RobustDock: Robust Generative Flexible Docking with Long-Tailed Data

Anonymous Authors[1]

Abstract

As generative modeling is increasingly applied to scientific challenges, an emerging issue is overfitting to noisy data, especially as synthetic data becomes more widely used for training. In this work, we provide the explicit observation of noise related overfitting in modern generative models for scientific applications particularly for flexible docking. In this task, the generative model is trained to model protein conformational shift from apo to holo state as well as ligand binding pose, where the apo structure is predicted by a large-scale deep learning model. The resulting apo-holo pairs exhibit a long-tailed structure shift distribution, which the model must learn effectively. Motivated by this observation, we propose robust training techniques for generative models, including asynchronized time schedule and inverse shift sigmoid reweighting, supported by both theoretical analysis and empirical results. By effectively reweighting the loss, model regularization, improved confidence model training and fine grained control of robustness over protein degrees of freedom, our ROBUSTDOCK achieves state-of-the-art performance on the PDBBind benchmark and increases by 13.6 percentage points (of ligand RMSD < 2 Å) on the large conformational shift subset.

1. Introduction

Molecular docking is a computational approach for determining the structure of a molecular complex by predicting how a small molecule binds to a macromolecule; typically a ligand to a protein receptor. Modeling such bindings is crucial for understanding cellular mechanisms and enabling structure-based drug design. While existing methods have achieved great success in rigid docking, they deviate from realistic scenarios where proteins are inherently flexible and undergo conformational changes upon ligand binding (Kessel & Ben-Tal, 2018). Instead, flexible docking jointly models the protein conformational shift from the apo (unbound) to the holo (bound) state, together with the ligand binding pose, providing explicit treatment of protein flexibility and thereby better agreement with realistic docking.

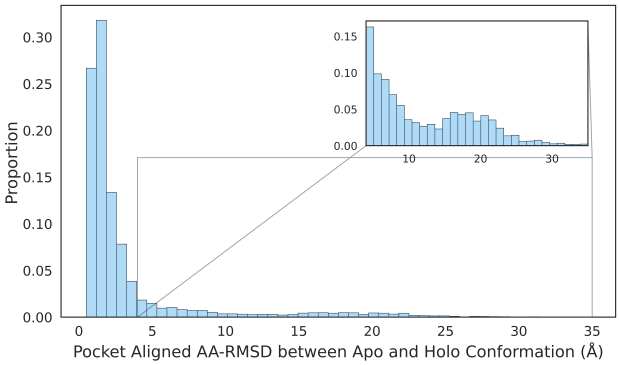


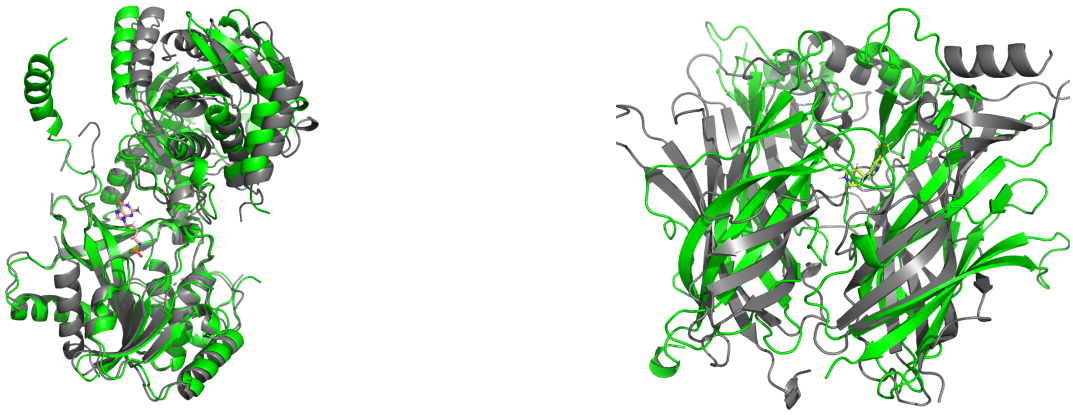
Figure 1. Structure shift distribution of pocket-aligned All-Atom RMSD between apo and holo conformation on the PDBBind Version 2020 training set, with large conformational samples (> 4 Å) is about 16.4%.

Generative modeling, which initially intrigued the community by generating natural images (Rombach et al., 2022; Esser et al., 2024), has emerged as a substantial tool for addressing scientific challenges of structural biology in recent years, including applications in protein design (Watson et al., 2023), modeling biomolecular interactions (Abramson et al., 2024), and drug design (Jin et al., 2018). Scientific data, however, is commonly *noisy*, hindering the training of generative models.

While noise in image data has been extensively studied, scientific datasets also exhibit noise arising from measurement uncertainty. There is also a shortage of data on scientific problems, due to the expense of data collection, which drives the usage of *synthetic* data produced by other large-scale deep learning model, whose data is inherently noisy. A prevalent case of this is the AlphaFold Data Bank (AFDB) (Varadi et al., 2022) which has been leveraged to train scalable generative models to capture diverse protein conformation ensembles (Geffner et al., 2025; Lewis et al., 2025). Overfitting to such noisy data present in the training

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.



(a) PDBID: 1V48. Apo Holo All-Atom RMSD: 12.323 Å

(b) PDBID: 3U8K. Apo Holo All-Atom RMSD: 18.709 Å

Figure 2. Visualization of two examples, apo structure in gray and holo structure in green.(a) 1V48: conformation being adjusted upon ligand binding. (b) 3U8K: ESMFold predict the monomer folds reasonably, but mispredicts the multimeric arrangement and the binding site is within the interchain regions. This can be verified by aligning chain separately with only small conformational shift for each chain.

sets undoubtedly limits the model's performance, leading to suboptimal generalization (Zhang et al., 2016).

Particularly, in flexible docking, the generative model is trained to produce the ligand's binding pose as well as the protein's conformational shift from apo to holo structure, *where the apo structure is typically produced by a folding model* such as AlphaFold (Jumper et al., 2021) or ESMFold (Lin et al., 2022). Such dependency on the folding model, when they mispredict, can result in large conformational shifts between the apo and holo structures. In Fig. 1, we can see that the structural difference of apo-holo pairs creates a long-tailed distribution of comformational shift which the generative model should effectively account for. Importantly, a central challenge is that large conformational shift may arise either from the noisy data or genuine cases where the protein undergoes substantial conformational changes upon binding. Fig. 2 further shows two examples with large structure shift. 1V48 undergoes the ligand induced conformational adjustment, whereas 3U8K is primarily mispredicted in the interchain regions around a binding site with an approximate plane of pseudosymmetry since ESMFold is not trained to model dimeric interfaces. This motivates robustifying the training of generative models against noisy and long-tailed data.

In this work, we highlight the impact of noisy training data in scientific generative modeling, an issue largely overlooked by prior work, with particular focus on flexible docking. Motivated by the long-tailed apo-holo shift distribution and its mixture of true conformational changes and noise, we propose RoBUSTDock, which improves training robustness of generative model via (i) asynchronized time schedule that reduces reliance on noisy apo inputs, (ii) inverse shift sigmoid reweighting that smoothly downweights extreme shift pairs with bounded control of the marginal distribution shift, (iii) improved confidence model training with pairwise ranking loss, and (iv) component-aware robustness control across protein's degrees of freedom and model regularization. Empirically, our RoBUSTDock achieves state-of-art performance on the PBDBind benchmark and increases by 13.6 percentage points (of ligand RMSD < 2 Å) on the large conformational shift subset.

2. Background and Related Work

Diffusion Models and Flow Matching. Diffusion models (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021) and flow matching (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2022) both adopt a regression-like objective that trains a neural network to approximate, respectively, the score of the data distribution $\nabla \log p(x)$ or the vector field $v(x)$ transporting noise to data distributions. In this work, we follow the notation of flow matching, while the same method and formal analysis can be similarly applied to diffusion models. For flow matching, since the true vector field $u(x_t)$ generating the probability path $p_t(x)$ is intractable, the following conditional flow matching (CFM) loss is leveraged for training which shares the same parameter gradient as the flow matching loss:

$$\mathcal{L}(\theta) = \mathbb{E}\left[w(t)||v_\theta(x_t) - u(x_t|x_0, x_1)||^2\right], \qquad (1)$$

where $w(t)$ is a time reweighting. It is explicitly governed by the time sampling and implicitly governed by the network parameterizations and interpolant schemes (Kingma & Gao, 2023). From Eq.(1), it is clear that noisy training data can cause the model to overfit their generative trajectories.

Generative Modeling on Corrupted Data. Training generative models with only access to noisy data has been recently explored (Aali et al., 2023; Daras et al., 2023; 2024), however, these works assume that the corruption is Gaussian. In tandem with the forward Gaussian diffusion process in generative models, it gives a relatively simple scenario where the noisy data x_t has only part of its diffusion time regions observed. Ambient protein (Daras et al., 2025) extends the noisy data to be low quality structure of AlphaFold2. It leverages the distribution merging property of the forward diffusion and trains the diffusion models over pLDDT dependent truncated time regions for each sample. Our work departs from the common setup of known structural noise and instead seeks *robust training under unknown corruption*.

Deep Learning-based Molecular Docking. Molecular docking, a task predicting the ligand 3D structure when binding to a receptor, has recently been explored with deep learning based solutions. Regression-based methods (Stärk et al., 2022; Lu et al., 2022; Pei et al., 2024) directly train a neural network to predict the target bound structure with various architecture designs. Generative-based methods formulate docking as a distribution learning problem, which naturally explains the aleatoric uncertainty in the biological data. Specifically, the diffusion or flow matching models can be either defined on the docking degrees of freedom (manifold docking), namely the global rotation, translation and torsion angles (Corso et al., 2023; 2024; Lu et al., 2024; Guo et al., 2025), or directly on the 3D Euclidean coordinates of the ligand (Stärk et al., 2023; Morehead & Cheng, 2025; Zhou et al., 2025). This problem setup has been expanded to model the protein conformational shift within the pocket, together with the ligand structure to better align with the realistic docking scenario (Plainer et al., 2023; Huang et al., 2024), known as flexible docking. Compared to the cofolding methods which co-generate the protein-ligand complex (Krishna et al., 2024; Bryant et al., 2024; Abramson et al., 2024), flexible docking enjoys substantial speedup for virtual screening (Lee et al., 2025), i.e., the process of finding/optimizing a binding ligand for a certain protein in structure-based drug discovery. Furthermore, they can be more realistic (Arai, 2018) (receptors often fold before conforming to incoming ligands), practical (critical applications commonly use experimentally-determined and/or precisely-simulated apo structures (Michino et al., 2025)), and data efficient (less model complexity is required for conformation than folding).

3. Methods

Task Formulation. Given a dataset $\mathcal{D} = \{x^i, y^i\}_{i=1}^{|\mathcal{D}|}$, the target distribution for flexible docking is the joint structure distribution of holo protein x and the bound ligand y. The generative model is trained to model the distribution shift from the apo $p(x_0)$ to the holo structure $p(x_1)$, and the RDKit ligand conformer $p(y_0)$ to its bound structure $p(y_1)$. The generative model is defined over the intrinsic coordinates of the molecule. Each protein x lies on the product manifold of backbone translation $\mathrm{T}(3)^N$, orientation $\mathrm{SO}(3)^N$ and side chain torsion angles $\mathrm{SO}(2)^m$, where N is the number of total residues and m is the number of rotatable side chain angles. This factorizes the protein representation into three components: per-residue translation, rotation, and side chain degrees of freedom. Similarly, the ligand y is defined over the global translation $\mathrm{T}(3)$, rotation $\mathrm{SO}(3)$ and torsion angles $\mathrm{SO}(2)^k$, where k is the number of rotatable bonds of the ligand. Next, we describe techniques we use to robustify the training of flow matching where we use the notation x for convenience instead of the joint (x, y).

3.1. Asynchronized Time Schedule

Training Time Distortion. To improve the training robustness regarding the noisy apo structure $p(x_0)$, we can bias the time sampling distribution $p(t)$ towards the holo protein ($t \to 1$), reducing the model's reliance on apo structure generated by a deep learning predictor. Given this, we propose the **asynchronized time schedule** for ligand and protein. Specifically, we sample the ligand time t uniformly from $U(0, 1)$, and propose to use the following uniform exponential transform for protein with normalized boundary conditions $f(0) = 0$ and $f(1) = 1$:

$$f_\alpha(t) = \frac{e^{\alpha t} - 1}{e^\alpha - 1}, \quad \alpha < 0, \quad t \sim U(0, 1). \tag{2}$$

For $\alpha < 0$, f is monotone increasing and induces a non-uniform time distribution, concentrating more samples towards the holo structure. Importantly, compared to decoupled sampling for ligand and protein, our mechanism strengthens the time coherence by first sampling ligand t uniformly and then calculating $f(t)$ for protein. Formally, it biases training toward specific time regions ($t \to 1$) by reweighting the CFM loss with its empirical PDF (Qin et al., 2024). Generally, this corresponds to the well-known inverse sampling (Devroye, 2006), a technique which samples from arbitrary univariate distribution given its CDF (Details in Appendix D). To allow for a monotonic increase of skewness with the increase of $\alpha > 0$, the transform can be flipped with the flipped boundary condition $f(1) = 0$ and $f(0) = 1$:

$$m := f_\alpha(t) = 1 - \frac{e^{\alpha t} - 1}{e^\alpha - 1} = \frac{e^\alpha - e^{\alpha t}}{e^\alpha - 1}, \quad \alpha > 0 \tag{3}$$

Applying the change of variable formula, we have:

$$p_M(m) = p_T\big(f^{-1}(m)\big) \left| \frac{dt}{dm} \right|$$

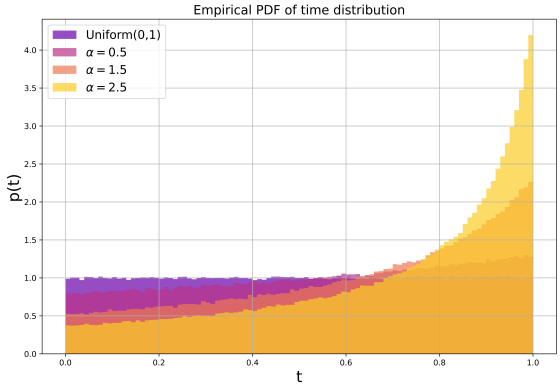


Figure 3. Empirical PDF of the skewed time distribution Eq.(4).

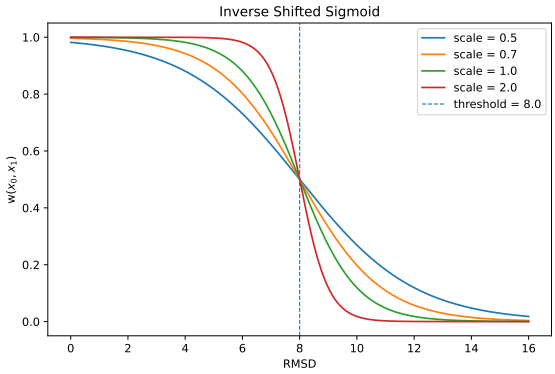


Figure 4. Inverse Shift Sigmoid Reweighting

$$\frac{dt}{dm} = \frac{1 - e^{\alpha}}{\alpha\big(e^{\alpha} + (1 - e^{\alpha})m\big)} < 0$$

The resulting probability density function of $m = f_\alpha(t)$ is a uniform generated distribution:

$$\boxed{p_M(m;\alpha) = \frac{e^{\alpha} - 1}{\alpha\left(e^{\alpha} + (1 - e^{\alpha})m\right)}, \qquad m \in (0,1)} \quad (4)$$

The empirical PDF is visualized in Fig. 3. As α increases, $p_M(m;\alpha)$ becomes increasingly skewed toward $m = 1$. In the limit $\alpha \to 0$, we recover the uniform distribution: $\lim_{\alpha\to 0} p_M(m;\alpha) = 1 \quad \text{for} \quad m \in (0,1)$.

Sampling Time Distortion. One advantage of generative models is the decoupling of training and sampling, and thus we can freely choose the discretization scheme separately for protein and ligand. To better match training and sampling, we adopt a similar time distortion during sampling. Concretely, we consider an adaptive step size schedule for protein to align with the training time distortion, and use an equal step size for the ligand pose. This essentially creates an asynchronized sampling speed for protein and ligand. With protein stabilized to its bound state faster than ligand, it interestingly relates to the classical *conformational selection* model (Monod et al., 1965) where the protein fluctuates among multiple conformations and the ligand selectively binds the conformation that matches it best. This interpretation can motivate another strategy of "fast ligand sampling" rather than protein, relating to the *Induced Fit* model (Koshland Jr, 1958) where the conformation shift is induced by the ligand binding. This interesting connection is summarized in Table 7. In this work, we use the "protein-fast" sampling to match the training time distortion. This choice does not map cleanly onto one of the alternatives. Because, even with asynchronized sampling, the protein and ligand still jointly converge to the bound conformation. However, since the protein is driven toward its holo state faster than the ligand, the behavior is leaning more towards conformational selection. It also better reflects realistic docking, which typically combines induced fit and conformational selection rather than adhering to a single model.

3.2. Loss Reweighting

Asynchronized time schedule reduces the training impact of apo structures for all samples. To directly address the noisy data, an explicit loss reweighting can be introduced over the sample pairs $w(x_0, x_1)$. This yields the following CFM loss extended with the sample pair weight $w(x_0, x_1)$ from the Eq.(1):

$$\mathcal{L}(\theta) = \mathbb{E}\left[w(x_0,x_1)\cdot w(t)\cdot ||v_\theta(x_t) - u(x_t|x_0,x_1)||^2\right].$$

Ideally, $w(x_0, x_1) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ operates by assigning low weight to noisy data pairs and high weight to clean data pairs to achieve robust training. A change of measure gives the tilted coupling distribution $p_w(x_0, x_1)$ as:

$$p_w(x_0,x_1) \sim w(x_0,x_1)p(x_0)p(x_1).$$

Notably, with this interpretation, the recent work of unbalanced flow matching (UFM) (Corso et al., 2025) can be interpreted as loss reweighting depending on the protein conformational shift measured by the RMSD:$w(x_0, x_1) = \mathbf{1}_{\text{RMSD}(x_0,x_1)<c}$. However, it does not fully align with some realistic docking cases, for example, the DFG in-out transition in kinase protein and the formation of the cryptic pocket, where the protein undergoes drastic conformational changes. This shortcoming could potentially hinder wider applications of the method.

To encode the domain knowledge that the apo-holo pairs with large structure deviations potentially indicate a folding model error or measurement noise, we propose the **Inverse Shift Sigmoid Reweighting** (ISSR) with $c(x_0, x_1) = \text{RMSD}(x_0, x_1)$:

$$w(x_0,x_1) = \text{Sigmoid}\left(-\beta\left(\text{RMSD}(x_0,x_1) - c\right)\right). \quad (5)$$

As shown in Fig. 4, β allows ISSR to smoothly interpolate between uniform weighting and threshold based rejection.

As $\beta \to +\infty$, ISSR approaches rejection sampling at the threshold c, whereas as $\beta \to 0$, all samples are weighted approximately equally. This connects naturally to robust regression under long-tailed data, where extreme or misspecified observations are downweighted rather than explicitly identified and removed. Analogously, ISSR softly limits the contribution of large apo-holo shifts while retaining these samples during training. Besides, the inverse sigmoid induces a probabilistic interpretation of $\mathrm{RMSD}(x_0, x_1)$, assigning lower probabilities to pairs with larger apo-holo deviations.

Marginal Tilting with ISSR. Sampling source and target distributions independently and applying weights to the CFM loss replaces the independent coupling $p(x_0, x_1) = p(x_0)p(x_1)$ with the tilted coupling $p_w(x_0, x_1) = \frac{w(x_0,x_1)}{Z_w} p(x_0)p(x_1)$. Marginalizing $p_w(x_0, x_1)$ gives the tilted marginal $p_{w,0}(x_0)$, we refer to as *marginal tilting*.

$$p_{w,0}(x_0) = \frac{\psi_w(x)}{Z_w} p(x_0)$$
$$\psi_w(x_0) = \int w(x_0, x_1) p(x_1) dx_1$$

where $\psi_w(x_0)$ is the marginal potential and Z_w is the normalization constant of the tilted coupling $p_w(x_0, x_1)$. The same analysis also applies to $p_{w,1}$. With this, one can further quantify the tilting effect via the relative variance (RV), which upperbounds the KL divergence between the tilted marginal and original marginal:

$$\mathrm{KL}(p_{w,0} \| p) = \int p_{w,0} \log \frac{p_{w,0}}{p} \, dx_0$$
$$<= \frac{\mathrm{Var}_p(\phi_w)}{Z_w^2} := \mathrm{RV}_0$$

The inverse shift sigmoid enables smooth control over the RV, as opposed to UFM where the induced marginal tilt can have unbounded RV. The unbounded RV implies the poor coverage of tail with large shift events being discarded, while exact matching of the original marginal $p(x_0)$ overweights the tail. In contrast, the ISSR avoids both failure modes by smoothly downweighting their influence, providing controlled coverage of the tail when the threshold c is bounded away from 0. Specifically, ISSR admits a theoretical RV bound with quadratic dependence on β formalized in the following theorem (Details in Appendix E).

Theorem 3.1 (Controlled RV for Inverse Shift Sigmoid). *Let $w_{\beta,c}(x_0, x_1) = \mathrm{Sigmoid}\left(-\beta\left(\mathrm{RMSD}(x_0, x_1) - c\right)\right)$. Define the marginal potential $\phi_{\beta,c}$, normalization constant $Z_{\beta,c}$ and density ratio r_0 as:*

$$\phi_{\beta,c}(x_0) := \mathbb{E}_{X_1}[w_{\beta,c}(x_0, X_1)]$$
$$Z_{\beta,c} := \mathbb{E}_{X_0}[\phi_{\beta,c}(X_0)], \qquad r_0(x_0) := \frac{\phi_{\beta,c}(x_0)}{Z_{\beta,c}}$$

With $\mathrm{RMSD}(x_0, x_1)$ to be x_1-uniformly L_d Lipschitz bounded and assuming x_0 has finite second moment, then $\phi_{\beta,c}$ is Lipschitz with constant $L_\phi \le \beta L_d/4$, and RV is upper bounded by:

$$\mathrm{RV}_0 := \mathrm{Var}_p(r_0) = \frac{\mathrm{Var}_p(\phi_{\beta,c}(X_0))}{Z_{\beta,c}^2}$$
$$\le \frac{\left(\frac{\beta L_d}{4}\right)^2 \mathbb{E}\|X_0 - \mu\|^2}{Z_{\beta,c}^2}, \qquad \mu := \mathbb{E}[X_0]$$

Proof sketch. The proof uses the uniform bound of the sigmoid function $\sigma'(z) = \sigma(z)(1 - \sigma(z)) \le \frac{1}{4}$. By the mean value theorem, the inverse shift sigmoid is therefore globally Lipschitz. This Lipschitz control yields an upper bound on the marginal potential. The final result follows by combining this bound with the assumptions and plugging into the definition of the RV.

3.3. Pairwise Ranking Loss for Confidence Model

To assess the quality of the predictions from generative model, a confidence module can be trained as a classifier whether a predicted pose has ligand RMSD $< c$ Å(Corso et al., 2023). At test time, the logit of the classifier is used to score the poses, and the pose with the highest score is leveraged as the final prediction. This method requires manually tuning the threshold c, and more critically, completely neglecting the interior ranking relation among the poses for each complex during training. To mitigate this, we propose the pairwise ranking loss (Burges et al., 2005):

$$\mathcal{L}_{\mathrm{conf}} = \sum_{i,j} \mathbf{1}[r_i < r_j] \, \log\big(1 + \exp(-(s_i - s_j))\big)$$

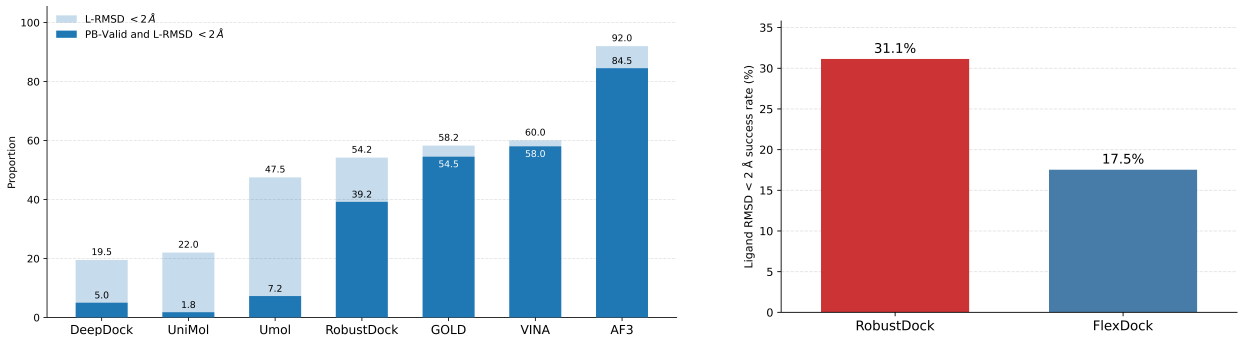
where (i, j) denotes two randomly sampled poses for one complex, and s_i and r_i are the confidence score and RMSD for pose i, respectively. The pairwise ranking loss converts the score difference between two poses into a probabilistic prediction that one pose is of higher quality than the other. Furthermore, the pairwise ranking loss removes the need to hand-tune the RMSD threshold c and better aligns training with the downstream ranking utility. By leveraging relative comparisons among multiple poses for each complex, it provides denser supervision, yielding up to $\mathcal{O}(n^2)$ pairwise training signals from n poses.

3.4. Training and Sampling

We follow the docking pipeline of a manifold docking flow followed by a structure relaxation flow (Corso et al., 2025), where the second relaxation flow is to refine the manifold docked poses for intermolecular potential minimization. We apply the asynchronized time schedule and ISSR to the manifold docking flow. Beyond these, we leverage **weight**

Table 1. Top-1 PDBBind ESMFold docking benchmark. Results of RobustDock are averaged over 5 seeds. The other baseline results are extracted from (Corso et al., 2025). The best results are in **bold**.

Method	Ligand RMSD		All-Atom RMSD	% PB valid	L-RMSD < 2Å	Runtime (s)
	% < 2Å (↑)	Median Å (↓)	% < 1Å (↑)	(↑)	and %PB valid (↑)	
SMINA (rigid)	6.6	7.7	N.A.	-	-	258
SMINA	3.6	7.3	5.2	-	-	1914
GNINA (rigid)	6.7	7.1	N.A.	**93.3**	6.7	260
GNINA	8.4	7.9	4.5	91.3	7.7	1575
DIFFDOCK-POCKET (rigid)	37.5	3.0	N.A.	25.9	4.9	17
DIFFDOCK-POCKET	41.8	2.5	32.4	30.1	5.9	17
REDOCK	39.0	2.5	39.8	-	-	15
FLEXDOCK	39.7	2.5	41.7	72.9	33.7	11
RobustDock	**45.9**	**2.2**	**43.4**	78.3	**39.2**	**10.3**



(a) Top-1 PoseBusters V2 Benchmark.

(b) Large conformational change subset of Top-1 PDBBind ESMFold Docking (106 examples; apo holo shift > 2 Å).

Figure 5. PoseBusters V2 benchmark and large conformation subset of the PDBBind ESMFold docking benchmark.

decay as a robustness-promoting regularization mechanism. By discouraging excessively large parameter values, weight decay imposes a simplicity bias on the learned model, reducing its tendency to fit spurious patterns from noisy examples. It has been shown to improve a neural network's generalization in the presence of static noise (Krogh & Hertz, 1991), and is also consistent with modern studies emphasizing the importance of regularization in preventing pure memorization in diffusion models (Baptista et al., 2025). In our initial experiments, we find that an appropriate weight decay setting delays generalization but ultimately improves docking performance (see Fig. 11). However, when regularization is excessive, ligand performance saturates while protein performance degrades.

For structure relaxation, since the training depends on the docking model's output, it displays a similar long-tailed structure shift shown in Fig. 10. We apply the ISSR to address this relatively moderate long tail. Beyond relaxation, we expect the added robustness can render the relaxation flow as a refiner to improve the docking accuracy. Since the relaxation flow is underparameterized with less than 1M. parameters compared to the docking model (∼ 76M.), we find that it does not benefit from weight decay. The confidence model is trained by the pairwise ranking loss instead of casting it as a binary classification task.

4. Experiments

Dataset. We train the model on the PDBBind version 2020 (Liu et al., 2017), which is a widely adopted benchmark for evaluating the docking performance. The training and validation contain 17k complexes before 2019 and testset has 363 complexes after 2019. While the PDBBind timesplit has relatively high protein ligand interaction similarity, we also evaluate on the PoseBusters V2 (Buttenschoen et al., 2024), a recent testset with 308 high quality structures carefully curated from the PDB, and DockGen-E (Corso et al., 2024; Morehead et al., 2025), which is considered as the most challenging testset with low pocket similarity to the training set. The apo structure is generated by the ESMFold (Lin et al., 2022) for each example with its PDB sequence as the input.

Evaluation Metrics. We adopt the fraction of the pocket aligned RMSD < 2Å, and RMSD median for the ligand performance. For the protein, we report the fraction of all-atom (AA) RMSD < 1Å. The pocket alignment operates by globally aligning the predicted pocket atoms to the ground truth pocket atoms. Except for the computational metrics, the chemical plausibility is also reported with the fraction of test examples passing the PoseBusters check (denoted as PB-Valid) and ligand RMSD < 2Å.

Table 2. DockGen-E benchmark. * indicates the crystal ligand is perturbed (translation, rotation, and torsion angles) for initializing the samples instead of an RDKit ligand pose, and thus the optimal ligand local bound geometry is preserved during denoising. ** indicates the runtime is calculated on a 25% subset of Astex Diverse using NVIDIA A100 80GB. Baseline results are extracted from (Morehead et al., 2025).

Method	L-RMSD $<$ 2 Å	L-RMSD $<$ 2 Å and PB valid	Runtime** (s)
P2Rank-Vina	1.3	1.8	1283.70
DiffDock-L(rigid)	10.0	0.7	88.33
DynamicBind	9.0	0.0	146.99
RFAA	4.0	0.0	3443.63
AF3-Single-Seq	9.7	5.7	–
AF3	28.7	21.3	3049.41
RobustDock	10.1	4.5	9.83
RobustDock* (oracle) (crystal ligand local structure)	27.5	23.3	9.83

Baselines. On PDBBind, we use SMINA (Koes et al., 2013) and GNINA (McNutt et al., 2021) as traditional methods, DiffDock-Pocket (Plainer et al., 2023), Flexdock (Corso et al., 2025) and ReDock (Huang et al., 2024) as deep learning methods for baseline comparisons. On PoseBusters V2 and DockGen-E, we extend the benchmark which includes the following baselines DeepDock (Liao et al., 2019), Unimol (Zhou et al., 2023), DynamicBind (Lu et al., 2024), RFAA (Krishna et al., 2024), Umol (Bryant et al., 2024), GOLD (Verdonk et al., 2003), VINA (Trott & Olson, 2010) and AlphaFold3 (Abramson et al., 2024).

4.1. Which protein components require robustness for long tail?

Section 3 describes robust training techniques for generative flexible docking. A practical question is which protein components most require robustness under long-tailed shifts between apo and holo structure, since the protein is decomposed into backbone translation, rotation, and side chain angles. Fig. 7 shows that all three components exhibit long-tailed shift, motivating a targeted study of where robustness most improves generalization.

Hypothesis. Within the binding site, backbone translation mainly captures global displacement between apo and holo structures and should benefit most from robustness, followed by backbone rotation. In contrast, side chain motions capture local flexibility and contain the dominant conformational variability (Clark et al., 2019), making them less suited to aggressive robustness.

To verify this, we ablate the asynchronized time schedule by applying it in isolation to each component (Details in Appendix F). The result shows that robustness improves performance for all components, with the largest gains from backbone translation. Side chains also benefit, but require careful tuning of robustness strength. Overall, the results support our hypothesis: backbone shift is more noise-prone, while side chain changes reflect a more challenging signal that the model should learn effectively.

4.2. Results

Results on PDBBind. As shown in Table 1, our RoboustDock demonstrates strong generalization on this benchmark, with substantial improvement $+6.2$ percentage points (pp) on the percent ligand RMSD $<$ 2Å and raising the RMSD median from 2.5 to 2.2. On protein metrics, we achieve a notable increase from 41.7% to 43.4% on the percentage of AA RMSD $<$ 1Å compared to FlexDock. When chemical plausibility is taken into account, the fraction of poses with ligand RMSD $<$ 2Å increases significantly from 33.7% to 39.2%, indicating improved docking performance is preserved after the protein ligand energy relaxation.

Results on PoseBusters V2 and DockGen-E. As shown in Fig. 5a, RobustDock generalizes better than the large scale screening method Umol, improving the percentage of ligand RMSD $<$ 2Å metric from 47.5 % to 54.2%. It greatly closes the gap to search-based methods such as Vina while remaining significantly faster. On the challenging DockGen-E benchmark shown in Table 2, where most methods achieve limited success, RobustDock outperforms most existing approaches and attains a success rate comparable to AF3-Single-Seq (no input MSAs). Notably, RobustDock is substantially more efficient, completing inference in 9.83 s compared with the much longer runtimes of most competing methods (e.g., 146.99 s for DynamicBind), highlighting its favorable accuracy efficiency trade-off particularly for virtual screening.

How does the *RobustDock* perform on large conformational samples? With full coverage of the training set,

Table 3. Ablations on the PDBBind ESMFold docking benchmark. Top panel reports Top-10 manifold docking; bottom panel reports Top-1 structure relaxation.

Method	L-RMSD < 2 Å	AA RMSD < 1 Å	L-RMSD < 2 Å and PB valid
Top-10 Manifold Docking			
Baseline FM	47.6 ± 1.0	47.2 ± 0.3	–
RobustDock	53.0 ± 0.8	48.1 ± 0.4	–
w/o ISSR	52.6 ± 0.5	47.1 ± 0.3	–
w/o WD	51.2 ± 0.8	48.2 ± 0.5	–
w/o ATS	49.4 ± 0.8	48.0 ± 0.4	–
Top-1 Structure Relaxation			
RobustDock	45.9 ± 1.0	43.4 ± 0.6	39.2 ± 1.3
w/o ISSR	44.7 ± 1.4	36.5 ± 0.8	38.2 ± 1.0

ROBUSTDOCK ideally should model the large conformational samples better than FLEXDOCK. To demonstrate this, we construct a subset of the PDBBind testset encompassing 106 complexes with relatively large conformational shift (CA RMSD $>$ 2Å). Fig. 5b shows the comparison that ROBUSTDOCK demonstrates stronger performance with $+13.6\%$ improvement on this subset. Figure 6 visualizes one such case in which the localized conformational rearrangement near the binding site is accurately predicted upon ligand binding.

4.3. Ablations and Analysis

Ablations on Docking and Structure Relaxation. We also train the baseline flow matching (FM) model with the same architecture and number of parameters. The upper panel of Table 3 shows the ablations of ROBUSTDOCK compared with the baselines on docking. It displays each proposed robustness component contributes to the final performance with the inverse shift sigmoid reweighting (ISSR) improving the holo prediction, asynchronized time schedule (ATS) and weight decay (WD) improving the ligand binding accuracy. For structure relaxation, the ablation can be done by keeping the manifold docking output the same and comparing the structure relaxation output with different training strategies. The bottom panel of Table 3 shows that the ISSR substantially improves protein prediction accuracy beyond just energy relaxation. Without it, the model would overfit the long tail, and the fraction of predictions with AA RMSD < 1Å drops from 43.4% to 36.5%.

Confidence Model Comparison. To show the effectiveness of the pairwise ranking loss, we keep the docking output and only alter the confidence module for sample selection. The oracle performance denotes using the ground truth to select the sample, corresponding to the performance ceiling of the confidence model. Table 4 shows that pairwise ranking improves selective accuracy over the ligand RMSD classifier, increasing Top-1 ligand RMSD $<$2Å while maintaining comparable protein accuracy.

Table 4. Confidence model comparison of ligand RMSD classifier and pairwise ranking loss.

Selector	L-RMSD < 2 Å	AA RMSD < 1 Å
Oracle	53.0 ± 0.8	48.1 ± 0.4
Classifier	42.8 ± 1.1	43.3 ± 0.8
Pairwise	43.8 ± 1.2	43.3 ± 0.7

Oracle Performance of RobustDock. To investigate the potential of flexible docking methods compared to the cofolding, we leverage the perturbed crystal ligand as the initial ligand conformer. In Table 2, ROBUSTDOCK* further approaches AF3-level performance. This suggests that, compared to the cofolding methods which infer the structure from scratch, flexible docking can exploit folding model derived protein structures and strong ligand conformer priors to provide more targeted structural guidance for difficult docking cases without being trained on a massive amount of data and remaining highly efficient at inference.

5. Conclusion

In this work, we propose ROBUSTDOCK, enhancing generative model training with robustness for flexible docking, where the unique challenge is that the long-tailed structure shift distribution between the apo and holo state may potentially arise from either noisy or difficult data. Building on this insight, we propose several training techniques, including asynchronized time schedule, inverse shift sigmoid reweighting and use of weight decay to improve the training robustness with fine-grained control on protein degrees of freedom. Additionally, to align the confidence model to its true ranking utility, we propose the pairwise ranking loss for the confidence model training. Empirically, ROBUSTDOCK achieves state-of-the-art performance on the PDBBind benchmark and strong generalization on PoseBusters V2. On the challenging DockGen-E benchmark, it matches AF3 in performance when provided with the crystal ligand local structure. Beyond flexible docking, the robustness perspective is a promising avenue to improve biomolecular modeling accuracy, especially when training on synthetic data (Lewis et al., 2025) or developing biomolecular foundation models using cross modal data (Abramson et al., 2024).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal

consequences of our work, none which we feel must be specifically highlighted here.

References

Aali, A., Arvinte, M., Kumar, S., and Tamir, J. I. Solving inverse problems with score-based generative priors learned from noisy data. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 837–843. IEEE, 2023.

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Arai, M. Unified understanding of folding and binding mechanisms of globular and intrinsically disordered proteins. *Biophysical reviews*, 10(2):163–181, 2018.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Baptista, R., Dasgupta, A., Kovachki, N. B., Oberai, A., and Stuart, A. M. Memorization and regularization in generative diffusion models. *arXiv preprint arXiv:2501.15785*, 2025.

Bryant, P., Kelkar, A., Guljas, A., Clementi, C., and Noé, F. Structure prediction of protein-ligand complexes from sequence information with umol. *Nature Communications*, 15(1):4536, 2024.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, 2005.

Buttenschoen, M., Morris, G. M., and Deane, C. M. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

Calvo-Ordonez, S., Meunier, M., Cartea, A., Reisinger, C., Gal, Y., and Hernandez-Lobato, J. M. Weighted conditional flow matching. *arXiv preprint arXiv:2507.22270*, 2025a.

Calvo-Ordonez, S., Meunier, M., Cartea, A., Reisinger, C., Gal, Y., and Hernandez-Lobato, J. M. Weighted conditional flow matching. *arXiv preprint arXiv:2507.22270*, 2025b.

Chen, R. T. Q. and Lipman, Y. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g7ohDlTITL.

Clark, J. J., Benson, M. L., Smith, R. D., and Carlson, H. A. Inherent versus induced protein flexibility: Comparisons within and between apo and holo structures. *PLoS computational biology*, 15(1):e1006705, 2019.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. S. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kKF8_K-mBbS.

Corso, G., Deng, A., Polizzi, N., Barzilay, R., and Jaakkola, T. S. Deep confident steps to new pockets: Strategies for docking generalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=UfBIxpTK10.

Corso, G., Somnath, V. R., Getz, N., Barzilay, R., Jaakkola, T., and Krause, A. Composing unbalanced flows for flexible docking and relaxation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gHLWTzKiZV.

Daras, G., Shah, K., Dagan, Y., Gollakota, A., Dimakis, A., and Klivans, A. Ambient diffusion: Learning clean distributions from corrupted data. *Advances in Neural Information Processing Systems*, 36:288–313, 2023.

Daras, G., Dimakis, A. G., and Daskalakis, C. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. *arXiv preprint arXiv:2404.10177*, 2024.

Daras, G., Ouyang-Zhang, J., Ravishankar, K., Daspit, W., Daskalakis, C., Liu, Q., Klivans, A., and Diaz, D. J. Ambient proteins: Training diffusion models on low quality structures. *bioRxiv*, pp. 2025–07, 2025.

Devroye, L. Nonuniform random variate generation. *Handbooks in operations research and management science*, 13:83–121, 2006.

Domingo-Enrich, C., Drozdzal, M., Karrer, B., and Chen, R. T. Q. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Geffner, T., Didi, K., Zhang, Z., Reidenbach, D., Cao, Z., Yim, J., Geiger, M., Dallago, C., Kucukbenli, E., Vahdat, A., and Kreis, K. Proteina: Scaling flow-based protein structure generative models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TVQLu34bdw.

Geiger, M. and Smidt, T. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

Guo, H., Liu, S., and Jing, B. ForceFM: Enhancing protein-ligand predictions through force-guided flow matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=e7HEbUVryj.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Huang, Y., Zhang, O., Wu, L., Tan, C., Lin, H., Gao, Z., Li, S., Li, S., et al. Re-dock: Towards flexible and realistic molecular docking with diffusion bridge. *arXiv preprint arXiv:2402.11459*, 2024.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35: 24240–24253, 2022.

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.

Kearns, M. J., Schapire, R. E., and Sellie, L. M. Toward efficient agnostic learning. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 341–352, 1992.

Kessel, A. and Ben-Tal, N. *Introduction to proteins: structure, function, and motion*. Chapman and Hall/CRC, 2018.

Kingma, D. and Gao, R. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

Koes, D. R., Baumgartner, M. P., and Camacho, C. J. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8):1893–1904, 2013.

Koshland Jr, D. E. Application of a theory of enzyme specificity to protein synthesis. *Proceedings of the National Academy of Sciences*, 44(2):98–104, 1958.

Krishna, R., Wang, J., Ahern, W., Sturmfels, P., Venkatesh, P., Kalvet, I., Lee, G. R., Morey-Burrows, F. S., Anishchenko, I., Humphreys, I. R., et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

Krogh, A. and Hertz, J. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Lee, J., Hao Nguyen, C., and Mamitsuka, H. Beyond rigid docking: deep learning approaches for fully flexible protein–ligand interactions. *Briefings in Bioinformatics*, 26(5):bbaf454, 2025.

Lewis, S., Hempel, T., Jiménez-Luna, J., Gastegger, M., Xie, Y., Foong, A. Y., Satorras, V. G., Abdin, O., Veeling, B. S., Zaporozhets, I., et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, pp. eadv9817, 2025.

Liao, Z., You, R., Huang, X., Yao, X., Huang, T., and Zhu, S. Deepdock: enhancing ligand-protein interaction prediction by a combination of ligand and structure information. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 311–317. IEEE, 2019.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T., Lopez-Paz, D., Ben-Hamu, H., and Gat, I. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., and Wang, R. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of chemical research*, 50(2):302–309, 2017.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu, W., Wu, Q., Zhang, J., Rao, J., Li, C., and Zheng, S. TANKBind: Trigonometry-aware neural networks for drug-protein binding structure prediction. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=MSBDFwGYwwt.

Lu, W., Zhang, J., Huang, W., Zhang, Z., Jia, X., Wang, Z., Shi, L., Li, C., Wolynes, P. G., and Zheng, S. Dynamicbind: predicting ligand-specific protein-ligand complex structure with a deep equivariant generative model. *Nature Communications*, 15(1):1071, 2024.

McNutt, A. T., Francoeur, P., Aggarwal, R., Masuda, T., Meli, R., Ragoza, M., Sunseri, J., and Koes, D. R. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):43, 2021.

Michino, M., Vendome, J., and Kufareva, I. Ai meets physics in computational structure-based drug discovery for gpcrs. *NPJ Drug Discovery*, 2(1):16, 2025.

Monod, J., Wyman, J., and Changeux, J.-P. On the nature of allosteric transitions: a plausible model. *Journal of molecular biology*, 12(1):88–118, 1965.

Morehead, A. and Cheng, J. Flowdock: Geometric flow matching for generative protein-ligand docking and affinity prediction. *ArXiv*, pp. arXiv–2412, 2025.

Morehead, A., Giri, N., Liu, J., Neupane, P., and Cheng, J. Assessing the potential of deep learning for protein–ligand docking. *Nature Machine Intelligence*, pp. 1–10, 2025.

Pei, Q., Gao, K., Wu, L., Zhu, J., Xia, Y., Xie, S., Qin, T., He, K., Liu, T.-Y., and Yan, R. Fabind: Fast and accurate protein-ligand binding. *Advances in Neural Information Processing Systems*, 36, 2024.

Plainer, M., Toth, M., Dobers, S., Stärk, H., Corso, G., Marquet, C., and Barzilay, R. DiffDock-Pocket: Diffusion for pocket-level docking with sidechain flexibility. 2023.

Qin, Y., Madeira, M., Thanou, D., and Frossard, P. Defog: Discrete flow matching for graph generation. *arXiv preprint arXiv:2410.04263*, 2024.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

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

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Stärk, H., Ganea, O., Pattanaik, L., Barzilay, R., and Jaakkola, T. Equibind: Geometric deep learning for drug binding structure prediction. In *International conference on machine learning*, pp. 20503–20521. PMLR, 2022.

Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. Harmonic self-conditioned flow matching for multi-ligand docking and binding site design. *arXiv preprint arXiv:2310.05764*, 2023.

Storn, R. Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical report, International Computer Science Institute*, 11, 1995.

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

Tong, A., FATRAS, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Trott, O. and Olson, A. J. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

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

Verdonk, M. L., Cole, J. C., Hartshorn, M. J., Murray, C. W., and Taylor, R. D. Improved protein–ligand docking using gold. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 2003.

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

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

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

Zhou, W., Sprague, C. I., Viliuga, V., Tadiello, M., Elofsson, A., and Azizpour, H. Energy-based flow matching for generating 3d molecular structure. *arXiv preprint arXiv:2508.18949*, 2025.

A. Case Study

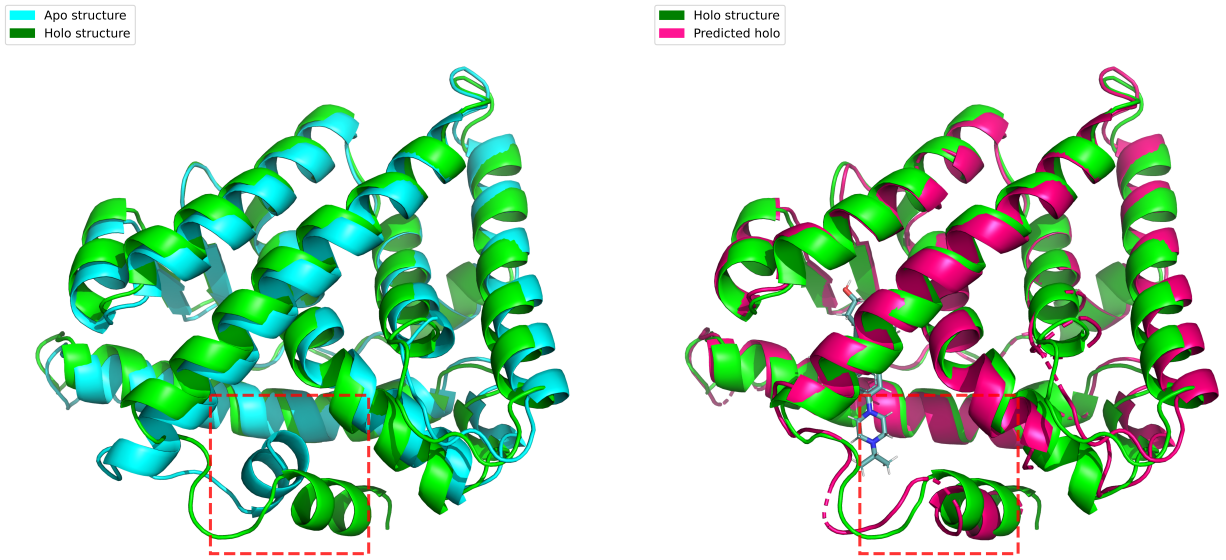


Figure 6. Conformational shift between apo, predicted holo and holo structures of 6a6k(PDBID). The pocket aligned apo holo RMSD: 3.92 Å. The AA RMSD between predicted holo and holo structure: 0.83 Å

B. Limitations

One limitation is that adding robustness components also introduces additional hyperparameters that must be co-optimized. A second limitation is even with enhanced robustness the model can still generate physically implausible molecule. However, this issue is largely orthogonal to robustness and can be improved with the reward finetuning (Domingo-Enrich et al., 2025) or inference time steering (Singhal et al., 2025).

C. Riemannian Flow Matching on Protein Conformation Shift

Conditional flow matching (CFM) is a simulation-free training method for continuous normalizing flow. To morph the distribution from source $p(x_0)$ to target $p(x_1)$, flow matching objective is defined to learn the vector field $u(x_t)$ that generates the probability path $p(x_t)$:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p(x_t)}\left[||v_\theta(x_t) - u(x_t)||^2\right]$$

However, this objective can not be leveraged for training since the true vector field is unknown. To alleviate this, the neural network is trained to match the conditional vector field $u(x_t|x_1, x_0)$ generating the conditional probability path $p(x_t|x_1, x_0)$, where x_0 and x_1 are samples from the source and target distribution. The marginalization of the conditional vector field yields the vector field $u(x_t)$ generating the path towards the data distribution (Lipman et al., 2023; Tong et al., 2024):

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,p(x_t|x_0,x_1),p(x_0,x_1)}\left[||v_\theta(x_t) - u(x_t|x_0, x_1)||^2\right]$$
$$u(x_t) = \int\int \frac{p(x_0, x_1)p(x_t|x_1, x_0)}{p(x_t)} u(x_t|x_1, x_0)dx_1dx_0$$

The conditional probability path $p(x_t|x_0, x_1)$ allows for flexible design. To reduce the transport cost during training, the conditional optimal transport path linearly transfers the source sample to the target sample:

$$p(x_t|x_0, x_1) = \mathcal{N}(x_t|tx_1 + (1-t)x_0, \sigma^2) \quad \mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,p(x_t|x_0,x_1),p(x_0,x_1)}\left[||v_\theta(x_t) - (x_1 - x_0)||^2\right]$$

Fortunately, the CFM loss shares the same parameter gradient with the flow matching loss:$\nabla_\theta \mathcal{L}_{\text{FM}} = \nabla_\theta \mathcal{L}_{\text{CFM}}$, and thus the ultimate learned conditional vector field $v_\theta(x_t)$ can be used for generating samples approximately from the data distribution

$x \sim p_{\text{model}}(x_1) \approx p_{\text{data}}(x_1)$ by simulating an ODE trajectory via $\frac{dx}{dt} = v_\theta(x_t)$. Compared to diffusion models, flow matching generalizes the source distribution from Gaussian to arbitrary, for instance, the apo structure distribution of protein.

Application to Protein Conformation Shift. As illustrated in section 3, the protein can be decomposed into backbone translation, rotation and side chain torsions, defining the product manifold of $\mathrm{T}(3)^N \times \mathrm{SO}(3)^N \times \mathrm{SO}(2)^m$, where N is the number of total residues and m is the number of side chain torsion angles. Because the translation $\mathrm{T}(3)^N$ is still Euclidean, the CFM loss can be directly exploited for training. For rotation, one must rely on the Riemannian flow matching (Chen & Lipman, 2024) for training:

$$r_t = \exp_{r_0}(t \log_{r_0}(r_1)) \quad \dot{r_t} = \frac{\log_{r_t}(r_1)}{1-t} \qquad \mathcal{L}_{\text{R-CFM}}(\theta) = \mathbb{E}_{t, p(r_t|r_0,r_1), p(r_0,r_1)} \left[||v_\theta(r_t) - \dot{r_t}||^2 \right]$$

where r_t is the geodesic interpolation of the apo r_0 and holo rotation r_1 and can be computed analytically with the Rodrigues formula, and $\dot{r_t}$ is the instantaneous vector field at time t following the geodesics. This formulation corresponds to the conditional optimal transport in the Euclidean case with the conditional path as the delta function over the geodesic interpolant r_t: $p(r_t|r_0, r_1) = \delta(r_t - (\exp_{r_0}(t \log_{r_0}(r_1))))$.

D. Inverse Sampling

The inverse sampling (Devroye, 2006) provides an efficient way of sampling the univariate random variable. It can be theoretically justified as follows. Let X be a continuous random variable with probability density $p_X(x)$ and cumulative distribution function (CDF) as:

$$F_X(x) = \int_{-\infty}^{x} p_X(t)\, dt$$

To generate samples from $p_X(x)$, inverse sampling draws uniform sample $U \sim \mathrm{Uniform}(0, 1)$ and compute the transformation with the inverse CDF:

$$X = F_X^{-1}(U)$$

Because X is a univariate random variable and F_X is monotonic and maps $\mathbb{R} \to [0, 1]$, its inverse F_X^{-1} readily exists. Hence, we have:

$$P(X \le x) = P(F_X^{-1}(U) \le x) = P(U \le F_X(x)) = F_U(F_X(x)) = F_X(x)$$

where the last equality is from $F_U(u) = u$ for a uniform random variable. Thus, X has CDF $F_X(x)$ and therefore PDF $p_X(x) = \frac{d}{dx} F_X(x)$. Equivalently, using the change of variable gives:

$$p_X(x) = p_U(F_X(x)) \left| \frac{dF_X(x)}{dx} \right| = 1 \cdot p_X(x)$$

This confirms the consistency. In a nutshell, one can sample from $p_X(x)$ by first sampling $U \sim \mathrm{Uniform}(0, 1)$ and compute X by $F_X^{-1}(U)$:

$$\boxed{X = F_X^{-1}(U), \quad U \sim \mathrm{Uniform}(0, 1) \;\Rightarrow\; X \sim p_X(x)}$$

In case of time distortion function $f(t)$ in Eq.(3) corresponds to the inverse CDF: $F_X^{-1}(t) = f(t)$ with $t \sim \mathrm{Uniform}(0, 1)$. This distortion function requires to be monotonic and preserve the boundary $f(0) = 0$, $f(1) = 1$ or $f(0) = 1$, $f(1) = 0$. Popular choices include *Polydec*: $f(t) = 2t - t^2$ and *Polyinc*: $f(t) = t^2$ (Qin et al., 2024).

E. Marginal Tilting of Inverse Shift Sigmoid

Notation. Let $x_0 \sim p(x_0)$ and $x_1 \sim p(x_1)$ with independent coupling as $p(x_0, x_1) = p(x_0)p(x_1)$. For an interpolant x_t and target conditional vector field $u(x_t|x_0, x_1)$, define the per-sample loss $\mathcal{L}_\theta(x, t) = \|v_\theta(x_t) - u(x_t|x_0, x_1)\|^2$. Let $w : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ be the weight. We expand the CFM loss of coupling $p(x_0, x_1)$ for illustration purpose:

$$\mathcal{L}(\theta) = \int_0^1 \iint \mathcal{L}_\theta(x, t)\, p(x_0, x_1)\, dx_0\, dx_1\, dt \tag{6}$$

Proposition E.1 (Adapted from Proposition 1 of (Calvo-Ordonez et al., 2025a)). *Define the tilted coupling density*

$$Z_w := \iint w(x_0, x_1)\, p(x_0)p(x_1)\, dx_0\, dx_1, \tag{7}$$

$$p_w(x_0, x_1) := \frac{w(x_0, x_1)}{Z_w}\, p(x_0)p(x_1), \tag{8}$$

where Z_w is the normalization constant. For any nonnegative, integrable per-sample CFM loss $\mathcal{L}_\theta(x_t, t)$ evaluated along the interpolant x_t from (x_0, x_1), the weighted CFM loss with independent coupling,

$$\mathcal{L}_w(\theta) := \int_0^1 \iint w(x_0, x_1)\, \mathcal{L}_\theta(x_t, t)\, p(x_0)p(x_1)\, dx_0\, dx_1\, dt, \tag{9}$$

is equivalent to the CFM loss in Eq. (6), up to the normalization constant Z_w.

Proof. By definition, $w(x_0, x_1)p(x_0)p(x_1) = Z_w\, p_w(x_0, x_1)$. Substitute into $\mathcal{L}_w(\theta)$:

$$\mathcal{L}_w(\theta) = \int_0^1 \iint \big[Z_w\, p_w(x_0, x_1)\big]\, \mathcal{L}_\theta(x_t, t)\, dx_0\, dx_1\, dt = Z_w \int_0^1 \iint \mathcal{L}_\theta(x_t, t)\, p_w(x_0, x_1)\, dx_0\, dx_1\, dt$$

Compared to the CFM loss Eq.6, training the weighted CFM loss with independent coupling is equivalent to (up to the normalization constant Z_w) the CFM loss drawing the source and target samples from the tilted coupling $p_w(x_0, x_1)$.

Tilted Coupling leads to Marginal Tilting. With the marginalization trick, we can obtain the marginal distribution of tilted coupling $p_w(x_0, x_1)$ denoted as $p_{w,0}(x_0)$ and $p_{w,1}(x_1)$:

$$p_{w,0}(x_0) = \int p_w(x_0, x_1)dx_1 = \frac{p(x_0)\phi_w(x_0)}{Z_w} \qquad \phi_w(x_0) = \int w(x_0, x_1)p(x_1)dx_1$$

$$p_{w,1}(x_1) = \int p_w(x_0, x_1)dx_0 = \frac{p(x_1)\psi_w(x_1)}{Z_w} \qquad \psi_w(x_1) = \int w(x_0, x_1)p(x_0)dx_0$$

where $\phi_w(x_0)$ and $\psi_w(x_1)$ are the one sided averages of the weight, known as the marginal potential. Obviously, compared to the original marginal $p(x_0)$ and $p(x_1)$, the density ratios $r_0(x_0)$ and $r_1(x_1)$ decide how much the marginal is tilted:

$$\frac{p_{w,0}(x_0)}{p(x_0)} = \frac{\phi_w(x_0)}{Z_w} = r_0(x_0) \qquad \frac{p_{w,1}(x_1)}{p(x_1)} = \frac{\psi_w(x_1)}{Z_w} = r_1(x_1)$$

This density ratio is the Radon–Nikodym derivative, determining how the probability mass is inflated and deflated.

Relative Variance. To explicitly quantify the tilting effect, relative variance (RV) can be used to measure how close the tilted marginal to the original marginal. Specifically, RV, the variance of the potential, upperbounds the KL divergence between the tilted marginal and original as follows:

$$p_{w,0}(x_0) = \frac{p(x_0)\phi_w(x_0)}{Z_w}, \qquad \mathbb{E}_p[r_0] = \int p(x_0)\frac{p_w(x_0)}{p(x_0)}dx_0 = \int p_{w,0}(x_0)dx_0 = 1$$

$$\begin{aligned}\mathrm{KL}\big(p_{w,0}\|p\big) &= \int p_{w,0} \log \frac{p_{w,0}}{p}\, dx_0 = \mathbb{E}_p[r_0 \log r_0] \le \mathbb{E}_p[r_0(r_0 - 1)] \\ &= \mathbb{E}_p[r_0^2] - 1 = \mathrm{Var}_p(r_0) = \frac{\mathrm{Var}_p(\phi_w)}{Z_w^2} := \mathrm{RV}_0(x_0).\end{aligned}$$

High or low RV indicates the tilting effect. More specifically, weighted conditional flow matching (Calvo-Ordonez et al., 2025b) adopts the exponential reweighting with Euclidean cost $w(x_0, x_1) = \exp\left(-||x_1 - x_0||/\epsilon\right)$ and show that in high dimension the choice of $\epsilon = \kappa\sqrt{d}$ with tuned κ barely changes the marginal and achieves entropy-regularized optimal transport between source and target distribution. However, our work targets to explicitly change the marginal through reweighting but with controlled and bounded influence in terms of RV. We demonstrate this with the following theorem.

Theorem 3.1 (Controlled RV for Inverse Shift Sigmoid) *Let* $d(x_0, x_1) = \mathrm{RMSD}(x_0, x_1)$ *and define*

$$w_{\beta,c}(x_0, x_1) := \sigma\big(-\beta(d(x_0, x_1) - c)\big), \qquad \sigma(z) = \frac{1}{1+e^{-z}},\ \beta > 0$$

Define

$$\phi_{\beta,c}(x_0) := \int w_{\beta,c}(x_0, x_1)\, p(x_1)\, dx_1, \qquad Z_{\beta,c} := \int \phi_{\beta,c}(x_0)\, p(x_0)\, dx_0$$

and $r_0(x_0) := \phi_{\beta,c}(x_0)/Z_{\beta,c}$ *so that* $\mathrm{RV}_0 := \mathrm{Var}_p(r_0) = \mathrm{Var}_p(\phi_{\beta,c})/Z_{\beta,c}^2$. *Assume:*

1. *There exists* $L_d < \infty$ *such that for all* x_1 *and all* x_0, x_0',

$$|d(x_0, x_1) - d(x_0', x_1)| \le L_d \|x_0 - x_0'\|$$

2. *For* $\mu := \mathbb{E}_p[X_0]$, $\mathbb{E}_p\|X_0 - \mu\|^2 < \infty$.

Then $\phi_{\beta,c}$ *is Lipschitz with constant* $L_\phi \le \beta L_d/4$, *and*

$$\mathrm{RV}_0 \ \le\ \frac{\left(\frac{\beta L_d}{4}\right)^2 \mathbb{E}_p\|X_0 - \mu\|^2}{Z_{\beta,c}^2}$$

Proof. We start by showing the bounded derivative of the sigmoid function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \Rightarrow \sigma'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = \sigma(z)(1-\sigma(z)) \le \frac{1}{4}$$

For $k \in (a, b)$, the mean value theorem implies:

$$\sigma(a) - \sigma(b) = \sigma'(k)(a-b) \Rightarrow |\sigma(a) - \sigma(b)| = |\sigma'(k)||b-a| \le \frac{1}{4}|a-b|$$

With this, plugging $f(d) := \sigma(-\beta(d-c))$ into the inequality above gives:

$$|f(d) - f(d')| \le \frac{\beta}{4}|d - d'|$$

Therefore, for any fixed x_1, the inverse shift sigmoid is bounded as:

$$|w_{\beta,c}(x_0, x_1) - w_{\beta,c}(x_0', x_1)| \le \frac{\beta}{4}|d(x_0, x_1) - d(x_0', x_1)| \le \frac{\beta L_d}{4}\|x_0 - x_0'\| \tag{10}$$

By the definition of marginal potential and linearity of integration, we have:

$$\phi_{\beta,c}(x_0) - \phi_{\beta,c}(x_0') = \int \big(w_{\beta,c}(x_0, x_1) - w_{\beta,c}(x_0', x_1)\big)\, p(x_1)\, dx_1$$

Taking absolute values and applying the triangle inequality for integrals gives:

$$\big|\phi_{\beta,c}(x_0) - \phi_{\beta,c}(x_0')\big| = \left|\int \Delta(x_1)\, p(x_1)\, dx_1\right| \le \int |\Delta(x_1)|\, p(x_1)\, dx_1$$
$$\Delta(x_1) := w_{\beta,c}(x_0, x_1) - w_{\beta,c}(x_0', x_1)$$

Using the bound Eq.(10) and $\int p(x_1)\, dx_1 = 1$ yield:

$$\big|\phi_{\beta,c}(x_0) - \phi_{\beta,c}(x_0')\big| \le \int \frac{\beta L_d}{4}\|x_0 - x_0'\|\, p(x_1)\, dx_1 = \frac{\beta L_d}{4}\|x_0 - x_0'\|$$

Thus $\phi_{\beta,c}$ is Lipschitz with constant $L_\phi \le \beta L_d/4$. Let $\mu = \mathbb{E}_p[X_0]$, by the definition of variance, we have:

$$\begin{aligned}\mathrm{Var}_p(\phi_{\beta,c}(X_0)) &= \mathbb{E}_p\big[(\phi_{\beta,c}(X_0) - \mathbb{E}[\phi_{\beta,c}(X_0)])^2\big] \\ &\le \mathbb{E}_p\big[(\phi_{\beta,c}(X_0) - \phi_{\beta,c}(\mu))^2\big] \\ &\le L_\phi^2\, \mathbb{E}_p\|X_0 - \mu\|^2 \\ &\le \left(\frac{\beta L_d}{4}\right)^2 \mathbb{E}_p\|X_0 - \mu\|^2\end{aligned}$$

where the first inequality is given that the variance is the minimizer of $\mathbb{E}[(f(x)-a)^2]$: $\mathrm{Var}(f(x)) \le \mathbb{E}[(f(x)-a)^2]$ for any constant a. Dividing by $Z_{\beta,c}^2$ yields the stated bound on RV_0. For x_1, a similar bound can be derived for RV_1. □

Assumptions. Assumption 1 is uniform-in-x_1 regularity condition that changing x_0 slightly must not change the RMSD to x_1 too wildly and same Lipschitz constant L_d works for every x_1, which is a relatively loose assumption in high dimension. Here, we demonstrate the case of $\mathrm{RMSD}(x_0, x_1)$.

Let $x_0, x_0', x_1 \in \mathbb{R}^{3N}$ be stacked Cartesian coordinates, written as:

$$x_0 = \begin{bmatrix} x_{0,1} \\ \vdots \\ x_{0,n} \end{bmatrix}, \qquad x_1 = \begin{bmatrix} x_{1,1} \\ \vdots \\ x_{1,n} \end{bmatrix}$$

with $x_{0,i}, x_{1,i} \in \mathbb{R}^3$ and i is the atom index. By the definition of RMSD:

$$\mathrm{RMSD}(x_0, x_1) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|x_{0,i} - x_{1,i}\|^2} = \frac{1}{\sqrt{N}}||x_0 - x_1||$$

For any fixed x_1, using the triangle inequality, we have:

$$|\mathrm{RMSD}(x_0, x_1) - \mathrm{RMSD}(x_0', x_1)| = \frac{1}{\sqrt{N}}\big|\,||x_0 - x_1|| - ||x_0' - x_1||\,\big| \le \frac{1}{\sqrt{N}}||x_0 - x_0'||$$

Since the same Lipschitz constant L_d should satisfy for every x_1, in practice, L_d can be chosen based on the smallest protein giving the largest $\frac{1}{\sqrt{N}}$. Hence, assumption 1 is satisfied for RMSD. Assumption 2 is the common assumption in generative models (Lipman et al., 2024).

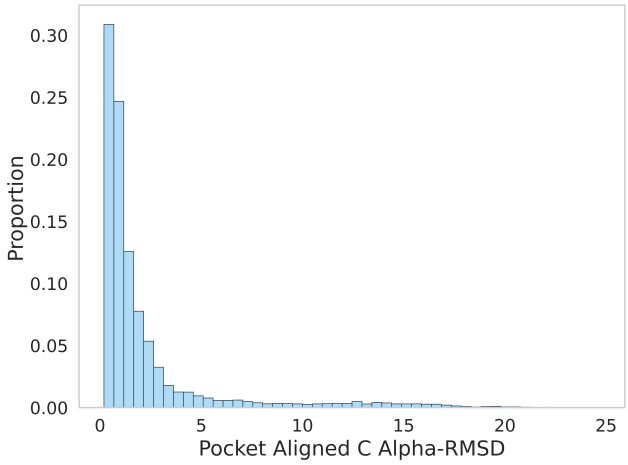
How β controls the Lipschitz RV bound. For inverse shift sigmoid reweighting $w_{\beta,c}(x_0, x_1) = \sigma(-\beta(d(x_0, x_1) - c))$, the Lipschitz-based relative variance bound takes the form:

$$\mathrm{RV}_0 = \frac{\mathrm{Var}_p(\phi_{\beta,c})}{Z_{\beta,c}^2} \;\le\; \frac{\left(\frac{\beta L_d}{4}\right)^2 \mathbb{E}_p\|X_0 - \mu\|^2}{Z_{\beta,c}^2}, \qquad Z_{\beta,c} = \mathbb{E}\big[\sigma(-\beta(d(X_0, X_1) - c))\big]$$

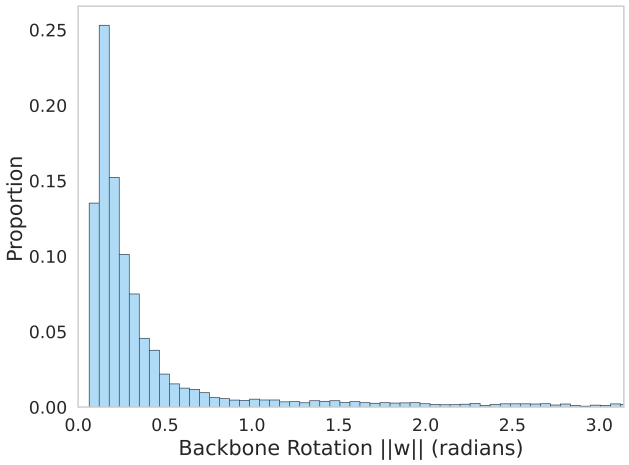
where $\phi_{\beta,c}(x_0) = \mathbb{E}_{p(x_1)}[w_{\beta,c}(x_0, X_1)]$ and L_d is the Lipschitz constant of $d(\cdot, x_1)$ in x_0. The numerator exhibits an explicit quadratic dependence on β. Besides, β also affects the normalization constant $Z_{\beta,c}$, so in general the bound depends on β through the combined factor $\beta^2/Z_{\beta,c}^2$. If the acceptance probability is small (small c), the term $1/Z_{\beta,c}^2$ can dominate and the bound can also blow up. In the high acceptance regime relevant to our setting (choosing c so that $Z_{\beta,c}$ stays bounded away from 0), i.e. $Z_{\beta,c} \ge Z_{\min} > 0$, the dependence becomes cleanly quadratic:

$$\mathrm{RV}_0 \;\le\; \frac{L_d^2\, \mathbb{E}_p\|X_0 - \mu\|^2}{16\, Z_{\min}^2}\, \beta^2$$

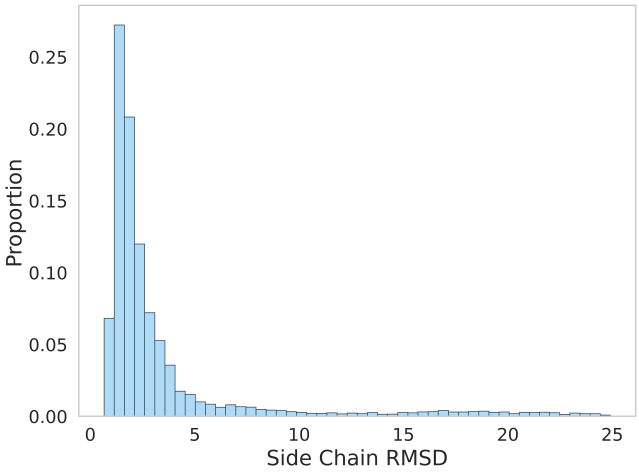
so decreasing β by a factor k decreases this Lipschitz RV upper bound by a factor k^2. Moreover, as $\beta \to 0$, $w_{\beta,c}(d) \to \sigma(0) = 1/2$, so $Z_{\beta,c} \to 1/2$ and the bound behaves like $O(\beta^2)$ up to a constant. Conversely, as $\beta \to \infty$, $w_{\beta,c} = \mathbf{1}\{d < c\}$ and $Z_{\beta,c} \to \mathbb{P}(d < c)$. *This recovers the unbalanced flow matching where the RV is uncontrolled with $\beta \to \infty$.*



(a) Backbone Translation



(b) Backbone Rotation



(c) Side Chains

Figure 7. Structure shift distributions from apo to holo for backbone translation, backbone rotation, and side-chain movement.

F. Which protein components require robustness for long tail?

To further visualize how large conformational shift manifest across different protein components, we plot fine-grained structure shift distributions Fig. 7 of (a) backbone translation, measured by the C-α RMSD between apo and holo structure, which reflects residue-wise displacement upon binding; (b) backbone rotation, which captures changes in the local rigid orientation of each residue; and (iii) side chains, quantified by the side chain all-atom RMSD. For backbone rotation, we use the rotation magnitude $||w||$ measuring the rotation shift. For side chain, we do not report the per-protein average side chain torsion change, because this summary can be misleading: the RMSD is often dominated by changes in the proximal torsion (typically χ_1), while the remaining torsions contribute less. As a result, a protein can exhibit a large side chain RMSD driven by a substantial shift in proximal angle χ_1 even when the average torsion angle change across χ_1–χ_4 remains small.

As shown in Fig. 7, the long tail of structure shift distribution appear for all three protein components contributing to the large conformational shift. This raises up a following question:

Which protein components require robustness for long tail?

Among these, backbone translation reflects the global displacement of the protein upon binding, while backbone rotation and side-chain torsion primarily capture local per-residue adjustments, namely, the residue local orientation and side chain flexibility.

Our hypothesis: Within the binding pocket, global misalignment is more likely to arise from noise in the apo conformation rather than true structural differences. Therefore, backbone translation should ideally be treated with greater robustness. Moreover, since most flexibility resides in side chains while backbone motions are typically limited (Clark et al., 2019), backbone rotations should require higher robustness compared to side chain.

To verify this hypothesis, we conduct an empirical study with several configurations. The robustness strength is controlled via α in the sample transform Eq.(3), with the increase of α for less dependence on the apo conformation for training. For all configurations, we only alter the setup of α for training and sampling time distortion while keeping all other setup the same as *RobustDock*.

Baseline setup. For all experiments, we consider the baseline setup where the backbone translation, rotation and side chain torsions are all sampled uniformly.

Config A: Backbone Translation. In this configuration, we sample backbone **translation** with time distortion ($\alpha = 1.5$), while backbone rotation and side chain torsions are uniformly sampled:

$$f_{\text{bb\_tr}}(t) = \frac{e^{1.5t} - e^{1.5}}{1 - e^{1.5}}$$

Config B: Backbone Rotation. In this configuration, we sample backbone **rotation** with time distortion ($\alpha = 1.5$), while backbone translation and side chain torsions are uniformly sampled:

$$f_{\text{bb\_rot}}(t) = \frac{e^{1.5t} - e^{1.5}}{1 - e^{1.5}}$$

Config C: Side Chain. In this configuration, we sample **side chain torsion** with time distortion ($\alpha = 1.5$), while backbone translation and rotation are uniformly sampled:

$$f_{\text{sc}}(t) = \frac{e^{1.5t} - e^{1.5}}{1 - e^{1.5}}$$

We point out that these experiments are to find out which protein components require robustness against noisy apo conformation and get a touch of each protein component's sensitivity to the robustness rather than find an optimal setup for α. We compare the validation metrics: fraction of Top5 ligand RMSD $< 2\mathring{A}$ and fraction of Top5 AA RMSD $< 1\mathring{A}$ with the baseline in Fig. 8 and 9.

Comments on Fig. 8. With robustness injected into backbone translation shown in Fig. 8a and 8b, the ligand validation metric shows a clear improvement over the baseline, with protein performance being maintained. For backbone rotation shown in Fig. 8c and 8d, ligand performance improves slightly, again without sacrificing protein performance. For the side chains in Fig. 8e and 8f, ligand performance remains comparable to the baseline, but protein performance degrades. These results suggest that injecting robustness into backbone translation and rotation is beneficial, especially for backbone translation, which shows the largest gains, while the side chain might not profit from adding robustness. However, there is a chance that setting $\alpha = 1.5$ imposes too strong robustness for side chain, since side chain typically contains more flexibility. Hence, we continue the experiments with the config D that α is decreased.

Config D: Side Chain. In this configuration, we sample **side chain torsion** with time distortion ($\alpha = 1.0$), while backbone translation and rotation are uniformly sampled:

$$f_{\text{sc}}(t) = \frac{e^{t} - e}{1 - e}$$

Comments on Fig. 9. By reducing the robustness strength from $\alpha = 1.5$ to 1.0, as shown in Fig. 9a and 9b, the protein performance matches the baseline, together with slightly improvement over the ligand performance. These observations demonstrate that while the most flexibility lies in the side chain, it can still benefit from robustness with careful control of robustness strength.

In Summary. Fig. 8 and 9 provide the empirical evidence of our hypothesis that the backbone translation which represents the global movement of residues, should be treated with greater robustness in contrast to the backbone rotation and side chain, which represent the local flexibility. Comparing to the backbone rotation, the side chain requires weaker robustness. Since the side chain usually exhibits more flexibility than the backbone, the large side chain shift has greater chance of being difficult rather than noisy data.

G. Structure Shift Distribution for Relaxation

Since the structure relaxation flow is trained on the output of the docking model which clearly contains a certain amount of "noise", we can similarly plot the structure shift distribution between the docking output and bound conformation. For each training example, five samples are generated resulting in total of approximately 81k samples shown in Fig. 10. As can be observed in Fig. 10, possibly due to insufficient network capacity and robust training of docking model, the relaxation flow exhibits a long-tailed distribution for both protein and ligand. Compared to the apo holo conformational shift in Fig. 1, the tail of structure relaxation appears to be tighter and more moderate.

H. Experimental Details

H.1. Data

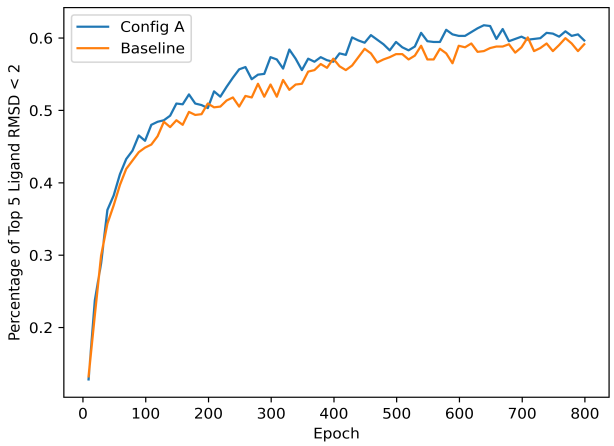
The PDBBind2020 used for training the model is downloaded from the git repository of DiffDock-Pocket[1] (Plainer et al., 2023). These data has been preprocessed by the PDBFixer to replace the non-standard residues, add missing atoms, and remove heterogens present in the PDB file. The ESMFold structure is predicted by the model `esmfold.pretrained.esmfoldv1()` from the esm repository[2] with the sequence derived from the PDB file after being processed by PDBFixer. The generated structure is also further processed by the PDBFixer for adding missing atoms and removing the hydrogen. The resulting ESMFold and PDB structures are used as the apo and holo conformation. For the PoseBusters V2 benchmark, test examples with PDBID 7FRX_088, 7M31_TDR, 7OP9_06K and 8F4J_PHO do not fit into the memory of 80GB RTX A100 after setting the chunk size to 2 for ESMFold, hence being filtered out.

Preprocessing. The dataset preprocessing follows the previous work (Corso et al., 2025). The key step involves in ligand and protein conformer matching. Because the local structure of the RDKit ligand conformer slightly differs from the ground truth, training the model with the ground truth local structure and then testing with the RDKit ones introduces a test time shift. Hence, the RDKit conformer should be aligned with the ground truth conformer to mitigate this issue. This is done using the differential evolution algorithms (Storn, 1995), searching the optimal rotation for each rotatable bonds of RDKit ligand for minimizing the RMSD with the ground truth conformer. After that, the optimized RDKit conformer is used as the bound ligand for training the generative model. This process is the same as the DiffDock (Corso et al., 2023) and TorsionDiff (Jing et al., 2022). For protein, to avoid the unnecessary confusion of global orientation and reduce the transport cost during training, the apo structure is first globally Kabsch aligned with the holo structure. The globally aligned apo and reference holo structures after pocket reduction are used to calculate the statistics in this work. In the same spirit of ligand conformer matching, the globally aligned apo structure is continued to be locally aligned for backbone rotation and side chain sequentially with the reference holo structure. The apo backbone plane is rotated by exactly aligning the frame normal vector and the CA–N bond direction. Similar to the ligand, side chain angles are aligned by optimizing the side chain RMSD to the holo structure using differential evolution. Finally, for training the flow matching model, the backbone rotation and side chain torsion derived from the fully aligned apo structure are used as the target samples, and the target sample of backbone translation is still from the holo structure.

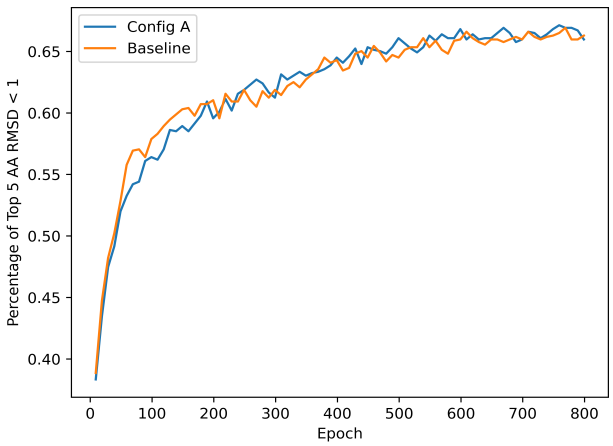
Pocket Definition. First, we define the holo pocket residues as those containing at least one atom within the atom-level cutoff (5Å) of any ligand atoms. We then compute the pocket center as the mean of the apo Cα positions of these residues. Given this pocket center, the pocket residues are further expanded with the residues whose Cα atoms in the apo structure lie within the residue-level cutoff (20Å) of the center. This apo based update is necessary to avoid missing the residues with large displacement, moving away from the pocket.

[1] https://github.com/plainerman/DiffDock-Pocket

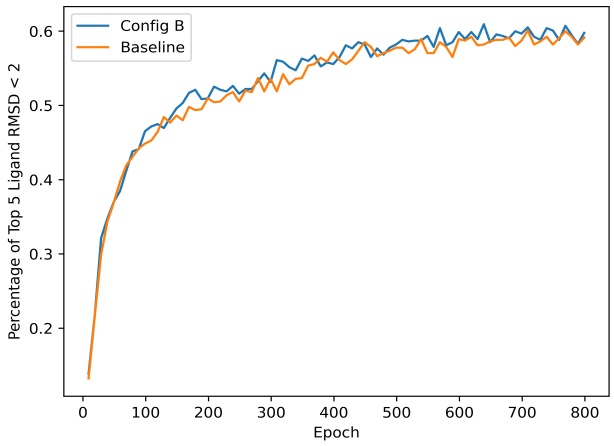
[2] https://github.com/facebookresearch/esm



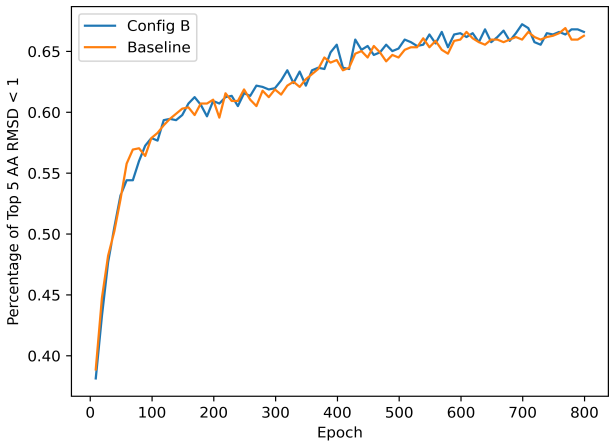
(a) Config A: fraction of Top5 ligand RMSD < 2Å



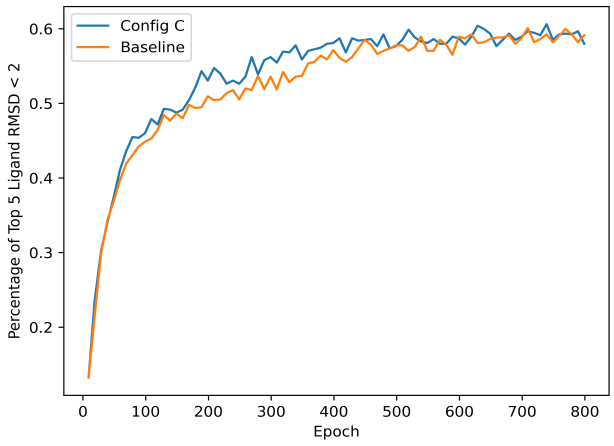
(b) Config A: fraction of Top5 AA RMSD < 1Å



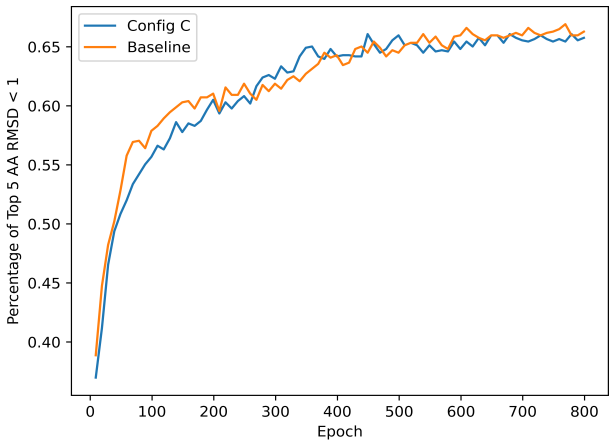
(c) Config B: fraction of Top5 ligand RMSD < 2Å



(d) Config B: fraction of Top5 AA RMSD < 1Å

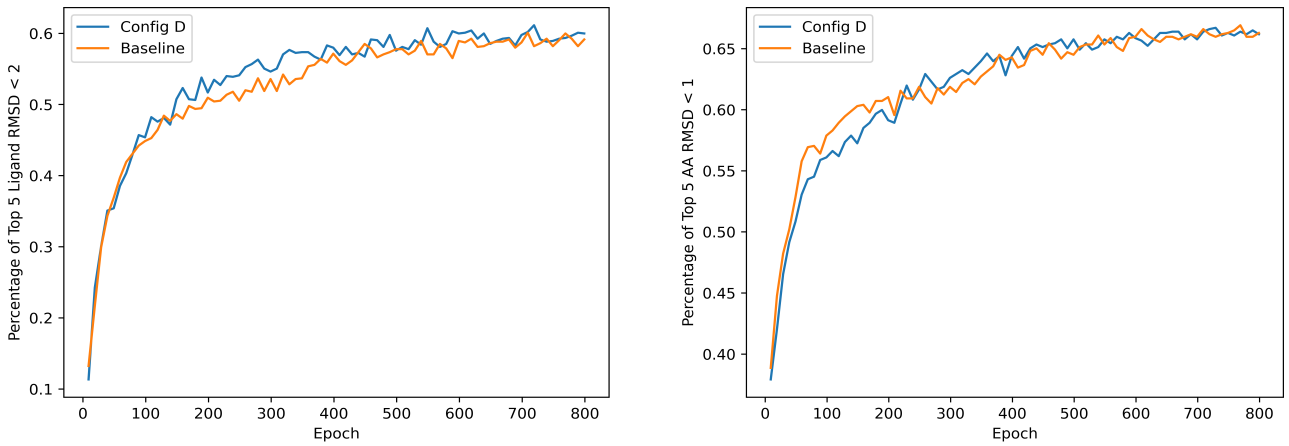


(e) Config C: fraction of Top5 ligand RMSD < 2Å



(f) Config C: fraction of Top5 AA RMSD < 1Å

Figure 8. Validation curve with robustness injected into different components $\alpha = 1.5$. Config A: Backbone Translation. Config B: Backbone Rotation. Config C: Side Chains.



(a) Config D: fraction of Top5 ligand RMSD < 2Å

(b) Config D: fraction of Top5 AA RMSD < 1Å

Figure 9. Validation curve with weaker robustness injected into side chains of Config D with $\alpha = 1.0$.

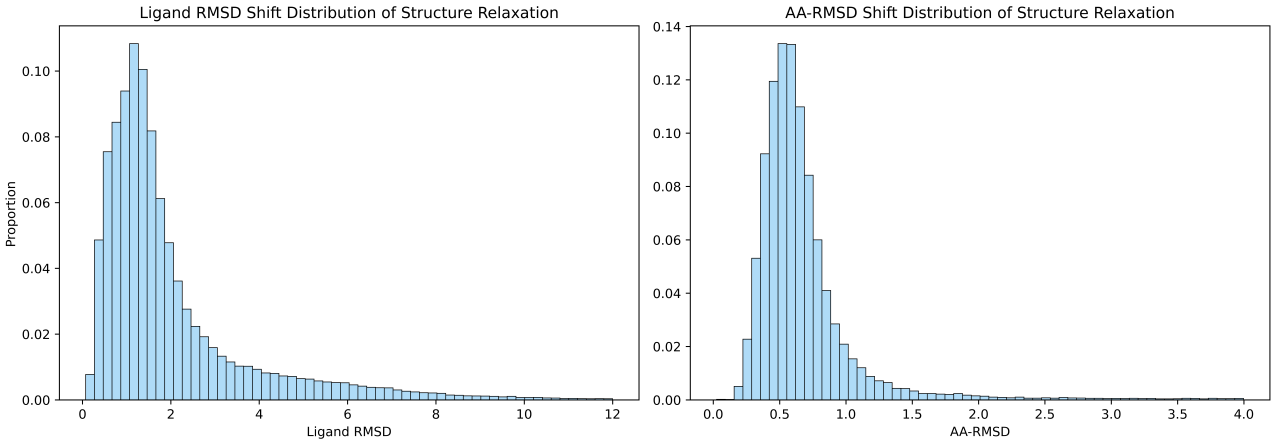


Figure 10. Ligand RMSD and pocket aligned AA RMSD shift distribution of docking output and holo conformation.

H.2. Architecture

As this work primarily studies the training robustness of generative modeling for flexible docking, we follow the architecture of the Flexdock. The network is the Tensor Field Network (Thomas et al., 2018) implemented with `e3nn` (Geiger & Smidt, 2022). The asynchronized time schedule needs slightly modifying the time embedding layer illustrated below. We summarize the architecture here, and more details can be found in Appendix C of DiffDock (Corso et al., 2023).

Feature Initialization and Embedding. The ligand node features are initialized with the typical atomic properties, such as the total number of atoms, chirality type, formal charge, total degree, implicit valence, etc. The ligand edge feature is the one-hot vector of the bond type. The protein node features are initialized with the residue type concatenated with esmfold embedding from `esm2.t36.3B.UR50D`. For time conditioning, the sinusoidal embedding with a predefined embedding scale is used. With asynchronized time schedule of backbone translation, rotation and side chains, we concatenate these time embeddings for protein receptor and atom graph. The time embedding and initialized node features of ligand and protein are then passed through separate MLP layers projecting to a hidden dimension.

Feature Extraction. The feature extraction layers are based on the tensor product convolution, where messages are computed as the tensor product between the spherical harmonics of edge vector, precomputed for each graph, and the node embedding. The spherical harmonics only contain the tensor with degree 1, and path weights are given by the edge features. The messages are then aggregated, followed by the layer normalization (Ba et al., 2016) and residual connection with the

node embedding before the tensor product convolution. The edges attributes are updated between the layers based on the updated node embedding.

Prediction Head. For the ligand, the global rotation and translation are predicted by the center convolution layer which is the same as the backbone tensor product convolution, but the graph is built with the center of mass for each ligand fully connected with all the ligand atoms. The $SE(3)$ equivariant output is set to be 2 vector features, corresponding to translation and rotation. For ligand bond prediction, a bond convolution layer with k $SE(3)$ invariant output is constructed by aggregating messages for each bond based on a radius graph between the bond positions and the ligand atom positions. For backbone prediction, an equivariant MLP layer is fed with the receptor node features output by the feature extraction layers and predict 2 vector outputs (translation and rotation) for each residue. For the flexible side chains, similar to the ligand bond, a side chain convolution layer is used for predict m $SE(3)$ invariant output based on the side chain radius graph.

This summarizes the architecture for manifold docking. For structure relaxation flow, the feature initialization and extraction layers remain to be the same, while prediction head is the tensor product convolution with 1 vector output for each atom.

H.3. Validation Scheme

For manifold docking, we conduct the validation inference over all 968 validation examples with 20 steps for sampling the protein ligand complex. To adhere more to realistic docking, we use the original apo structure rather than the one Kabsch aligned with the holo structure for inference. For each validation example, we generate 5 samples and monitor the percentage of Top5 (average) ligand RMSD < 2Å and Top5 (average) all-atom RMSD < 1Å for convergence. Generating 5 samples rather than just one sample can greatly reduce the variance of the validation metrics, but also causes 5 times more validation compute (around 9.5 mins). Additionally, we use the last checkpoint as the final model, rather than the one yielding the highest ligand or all-atom RMSD performance, to avoid overfitting on the validation set. For structure relaxation, the validation inference is conducted every 10 epochs with 5 sampling steps. For each validation example, we generate 5 samples and monitor the auxiliary energy loss and fraction of total samples with ligand RMSD < 2Å for convergence.

H.4. Training and Sampling Details

The configurations of the robustness component are here:

Table 5. Hyperparameters of robustness components.

(a) Manifold Docking

Component	Hyperparameter	Setup
ATS	Backbone translation α	1.5
	Backbone rotation α	0.5
	Side chain α	0.5
Inverse Shift Sigmoid	c	8.0
	β	1.0
Weight Decay	λ	0.02

(b) Structure Relaxation

Component	Hyperparameter	Setup
Ligand Sigmoid	c	3.0
	β	1.0
Protein Sigmoid	c	1.5
	β	1.0

Training. We train both the manifold docking, structure relaxation and confidence models using the AdamW optimizer (Loshchilov & Hutter, 2017) with an initial learning rate of 1×10^{-3} and BF16 mixed precision. For manifold docking, training runs for 800 epochs (approximately 4 days and 15 hours) on 8 NVIDIA RTX A100 80 GB GPUs with a batch size of 4 per GPU. The learning rate is decayed by a factor of 0.7 based on the inference result of the fraction of top5 ligand RMSD <2Å, with scheduler patience of 30 epochs. For structure relaxation, training runs for 100 epochs (approximately 7 hours) on 2 NVIDIA RTX A100 40 GB GPUs with a batch size of 64 per GPU. The learning rate is decayed by a factor of 0.7 based on the inference result of the fraction of top5 ligand RMSD <1Å, with scheduler patience of 20 epochs. For the confidence model, training runs for 120 epochs (approximately 11 hours) on 8 NVIDIA RTX A100 40 GB GPUs with a batch size of 5 per GPU, with three poses sampled for each batch example. For validation, we sample 10 poses for each example and compute the top1 ligand RMSD < 2 Å. This metric decays the learning rate with a scheduler patience of 30 epochs and decay factor of 0.7.

Table 6. Top-10 PDBBind ESMFold docking performance with different threshold values of c on a smaller model (7.9M).

Threshold c	Top-10 Ligand RMSD < 2 Å (↑)	Top-10 AA RMSD < 1 Å (↑)
8.0	52.6 ± 0.6	42.0 ± 0.8
7.0	52.9 ± 0.9	44.7 ± 0.2
5.0	53.0 ± 0.9	41.6 ± 0.4

Sampling. For each test complex, we generate 10 candidate samples, using 20 sampling steps for docking and 5 steps for structure relaxation, respectively. The confidence model ranks the docking output samples, and the top-scoring sample passing the energy filtering is selected as the ultimate prediction.

I. Additional Results

Inverse Shift Sigmoid on a Smaller Scale. One limitation of inverse shift sigmoid is that it universally downweights the large shift examples beyond the threshold without distinguishing the noisy and difficult data. More broadly, in deep learning, "noise" can extend beyond explicit label or input corruption to difficult examples due to "misspecification" (Kearns et al., 1992). In this view, examples that fall outside the hypothesis class exhibit irreducible uncertainty and thus require robustness methods that can handle both noisy and difficult examples. For example, robust loss functions limit the influence of such high loss samples. In our setting, examples with larger conformational shifts also tend to produce larger losses, and are therefore softly downweighted by the inverse shift sigmoid. To examine this, we trained a smaller RobustDock (7.9M) with different threshold values of c. As shown in Table 6, we observe that the smaller model performs better with a smaller c(=7) than a larger model ($c = 8$), suggesting that it benefits from stronger downweighting of large shift examples due to misspecification inducing more "difficult" examples.

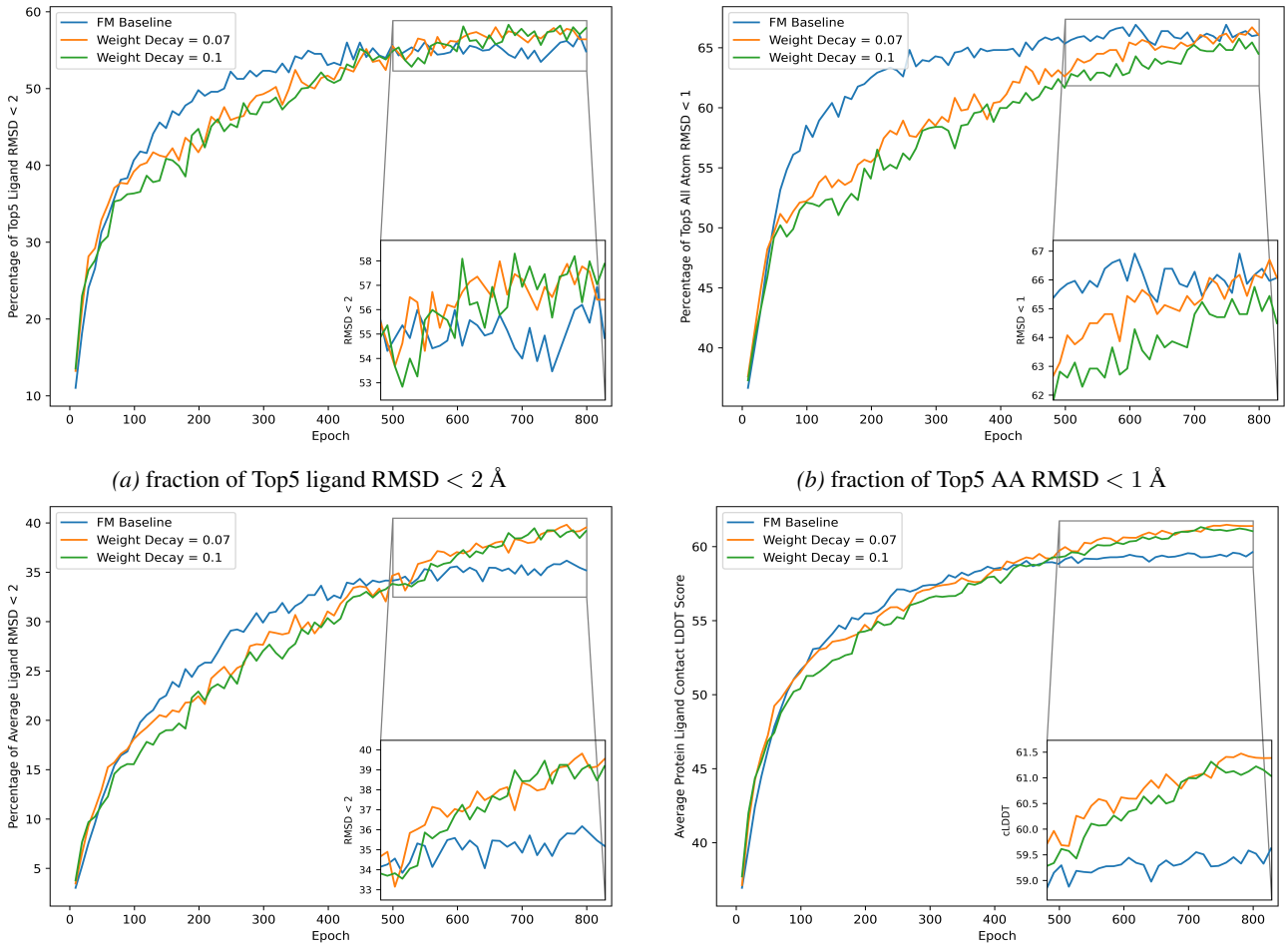
Delayed Generalization with Weight Decay. Fig. 11 visualizes the validation curve on different weight decay setups compared with the flow matching baseline (weight decay=0) on 4 docking metrics: the fraction of top5 ligand RMSD < 2Å, the fraction of average ligand RMSD < 2Å, top5 AA RMSD < 1Å and average protein ligand contact local difference distance tests (LDDT) score. It can be observed that with a proper setup of weight decay(=0.07), the generalization gets delayed and improved on ligand metrics and retain the protein metrics compared with the baseline. With an overregularized setup (=0.1), the ligand metrics saturate, while the protein metric degrades.

Validation Curve of Pairwise Ranking and Ligand RMSD Classifier. Since the classification and ranking are two separate tasks, it doesn't directly provide a common ground for validation comparison. Hence, to align the validation to the test case, we use the percentage of ligand RMSD < 2 Å of the top1 selected ligand, where 10 poses are scored for each validation example. Figure 12 shows the validation curve comparison that pairwise ranking loss improves over the ligand RMSD classification on the top selected prediction.

Conformational Selection and Induced Fit. Table 7 summarizes how asynchronized protein ligand sampling speed relates to classical docking mechanisms. In realistic docking, binding dynamics typically reflect a hybrid of conformational selection and induced fit.

Samples. Fig. 13 visualizes six randomly picked samples predicted by RoBustDock.

Figure 11. Validation curve of different weight decay setup compared with the flow matching baseline.



(a) fraction of Top5 ligand RMSD < 2 Å

(b) fraction of Top5 AA RMSD < 1 Å

(c) fraction of Average ligand RMSD < 2 Å

(d) Average protein ligand contact LDDT score

Figure 12. Validation Curve of Pairwise Ranking and ligand RMSD Classifier.

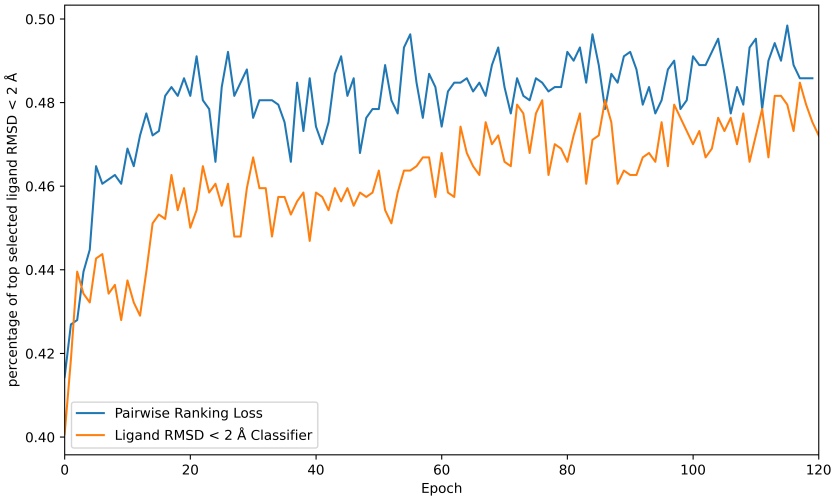
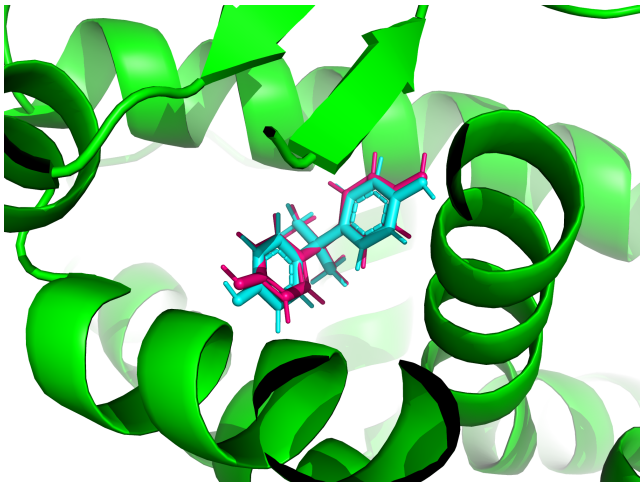
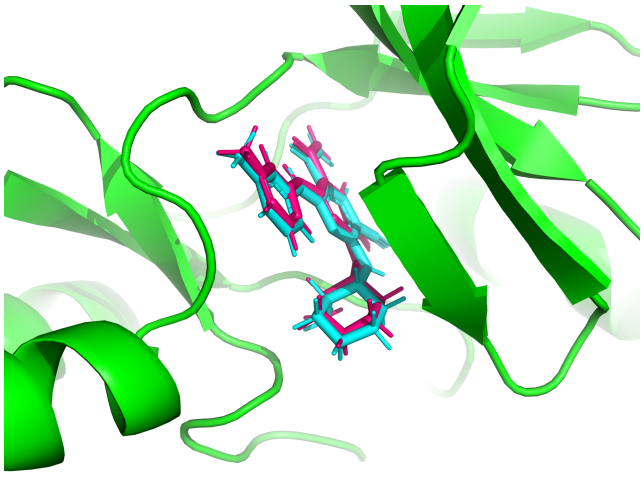


Table 7. Interpretation of asynchronized protein ligand sampling as classical docking mechanisms.

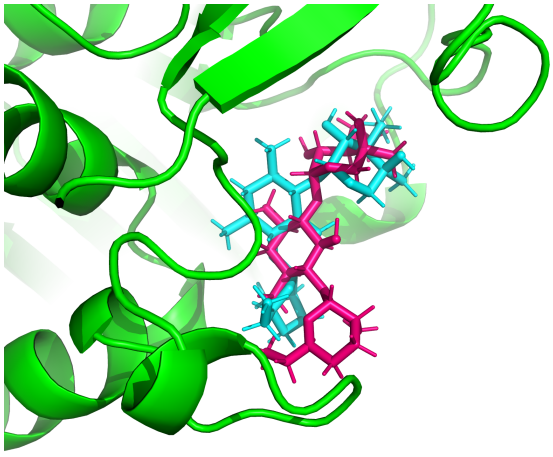
Sampling speed	**Docking model**	**Interpretation**
Protein fast, ligand slow	Conformational selection (Monod et al., 1965)	The protein preexists in an ensemble of conformations, and the ligand selectively binds the conformation that best matches it.
Ligand fast, protein slow	Induced fit (Koshland Jr, 1958)	Ligand binding drives subsequent protein conformational change toward the bound state.
Both / comparable	Hybrid / combined view	Binding typically reflects a combination of conformational selection and induced fit: partial pre-existing conformational preference followed by ligand-induced relaxation of the complex.



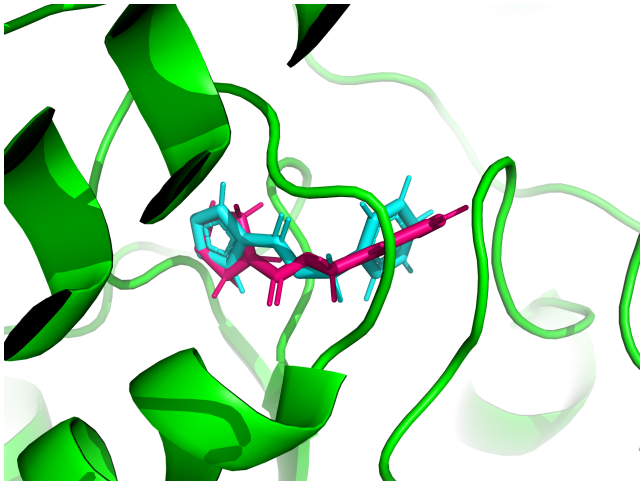
(a) PDBID: 6i63



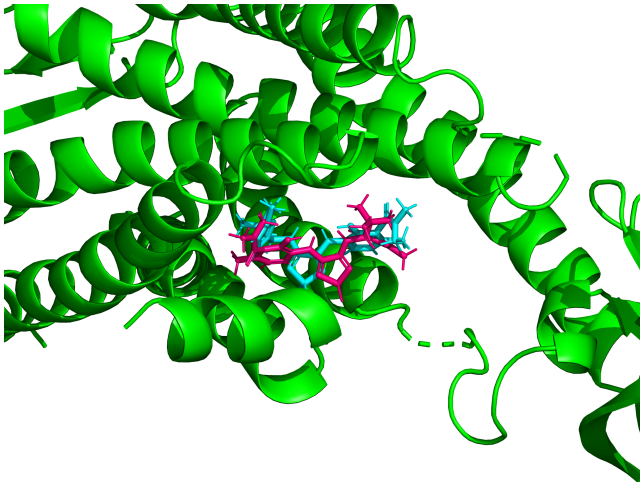
(b) PDBID: 6jmf



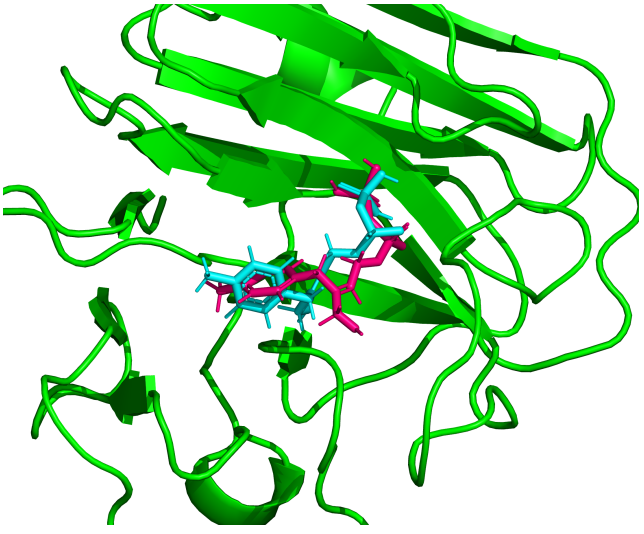
(c) PDBID: 6np3



(d) PDBID: 6qrf



(e) PDBID: 6qzh



(f) PDBID: 6j06

Figure 13. Visualization of randomly picked samples predicted by RoBUSTDock. Ground Truth in cyan and Predicted ligand pose in hotpink.