SE(3) denoising score matching for unsupervised binding energy prediction and nanobody design

Wengong Jin*, Xun Chen*, Amrita Vetticaden, Siranush Sarzikova,
Raktima Raychowdhury, Caroline Uhler[†], Nir Hacohen[†]
Broad Institute of MIT and Harvard

Abstract

Modeling the binding between proteins and other molecules is pivotal to drug discovery. Geometric deep learning is a promising paradigm for protein-ligand/protein-protein binding energy prediction, but its accuracy is limited by the size of training data as high-throughput binding assays are expensive. Herein, we propose an unsupervised binding energy prediction framework, named DSMBind, which does not need experimental binding data for training. DSMBind is an energy-based model that estimates the likelihood of a protein complex via SE(3) denoising score matching (DSM). This objective, applied at both backbone and side-chain levels, builds on a novel equivariant rotation prediction network derived from Euler's Rotation Equations. We find that the learned log-likelihood of protein complexes is highly correlated with experimental binding energy across multiple benchmarks, even matching the performance of supervised models trained on experimental data. We further demonstrate DSMBind's zero-shot binder design capability through a PD-L1 nanobody design task, where we randomize all three complementarity-determining regions (CDRs) and select the best CDR sequences based on DSMBind score. We experimentally tested the designed nanobodies with ELISA binding assay and successfully discovered a novel PD-L1 binder. In summary, DSMBind offers a versatile framework for binding energy prediction and binder design.

1 Introduction

The binding of proteins and other molecules forms the basis of many biological processes. Accurate prediction of binding affinity allows us, for instance, to engineer new drugs for therapeutic targets, design neutralizing antibodies for infectious diseases, and identify mutations that cause certain diseases. However, predicting binding affinity from structures remains challenging, especially when experimental data is lacking. Ideally, we would like to have an unsupervised learning approach that does not require any training data so that we can design molecules or proteins for any target.

Existing unsupervised binding energy prediction approaches, however, are either too expensive or inaccurate. Traditional physics-based models [18] use molecular dynamics to calculate the binding energy of a protein complex. However, it usually takes 4-6 hours to run molecular dynamics for one molecule. Thus, they are rarely used in large-scale virtual screening projects needed for drug discovery. More recently, unsupervised protein language models (PLMs) [6, 17] reveal that the learned likelihood of protein sequences are useful for predicting protein mutation effects. However, PLMs only work for protein sequences and are not applicable to protein-ligand binding (small molecules). More importantly, binding interface structures contain more information than sequences and its likelihood should be a better indicator of binding.

In this work, we propose DSMBind, an unsupervised binding energy prediction framework for both small molecules and proteins. The basic idea is to learn an energy-based model (EBM) that

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

maximizes the log-likelihood (or minimizing the energy) of crystal structures in a training set. This objective is implemented by a novel SE(3) denoising score matching (DSM) objective that extends the standard DSM objective [23]. In each training step, we first perturb a protein complex (crystal structure) by randomly rotating one of the proteins (or ligands) and its side-chain atoms. We then use the DSM objective to shape the energy function so that its gradient recovers the injected rotation noise. Formally speaking, this gradient matching procedure minimizes the Fisher divergence between the learned and true binding energy, even though we don't know the true binding energy of a given complex. At test time, we use the learned energy to compare different proteins or ligands to find or design the best binder for a given target.

We validate DSMBind on four benchmarks related to protein-ligand binding, antibody-antigen binding, and protein-protein binding mutation effect prediction. We compare DSMBind with state-of-the-art binding prediction models including protein language models (ESM) [6, 17], physics-based models [1, 4], and supervised deep learning models [16, 15, 28] trained on experimental binding affinity data in the public domain. We also explore alternative EBM training objectives (e.g. Gaussian DSM and contrastive learning) to verify the advantage of our SE(3) DSM objective. We find that DSMBind outperforms most of the unsupervised baselines and matches the performance of supervised models despite not using any binding affinity labels during training. We further showcase DSMBind's zero-shot design capability through a PD-L1 nanobody design task. We experimentally evaluated 48 designed nanobodies, where one of them showed positive, PD-L1 specific binding in ELISA. Our code is publicly available at `github.com/wengong-jin/DSMBind`.

2 Unsupervised Binding Energy Prediction

Notation. A protein-ligand (or a protein-protein) complex is denoted as a tuple $(\boldsymbol{A}, \boldsymbol{X})$, with atom features $\boldsymbol{A} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n]$ and atom coordinates $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]$ (column-wise concatenation). The key property of a complex is its binding energy $E(\boldsymbol{A}, \boldsymbol{X})$. A lower binding energy means a ligand binds more strongly to a protein. In this section, we describe how to parameterize $E(\boldsymbol{A}, \boldsymbol{X})$ and design proper training objectives to infer the true binding energy function from a list of crystal structures without binding affinity labels.

Energy function architecture. An energy function $E(\boldsymbol{A}, \boldsymbol{X})$ is composed of a protein encoder and an output layer. The encoder is a frame averaging neural network [19] that learns a SE(3)-invariant representation $\boldsymbol{h}_i$ for each atom. We choose this architecture because of its simplicity (detailed in the appendix). The output layer ϕ_o is a feed-forward neural network with one hidden layer. It predicts the interaction energy $\phi_o(\boldsymbol{h}_i, \boldsymbol{h}_j)$ for each pair of atoms. Finally, we define $E(\boldsymbol{A}, \boldsymbol{X})$ as the sum of pairwise interaction energies: $E(\boldsymbol{A}, \boldsymbol{X}) = \sum_{i,j:d_{ij}<d} \phi_o(\boldsymbol{h}_i, \boldsymbol{h}_j)$. Since atomic interaction vanishes beyond certain distance, we only consider atom pairs with distance $d_{ij} < d$. So far, we have described our method in generic terms. We now specify the input features and preprocessing steps tailored to small molecules and antibodies.

DSMBind algorithm. Given that our training set does not have binding affinity labels, we need to design an unsupervised training objective different from a supervised regression loss. Our key hypothesis is that we can infer the true binding energy function (up to affine equivalence) by maximizing the likelihood of crystal structures in our training set. The motivation of our hypothesis is that a crystal structure is the lowest energy state of a protein-ligand complex. The maximum likelihood objective seeks to minimize the energy of crystal structures since the likelihood of a complex is $p(\boldsymbol{A}, \boldsymbol{X}) \propto \exp(-E(\boldsymbol{A}, \boldsymbol{X}))$.

While maximum likelihood estimation (MLE) is difficult for EBMs due to marginalization, recent works [23, 24] has successfully trained EBMs using denoising score matching (DSM) and proved that DSM is a good approximation of MLE. In standard DSM, we create a perturbed complex by adding Gaussian noise to ligand atom coordinates. However, adding Gaussian noise is not ideal for protein complexes because it creates nonsensical conformations that violate physical constraints. A better solution is to create a perturbed complex $(\boldsymbol{A}, \tilde{\boldsymbol{X}})$ via random rotation and translation. Given a protein complex, our perturbation procedure consists of two steps (Figure 1):

1. **Backbone rotation/translation**: Sample a random translation $\boldsymbol{t} \sim p(\boldsymbol{t}) = \mathcal{N}(0, \boldsymbol{I})$ and a random rotation vector $\boldsymbol{\omega} \sim p(\boldsymbol{\omega}) = \mathcal{N}_{SO(3)}$, an isotropic Gaussian distribution over the SO(3) rotation group [12]. Apply rigid transformation to the entire ligand (including backbone and side-chain atoms). This step transform the ligand as a rigid body.

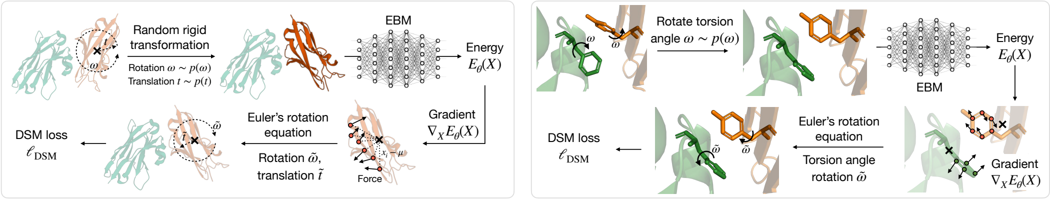
Random rigid transformation
Rotation $\omega \sim p(\omega)$
Translation $t \sim p(t)$
EBM
Energy $E_\theta(X)$
Gradient $\nabla_X E_\theta(X)$
$x_i - \mu$
Force
Euler's rotation equation
Rotation $\tilde{\omega}$, translation $\tilde{t}$
t
ω
$\tilde{t}$
$\tilde{\omega}$
DSM loss ℓ_{DSM}
Rotate torsion angle $\omega \sim p(\omega)$
EBM
Energy $E_\theta(X)$
Gradient $\nabla_X E_\theta(X)$
Euler's rotation equation
Torsion angle rotation $\tilde{\omega}$
DSM loss ℓ_{DSM}

Figure 1: DSMBind training algorithm. Left: Backbone translation/rotation DSM procedure. Right: Side-chain rotation DSM procedure.

Algorithm 1 DSMBind Training Procedure (single step)

```
Require: A training complex $(\boldsymbol{A}, \boldsymbol{X})$.
1: Sample a noise level $\sigma$.
2: Sample backbone rotation vector $\boldsymbol{\omega} \sim \mathcal{N}_{SO(3)}$ with variance $\sigma^2$
3: Sample backbone translation vector $\boldsymbol{t} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.
4: Sample a side-chain rotation vector $\boldsymbol{\chi}_i \sim \mathcal{N}_{SO(3)}$ for each residue $i$.
5: Perturb the coordinates $\tilde{\boldsymbol{X}}$ by applying rigid transformation $(\boldsymbol{\omega}, \boldsymbol{t}, \{\boldsymbol{\chi}_i\})$ to the ligand.
6: Compute the score of energy function $(\tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{t}}, \{\tilde{\boldsymbol{\chi}}_i\})$ based on its gradient (force) $-\partial E / \partial \tilde{\boldsymbol{X}}$.
7: Minimize DSM objective $\ell_{\text{se3}}$.
```

2. **Side-chain rotation**: Sample a rotation vector $\boldsymbol{\chi}_i \sim p(\boldsymbol{\chi}_i) = \mathcal{N}_{SO(3)}$ for each ligand residue i. Rotate the side-chain of each residue by $\boldsymbol{\chi}_i$. Each side-chain group is rotated independently.

Under this perturbation scheme, DSM aims to match the model score $-\partial E / \partial \tilde{\boldsymbol{X}}$ with the score of backbone rotation, translation, and side-chain rotation noise $\nabla_{\boldsymbol{\omega}} \log p(\boldsymbol{\omega}), \nabla_{\boldsymbol{t}} \log p(\boldsymbol{t}), \nabla_{\boldsymbol{\chi}_i} \log p(\boldsymbol{\chi}_i)$. Our SE(3) DSM objective is a sum of three losses: $\ell_{\text{se3}} = \ell_t + \ell_r + \ell_s$, where ℓ_t, ℓ_r, and ℓ_s correspond to the translation, rotation DSM, and side-chain DSM loss. The translation DSM is straightforward since $\boldsymbol{t}$ follows a normal distribution and $\nabla_{\boldsymbol{t}} \log p(\boldsymbol{t}) = -\boldsymbol{t}/\sigma^2$:

$$\ell_t = \mathbb{E}\big[\|\tilde{\boldsymbol{t}} - \nabla_{\boldsymbol{t}} \log p(\boldsymbol{t})\|^2\big], \qquad \tilde{\boldsymbol{t}} = -\sum\nolimits_i \partial E / \partial \tilde{\boldsymbol{x}}_i \tag{1}$$

For the rotation DSM, we sample random rotation $\boldsymbol{\omega} = \theta \hat{\boldsymbol{\omega}}$. As $\hat{\boldsymbol{\omega}}$ is sampled from a uniform distribution over a sphere (whose density is constant), the density and score of $p(\boldsymbol{\omega})$ is

$$p(\boldsymbol{\omega}) \propto f(\theta), \quad \nabla_{\boldsymbol{\omega}} \log p(\boldsymbol{\omega}) = \nabla_\theta \log f(\theta) \cdot \hat{\boldsymbol{\omega}} \tag{2}$$

In practice, we calculate the density and score by precomputing truncated infinite series in $f(\theta)$. However, the main challenge is that the model score $-\partial E / \partial \tilde{\boldsymbol{X}} \in \mathbb{R}^{n \times 3}$ is defined over atom coordinates, which is not directly comparable with $\nabla_{\boldsymbol{\omega}} \log p(\boldsymbol{\omega}) \in \mathbb{R}^3$ as they have different dimensions. To address this issue, we map $-\partial E / \partial \tilde{\boldsymbol{X}}$ to a rotation vector $\tilde{\boldsymbol{\omega}}$ using the Neural Euler's Rotation Equation F_{NERE} (defined in the appendix) and perform DSM in the rotation space:

$$\ell_r = \mathbb{E}\big[\|\tilde{\boldsymbol{\omega}} - \nabla_{\boldsymbol{\omega}} \log p(\boldsymbol{\omega})\|^2\big], \qquad \tilde{\boldsymbol{\omega}} = F_{\text{NERE}}(-\partial E / \partial \tilde{\boldsymbol{X}}) \tag{3}$$

For the side-chain DSM, we sample random rotation $\boldsymbol{\chi}_i \sim \mathcal{N}_{SO(3)}$ for each residue i and rotate its side chain by $\boldsymbol{\chi}_i$. The side-chain DSM loss is defined as

$$\ell_s = \sum\nolimits_i \mathbb{E}\big[\|\tilde{\boldsymbol{\chi}}_i - \nabla_{\boldsymbol{\chi}_i} \log p(\boldsymbol{\chi}_i)\|^2\big], \qquad \tilde{\boldsymbol{\chi}}_i = F_{\text{NERE}}(-\partial E / \partial \tilde{\boldsymbol{X}}_{A(i)}) \tag{4}$$

where $A(i)$ is the set of side-chain atoms belonging to residue i and $\tilde{\boldsymbol{X}}_{A(i)}$ represents their coordinate matrix. The overall training procedure is summarized in Algorithm 1.

3 Experiments

Protein-ligand binding prediction. We test the performance of DSMBind on a free energy perturbation benchmark (FEP) developed by Merck [20]. This benchmark has 264 protein-ligand complexes targeting eight proteins (cdk8, cmet, eg5, hif2a, pfkfb3, shp2, syk, and tnks2). Each protein-ligand complex is predicted by Glide core-constrained docking and labeled by experimental binding affinity.

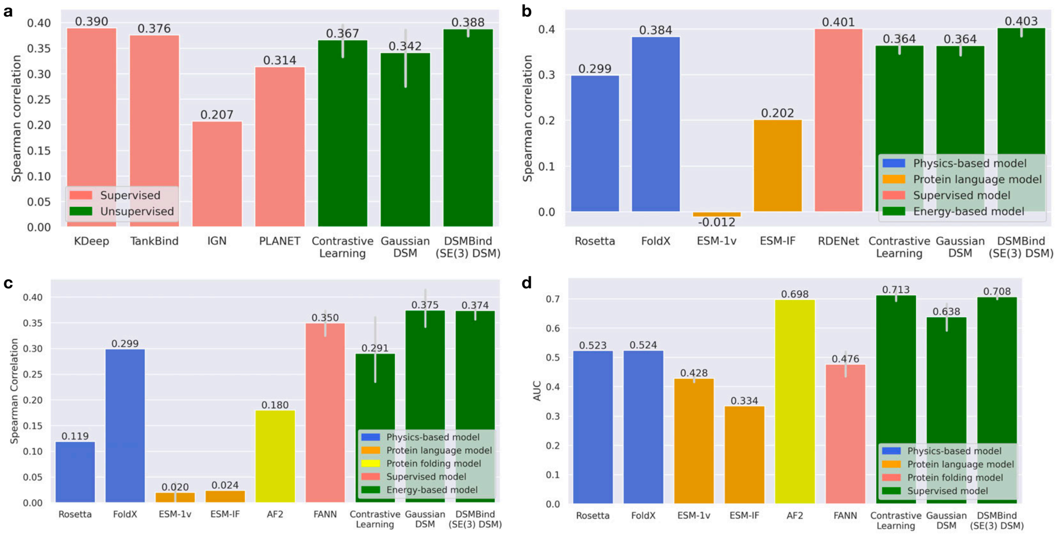

Figure 2: Results on four benchmarks related to (a) protein-ligand binding, (b) protein-protein binding mutation effect, and (c-d) antibody-antigen binding prediction.

We train DSMBind on 4806 protein-ligand complexes from the refined subset of PDBbind v2020 database [26] with their binding affinity labels excluded. We removed all instances from the training set whose ligand appeared the validation and test set. We calculate the Spearman correlation between the learned binding energy and experimental binding affinity for each target and report the average. We compare with four supervised learning baselines (KDeep [9], IGN [8], TankBind [15], and PLANET [28]) and two alternative EBM training methods (contrastive learning and Guassian DSM). As shown in Figure 2, DSMBind outperforms all unsupervised baselines and matches the best supervised baseline (KDeep), although it is not trained on any binding affinity labels.

Protein-protein binding mutation effect ($\Delta\Delta G$) **prediction**. We test the performance of DSMBind on SKEMPI [7]. It has 348 protein complexes and approximately 6000 $\Delta\Delta G$ data points. The training set of DSMBind has approximately 27000 non-redundant protein-protein complexes downloaded from PDB (with no binding affinity labels). We compare DSMBind against two physics-based models (Rosetta [1] and FoldX [4]), two protein-language model baselines (ESM-1v [17] and ESM-IF [6]), and the state-of-the-art supervised model (RDENet [16] trained on the SKEMPI database). We select 10% of the SKEMPI data for validation and the rest of 90% for testing. We run the model with five random seeds and report the average per-target Spearman correlation on the test set. As shown in Figure 2b, DSMBind outperforms all unsupervised baselines and matches the best supervised baseline (RDENet), although it is not trained on any $\Delta\Delta G$ labels.

Antibody-antigen binding prediction. We test DSMBind's performance on two benchmarks. The first test set comes from the Structural Antibody Database (SAbDab) [21], which has 566 antibody-antigen complexes that have binding affinity labels. The rest of the complexes in SAbDab (those without binding affinity labels) form our training set. After removing antigen/antibody sequences that appear in the test set, our training set has 3416 non-redundant complexes. We compare DSMBind with the same set of unsupervised baselines in the SKEMPI benchmark, with an additional baseline based on AlphaFold2 (AF2) [10]. Recently, Bennett et al. [2] discovered that AF2's predicted aligned error (PAE) is an useful indicator of protein binding. The supervised baseline (FANN) is trained on all the binding affinity labels in SKEMPI except those appearing in the test set. We run the model with five random seeds and report the average Spearman correlation on the test set. As shown in Figure 2c, DSMBind achieves the state-of-the-art performance on this benchmark.

The second test set comes from an HER2 binder design assay [22]. It has 424 designed variants of trastuzumab with experimentally measured binding affinity to the HER2 antigen. The structure of trastuzumab-HER2 complex has been crystallized (PDB: 1n8z). Among the 424 designed antibodies, four of them showed better binding affinity than wild type trastuzumab. We use this benchmark as a classification task, with the four successful designs as positive cases and the others as negative cases. As shown in Figure 2d, DSMBind achieves the best classification performance (AUROC) on this benchmark, i.e. it is able to identify variants with improved binding affinity more accurately.

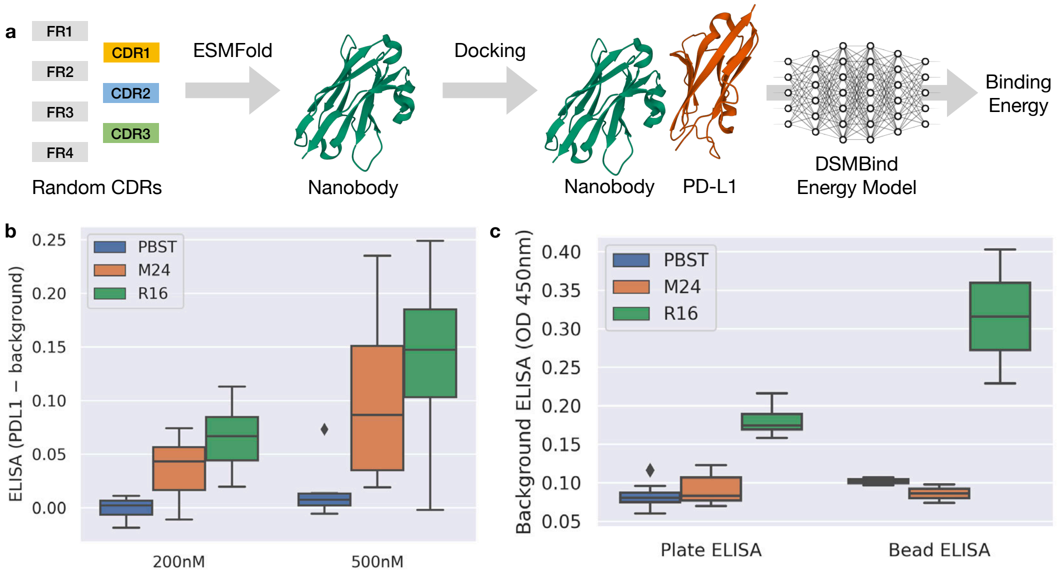

Figure 3: a) The workflow of DSMBind for PD-L1 nanobody design. b) ELISA binding assay results of R16, M24, and PBST (negative control). R16 and M24 binds PD-L1 at 200nM concentration (p-value < 0.005 when compared with PBST). c) Background binding of R16, M24, and PBST (ELISA OD 450nm) suggests that M24 binding is PD-L1 specific (low background binding).

4 Using DSMBind for PD-L1 nanobody design

The use of antibodies/nanobodies to target immune checkpoints, particularly PD-1/PD-L1, has made a profound impact in the field of cancer immunotherapy. Herein, we apply DSMBind to a PD-L1 nanobody design task to explore its zero-shot design capability. Unlike antibodies, nanobodies have only three CDRs because they do not have a light chain. Therefore, we seek to computationally design CDR1, CDR2, and CDR3 sequences simultaneously.

Our design workflow is illustrated in Figure 3a. Given a nanobody with random CDR sequences, we predict its 3D structure with ESMFold [14] and apply template-based docking to predict a PD-L1-nanobody complex for each sequence. The template structure is a crystallized PD-L1-nanobody complex in PDB (ID: 5jds). The docked structure is then passed to the DSMBind model to calculate the binding energy of a nanobody. We calculate the DSMBind score for each docked structure and select the best 48 sequences for experimental validation. We clone and purify the selected nanobodies and test their binding with PD-L1 using enzyme-linked immunosorbent assay (ELISA) (10 replicates). In each replicate, we measured the binding activity as the difference between the ELISA reading from PD-L1 coated plates and from plates without coating (background). As shown in Figure 3b, two of the tested nanobodies (R16 and M24) bind PD-L1 at 200nM and 500nM (background subtracted ELISA > 0.02), where R16 comes from the first library (rank 16) and M24 comes from the second library (rank 24). The binding of both nanobodies are statistically significant (p-value < 0.005 compared to negative control PBST). The designed CDRs are novel because their sequence identity between R16/M24 and all antibodies in the training set is lower than 30%.

Next, we compare the background ELISA reading of R16 and M24 with PBST to investigate if any of these signals come from non-specific binding (e.g., binding to the plate). As shown in Figure 3c, M24 and PBST has low background binding as expected while R16 has a much higher background binding, which suggest that R16 may be a non-specific binder even though its binding to PD-L1 is stronger than the background. To further analyze the background binding of R16 and M24, we implemented a more sensitive ELISA assay based on magnetic beads instead of plates. Different from R16, M24 still shows low background binding under this bead-based ELISA (Figure 3c), which suggests that M24 is a PD-L1-specific binder. In summary, these results provide a proof-of-concept for DSMBind's nanobody design capability. To systematically evaluate the performance of DSMBind, we plan to experimentally test much larger collections of designed nanobodies and random nanobodies.

Acknowledgements

We would like to thank the generous support from the BroadIgnite Award and the Eric and Wendy Schmidt Center at Broad Institute.

References

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

[2] N. R. Bennett, B. Coventry, I. Goreshnik, B. Huang, A. Allen, D. Vafeados, Y. P. Peng, J. Dauparas, M. Baek, L. Stewart, et al. Improving de novo protein binder design with deep learning. *Nature Communications*, 14(1):2625, 2023.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

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

[5] R. A. Friesner, R. B. Murphy, M. P. Repasky, L. L. Frye, J. R. Greenwood, T. A. Halgren, P. C. Sanschagrin, and D. T. Mainz. Extra precision glide: Docking and scoring incorporating a model of hydrophobic enclosure for protein- ligand complexes. *Journal of medicinal chemistry*, 49(21):6177–6196, 2006.

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

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

[8] D. Jiang, C.-Y. Hsieh, Z. Wu, Y. Kang, J. Wang, E. Wang, B. Liao, C. Shen, L. Xu, J. Wu, et al. Interactiongraphnet: A novel and efficient deep graph representation learning framework for accurate protein–ligand interaction predictions. *Journal of medicinal chemistry*, 64(24): 18209–18232, 2021.

[9] J. Jiménez, M. Skalic, G. Martinez-Rosell, and G. De Fabritiis. K–deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

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

[11] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[12] A. Leach, S. M. Schmon, M. T. Degiacomi, and C. G. Willcocks. Denoising diffusion probabilistic models on so (3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.

[13] T. Lei. When attention meets fast recurrence: Training language models with reduced compute. *arXiv preprint arXiv:2102.12459*, 2021.

[14] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[15] W. Lu, Q. Wu, J. Zhang, J. Rao, C. Li, and S. Zheng. Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. *bioRxiv*, 2022.

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

[17] J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

[18] B. R. Miller III, T. D. McGee Jr, J. M. Swails, N. Homeyer, H. Gohlke, and A. E. Roitberg. Mmpbsa.py: an efficient program for end-state free energy calculations. *Journal of chemical theory and computation*, 8(9):3314–3321, 2012.

[19] O. Puny, M. Atzmon, H. Ben-Hamu, E. J. Smith, I. Misra, A. Grover, and Y. Lipman. Frame averaging for invariant and equivariant network design. *arXiv preprint arXiv:2110.03336*, 2021.

[20] C. E. Schindler, H. Baumann, A. Blum, D. Bose, H.-P. Buchstaller, L. Burgdorf, D. Cappel, E. Chekler, P. Czodrowski, D. Dorsch, et al. Large-scale assessment of binding free energy calculations in active drug discovery projects. *Journal of Chemical Information and Modeling*, 60(11):5457–5474, 2020.

[21] C. Schneider, M. I. Raybould, and C. M. Deane. Sabdab in the age of biotherapeutics: updates including sabdab-nano, the nanobody structure tracker. *Nucleic acids research*, 50(D1):D1368–D1372, 2022.

[22] A. Shanehsazzadeh, S. Bachas, M. McPartlon, G. Kasun, J. M. Sutton, A. K. Steiger, R. Shuai, C. Kohnert, G. Rakocevic, J. M. Gutierrez, et al. Unlocking de novo antibody design with generative artificial intelligence. *bioRxiv*, pages 2023–01, 2023.

[23] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] Y. Song, C. Durkan, I. Murray, and S. Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021.

[25] H. Stärk, O. Ganea, L. Pattanaik, R. Barzilay, and T. Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning*, pages 20503–20521. PMLR, 2022.

[26] M. Su, Q. Yang, Y. Du, G. Feng, Z. Liu, Y. Li, and R. Wang. Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59(2):895–913, 2018.

[27] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[28] X. Zhang, H. Gao, H. Wang, Z. Chen, Z. Zhang, X. Chen, Y. Li, Y. Qi, and R. Wang. Planet: A multi-objective graph neural network model for protein-ligand binding affinity prediction. *bioRxiv*, pages 2023–02, 2023.

A Appendix

A.1 Neural Euler's Rotation Equations (NERE)

NERE is a crucial component of SE(3) DSM that converts the gradient of a binding energy function $E(\boldsymbol{A}, \boldsymbol{X})$ to a rotation matrix. In classical mechanics, Euler's rotation equation is a first-order ordinary differential equation that describes the rotation of a rigid body. Suppose a ligand rotates around its center mass $\boldsymbol{\mu}$ with angular velocity $\boldsymbol{\omega}$. Euler's rotation equation in an inertial reference frame is defined as

$$\boldsymbol{I}_N \frac{d\boldsymbol{\omega}}{dt} = \boldsymbol{\tau}, \qquad \boldsymbol{\tau} = \sum_i (\boldsymbol{x}_i - \boldsymbol{\mu}) \times \boldsymbol{f}_i \tag{5}$$

$$\boldsymbol{I}_N = \sum_i \|\boldsymbol{x}_i - \boldsymbol{\mu}\|^2 \boldsymbol{I} - (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top \tag{6}$$

where $\boldsymbol{I}_N \in \mathbb{R}^{3\times 3}$ is the inertia matrix of a ligand, $\boldsymbol{\tau}$ is the torque it received, and $\boldsymbol{f}_i$ is the force applied to a ligand atom i. The force is defined as the gradient of a binding energy function $\boldsymbol{f}_i = -\partial E(\boldsymbol{A}, \boldsymbol{X}) / \partial \boldsymbol{x}_i$. The inertia matrix describes the mass distribution of a ligand and the torque needed for a desired angular acceleration. For a short period of time Δt, we can approximate the new angular velocity $\boldsymbol{\omega}_{t=\Delta t}$ by

$$\frac{d\boldsymbol{\omega}}{dt} \approx \frac{\boldsymbol{\omega}_{t=\Delta t} - \boldsymbol{\omega}_{t=0}}{\Delta t} = \boldsymbol{I}_N^{-1} \boldsymbol{\tau} \tag{7}$$

Since we assume the system is in an inertial reference frame ($\boldsymbol{\omega}_{t=0} = 0$), we have $\boldsymbol{\omega}_{t=\Delta t} = \boldsymbol{I}_N^{-1} \boldsymbol{\tau} \Delta t$ (we set $\Delta t = 0.1$). We note that calculating the inverse $\boldsymbol{I}_N^{-1}$ is cheap because it is a 3×3 matrix. In summary, NERE is a function that converts the force to a rotation vector $\boldsymbol{\omega}$.

$$\boldsymbol{\omega} = F_{\mathrm{NERE}}(-\partial E(\boldsymbol{A}, \boldsymbol{X}) / \partial \boldsymbol{X}) = \boldsymbol{I}_N^{-1} \boldsymbol{\tau} \Delta t \tag{8}$$

A.2 Random Rigid Transformations

To construct a random rotation, we sample a rotation vector $\boldsymbol{\omega}$ from $\mathcal{N}_{SO(3)}$, an isotropic Gaussian distribution over $SO(3)$ rotation group [12] with variance σ^2. Each $\boldsymbol{\omega} \sim \mathcal{N}_{SO(3)}$ has the form $\boldsymbol{\omega} = \theta \hat{\boldsymbol{\omega}}$, where $\hat{\boldsymbol{\omega}}$ is a vector sampled uniformly from a unit sphere and $\theta \in [0, \pi]$ is a rotation angle with density

$$f(\theta) = \frac{1 - \cos\theta}{\pi} \sum_{l=0}^{\infty} (2l+1) e^{-l(l+1)\sigma^2} \frac{\sin((l+1/2)\theta)}{\sin(\theta/2)} \tag{9}$$

Likewise, we sample a random translation vector $\boldsymbol{t}$ from a normal distribution $\boldsymbol{t} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Finally, we apply this rigid transformation to the ligand and compute its perturbed coordinates $\tilde{\boldsymbol{X}} = \boldsymbol{R}_{\boldsymbol{\omega}} \boldsymbol{X} + \boldsymbol{t}$, where $\boldsymbol{R}_{\boldsymbol{\omega}}$ is the rotation matrix given by the rotation vector $\boldsymbol{\omega} = (\boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z)$.

$$\boldsymbol{R}_{\boldsymbol{\omega}} = \exp(\boldsymbol{W}_{\boldsymbol{\omega}}), \ \boldsymbol{W}_{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\boldsymbol{\omega}_z & \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z & 0 & -\boldsymbol{\omega}_x \\ -\boldsymbol{\omega}_y & \boldsymbol{\omega}_x & 0 \end{pmatrix}. \tag{10}$$

Here $\exp$ means matrix exponentiation and $\boldsymbol{W}_{\boldsymbol{\omega}}$ is an infinitesimal rotation matrix. Since $\boldsymbol{W}_{\boldsymbol{\omega}}$ is a skew symmetric matrix, its matrix exponential has the following closed form

$$\boldsymbol{R}_{\boldsymbol{\omega}} = \exp(\boldsymbol{W}_{\boldsymbol{\omega}}) = \boldsymbol{I} + c_1 \boldsymbol{W}_{\boldsymbol{\omega}} + c_2 \boldsymbol{W}_{\boldsymbol{\omega}}^2 \tag{11}$$

$$c_1 = \frac{\sin\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|}, \quad c_2 = \frac{1 - \cos\|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2} \tag{12}$$

Moreover, we do not need to explicitly compute the matrix exponential $\boldsymbol{R}_{\boldsymbol{\omega}}$ since $\boldsymbol{W}_{\boldsymbol{\omega}}$ is the linear mapping of cross product, i.e. $\boldsymbol{\omega} \times \boldsymbol{r} = \boldsymbol{W}_{\boldsymbol{\omega}} \boldsymbol{r}$. Therefore, applying a rotation matrix only involves cross product operations that are very efficient:

$$\boldsymbol{R}_{\boldsymbol{\omega}} \boldsymbol{x}_i = \boldsymbol{x}_i + c_1 \boldsymbol{\omega} \times \boldsymbol{x}_i + c_2 \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{x}_i) \tag{13}$$

A.3 EBM Architecture

We parameterize $E(\boldsymbol{A}, \boldsymbol{X})$ as an energy-based model (EBM), which consists of a protein encoder and an output layer. The encoder is a frame averaging neural network (FANN) [19] that learns a SE(3)-invariant representation $\boldsymbol{h}_i$ for each atom:

$$\boldsymbol{H} = [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n] = \frac{1}{|\mathcal{F}(\boldsymbol{X})|} \sum_{\boldsymbol{U} \in \mathcal{F}(\boldsymbol{X})} \phi_h(\boldsymbol{A}, (\boldsymbol{X} - \boldsymbol{\mu})\boldsymbol{U}) \tag{14}$$

ϕ_h is a recurrent neural network with a simple recurrent unit (SRU) and self attention [13]. In the input layer, we project the coordinate matrix $\boldsymbol{X}$ onto a set of eight frames $\boldsymbol{U} \in \mathcal{F}(\boldsymbol{X})$ constructed by Principal Component Analysis (PCA). Suppose $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3$ are the three principle components of a covariance matrix $\Sigma = (\boldsymbol{X} - \boldsymbol{\mu})^\top (\boldsymbol{X} - \boldsymbol{\mu})$ ($\boldsymbol{\mu}$ is the center mass of $\boldsymbol{X}$). The frame set $\mathcal{F}(\boldsymbol{X})$ is defined as

$$\mathcal{F}(\boldsymbol{X}) = \{[\pm\boldsymbol{u}_1, \pm\boldsymbol{u}_2, \pm\boldsymbol{u}_3]\} \tag{15}$$

The frame averaging operation at the end ensures the encoded representation $\boldsymbol{h}_i$ is SE(3)-invariant. The output layer ϕ_o is a feed-forward neural network with one hidden layer. It predicts the interaction energy $\phi_o(\boldsymbol{h}_i, \boldsymbol{h}_j)$ for each pair of atoms. Finally, we define $E(\boldsymbol{A}, \boldsymbol{X})$ as the sum of pairwise interaction energies:

$$E(\boldsymbol{A}, \boldsymbol{X}) = \sum_{(i,j):d_{ij} < d} \phi_o(\boldsymbol{h}_i, \boldsymbol{h}_j) \tag{16}$$

Since atomic interaction vanishes beyond certain distance, we only need to consider atom pairs in the binding interface (with distance $d_{ij} < d$). The binding interface and atom features are defined as follows:

- For a protein-ligand complex, the binding interface includes the entire ligand and top 50 protein residues closest to the ligand. On the protein side, each atom is represented by a one-hot encoding of its atom name (C_α, C_β, N, O, etc.) and a 2560-dimensional residue embedding learned by ESM-2 [14]. On the ligand side, the atom features are learned by a message passing network (MPN) [27] based on the ligand molecular graph. The MPN and EBM are optimized jointly during training.
- For an antibody-antigen complex, the binding interface consists of residues in the antibody complementarity determining region (CDR) and top 50 antigen residues closest to the CDR. All atoms are represented by the one-hot encoding of its atom name and ESM-2 embedding.
- For a protein-protein complex, we crop each protein to the top 50 residues closet to the other protein. All atoms are represented by the one-hot encoding of its atom name and ESM-2 embedding.

A.4 Contrastive Learning Baseline

The maximum likelihood objective seeks to minimize the energy of crystal structures where the likelihood of a complex is $p(\boldsymbol{A}, \boldsymbol{X}) \propto \exp(-E(\boldsymbol{A}, \boldsymbol{X}))$. However, maximum likelihood estimation (MLE) is difficult for EBMs due to marginalization. One solution is to approximate MLE via contrastive learning [3]. For each crystal structure $(\boldsymbol{A}, \boldsymbol{X})$, we apply K random rigid transformations to obtain K perturbed protein-ligand complexes $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_K$ as negative samples. Suppose $-E(\boldsymbol{A}, \boldsymbol{X}_i)$ is the predicted energy for $\boldsymbol{X}_i$, we train our EBM to maximize the likelihood of the crystal structure.

$$\ell_{\text{contrastive}} = \mathbb{E}_{(\boldsymbol{A}, \boldsymbol{X}) \sim \mathcal{D}} \left[\frac{\exp(-E(\boldsymbol{A}, \boldsymbol{X}))}{\sum_{i=1}^{K} \exp(-E(\boldsymbol{A}, \boldsymbol{X}_i))} \right] \tag{17}$$

A.5 Gaussian DSM Baseline

Recent works [23, 24] has successfully trained EBMs using denoising score matching (DSM) and proved that DSM is a good approximation of MLE. In standard DSM, we create a perturbed complex by adding Gaussian noise to ligand atom coordinates, i.e., $\tilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) = \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. DSM objective tries to match the score of our model $-\partial E / \partial \tilde{\boldsymbol{X}}$ and the score of the noise distribution $\nabla_{\boldsymbol{\epsilon}} \log p(\boldsymbol{\epsilon}) = -\boldsymbol{\epsilon}/\sigma^2$:

$$\ell_{\text{gaussian}} = \mathbb{E}_{(\boldsymbol{A}, \boldsymbol{X}), \epsilon \sim \mathcal{N}(0, \sigma^2 I)} \left[\| \partial E(\boldsymbol{A}, \tilde{\boldsymbol{X}}) / \partial \tilde{\boldsymbol{X}} - \boldsymbol{\epsilon}/\sigma^2 \|^2 \right] \tag{18}$$

Intuitively, ℓ_{g} forces the gradient to be zero when the input complex is a crystal structure ($\boldsymbol{\epsilon} = 0$). As a result, a crystal structure pose will be at the local minima of an EBM under the DSM objective.

A.6 Protein-Ligand Binding Experimental Details

Data. Our training data comes from the refined subset of PDBbind v2020 database [26] with their binding affinity labels excluded. After removing overlapping instances with the validation and test set, our training set has 4806 protein-ligand complexes in total. Our validation set has 363 complexes randomly sampled from PDBbind by Stärk et al. [25] with binding affinity labels. Our test set comes from the FEP+ benchmark [20], which has 264 protein-ligand complexes targeting eight proteins (cdk8, cmet, eg5, hif2a, pfkfb3, shp2, syk, and tnks2). Each protein-ligand complex is predicted by Glide core-constrained docking and labeled by experimental binding affinity.

Metric. We report the Spearman and Pearson correlation between true binding affinity and predicted energy $E(\boldsymbol{A}, \boldsymbol{X})$. We do not report root mean square error (RMSE) because our model does not predict absolute affinity values. In fact, shifting $E(\boldsymbol{A}, \boldsymbol{X})$ by any constant will be equally optimal under the DSM objective. We run our model with five different random seeds and report their average.

Hyperparameters. Our model consists of a MPN molecule encoder and a FANN protein encoder. For the MPN encoder, we use the default hyperparameter from Yang et al. [27]. For the FANN protein encoder, we set the hidden layer dimension to be 256, distance threshold $d = 10$, and try encoder depth $L \in \{1, 2, 3\}$. We use the Spearman correlation on the validation set to select the best hyperparameter.

Baselines. We consider three sets of baselines for comparison:

- **Physics-based models** calculate binding affinity based on molecular dynamics. We consider three popular methods: Glide [5], MM/GBSA [18], and Schrodinger FEP+ software. The performance of these baselines are provided by the authors of the FEP benchmark. Among these methods, Schrodinger FEP+ is the most accurate but most computationally expensive. It takes six hour to calculate energy for just one complex on a 64-core CPU server with 8 GPUs.
- **Unsupervised models**. Since unsupervised learning is relatively under-explored in this area, we implement two unsupervised EBMs trained with contrastive learning and Gaussian DSM. These two baselines use the same encoder architecture as DSMBind but different training objectives.
- **Supervised models**. Most of the existing deep learning models for binding affinity prediction belong to this category. They are trained on the entire PDBBind database with approximately 19000 binding affinity data points. We include four top-performing methods based on a recent survey [28]: KDeep [9], IGN [8], TankBind [15], and PLANET [28]. TankBind is currently the state-of-the-art model on our validation set. We use their open-source implementation and pre-trained model checkpoint on GitHub or online servers to evaluate their performance.

A.7 Protein-Protein Binding Experimental Details

Data. The training set of DSMBind are downloaded from PDB and filtered by the following criteria:

- It is not a homomer complex (i.e., a complex composed of multiple instances of the same protein.
- It does not have more than eight chains.
- If the complex contains an antibody, it must also have its antigen.
- If the complex contains a T cell receptor (TCR), it must also have its antigen.

For each downloaded complex, we further decompose it into pairs of two chains and remove pairs of chains with buried surface area less than 500Å^2. After these filtering steps, we derived approximately 27000 non-redundant protein-protein complexes as our training set. Each complex has exactly two chains. During training, we randomly rotate one of the proteins while keeping the other fixed.

The test set comes from the SKEMPI 2.0 database, which has 348 protein complexes and approximately 6000 $\Delta\Delta G$ data points. Similar to Luo et al. [16], we randomly select 10% of the data for validation and the rest of 90% for testing. We run the model with five random seeds and report the average performance on the test set.

Hyperparameters. For our FANN protein encoder, we set the hidden layer dimension to be 256, try distance threshold $d = [10, 16]$, and encoder depth $L \in \{1, 2, 3\}$. We use the Spearman correlation on the validation set to select the best hyperparameter.

Baselines. Luo et al. [16] reported a comprehensive list of baselines on the SKEMPI test set. They can be categorized into three groups: physics-based models, protein-language models, and supervised models. Their performance are directly copied from Luo et al. [16] and we briefly review these methods and their evaluation protocol here for reference.

- **Physics-based models**: We use two popular protein energy functions: Rosetta [1] and FoldX 5.0 [4]. For Rosetta, we use its default scoring function (ref2015) and apply `fast_relax` protocol to minimize the energy of each input complex. We then calculate the interaction energy between the two proteins in both wildtype and mutant complex. For FoldX, we apply `RepairPDB` function to minimize the energy of each input complex and calculate $\Delta\Delta G$ using its `BuildModel` function.
- **Protein language models (PLMs)**. We consider two PLMs for comparison: ESM-1v [17], ESM-IF [6]. For ESM-1v, we run its inference code with `masked-marginals` mode. For ESM-IF, we score the log-likelihood of each protein sequence with `multichain_backbone` flag so that the model see the whole protein-protein complex. For both models, we score the likelihood of wild-type and mutant sequences and use their difference as the estimation of $\Delta\Delta G$. We use the implementation provided in the ESM GitHub repository to calculate their performance.
- **Supervised models**. RDENet is pre-trained on 38413 unlabelled protein clusters with a rotamer density estimation (RDE) task and fine-tuned on the SKEMPI data with three-fold cross validation.
- **Energy-based models**. In addition to the above baselines, we report the performance of energy-based models trained with contrastive learning and Gaussian DSM. They are trained on the same training set and have the same model architecture as DSMBind.

A.8 Antibody-Antigen Binding Experimental Details

SabDab Data. Our training data comes from the Structural Antibody Database (SAbDab) [21], which contains 4883 non-redundant antibody-antigen complexes. Our test set has 566 complexes from SAbDab that have binding affinity labels. After removing antigen/antibody sequences that appear in the test set, our training set has 3416 complexes without binding affinity labels. Our validation set has 116 complexes (with binding affinity labels) after removing antibodies or antigens overlapping with the test set.

HER2 Data. This dataset has 424 designed variants of trastuzumab with experimentally measured binding affinity to the HER2 antigen. The structure of trastuzumab-HER2 complex has been crystallized (PDB: 1n8z) and the affinity of trastuzumab is already known (K_d = 0.194nM). For each designed antibody, we use the following template-based docking algorithm to predict the structure of HER2-antibody complex:

1. Use the ESMFold algorithm [14] to predict the antibody structure (including side-chain atoms).
2. Run the Kabsch algorithm [11] to superimpose the predicted antibody structure to the trastuzumab crystal structure in the complex 1n8z (which is used as our docking template).

Hyperparameters. For our FANN protein encoder, we set the hidden layer dimension to be 256, distance threshold $d = 20$, and try encoder depth $L \in \{1, 2, 3\}$. We use the Spearman correlation on the validation set to select the best hyperparameter. We run the model with five random seeds and report the average performance on each test set.

Baselines. Similar to the previous section, we consider four sets of baselines for comparison:

- **Physics-based models**. We consider two physics-based potentials (Rosetta [1] and FoldX [4]) that are commonly used in protein engineering.
- **Protein-language models**. The PLM baselines are implemented slightly differently from the previous experiment. The input to ESM-1v is the concatenation of an antibody and an antigen sequence and we take the pseudo log-likelihood of antibody CDR residues as the binding affinity of an antibody-antigen pair. The input to ESM-IF is the crystal structure of an antibody-antigen complex and we take the conditional log-likelihood of antibody CDR residues given the backbone structure as its binding affinity.
- **Energy-based models**. Contrastive learning and Gaussian DSM baselines are implemented in the same way as protein-protein binding prediction, except that it is trained on the SAbDab training set.

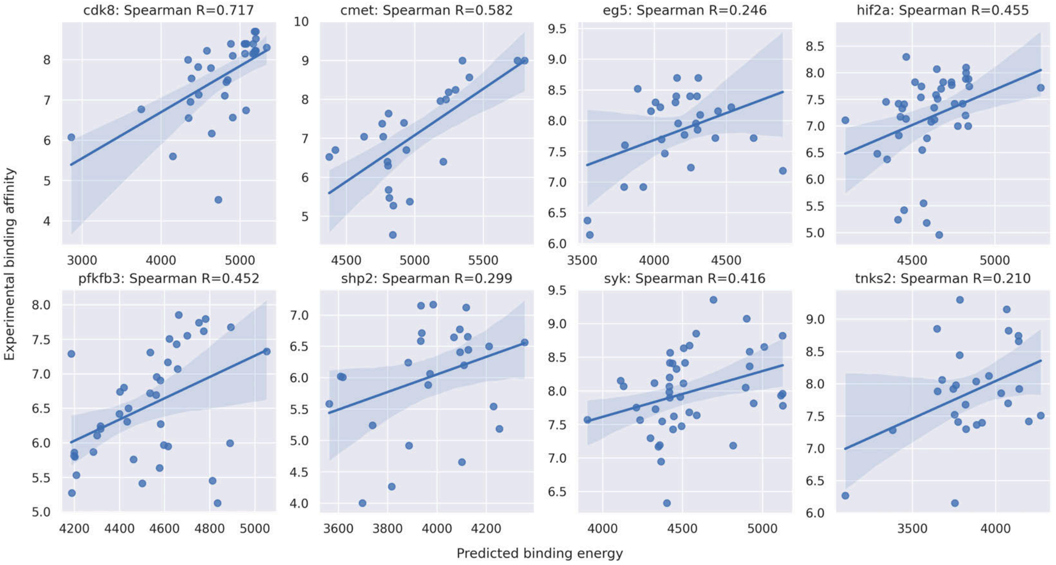

Figure 4: Spearman correlation of DSMBind for each of the eight targets in the FEP benchmark.

- **Supervised models**. We also compare DSMBind with a supervised model with the same frame averaging neural network (FANN) encoder and pre-trained ESM-2 residue embedding as DSMBind. Since all labeled data from SAbDab are in the validation and test sets, we draw additional data from the SKEMPI database [7]. We obtain 5427 binding affinity data points after removing all complexes appeared in the test set. To maximize the model performance, we train the model on 5427 data points first and then finetuning on 273 antibody-antigen data points.

A.9 Ablation studies of DSMBind

Lastly, we perform three ablation studies to understand the importance of different modules of DSMBind.

Removing ESM-2 embedding. The default DSMBind model uses ESM-2 sequence embedding as residue features. To understand its importance, we train DSMBind with one-hot amino acid embedding instead. We evaluate this modified version (No ESM) on protein-ligand binding (FEP test set) and antibody-antigen binding (SabDab test set). As shown in Fig.5a, removing ESM-2 embedding had almost no effect in the FEP test set (0.380 vs. 0.388), but the difference becomes much more noticeable in the SabDab test set (0.314 vs. 0.374, Fig.5b). Therefore, we conclude that using ESM-2 embedding is useful for modeling binding energy.

Removing backbone/side-chain DSM. Our SE(3) DSM objective includes both backbone and side-chain DSM. To understand their importance, we train DSMBind with only backbone DSM. As shown in Fig.5a-c, we find that removing side-chain DSM has moderate effect on the FEP (0.367 vs 0.388) and SKEMPI dataset (0.403 vs 0.380), but has notable effect on the SAbDab test set (0.314 vs 0.374). We also train DSMBind with only side-chain DSM (i.e., without backbone rotation/translation noise). It has significant impact on the FEP (0.242 vs 0.388) and SAbDab test set (0.334 vs 0.374), but almost no effect on the SKEMPI test set. This result is expected since side-chain flexibility plays the most important role in $\Delta\Delta G$.

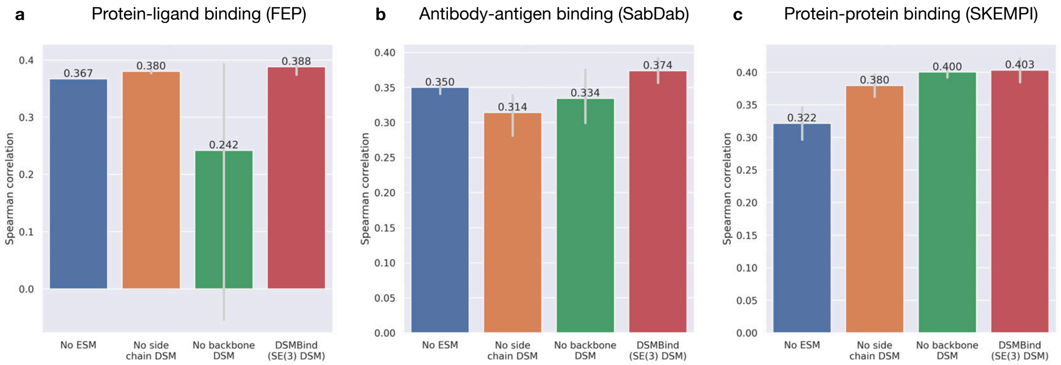

Figure 5: Ablation study of DSMBind on the FEP, SAbDab, and SKEMPI test set.

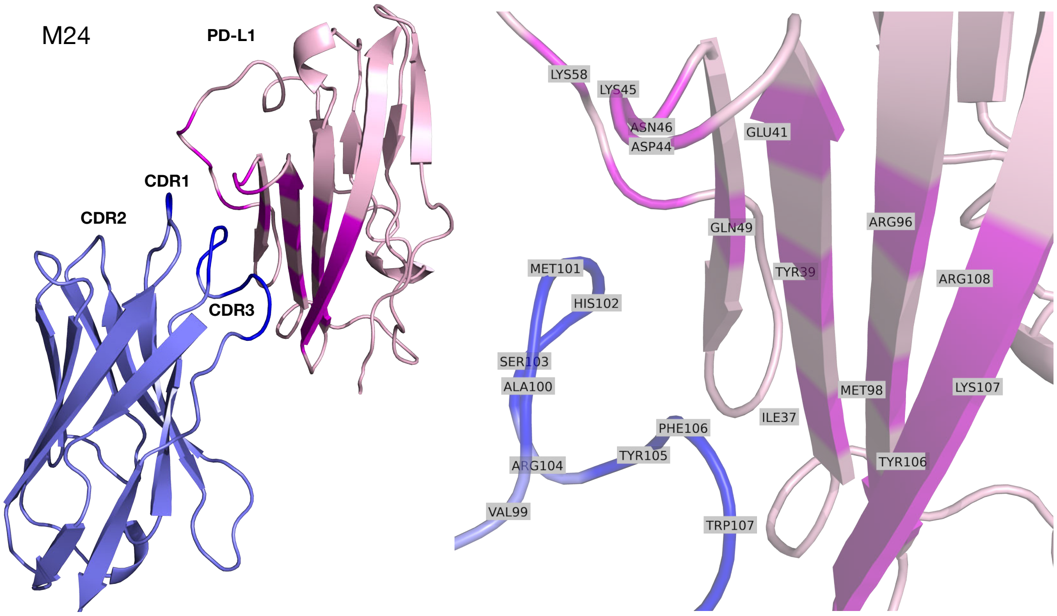

Figure 6: Structure of nanobody M24 in complex with PD-L1.

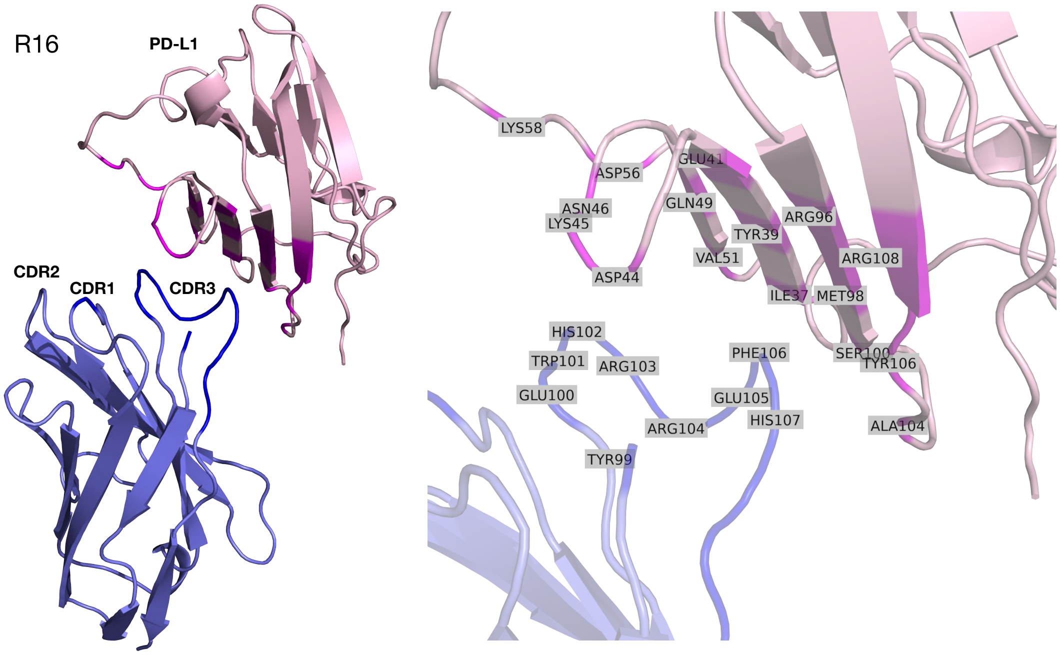

Figure 7: Structure of nanobody R16 in complex with PD-L1.