu, Z., Wu, D., Tan, C., Zheng, J., Huang, Y., and Li, S. Z. Vqdna: Unleashing the power of vector quantization for multi-species genomic sequence modeling. *arXiv preprint arXiv:2405.10812*, 2024.

Lin, H., Huang, Y., Zhang, O., Liu, Y., Wu, L., Li, S., Chen, Z., and Li, S. Z. Functional-group-based diffusion for pocket-specific molecule generation and elaboration. *Advances in Neural Information Processing Systems*, 36, 2024.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Liu, S., Zhu, T., Ren, M., Yu, C., Bu, D., and Zhang, H. Predicting mutational effects on protein-protein binding via a side-chain diffusion probabilistic model. *arXiv preprint arXiv:2310.19849*, 2023.

Lu, H., Zhou, Q., He, J., Jiang, Z., Peng, C., Tong, R., and Shi, J. Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials. *Signal transduction and targeted therapy*, 5(1):213, 2020.

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.

Luo, Y., Jiang, G., Yu, T., Liu, Y., Vo, L., Ding, H., Su, Y., Qian, W. W., Zhao, H., and Peng, J. Ecnet is an evolutionary context-integrated deep learning framework for protein engineering. *Nature communications*, 12(1): 5743, 2021.

Marchand, A., Van Hall-Beauvais, A. K., and Correia, B. E. Computational design of novel protein–protein interactions–an overview on methodological approaches and applications. *Current Opinion in Structural Biology*, 74:102370, 2022.

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.

Murphy, K. and Weaver, C. *Janeway's immunobiology*. Garland science, 2016.

Notin, P., Dias, M., Frazer, J., Hurtado, J. M., Gomez, A. N., Marks, D., and Gal, Y. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pp. 16990–17017. PMLR, 2022.

Park, H., Bradley, P., Greisen Jr, P., Liu, Y., Mulligan, V. K., Kim, D. E., Baker, D., and DiMaio, F. Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. *Journal of chemical theory and computation*, 12(12):6201–6212, 2016.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021.

Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F., and Serrano, L. The foldx web server: an online force field. *Nucleic acids research*, 33(suppl_2):W382–W388, 2005.

Shan, S., Luo, S., Yang, Z., Hong, J., Su, Y., Ding, F., Fu, L., Li, C., Chen, P., Ma, J., et al. Deep learning

guided optimization of human antibody against sars-cov-2 variants with broad neutralization. *Proceedings of the National Academy of Sciences*, 119(11):e2122954119, 2022.

Tan, C., Gao, Z., Wu, L., Xia, J., Zheng, J., Yang, X., Liu, Y., Hu, B., and Li, S. Z. Cross-gate mlp with protein complex invariant embedding is a one-shot antibody designer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15222–15230, 2024.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Wang, Z., Zhang, Q., HU, S.-W., Yu, H., Jin, X., Gong, Z., and Chen, H. Multi-level protein structure pre-training via prompt learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XGagtiJ8XC.

Wu, L., Huang, Y., Tan, C., Gao, Z., Hu, B., Lin, H., Liu, Z., and Li, S. Z. Psc-cpi: Multi-scale protein sequence-structure contrasting for efficient and generalizable compound-protein interaction prediction. *arXiv preprint arXiv:2402.08198*, 2024a.

Wu, L., Tian, Y., Huang, Y., Li, S., Lin, H., Chawla, N. V., and Li, S. MAPE-PPI: Towards effective and efficient protein-protein interaction prediction via microenvironment-aware protein embedding. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=itGkF993gz.

Yang, F., Fan, K., Song, D., and Lin, H. Graph-based prediction of protein-protein interactions with attributed signed graph embedding. *BMC bioinformatics*, 21(1): 1–16, 2020.

Yang, K. K., Zanichelli, N., and Yeh, H. Masked inverse folding with sequence transfer for protein representation learning. *bioRxiv*, 2022.

Appendix

A. E(3)-equivariant Graph Neural Networks

Since developing a new E(3)-equivariant architecture is not the focus of this paper, we directly adopt an E(3)-equivariant Graph Neural Network similar to MEAN (Kong et al., 2022) for updating node features and coordinates. Suppose the node feature and coordinates of residue v_i in the l-th layer are $\mathbf{h}_i^{(l)}$ and $\mathbf{Z}_i^{(l)}$, respectively. We denote the relative coordinates between residue v_i and v_j as $\mathbf{Z}_{i,j}^{(l)} = \mathbf{Z}_i^{(l)} - \mathbf{Z}_j^{(l)}$. Then, the message aggregation and updating of the l-th layer ($0 \leq l \leq L-1$) for node v_i can be defined as follows

$$\mathbf{m}_{i,j}^{(l)} = \phi_m\Big(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \frac{(\mathbf{Z}_{i,j}^{(l)})^\top \mathbf{Z}_{i,j}^{(l)}}{\big\|(\mathbf{Z}_{i,j}^{(l)})^\top \mathbf{Z}_{i,j}^{(l)}\big\|_F}, \mathbf{E}_{i,j}^{(l)}\Big), \tag{A.1}$$

$$\mathbf{h}_i^{(l+1)} = \phi_h\Big(\mathbf{h}_i^{(l)}, \sum_{j\in\mathcal{N}(i|\mathcal{E}_{\text{in}})} \mathbf{m}_{i,j}^{(l)}, \sum_{j\in\mathcal{N}(i|\mathcal{E}_{\text{ex}})} \mathbf{m}_{i,j}^{(l)}\Big), \tag{A.2}$$

$$\mathbf{E}_{i,j}^{(l+1)} = \phi_e\Big(\mathbf{h}_i^{(l+1)}, \mathbf{E}_{i,j}^{(l)}, \mathbf{h}_j^{(l+1)}\Big), \tag{A.3}$$

$$\mathbf{Z}_i^{(l+1)} = \mathbf{Z}_i^{(l)} + \frac{1}{|\mathcal{N}(i|\mathcal{E}_{\text{in,ex}})|} \sum_{j\in\mathcal{N}(i|\mathcal{E}_{\text{in,ex}})} \mathbf{Z}_{i,j}^{(l)} \phi_z(\mathbf{m}_{i,j}^{(l)}). \tag{A.4}$$

where $\mathcal{N}(i|\mathcal{E}_{\text{in}})$, $\mathcal{N}(i|\mathcal{E}_{\text{ex}})$, and $\mathcal{N}(i|\mathcal{E}_{\text{in,ex}})$ denote the neighbors of node v_i regarding the internal connections, external connections, and both. Besides, $\phi_m(\cdot)$, $\phi_h(\cdot)$, $\phi_e(\cdot)$, and $\phi_z(\cdot)$ are all implemented as one- or two-layer MLPs with $\text{SiLU}(\cdot)$ (Elfwing et al., 2018) as the activation function. Finally, we output $\widehat{\mathbf{O}}_i = \mathbf{Z}_i^{(0)} - \mathbf{Z}_i^{(L)}$ as the prediction on structural noise $\mathbf{O}_i$, i.e., local conformational changes.

B. Huber Loss Function

The Huber loss (Huber, 1992) helps to lead to a more stable training procedure, which is defined as follows:

$$l(x,y) = \begin{cases} 0.5(x-y)^2, & \text{if } |x-y| < \delta \\ \delta \cdot (|x-y| - 0.5\cdot\delta), & \text{else} \end{cases} \tag{A.5}$$

where we set $\delta = 1$ in this paper.

C. Comparison with Related Work

A recent work, MAPE-PPI (Wu et al., 2024b), is the first computational approach for microenvironment discovery and encoding. Our Prompt-PDG is partially inspired by it but differs from it in several aspects: (1) MAPE-PPI constructs one *single codebook* characterizing the sequence and structural context of the entire microenvironment, while our Prompt-DDG constructs a *hierarchical codebook* to separately record common microenvironmental patterns at three different structural scales, including 1D residue types, 2D geometric angles, and 3D backbone conformations. (2) MAPE-PPI encodes the *microenvironments around all residues* and utilizes the codebook to generate pre-trained representations for downstream tasks, while our Prompt-DDG encodes only the *microenvironments around mutations* and generates several prompts for $\Delta\Delta G$ prediction. (3) MAPE-PPI pre-trains the codebook via a *masked codebook* modeling task. However, our Prompt-DDG directly *masks the input microenvironment* (including its residue types, angular statistic, and conformational coordinates), and then trains each sub-codebook with an individual task, aimed at capturing the joint distribution of each residue mutation with three structural scales of the microenvironment.

Algorithm 1 Algorithm for the Prompt-DDG

Randomly initializing the parameters of microenvironment encoder, decoder, and prompt codebook $\mathcal{A}$.
 # Prompt Codebook Pre-training
for each iteration **do**
  Masking the microenvironment $\mathcal{G}_m$ as $\widetilde{\mathcal{G}}_m$.
  Encoding microenvironment $\widetilde{\mathcal{G}}_m$ around each mutation $m \in \mathcal{M}$ into representations $\mathbf{h}_m$ by Eq. (3);
  Performing vector quantization on $\mathbf{h}_m$ into prompt codes z_i by the prompt codebook $\mathcal{A}$ by Eq. (4);
  Reconstructing the inputs from prompt embeddings;
  Optimizing the encoder, decoder, and prompt codebook $\mathcal{A}$ jointly by minimizing the loss of Eq. (8).
end for
 # Prompt-Guided $\Delta\Delta G$ Prediction
Freezing the encoder $f_\theta(\cdot)$ and codebook $\mathcal{A}$, and randomly initializing the parameters of $\Delta\Delta G$ predictor.
for each iteration **do**
  Combining three prompts of different structural scales by a lightweight prompt adapter by Eq. (9).
  Add microenvironment-aware prompts to residues in the microenvironment around each mutation.
  Pooling the structural representations of wild-type and mutant complexes for predicting $\Delta\Delta G$s.
  Optimizing $\Delta\Delta G$ predictor by minimizing the MSE loss between the predicted and ground-truth $\Delta\Delta G$s.
end for
return Trained $\Delta\Delta G$ predictor.

D. Training Time Complexity Analysis

The training time complexity of Prompt-DDG comes from four parts: (1) microenvironment encoding $\mathcal{O}(|\mathcal{V}|F^2 + |\mathcal{E}|F)$; (2) vector quantization $\mathcal{O}(|\mathcal{M}|\cdot|\mathcal{A}|F)$; (3) prompt combination $\mathcal{O}(|\mathcal{M}|F)$; (4) $\Delta\Delta G$ prediction $\mathcal{O}(|\mathcal{V}|F^2 + |\mathcal{E}|F)$, where $|\mathcal{V}|$ and $|\mathcal{E}|$ are the number of nodes and edges, F is the hidden dimension, $|\mathcal{M}|$ is the number of mutations, and $|\mathcal{A}|$ is the size of codebook. The total training time complexity of Prompt-DDG is $\mathcal{O}(|\mathcal{V}|F^2 + |\mathcal{E}|F + |\mathcal{M}|\cdot|\mathcal{A}|F)$, which is linear with respect to all $|\mathcal{V}|$, $|\mathcal{E}|$, $|\mathcal{A}|$, and $|\mathcal{M}|$.

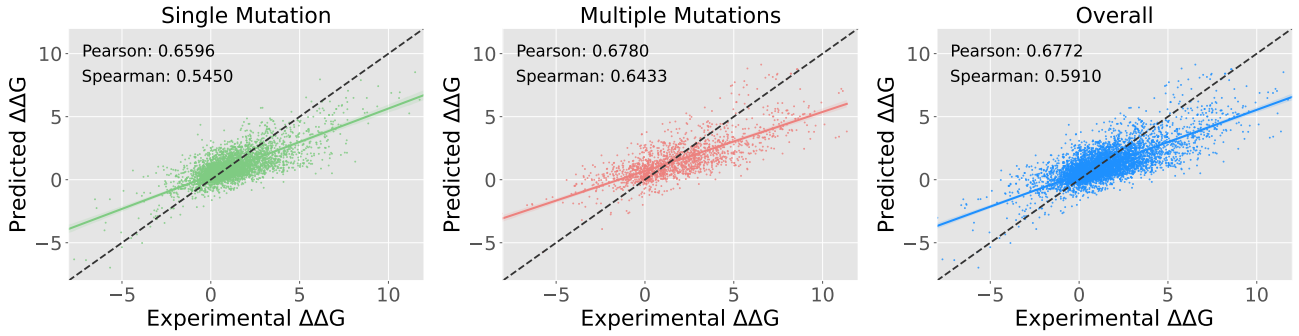

Figure A1. A visualization comparison of correlations between experimental $\Delta\Delta G$ and $\Delta\Delta G$ predicted by three mutation settings.

E. Pseudo-code of Prompt-DDG

The pseudo-code of the proposed Prompt-DDG framework for $\Delta\Delta G$ prediction is summarized in Algorithm 1.

F. Details about Baselines

The results of all baselines except DiffAffinity in Table. 1 and Table. 2 are copied from a previous work (Luo et al., 2023), which has provided details of various baseline implementations. We refer the interested reader directly to their descriptions in the subsection of *"A.1 Baselines Implementations"*. Besides, the results of DiffAffinity (Liu et al., 2023) are taken from their original paper. For a fair comparison, our Prompt-DDG adopts the same self-attention-based network from (Jumper et al., 2021) as RDE (Luo et al., 2023) does. The only difference is that Prompt-DDG adapts it to graph data by restricting the original global attention computation and message passing to the local microenvironment.

G. Hyperparameters and Implementation Details

The following hyperparameters are determined by an AutoML toolkit NNI with the hyperparameter search spaces as: Adam optimizer (Kingma & Ba, 2014) with $lr = 0.0003$, batch size $B = 32$, codebook iteration $T_{\text{code}} = 2,000$, $\Delta\Delta G$ iteration $T_{\Delta\Delta G} = 7,000$, thresholds $d_s = 2$, $d_r = 15\mathring{A}$, and neighbor number $K = 15$ for graph construction, hidden dimension $F = \{128, 256\}$, codebook size $|\mathcal{A}| = \{128, 256, 512\}$, mask ratio $r = \{0.1, 0.2\}$, and loss weights $\eta = 0.25$, $\lambda = \{0.01, 0.001\}$. In addition, $\{\phi_{\omega}^{(k)}(\cdot)\}_{k=1}^{3}$ are implemented as one-layer linear transformation. Besides, We crop structures into patches containing 128 residues by first choosing a seed residue, and then selecting its 127 nearest neighbors based on C-beta distances.

H. More Correlation Visualizations

The visualizations of Prompt-DDG for single-point, multi-point, and all-point mutations are provided in Figure. A1.