LATENTDOCK: Protein-Protein Docking with Latent Diffusion

Matt McPartlon[1], **Céline Marquet**[1,2], **Tomas Geffner**[1,3], **Daniel Kovtun**[1], **Alexander Goncearenco**[1], **Zachary Carpenter**[1], **Luca Naef**[1], **Michael Bronstein**[1,4], **Jinbo Xu**[5]
[1]VantAI, [2]TU Munich, [3]UMass Amherst, [4]Oxford, [5]TTI Chicago
{matt,celine,tomas,danny,alex,zach,luca,michael}@vant.ai,
jinbo.xu@gmail.com

Abstract

Interactions between proteins form the basis for many biological processes, and understanding their relationships is an area of active research. Computational approaches offer a way to facilitate this understanding without the burden of expensive and time-consuming experiments. Here, we introduce LATENTDOCK, a generative model for protein-protein docking. Our method leverages a diffusion model operating within a geometrically-structured latent space, derived from an encoder producing roto-translational invariant representations of protein complexes. Critically, it is able to perform flexible docking, capturing both backbone and side-chain conformational changes. Furthermore, our model can condition on binding sites, leading to significant performance gains. Empirical evaluations show the efficacy of our approach over relevant baselines, even outperforming models that do not account for flexibility.

1 Introduction

Protein complexes serve as vital components for various cellular functions, with protein-protein interactions playing a pivotal role in understanding these processes. The application of machine learning methods in this context has proven to be both highly effective and computationally efficient. However, it is imperative to recognize the inherent flexibility associated with protein complexes during their interactions, characterized by two critical aspects: (i) the absence of a fixed or singular position for the complex, and (ii) the propensity for conformational changes in proteins upon docking.

Recently proposed machine learning-based methods often face challenges in effectively addressing both aspects concurrently. For instance, while current regression-based methods (Evans et al., 2021; McPartlon & Xu, 2023) enable flexible docking, they operate deterministically, failing to produce alternative solutions when binding sites are hard to discern. In essence, these approaches offer no way to reconcile incorrect predictions. On the other hand, the recent diffusion-based method DIFFDOCK-PP (Ketata et al., 2023) naturally produces a variety of poses, but requires treating individual chains in a complex as rigid bodies.

We propose LATENTDOCK, a method designed to address both aspects simultaneously. Our approach is based on a diffusion model operating on a geometrically-structured latent space. LATENTDOCK performs protein-protein docking at a *full-atom resolution*, enabling *flexible* backbone and side chain conformations. In contrast to deterministic methods, LATENTDOCK leverages diffusion models to offer the advantage of producing multiple potential complex structures. In empirical evaluations, LATENTDOCK demonstrates state-of-the-art performance in protein-protein docking. It notably outperforms DiffDock-PP by a significant margin and shows similar performance to DockGPT.

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

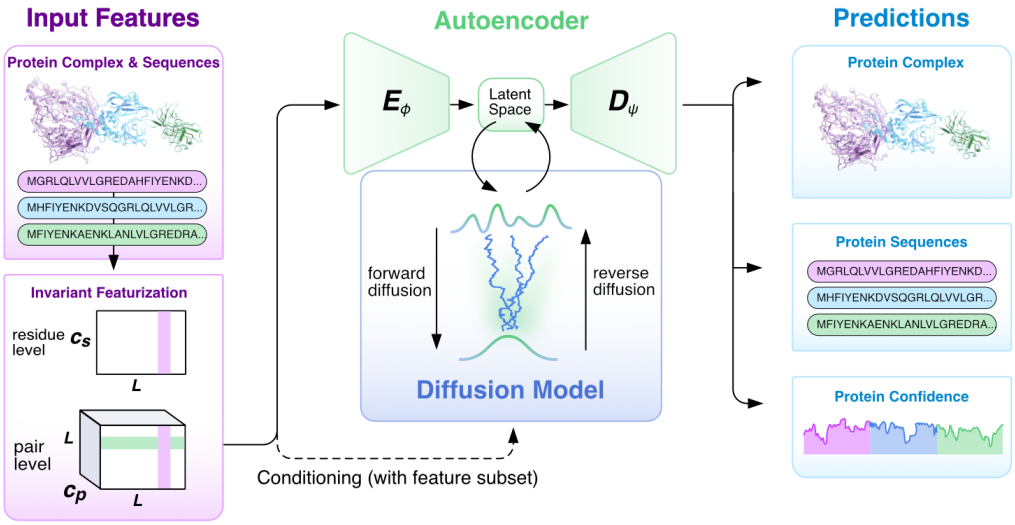

Figure 1: **LatentDock overview**. LatentDock extracts sequence and roto-translational invariant features from each individual chain, and uses a conditional latent diffusion model (+ decoder) to produce the final protein complex conformation.

2 Related Work

Protein docking has been a longstanding challenge in structural biology. Traditional physics-based models treat this as a search problem, evaluating the energy of each conformation in a massive set of potential candidates (Chen et al., 2003; De Vries et al., 2010; Yan et al., 2020).

Multiple machine learning methods have been introduced to lighten the computational costs of traditional protein-protein docking methods. Regression-based methods produce competitive results (Ganea et al., 2021; McPartlon & Xu, 2023), but only propose a single conformation for the final complex. Further, the regression methods inherently converge to the mean, which may be problematic when the true complex is difficult to discern, or multiple valid poses are possible (Corso et al., 2022).

Recently, diffusion models have been applied to the protein-protein docking task to address this limitation (Ketata et al., 2023). However, existing approaches are subject to certain limitations, as they either only handle docking of two proteins, and/or treat them as rigid bodies, which is inconsistent with the flexible nature of protein interactions.

3 Latent Diffusion for Protein-Protein Docking

This section introduces LatentDock (fig. 1), a generative model for flexible protein-protein docking. It consists of a latent diffusion model (Rombach et al., 2022; Vahdat et al., 2021) operating on the geometrically structured latent space of a pre-trained autoencoder. The training of LatentDock follows a two-stage approach (Rombach et al., 2022). First, we train an autoencoder tailored for protein complexes (described in section 3.2), using roto-translational invariant features (section 3.1). Second, with the autoencoder frozen, we train a diffusion model that operates in this latent space (section 3.3). Theoretical considerations concerning properties of the learned distribution are discussed in appendix C.

3.1 Roto-translational Invariant Features for Protein Complexes

Given a protein complex, we extract roto-translational invariant features to coarsely characterize the sequences and structures of its individual chains. These features are then used to train the autoencoder and the latent diffusion. We follow the features used by DockGPT (McPartlon & Xu, 2023), which are split into three types: **residue-level** (including amino acid type, sequence position, and backbone angles), **intra-chain-level** (information for pairs of residues i, j in the same chain,

including their distance, relative orientation, and sequence separation), and **inter-chain-pair** (information for each pair of residues i, j in different chains). To train the autoencoder, we include all relevant distance and angle information as inter-chain features, but remove this when training the latent diffusion model for docking.

For each residue i, or residue pairs i, j, all the aforementioned features are one-dimensional arrays. We provide details on how all these features are generated in appendix B. Given a protein complex with L residues, residue-level features are combined into an $L \times c_s$ matrix $\mathbf{s}$, and pair-level features are combined into an $L \times L \times c_p$ tensor $\mathbf{p}$.

3.2 Autoencoder with Roto-translational Invariant Latent Space

The first component of LATENTDOCK is an **autoencoder**, consisting of a stochastic **encoder** (which maps a protein complex to a roto-translational invariant latent representation) and a **decoder** (which reconstructs the input complex structure given its latent representation). As detailed in appendix D, the autoencoder is trained independently of the diffusion, by minimizing a combination of the reconstruction loss (for the predicted structure) and the cross-entropy loss (for the predicted sequence).

Encoder $\mathcal{E}_\phi$ Given a protein complex, the encoder computes the mean and variance of a Gaussian distribution over the latent space, which is sampled to produce the latent representation. Each layer (we use 8 layers) in the encoder is given by

$$\mathbf{s}_i \leftarrow \mathrm{PairBiasAtt}(\mathbf{s}, \mathbf{p}_{i:}), \tag{1}$$

where $\mathbf{s}$ and $\mathbf{p}$ are the residue-level and pair features, respectively, and $\mathrm{PairBiasAtt}$ is the pair-biased attention layer from Jumper et al. (2021). The mean and log-scale of the Gaussian distribution are then obtained as

$$\mu_i \leftarrow \mathrm{Linear}(\mathbf{s}_i), \quad \log \sigma_i \leftarrow \mathrm{Linear}(\mathbf{s}_i). \tag{2}$$

Finally, the latent representation $\mathbf{z}$ (an $L \times 16$ matrix) is obtained as $\mathbf{z}_i = (\tilde{\mathbf{z}}_i - \mathrm{mean}(\tilde{\mathbf{z}}_i))/\mathrm{std}(\tilde{\mathbf{z}}_i)$, where $\tilde{\mathbf{z}} \sim \mathcal{N}(\mu, \sigma^2)$. This latent representation jointly captures structural and sequence information for each residue in the input complex.

Decoder $\mathcal{D}_\psi$ Given a latent representation $\mathbf{z}$, the decoder is designed to reconstruct the input protein complex. We use the structure module from AlphaFold2 (Jumper et al., 2021), where each residue in the reconstructed backbone (represented as the frame formed by the N-C_α-C atoms) is identified with a rigid transformation $\mathbf{T}_i$ consisting of a translation and a rotation. Using $\mathbf{q}$ to denote the $L \times L \times c_q$ tensor obtained by combining two *pair* features (sequence separation and relative chain information, see section 3.1), each layer (we use 8 layers) in the decoder is given by

$$\mathbf{z}_i \leftarrow \mathrm{IPA}(\mathbf{z}, \mathbf{T}_{:}, \mathbf{q}_{i:}), \quad \mathbf{z}_i \leftarrow \mathrm{MLP}(\mathbf{z}_i), \quad \mathbf{T}_i \leftarrow \mathbf{T}_i \circ \mathrm{BackboneUpdate}(\mathbf{z}_i). \tag{3}$$

The invariant point attention (IPA) and backbone update operations are described in detail in Jumper et al. (2021). The side chain angles, amino acid type (logits over the 20 natural amino acids), and confidence score are then predicted as (for each residue i in the backbone)

$$\mathrm{angles}_i = \mathrm{MLP}(\mathbf{z}_i), \quad \mathrm{aa}_i = \mathrm{Linear}(\mathbf{z}_i), \quad \mathrm{conf}_i = \mathrm{Linear}(\mathbf{z}_i). \tag{4}$$

3.3 Latent Space Diffusion

The second component of LATENTDOCK is a **conditional diffusion model** operating in the autoencoder latent space. This diffusion is trained in a second step, after training and freezing the autoencoder (Rombach et al., 2022).

As detailed in appendix A, diffusion models define a forward process that gradually diffuses samples $\mathbf{z}$ (in our case latent representations) by running a diffusion process ("noising"). Then, they generate samples by reversing this process ("denoising"). This requires training a *score network* $s_\theta(z_t, t, c)$, where c represents the conditioning information available to the model.

Score network Our score network $s_\theta(\mathbf{z}_t, t, c)$ resembles the encoder architecture, with extra updates for the *pair* features through triangular multiplicative layers (Jumper et al., 2021). Using $\mathbf{u}^t$ to denote the $L \times (16 + c_u)$ matrix obtained by concatenating $\mathbf{z}_t$ and the *residue-level* features in c,

and $\mathbf{r}$ to denote the $L \times L \times c_r$ tensor obtained by combining *pair-level* features in c, each layer in the score network consists of (we use 12 layers)

$$\mathbf{u}_i^t \leftarrow \mathrm{PairBiasAtt}^*(\mathbf{u}^t, \mathbf{r}_{i:}, t_{\mathrm{enc}}), \quad \mathbf{r}_{ij} \leftarrow \mathbf{r}_{ij} + \mathrm{OutSum}(\mathbf{u}_i^t, \mathbf{u}_j^t), \quad \mathbf{r} \leftarrow \mathbf{r} + \mathrm{TriangMult}(\mathbf{r}), \tag{5}$$

where t_{enc} denotes the sinusoidal encoding of t (Vaswani et al., 2017), and $\mathrm{PairBiasAtt}^*$ is a variant of the pair-biased attention layer (Jumper et al., 2021) that uses t_{enc} to compute attention weights. The final score is obtained through a linear layer $\mathrm{score}_i = \mathrm{Linear}(\mathbf{u}_i^t)$.

4 Empirical Evaluation

We briefly introduce the datasets and metrics used for each task, then present our empirical results. Full details, and an explanation of data collection can be found in appendices E and F. Extended results are provided in appendix G. In all tables, we use bold to denote the best performing method, and underline the second-best.

Protein-protein docking. Our dataset contains all available chains in the Protein Data Bank (PDB, March 2023, 199k proteins). Splits are generated by performing FoldSeek all-vs-all structural alignments of protein binding sites (Berman et al., 2003; van Kempen et al., 2023). This novel split is introduced to address significant potential data leakage found in the DIPS splits. Full details and evidence of leakage are provided in appendix E. Our test set is selected from cluster representatives among the top 10% highest resolution, which contain at least one high quality representative PPI. We randomly chose a subset of 150 dimers, 100 heterodimers and 50 homodimers. We remark that all methods in the comparison are retrained on the same splits so as to limit potential bias. Retraining details are discussed in appendix F.

We evaluate protein-protein docking methods by measuring differences between predicted and ground truth structures in terms of root mean square deviation (RMSD), RMSD for interface residues (I-RMSD), and RMSD for ligand residues (L-RMSD). We report 25th and 50th percentiles, and the proportion of predictions with I-RMSD $\leq$3Å and L-RMSD $\leq$6Å. We also report DockQ, which is a composite score of I-RMSD, L-RMSD and fraction of recovered native contacts (Basu & Wallner, 2016). The score (range 0 to 1) can be used to reproduce the Critical Assessment of PRediction of Interactions (CAPRI) classification of Incorrect (DockQ < 0.23), Acceptable ($0.23 <$ DockQ < 0.49), Medium ($0.49 \leq$ DockQ < 0.8) and High (DockQ ≥ 0.8) quality predictions (Vajda et al., 2002).

4.1 Autoencoder Evaluation

We begin our empirical evaluation by studying the autoencoder's accuracy, as it may limit LATENTDOCK's performance on downstream tasks. We do this by measuring its capacity to reconstruct input complexes using the metrics described above. Samples from LATENTDOCK's encoder are decoded with an average full-atom Complex RMSD of 1.3 ± 0.3Å, average I-RMSD is 1.3 ± 0.3Å, and L-RMSD is 2.1 ± 0.5Å. With these RMSD statistics, the average DockQ score of predicted complexes is 0.75 ± 0.05Å, which is at the upper threshold of medium quality. Although recovered structures are of relatively high quality, in section 4.2, we show that LATENTDOCK is capable of generating structures at the lower bound of the autoencoder's recovery range. This suggests that improvements to the autoencoder could directly translate to performance gains on design tasks.

4.2 Protein-Protein docking

We compare LATENTDOCK to three machine learning approaches for protein-protein docking, the regression-based methods EQUIDOCK (Ganea et al., 2021) and DOCKGPT (McPartlon & Xu, 2023), and the diffusion-based method DIFFDOCK-PP (Ketata et al., 2023). We re-trained each of these methods on the dataset described above. Training details for baselines are provided in appendix F.

Table 1 reports the results achieved by all methods on the 150 dimers in the test set. When reporting results for generative methods, LATENTDOCK and DIFFDOCK-PP, we sample 20 structures per target and report "oracle" statistics, by selecting the prediction with the lowest RMSD from the ground truth. Although this biases performance in favor of diffusion models, it provides clear and simple

	I-RMSD(Å)↓			L-RMSD(Å)↓			DockQ↑		
	25th	50th	%≤ 3Å↑	25th	50th	%≤ 6Å↑	≥accep.	≥med.	≥high
EQUIDOCK	14.5	18.14	0.0%	29.3	35.0	0.0%	0.0%	0.0%	0.0%
DOCKGPT	**0.76**	2.13	55.3%	**1.86**	5.96	50.1%	63.3%	52.0%	**31.3%**
DIFFDOCK-PP (20)†	2.63	5.01	31.3%	6.1	13.2	24.6%	44.6%	24.0%	4.6%
LATENTDOCK (20)†	1.45	**1.92**	**64.7%**	2.61	**3.82**	**60.7%**	**70.0%**	**56.7%**	1.3%
LATENTDOCK + 1C (20) †	1.31	1.50	94.0%	2.11	2.73	92.0%	97.3%	90.7%	3.3%

Table 1: **Results on 150 protein dimers**. Results for four ML-based docking methods are shown for the test set. Here, we use 25th and 50th to denote 25th and 50th percentile values. Each method was re-trained and evaluated on the same splits. For diffusion models, the number of sampled poses is shown in parentheses. In an effort to fairly compare our method with DIFFDOCK-PP, we report only oracle statistics, denoted with †, which refers to the setting where we can perfectly select the best pose out of the sampled ones. We distinguish our performance on blind docking (LATENTDOCK) and our performance on site-conditioned docking given one C_α-C_α contact (LATENTDOCK + 1C)

criteria that is easy to apply across both methods. (Regression based methods are deterministic, producing a single structure per target.) As an ablation study, we also measure performance for a varying number of sampled structures (5, 10, 20), with results shown in table G.1.

Table 1 shows that LATENTDOCK achieves competitive lower-quartile I-RMSD and L-RMSD with DOCKGPT, and significantly outperforms DIFFDOCK-PP and EQUIDOCK on all metrics. In terms of DockQ score, LATENTDOCK finds the largest fraction of medium or better quality poses, but falls short of DOCKGPT in terms of high-quality predictions. This is not surprising given that the autoencoder has an average DockQ of 0.75 – marginally below the high quality threshold. We expect improvements to the autoencoder to translate to improvements in LATENTDOCK's performance.

We also evaluate LATENTDOCK's and DIFFDOCK-PP's performance when generating a different number of samples per target. Results are shown in table G.1. We observe that LATENTDOCK tends to converge on low-RMSD solutions with significantly less samples than DIFFDOCK-PP. In fact, LATENTDOCK with 5 samples per target significantly outperforms DIFFDOCK-PP with 20 samples *across all metrics*. Considering the best pose across five samples, LATENTDOCK achieves median oracle I-RMSD of 2.37 Å and median oracle L-RMSD of 5.53 Å. Given the same number of samples, DIFFDOCK-PP's median I-RMSD and L-RMSD is 8.67 Å and L-RMSD is 19.78 Å.

We also assess LATENTDOCK's ability to incorporate binding site information in the form of pairwise C_α contacts (included as the contact information inter-chain-pair feature). In line with the results in McPartlon & Xu (2023), we observe that providing even a single inter-chain contact significantly improves docking performance (fig. G.1). In fact, with a single contact, LATENTDOCK achieves an oracle (out of 20 samples), 90% of LATENTDOCK's predictions achieve a medium or high DockQ score. We remark that the 25-th percentile I-RMSD is roughly equal to the error rate of the autoencoder, showing again that LATENTDOCK's is performing at the limit imposed by the autoencoder, and that improvements made to the autoencoder could directly translate to improvements in LATENTDOCK's performance.

5 Conclusion

We present LATENTDOCK, a latent diffusion model for general protein-protein docking. Unlike existing diffusion-based methods, LATENTDOCK offers a methodology for incorporating full-atom conformational flexibility, and for simultaneously docking more than two chains. LATENTDOCK also offers a straight-forward way to sample multiple conformations, outperforming regression-based methods like DOCKGPT in terms of DockQ hit-rate.

References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Sankar Basu and Björn Wallner. Dockq: a quality measure for protein-protein docking models. *PloS one*, 11(8):e0161879, 2016.

Helen Berman, Kim Henrick, and Haruki Nakamura. Announcing the worldwide protein data bank. *Nature Structural & Molecular Biology*, 10(12):980–980, December 2003.

Rong Chen, Li Li, and Zhiping Weng. Zdock: an initial-stage protein-docking algorithm. *Proteins: Structure, Function, and Bioinformatics*, 52(1):80–87, 2003.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

Sjoerd J De Vries, Marc Van Dijk, and Alexandre MJJ Bonvin. The haddock web server for data-driven biomolecular docking. *Nature protocols*, 5(5):883–897, 2010.

Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with alphafold-multimer. *biorxiv*, pp. 2021–10, 2021.

Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. *arXiv preprint arXiv:2111.07786*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Mohamed Amine Ketata, Cedrik Laue, Ruslan Mammadov, Hannes Stärk, Menghua Wu, Gabriele Corso, Céline Marquet, Regina Barzilay, and Tommi S Jaakkola. Diffdock-pp: Rigid protein-protein docking with diffusion models. *arXiv preprint arXiv:2304.03889*, 2023.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Matt McPartlon, Ben Lai, and Jinbo Xu. A deep se (3)-equivariant model for learning inverse protein folding. *bioRxiv*, pp. 2022–04, 2022.

Matthew McPartlon and Jinbo Xu. Deep learning for flexible and site-specific protein docking and design. *bioRxiv*, pp. 2023–04, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 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.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Raphael Townshend, Rishi Bedi, Patricia Suriana, and Ron Dror. End-to-end learning on 3d protein structure for interface prediction. *Advances in Neural Information Processing Systems*, 32, 2019.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

S Vajda, IA Vakser, MJ Sternberg, and J Janin. Capri: Critical assessment of prediction of interactions. *PROTEINS: Structure, Function, and Genetics*, 47(4):444–446, 2002.

Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, pp. 1–4, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Thom Vreven, Iain H Moal, Anna Vangone, Brian G Pierce, Panagiotis L Kastritis, Mieczyslaw Torchala, Raphael Chaleil, Brian Jiménez-García, Paul A Bates, Juan Fernandez-Recio, et al. Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. *Journal of molecular biology*, 427(19):3031–3041, 2015.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Yumeng Yan, Huanyu Tao, Jiahua He, and Sheng-You Huang. The hdock server for integrated protein–protein docking. *Nature protocols*, 15(5):1829–1852, 2020.

Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, 2020.

A Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) represent a powerful generative modeling technique. Given a target distribution $p_{\text{data}}(z)$, they define a forward process that gradually transforms this distribution into a tractable reference. For instance, the variance preserving formulation from Song et al. (2020) defines this process using an SDE,

$$\mathrm{d}z_t = -\tfrac{1}{2}\beta(t)\, z_t \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}w, \quad \text{where} \quad t \in [0,1] \quad \text{and} \quad z_0 \sim p_{\text{data}}(z). \tag{6}$$

Essentially, this process takes a dataset of samples from $p_{\text{data}}(z)$ and progressively transforms them into random noise. Critically, it can be simulated exactly for any time t: Given $z_0 \sim p_{\text{data}}$ we have

$$z_t \sim p_t(z_t \,|\, z_0) = \mathcal{N}\left(z_t \;\middle|\; z_0\, e^{-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s}, I - I\, e^{-\int_0^t \beta(s)\mathrm{d}s}\right). \tag{7}$$

For an appropriate choice for $\beta(t)$ (Song et al., 2020), this shows that samples z_1 (obtained by running eq. (6) up to time $t = 1$) approximately satisfy $z_1 \sim \mathcal{N}(0, I)$. Therefore, new samples from p_{data} can be obtained by simulating the time-reversal (Anderson, 1982) of eq. (6), given by

$$\mathrm{d}z_t = -\tfrac{\beta(t)}{2}\Big(z_t + 2\,\nabla \log p_t(z_t)\Big)\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\bar{w}, \quad z_1 \sim \mathcal{N}(0, I), \tag{8}$$

from $t = 1$ to $t = 0$. Unfortunately, the "score" $\nabla \log p_t(z_t)$ is often intractable. Diffusion models address this training a *score network* $s_\theta(z_t, t)$ to approximate it, minimizing the denoising score matching objective (Hyvärinen & Dayan, 2005; Vincent, 2011)

$$\mathcal{L}(\theta) = \mathbb{E}_{t, z_0, z_t|z_0}\left[w(t)\, \|s_\theta(z_t, t) - \nabla_{z_t} \log p_t(z_t \,|\, z_0)\|^2\right]. \tag{9}$$

Finally, new samples from $p_{\text{data}}(z)$ can be obtained (approximately) by simulating the reverse process from eq. (8) using $s_{\theta^*}(z_t, t) \approx \nabla \log p_t(z_t)$.

Conditional diffusion models are a natural extension of the formulation above, in which a diffusion model is trained to approximate conditional distributions $p_{\text{data}}(z\,|c)$, where c is the conditioning variable. In this case, the dataset consists on tuples (z, c), the score network is given by $s_\theta(z_t, t, c)$, and the reverse process produces samples from $p_{\text{data}}(z \,|\, c)$ (for any given c).

B Roto-translational Invariant Features

Residue-level features (E_{aa}(i), $E_{\text{pos}}(i)$, and $E_{\text{angle}}(\theta_i)$) include amino acid type, sequence position, and backbone angles, respectively.

$E_{\text{aa}}(i)$ encodes the type of residue i (as a one-hot encoding of the 20 natural amino acids in the autoencoder, or as the residue ESM embedding (Lin et al., 2023) in the diffusion). $E_{\text{pos}}(i)$ encodes the ith residue relative sequence position as a one-hot vector using ten equal-width bins. $E_{\text{angle}}(\theta_i)$ encodes the backbones torsional angles $\theta_i \in \{\phi_i, \psi_i\}$ as a one-hot encoding by splitting $\theta \in [-180°, 180°]$ into 18 equal-width bins.

Intra-chain pair features ($E_{\text{dist}}(i,j)$, $E_{\text{angle}}(\theta_{ij})$, and $E_{\text{sep}}(i,j)$) include distance, relative orientation, and sequence separation, respectively.

$E_{\text{dist}}(i,j)$ bins the distance between the i-th residue C_α atom and the j-th residue backbone atom $a \in \{N, C_\alpha, C, C_\beta\}$ into six equal-width groups between 2Å and 16Å. $E_{\text{angle}}(\theta_{ij})$ encodes the angles $\theta_{ij} \in \{\phi_{ij}, \psi_{ij}, \omega_{ij}\}$ of pairwise residue orientations (Yang et al., 2020). $E_{\text{sep}}(i,j)$ produces a one-hot encoding of relative sequence separation between residues i and j into 32 classes (McPartlon et al., 2022). The pairwise features for each chain are stacked to form a block-diagonal input matrix with an additional learned parameter filling the missing off-diagonal entries.

Inter-chain pair features

$E_{\text{contact}}(i,j)$ is a binary flag indicating whether the distance between the C_α atoms of residues i and j is less than 10Å. $E_{\text{chain}}(i,j)$ is a three-class one-hot encoding indicating whether the index of the chain containing residue i is greater than, equal, or less than the index of the chain containing residue j. (The distance and angle features are generated as explained above for the *intra-chain-pair* features.)

C LATENTDOCK analysis

After training the autoencoder, LATENTDOCK's performs protein-protein docking by (i) extracting roto-translational invariant features from the input chains; (ii) running the latent diffusion conditioned on these features; (iii) feeding the resulting latent sample z through the decoder. This process defines a distribution over protein complex structures. This section briefly studies this distribution's properties.

Proposition 1. *Let x denote a protein complex structure, and $p(x \mid c)$ denote the distribution defined by* LATENTDOCK*, where c denote the conditioning features for all individual proteins in the complex (sequence and individual structures). If the decoder initializes backbone frames with a global rotation chosen uniformly at random, then $p(x \mid c_1, ..., c_N) = p(R\,x \mid c_1, ..., c_N)$ for any three-dimensional rotation R.*

We prove proposition 1 in appendix C.1. The proposition states that the distribution defined by LATENTDOCK over the complex structure, denoted by $p(x \mid c)$, is invariant w.r.t. rotations. This is a property of the true data distribution and it has been observed that methods that enforce symmetries of the true data distribution often yield better generalization (Jumper et al., 2021; Xu et al., 2022).)

Another desirable property satisfied by LATENTDOCK is its invariance w.r.t. rigid transformations of the individual chains. This is desirable, as the distribution over full complex structure should not be affected by rigid transformations of the chains provided as input.

C.1 Proof of proposition 1

Proof. Without loss of generality, we assume that both the input and output structures have mean $\mathbf{0}$. This follows from the fact that IPA is translation equivariant, and subtracting the structure's center of mass results in an equivalent update to the output.

The proposition is a consequence of the architecture used for LATENTDOCK's decoder. The updates from the invariant point attention layer (IPA) are invariant to global rigid transformations of the frames, while the backbone update is equivariant to such transformations. As a result, for a fixed latent representation $\mathbf{z}$, initializing all frames with the same random rotation and running the decoder is equivalent to initializing the frames with the identity rotation and applying the random rotation on the decoder's output. Since this rotation is chosen uniformly at random, we have $p(x \mid \mathbf{z}) = p(R\,x \mid \mathbf{z})$ for any R. This is the key property in the derivation below.

Letting z denote the sample produced by the latent diffusion, and $p_{\text{diff}}(\mathbf{z} \mid c_1, ..., c_N)$ its distribution, we have

$$
\begin{aligned}
p(R\,x \mid c_1, ..., c_N) &= \int p(R\,x\mathbf{z} \mid c_1, ..., c_N)\mathrm{d}\mathbf{z} && (10)\\
&= \int p(R\,x \mid \mathbf{z}, c_1, ..., c_N)\, p_{\text{diff}}(\mathbf{z} \mid c_1, ..., c_N)\mathrm{d}\mathbf{z} && (11)\\
&= \int p(R\,x \mid \mathbf{z})\, p_{\text{diff}}(\mathbf{z} \mid c_1, ..., c_N)\mathrm{d}\mathbf{z} && (12)\\
&= \int p(x \mid \mathbf{z})\, p_{\text{diff}}(\mathbf{z} \mid c_1, ..., c_N)\mathrm{d}\mathbf{z} && (13)\\
&= p(x \mid c_1, ..., c_N), && (14)
\end{aligned}
$$

where eq. (12) uses the fact that, given $\mathbf{z}$, (x, s) is independent of $c_1, ..., c_N$ (i.e. $c_1, ..., c_N$ is only used to generate $\mathbf{z}$ by running the reverse diffusion; given $\mathbf{z}$, the decoder does not use $c_1, ..., c_N$ in any way.) □

D Autoencoder training details

The loss used to train the autoencoder is given by

$$
\begin{aligned}
\mathcal{L}_{\text{ae}}(\phi, \psi) = \text{CrossEntropy}(\hat{S}, S) + \text{FAPE}(\hat{X}, X_{\text{true}}) + \\
10^{-3} \cdot \text{KL}(\mathcal{N}(\mu, \sigma^2) \,\|\, \mathcal{N}(0, I)) + \text{plDDT}(\hat{X}^{C\alpha}, X^{C\alpha}_{\text{true}}), \quad (15)
\end{aligned}
$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the distribution in the latent space produced by the encoder, $\hat{X}^a$ is the full-atom three-dimensional structure reconstructed by the decoder and indexed by atom type a, and $\hat{S}$ is the reconstructed sequence. The final term plDDT is taken from (Jumper et al. (2021), Supplemental Algorithm 29). The FAPE loss (Jumper et al., 2021) measures the quality of the produced structure by aligning predicted per-residue predicted and ground truth rigid frames. To account for limited or unknown knowledge of binding interfaces in a protein complex, we mask the contact features $E_{\mathrm{contact}}(i, j)$ when producing the pair representation used as input for the encoder with probability $1/2$. Therefore, half of the samples encountered during training do not contain inter-chain contact information. When these features are not masked, we subsample the number of contacts included as $N_{\text{contact}} \sim \text{Geometric}(1/3)$.

E Dataset

Most recent protein-protein docking methods have been evaluated on the Docking Benchmark 5 (DB5) (Vreven et al., 2015), and trained with complexes from the from Protein Data Bank (PDB) (Berman et al., 2003), such that no protein had more than 30% sequence homology to any protein in the DB5 as proposed by DIPS Townshend et al. (2019). This approach has some limitations. For instance, the size of DB5 is rather small when compared to DIPS, which means the structural diversity of the test set may not be representative, and thus sequence similarity is not always a good proxy to differentiate structurally similar proteins. Even more concerning, however, when performing interface clustering between train, validation and test set of the frequently used DIPS splits Ganea et al. (2021); Ketata et al. (2023), we found that a large majority of the test dataset had structural overlap with the training data set, as evidenced by fig. E.1.

Therefore, to avoid overreporting performance, we train LATENTDOCK with splits generated by a structural interface clustering using FoldSeek all-vs-all structural alignments on all available chains in the PDB (March 2023, 199k proteins), focusing on respective protein binding sites (Berman et al., 2003; van Kempen et al., 2023), and retrain all existing methods on these splits for fair comparison. Foldseek stores local alignment positions and normalizes the alignment scores as TM-score, which is used to filter out alignments with lower structural similarity (< 0.60 TM-score). Binding site residues were identified based on a criterion of an 6Å C_α distance threshold between chains. A pair of chains was classified as interacting if there were a minimum of 6 binding residues, and at least 50% were encompassed by the Foldseek alignment. Subsequently, a graph representation encoding interface similarity of the interacting chain pairs, where TM-scores served as the weights for the edges, was used to perform community clustering to delineate interface clusters.

The test set consists of the cluster representative with the highest resolution for 10% of the clusters, which contained at least one high quality representative protein-protein interaction (1973 proteins). All representative PPIs in the test set have a minimum resolution of 4.5A, a minimum of 5 atom types, a dimeric state, an interface without any missing residues (gaps), either chain with a maximum of 550 residues and minimum of 25 residues in length, and are solved by X-ray crystallography. The validation set consists of 190 proteins with the same restrictions, except that they may contain gaps. The training data consists of the remaining clusters without any quality-based filtering.

F Data Collection

We retrained each method using the same splits, describe in appendix E. All methods were retrained using the exact parameters described in the corresponding manuscripts. Additional instructions for training and inference were gathered through correspondence with the authors of DOCKGPT and DIFFDOCK-PP. The implementation of DOCKGPT was modified slightly to use 1Å width bins for pairwise distance features (original paper used $2\mathring{A}$ bin-width). This was done to improve performance on rigid docking.

G Extended Results

We show some additional results for protein docking in table G.1 and fig. G.1.

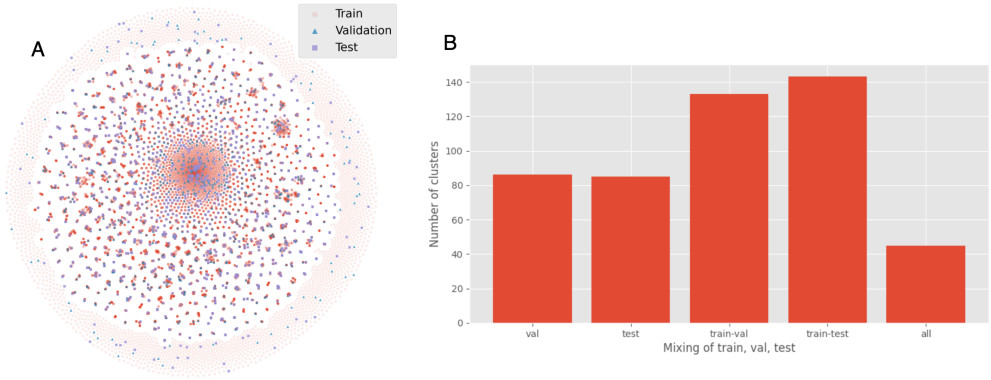

Figure E.1: **Leakage between training, validation and test splits in the DIPS benchmark set (Townshend et al., 2019)**. All-vs-all pairwise structural alignments of respective binding sites performed with Foldseek (van Kempen et al., 2023). (A) TSNE plot of pairwise TM-alignment scores for all chains in DIPS, showing mixed clusters of train (red), validation (blue), and test (purple). (B) Bar plot showing the number of Foldseek clusters against members of DIPS (bars from left to right: only validation, only test, both training and validation, both training and test, or all).

	I-RMSD↓			L-RMSD↓		
	25	50	%≤ 3Å↑	25	50	%≤ 6Å↑
Diffdock-PP (5)†	4.88	8.67	15.3%	11.57	19.78	12.6%
Diffdock-PP (10)†	3.87	6.58	20.0%	9.17	16.49	15.3%
DIFFDOCK-PP (20)†	2.63	5.01	31.3%	6.16	13.20	24.6%
LATENTDOCK (5)†	1.65	2.37	53.3%	3.41	5.53	50.7%
LATENTDOCK (10)†	1.54	2.05	58.0%	2.72	4.36	54.7%
LATENTDOCK (20)†	**1.45**	**1.92**	**64.7%**	**2.61**	**3.82**	**60.7%**

Table G.1: **Results for DIFFDOCK-PP and LATENTDOCK with varying number of samples** For diffusion models, the number of sampled poses is shown in parentheses. In an effort to fairly compare our method with Diffdock-PP, we report only oracle statistics, denoted with †, which refers to the setting where we can perfectly select the best pose out of the sampled ones.

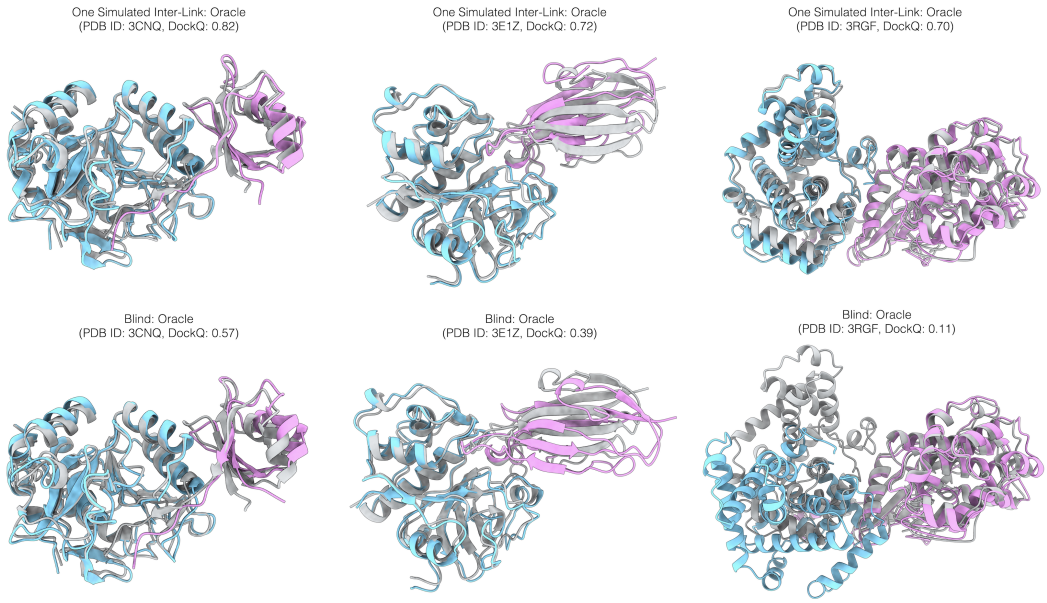

Figure G.1: **Protein-Protein Docking with LATENTDOCK for three complexes** with PDB identifier: 3CNQ, 3E1Z, 3RGF from left to right. Top row shows LATENTDOCK Oracle docking predictions (40 sampled poses) with contact information (one simulated inter-link). Bottom row shows LATENTDOCK Oracle docking predictions (40 sampled poses) without additional contact information (blind). The respective ground truth structures are displayed in gray.