Design of peptides with non-canonical amino acids using flow matching

Jin Sub Lee
University of Toronto
jinsub.lee@mail.utoronto.ca

Philip M. Kim
University of Toronto
pi@kimlab.org

Abstract

The canonical vocabulary of twenty amino acids limits the chemical space available to proteins and peptides. Expanding this vocabulary to hundreds of non-canonical amino acids allows the engineering of proteins with novel function and activity, and is of great interest for the discovery of novel drugs such as macrocyclic peptides. Here we present NCFlow, a flow-based generative model capable of incorporating any arbitrary non-canonical amino acid into a given protein. To supplement sparse training data in the Protein Data Bank, NCFlow is pretrained on millions of small molecule structures and a large set of protein-ligand complexes before finetuning on native non-canonicals found within proteins in the Protein Data Bank. We show that NCFlow outperforms AlphaFold3-based methods in the structure prediction of unseen non-canonical amino acids. We present a peptide design pipeline akin to *in silico* deep mutational scanning, and propose a novel scoring strategy using a combination of deep learning-based and molecular dynamics-based alchemical binding free energy calculations to identify improved peptide variants. We apply the method on four protein-peptide complex test cases, and observe that incorporating non-canonicals can significantly improve binding affinity by up to -7.0 kcal/mol. Thus, NCFlow can be easily integrated into existing protein design platforms to further improve its properties outside of what is capable with standard amino acids.

1 Introduction

Proteins typically consist of a polypeptide chain of amino acids, where a standard vocabulary of 20 canonical amino acids build the immense diversity of protein functions necessary for life. However, nature also encodes hundreds of non-proteinogenic non-canonical amino acids (ncAAs) in nature as metabolic intermediates, alterations in the translation pathway, or post-translational modifications of amino acids that allow even greater flexibility in the function and biochemical properties of proteins [1]. Thus, the design of proteins with ncAAs holds significant potential as they can introduce novel functions and capabilities, such as in biocatalysis to develop artificial enzymes, or enzymes with novel reactions not observed in nature [2, 3]. A key area of interest is the design of improved peptide variants with nonstandard backbone or side-chain chemistries. Current approaches to designing such peptides are largely limited to experimental screening via genetic code reprogramming, where modified tRNAs with diversified amino acid payloads are used in conjunction with *in vitro* display technologies such as mRNA display [4]. With the rise in popularity of peptide drugs, and more specifically, macrocyclic peptides containing amino acid modifications, an effective computational approach to nonstandard peptide design would greatly benefit therapeutic drug discovery campaigns.

Protein design with deep learning has been revolutionized in recent years with the development of expressive architectures and rise in data and computing power. AlphaFold2 [5] and its successor AlphaFold3 [6] have facilitated the protein structure prediction boom with near experimental accuracy across many targets and enabled various applications in structure-based protein design [7]. Moreover, the development of diffusion-based generative models [8, 9, 10] and their applications to protein

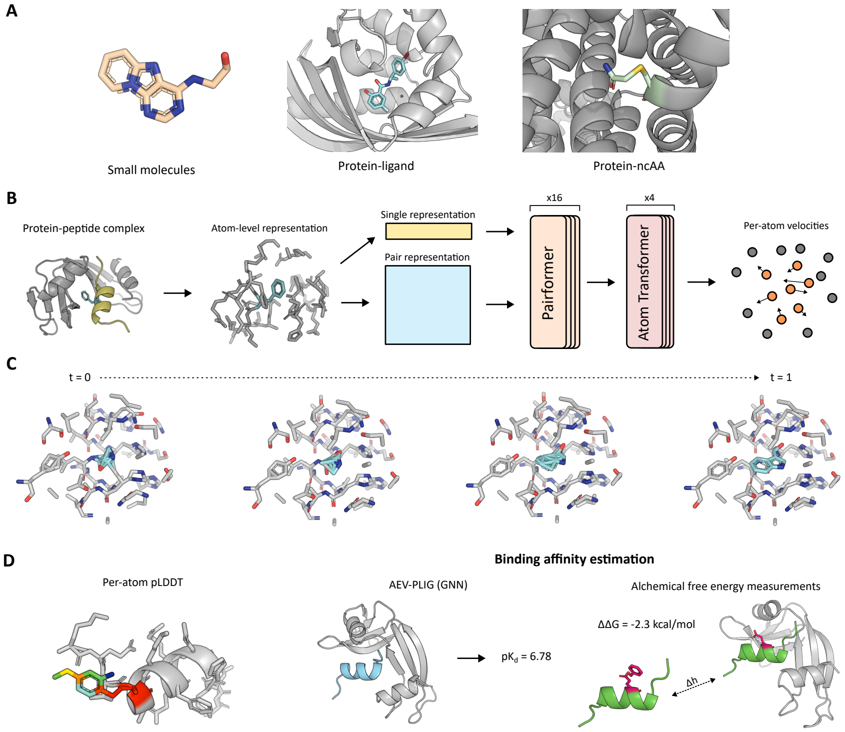

Figure 1: Overview of NCFlow. A) NCFlow is trained from three datasets: 1. PubChem3D, containing low-energy conformers of small molecules, 2. Plinder, a curated dataset of protein-small molecule complexes, and 3. a dataset of ncAA-protein environments found in the Protein Data Bank. B) NCFlow simply takes in a protein-peptide complex and the chemical description of a ncAA (atom types and bond connectivity), and predicts the 3D conformation of the ncAA in the given residue and pocket. C) An example sampling trajectory of a ncAA using NCFlow from $t = 0$ to $t = 1$. D) To enable design, we rely on three filters: per-atom pLDDT predicted by a confidence module, a deep learning-based protein-ligand binding affinity predictor named AEV-PLIG [21], and MD-based relative binding free energy calculations via an alchemical coordinate transformation Δh [22].

design [11, 12, 13, 14] allows rapid and effective design at unprecedented scale that rivals experimental screening approaches at a fraction of time and cost. However, the adaptation of existing computational protein design tools for nonstandard peptides is non-trivial - AlphaFold2 and RosettaFold-based methods such as RFDiffusion and BindCraft are unable to model ncAAs. Moreover, applying AlphaFold3-based tools such as Boltz [15, 16] and Chai1 [17] face another issue since modeling non-standard residues require conditional information (ex. number of atoms per token, RDKit conformer positions) that are specific to each ncAA. Thus, a standard design framework using gradient descent is difficult to apply when the modified residue is not known *a priori*. Another limitation in modeling ncAAs is the scarcity and bias of ncAA data in the Protein Data Bank, as less than 0.02% of deposited residues correspond to ncAAs. Moreover, a large fraction of these are non-functional, where modified residues such as selenomethionines and selenocysteines are routinely used to aid solving the phase problem in X-ray crystallography. Thus, the fraction of ncAAs that participate in native interactions is severely limited. Finally, scoring ncAA variants is another issue, since most protein-protein and protein-peptide scoring methods [18, 19, 20] only support canonical amino acids, so additional methods to identify high-fitness ncAA variants must be explored.

In this work, we present NCFlow, a model that learns to place any arbitrary ncAA into a given protein backbone. Due to the scarcity of ncAA data, we formulate the task as a single-residue structure prediction problem for which the PDB contains on the order of 10^4 data points, allowing reasonable training of neural networks. To augment the data, we also employ pretraining on small molecule structures (PubChem [23]) and protein-ligand complexes (Plinder [24]), increasing the available training data by over three orders of magnitude to improve model performance and generalization to

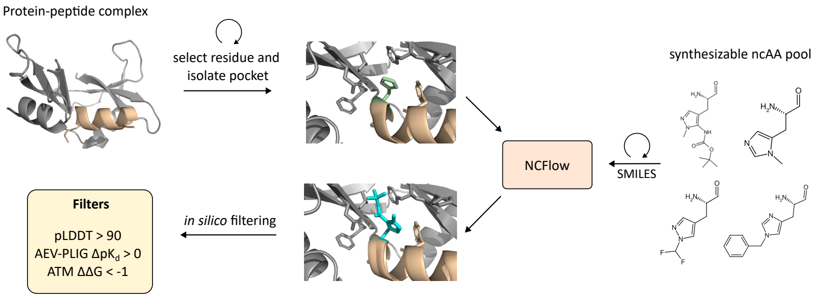

Figure 2: Design workflow of peptide-ncAA variants using NCFlow.

ncAAs not found in the PDB. To apply NCFlow in a design setting, we propose an approach similar to deep mutational scanning, where each residue of a given peptide is mutated to a ncAA variant and assessed for improved activity or fitness. We identify improved variants with a combination of uncertainty estimates (pLDDT), deep learning-based affinity prediction (AEV-PLIG [21]), and alchemical relative binding free energy calculations (Alchemical Transfer Method [22]) to identify promising candidates that improve binding affinity to the protein targets. We validate this scoring approach on a comprehensive dataset of experimentally-derived protein-peptide mutational scanning datasets, and observe that across many systems, the proposed combination of deep learning and alchemical methods can effectively isolate improved peptide variants with high precision. We apply the pipeline on four unique test cases including head-to-tail and disulfide-cyclized cyclic peptides, and show that we can obtain peptide variants with ncAAs displaying improved predicted binding affinity across most cases.

2 Results

First, we evaluated the respective pretrained models on PubChem3D and Plinder on their ability to recover the ground truth small molecule conformations, using Kabsch-aligned ground truth and sample structures for PubChem3D, and symmetry-corrected RMSD [25] for Plinder (Figure S1A). We observe that the model generates small molecule conformers with high fidelity and exhibit various reasonable interactions with its target protein (Figure S2B). We then tested whether the PubChem3D and Plinder pretraining tasks improves performance on prediction of single-residue ncAA structures given its protein pockets (Figure S1C). We observe that no pretraining attains strong performance with a mean RMSD of 1.58Å across all test set ncAAs, with lower RMSDs observed for core residues (1.08Å) than surface ones (1.79Å) due to increased contextual information and therefore rigidity of the ncAA in the protein pocket. Pretraining on PubChem and Plinder further decreases RMSD across the test set by 0.21, 0.14, and 0.18Å in all, core, and surface residues, respectively, suggesting the effectiveness of pretraining on small molecules for ncAA structure prediction. We then analyzed the intrinsic uncertainty estimates (pLDDT) predicted by the confidence model for ranking predictions (Figure S1D), and observed a strong negative relationship between pLDDT and RMSD. This suggests that high-confidence samples ranked by pLDDT can be used to filter out poor samples generated by NCFlow.

We also report comparisons to Boltz [15], an open-source implementation of AlphaFold3 that is capable of predicting protein structures with ncAAs (Figure S1B). Direct performance comparison is not possible since AlphaFold3-like models are trained for the much more difficult task of general biomolecular structure prediction and the non-ncAA protein residues are co-folded with the ncAAs, while NCFlow considers only the pocket and neighboring side-chains and assumes the protein backbone to be fixed. Thus, poor predictions of the overall protein scaffold by Boltz may lead to large deviations in RMSD. We ameliorate this issue by 1. superimposing the predicted ncAA backbone atoms (N, CA, and C) by Boltz with the ground-truth ncAA backbone atoms prior to RMSD calculation, and 2. running NCFlow in single-chain mode by isolating the chain containing the ncAA. We observe that Boltz1 and Boltz2 performs significantly worse than NCFlow for single ncAA structure prediction with mean RMSDs of 3.29Å, 3.24Å, and 1.43Å respectively, suggesting that NCFlow predicts more accurate ncAA conformations over AlphaFold3-based methods. We report a few ground-truth and predicted ncAA structures from the test set in (Figure S2A), and observe

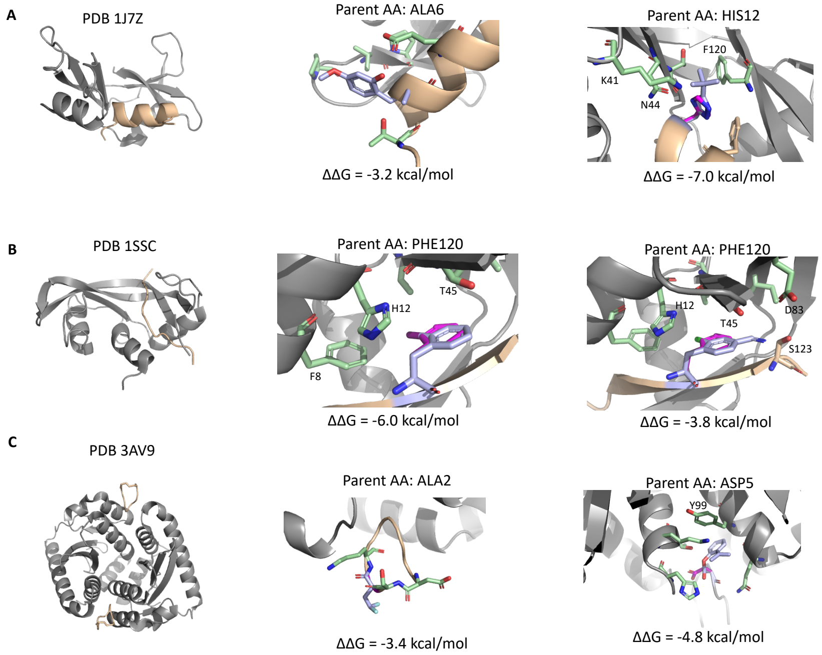

Figure 3: Selected ncAA variants with improved binding affinity for each test case. The ground truth pocket is colored in gray with selected interacting pocket residues in green, the wild type residue to be replaced in magenta, and the ncAA variant in blue.

that the predicted ncAA structures are structurally valid and align closely with the ground-truth conformations.

To apply NCFlow for engineering peptides with ncAAs, we use *in silico* deep mutational scanning with virtual screening via computational binding affinity predictions to identify peptide variants with improved predicted binding affinity (Figure 2). We start with a bound protein-peptide complex, and isolate the pocket atoms based on the CA atom of the peptide residue of interest. Then, we extract a pool of ncAAs based on the wild-type parent amino acid in the form of SMILES strings, which can be used to extract atom types and bond connectivity information. NCFlow takes in the pocket structural data and ncAA atom type and bond information to replace the parent canonical amino acid to the ncAA variant, resulting in a protein-peptide complex containing the selected ncAA. This process is iterated over all residue positions and the entire ncAA pool to obtain hundreds of protein-peptide complex variants, each containing a unique peptide-ncAA variant bound to the target protein. These variants are filtered to exclude low-confidence predictions by a pLDDT filter of > 90, and AEV-PLIG [21] is used to predict binding affinity of the wild type peptide and the ncAA-containing variant, which has been shown to exhibit state-of-the-art performance and outperforms ipTM confidence scores of Boltz-1x across most metrics [26]. We exclude all variants with a AEV-PLIG predicted $\Delta pK_d < 0$ based on the wild-type predicted pK_d, and select the top 50 variants by highest ΔpK_d for further evaluation. We further validate these variants by measuring the relative binding free energy using the Alchemical Transfer Method [22], an alchemical free energy method that estimates binding affinity by alchemical coordinate transformations between two ligands. Thus, we obtain tens of variants per protein-peptide complex that are of high-confidence by pLDDT, and are predicted to increase binding affinity by two orthogonal methods - a deep learning-based model (AEV-PLIG) and a more rigorous MD-based alchemical relative binding free energy measurement (ATM). We perform a more rigorous evaluation of this scoring method in *Methods*.

We ran the design pipeline on a total of four test cases, ranging from standard linear helical peptides to head-to-tail or disulfide-bridged cyclic peptides (Figure S3A). We plot the pLDDT with AEV-PLIG ΔpK_d for all variants to analyze the variant distribution for each test case (Figure S3B). Filtering for high-confidence samples removes 22.4% of the samples across all test cases, suggesting that most generated samples are predicted with low RMSD. Moreover, 47.1% of the variants exhibit a ΔpK_d > 0, indicating that more than half of the ncAA variants are predicted to decrease binding affinity.

We measured the Spearman correlation between pLDDT and ΔpK_d and observe large variance ranging from -0.40 (3AV9) to 0.39 (1SFI), suggesting that internal confidence metrics cannot be used for binding affinity estimation. For some test cases (PDB 1J7Z and 1SFI), we observe that most of the variants have a predicted $\Delta pK_d < 0$ with 25.8% and 0.64%, respectively, suggesting that most ncAA variants do not improve binding affinity relative to the wild-type peptide. Interestingly, the head-to-tail and disulfide-cyclized peptide in PDB 1SFI contain only 5 variants that minimally improve binding affinity. This can be explained by the tight binding affinity of the native cyclic peptide to trypsin - its cognate receptor - with an experimentally measured binding affinity of 0.1nM (and in AEV-PLIG's PDBBind training set), while the other test cases are measured and/or predicted to exhibit micromolar affinity. Thus, it may be increasingly difficult to identify variants that further improve binding affinity of strong binders, especially with just single mutants.

Finally, we ran the top 50 variants ranked by AEV-PLIG ΔpK_d for each test case with ATM to further computationally validate these peptide-ncAA variants (Figure S3C). We observed that 35.0% of samples exhibit ATM $\Delta\Delta G$ < -1.0 kcal/mol, indicative of variants that are predicted to increase binding affinity by both binding affinity prediction tools. Interestingly, the 3 ncAA variants of PDB 1SFI where ATM successfully ran yielded no variants with ATM $\Delta\Delta G$ < -1.0 kcal/mol, so we removed this complex from further downstream analysis. We visualize some of the top variants per test case in Figure 3. We observe that the variants predicted to increase binding affinity exhibit either increased noncovalent interactions with pocket residues or increased polar interactions with the solvent. For instance, the ALA6 variant of PDB 1J7Z (Figure 3A, middle) contains a 2-hydroxy-4-methoxybenzaldehyde group linked to the peptide nitrogen atom that induces polar interactions with the solvent, suggesting it may stabilize native binding interactions with the target protein. Moreover, the PHE120 variants of PDB 1SSC (Figure 3B) exhibit increased side-chain interactions with neighboring pocket residues in distinct ways: the left variant is linked to an iodine atom that exhibit polar interactions with the neighboring H12 residue, while the right variant contains a chlorine atom in the same position for polar interactions in addition to a C-N group that extends into a pocket consisting of polar oxygen atoms of T45, D83, and S123 residues of the target protein for increased hydrogen-bonding interactions. The ALA2 variant of PDB 3AV9 contains two solvent-exposed polar fluorine atoms for increased solvent interactions, and the ASP5 variant appears to exhibit $\pi - \pi$ interactions with the protein's Y99 residue (Figure 3C). Thus, we show that the design workflow can generate ncAA variants for protein-peptide test cases that are biophysically plausible to increase binding affinity to its target protein.

3 Conclusion

In this work, we present a novel framework to incorporate ncAAs into protein-peptide complexes to improve binding affinity. We developed NCFlow, a flow matching generative model that learns to predict the 3D conformations of any input ncAA within a given protein pocket, and apply the model to generate hundreds of peptide-ncAA variants that are scored with deep learning and alchemical methods to predict binding affinity. We envision many possible avenues of future work such as experimental validation and extension to peptide variants with more than one ncAA.

References

[1] A. Ambrogelly, S. Palioura, and D. Söll, "Natural expansion of the genetic code," *Nature chemical biology*, vol. 3, no. 1, pp. 29–35, 2007.

[2] Z. Birch-Price, F. J. Hardy, T. M. Lister, A. R. Kohn, and A. P. Green, "Noncanonical amino acids in biocatalysis," *Chemical Reviews*, vol. 124, no. 14, pp. 8740–8786, 2024.

[3] B. Brouwer, F. Della-Felice, J. H. Illies, E. Iglesias-Moncayo, G. Roelfes, and I. Drienovská, "Noncanonical amino acids: Bringing new-to-nature functionalities to biocatalysis," *Chemical reviews*, vol. 124, no. 19, pp. 10 877–10 923, 2024.

[4] S. L. Richardson, K. K. Dods, N. A. Abrigo, E. S. Iqbal, and M. C. Hartman, "In vitro genetic code reprogramming and expansion to study protein function and discover macrocyclic peptide ligands," *Current opinion in chemical biology*, vol. 46, pp. 172–179, 2018.

[5] J. Jumper et al., "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[6] J. Abramson et al., "Accurate structure prediction of biomolecular interactions with alphafold 3," *Nature*, pp. 1–3, 2024.

[7] M. Pacesa et al., "Bindcraft: One-shot design of functional protein binders," *bioRxiv*, pp. 2024–09, 2024.

[8] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*, pmlr, 2015, pp. 2256–2265.

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[10] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[11] J. S. Lee, J. Kim, and P. M. Kim, "Score-based generative modeling for de novo protein design," *Nature Computational Science*, vol. 3, no. 5, pp. 382–392, 2023.

[12] J. L. Watson et al., "De novo design of protein structure and function with rfdiffusion," *Nature*, vol. 620, no. 7976, pp. 1089–1100, 2023.

[13] J. B. Ingraham et al., "Illuminating protein space with a programmable generative model," *Nature*, vol. 623, no. 7989, pp. 1070–1078, 2023.

[14] T. Geffner et al., "Proteina: Scaling flow-based protein structure generative models," *arXiv preprint arXiv:2503.00710*, 2025.

[15] J. Wohlwend et al., "Boltz-1: Democratizing biomolecular interaction modeling," *bioRxiv*, pp. 2024–11, 2024.

[16] Y. Cho, M. Pacesa, Z. Zhang, B. E. Correia, and S. Ovchinnikov, "Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design," *bioRxiv*, pp. 2025–04, 2025.

[17] C. D. team et al., "Chai-1: Decoding the molecular interactions of life," *BioRxiv*, pp. 2024–10, 2024.

[18] M. McFee and P. M. Kim, "Gdockscore: A graph-based protein–protein docking scoring function," *Bioinformatics advances*, vol. 3, no. 1, vbad072, 2023.

[19] M. McFee, J. Kim, and P. M. Kim, "Eudockscore: Euclidean graph neural networks for scoring protein–protein interfaces," *Bioinformatics*, vol. 40, no. 11, btae636, 2024.

[20] N. Manshour, J. Z. Ren, F. Esmaili, E. Bergstrom, and D. Xu, "Comprehensive evaluation of alphafold-multimer, alphafold3 and colabfold, and scoring functions in predicting protein-peptide complex structures," *bioRxiv*, pp. 2024–11, 2024.

[21] Í. Valsson, M. T. Warren, C. M. Deane, A. Magarkar, G. M. Morris, and P. C. Biggin, "Narrowing the gap between machine learning scoring functions and free energy perturbation using augmented data," *Communications Chemistry*, vol. 8, no. 1, p. 41, 2025.

[22] S. Azimi, S. Khuttan, J. Z. Wu, R. K. Pal, and E. Gallicchio, "Relative binding free energy calculations for ligands with diverse scaffolds with the alchemical transfer method," *Journal of Chemical Information and Modeling*, vol. 62, no. 2, pp. 309–323, 2022.

[23] E. E. Bolton et al., "Pubchem3d: A new resource for scientists," *Journal of cheminformatics*, vol. 3, pp. 1–15, 2011.

[24] J. Durairaj et al., “Plinder: The protein-ligand interactions dataset and evaluation resource,” *bioRxiv*, pp. 2024–07, 2024.

[25] R. Meli and P. C. Biggin, “Spyrmsd: Symmetry-corrected rmsd calculations in python,” *Journal of cheminformatics*, vol. 12, no. 1, p. 49, 2020.

[26] P. Lemos et al., “Sair: Enabling deep learning for protein-ligand lnteractions with a synthetic structural dataset,” *bioRxiv*, pp. 2025–06, 2025.

[27] P. C. Hawkins, A. G. Skillman, G. L. Warren, B. A. Ellingson, and M. T. Stahl, “Conformer generation with omega: Algorithm and validation using high quality structures from the protein databank and cambridge structural database,” *Journal of chemical information and modeling*, vol. 50, no. 4, pp. 572–584, 2010.

[28] A. Tong et al., “Improving and generalizing flow-based generative models with minibatch optimal transport,” *arXiv preprint arXiv:2302.00482*, 2023.

[29] M. Plainer et al., “Diffdock-pocket: Diffusion for pocket-level docking with sidechain flexibility,” 2023.

[30] R. Singhal et al., “A general framework for inference-time scaling and steering of diffusion models,” *arXiv preprint arXiv:2501.06848*, 2025.

[31] R. Wang, X. Fang, Y. Lu, C.-Y. Yang, and S. Wang, “The pdbbind database: Methodologies and updates,” *Journal of medicinal chemistry*, vol. 48, no. 12, pp. 4111–4119, 2005.

[32] P. Kunzmann et al., “Biotite: New tools for a versatile python bioinformatics library,” *BMC bioinformatics*, vol. 24, no. 1, p. 236, 2023.

[33] A. Jakalian, D. B. Jack, and C. I. Bayly, “Fast, efficient generation of high-quality atomic charges. am1-bcc model: Ii. parameterization and validation,” *Journal of computational chemistry*, vol. 23, no. 16, pp. 1623–1641, 2002.

[34] S. Doerr, M. J. Harvey, F. Noé, and G. De Fabritiis, “Htmd: High-throughput molecular dynamics for molecular discovery,” *Journal of chemical theory and computation*, vol. 12, no. 4, pp. 1845–1852, 2016.

[35] F. Sabanés Zariquiey, S. E. Farr, S. Doerr, and G. De Fabritiis, “Quantumbind-rbfe: Accurate relative binding free energy calculations using neural network potentials,” *Journal of Chemical Information and Modeling*, 2025.

[36] J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser, and C. Simmerling, “Ff14sb: Improving the accuracy of protein side chain and backbone parameters from ff99sb,” *Journal of chemical theory and computation*, vol. 11, no. 8, pp. 3696–3713, 2015.

[37] J. Li, K. Yanagisawa, M. Sugita, T. Fujie, M. Ohue, and Y. Akiyama, “Cycpeptmpdb: A comprehensive database of membrane permeability of cyclic peptides,” *Journal of Chemical Information and Modeling*, vol. 63, no. 7, pp. 2240–2250, 2023.

[38] S. T. Runyon et al., “Structural and functional analysis of the pdz domains of human htra1 and htra3,” *Protein Science*, vol. 16, no. 11, pp. 2454–2471, 2007.

[39] J. Phan et al., “Structure-based design of high affinity peptides inhibiting the interaction of p53 with mdm2 and mdmx,” *Journal of Biological Chemistry*, vol. 285, no. 3, pp. 2174–2183, 2010.

[40] M. van Rosmalen et al., “Affinity maturation of a cyclic peptide handle for therapeutic antibodies using deep mutational scanning,” *Journal of Biological Chemistry*, vol. 292, no. 4, pp. 1477–1489, 2017.

[41] J. M. Rogers, T. Passioura, and H. Suga, “Nonproteinogenic deep mutational scanning of linear and cyclic peptides,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 43, pp. 10 959–10 964, 2018.

Supplementary Information

Methods

1.1 Data curation

The small molecule dataset is obtained from PubChem3D [23], which uses the OpenEye OMEGA software [27] to generate conformers for >170 million compounds in PubChem. Due to resource constraints, we download a random subset of 13.8 million compounds for pretraining using the single conformer found in PubChem's FTP site (`https://ftp.ncbi.nlm.nih.gov/pubchem/Compound_3D/01_conf_per_cmpd/`). We use a 90:10 train:test split and evaluate RMSD on the test set. For protein-ligand finetuning, protein-ligand complexes from Plinderv2 [24] are used, which results in a total of 293,591 complexes after filtering for ligands < 64 heavy atoms and systems for which a *holo* (bound) complex PDB file exists. We set aside 100 randomly selected structures for performance evaluation. The final ncAA-protein dataset is extracted from a snapshot of PDB (dated 2023-07-28) by filtering for the presence of non-canonical residues. This is performed by filtering for all amino acids that are not 'peptide-linking' or 'terminus' residues as defined in the chemical component dictionary described in PDBeChem (`https://www.ebi.ac.uk/pdbe-srv/pdbechem/`). We also remove selenomethionine and selenocysteine as these amino acids are routinely used to aid resolving X-ray crystal structures, and therefore are heavily biased in the PDB. We use any ncAA present in the PDB in at least 10 instances to curate the train/validation splits using a random 90:10 split, and test on ncAAs found in less than 10 instances. This results in a total of 38,618, 4,283, and 593 train, validation, and test ncAA examples, respectively, corresponding to 254 unique ncAAs in the train/validation sets, and 231 unique ncAAs in the test set. We note that some of the ncAAs extracted from the PDB exist as ligands in isolation rather than a part of the protein's polypeptide chain - we chose to use all examples for training due to data scarcity concerns.

1.2 Model Training

NCFlow is trained in three stages: first, small molecule conformers in PubChem3D are used to learn general chemical validity given a set of atom types and bond connectivities. Then, the model is finetuned on protein-ligand complexes to learn binding interface biophysics given a protein pocket. Finally, the model is further finetuned on ncAA-protein environments to learn ncAA-specific interactions and placing ncAAs as part of a polypeptide chain. The details for each training stage are described below:

1.2.1 General details

The core framework of NCFlow is based on the I-CFM formulation of flow matching depicted in [28], which we briefly describe here. Flow matching seeks to learn a vector (velocity) field u_t that defines an ODE, where the solutions to this ODE are defined by a flow ψ_t. X is defined as the trajectory that follow the vector field from X_0 to X_1, where $X_0 \sim p$ and $X_1 \sim q$, where p and q are the prior and data distributions, respectively. Thus, the ODE is defined by:

$$\frac{d}{dt}X_t = u_t(X_t), \text{where } X_t = \psi_t(X_0) \tag{1}$$

For generative modeling, we want to build a flow ψ_t that yields some probability path $X_t \sim p_t$ that transforms a sample from a simple source distribution (typically $X_0 \sim \mathcal{N}(0, \mathbb{1})$) to one from the complex target distribution $X_1 \sim p_{data}(X)$, and solve the ODE determined by the vector field u_t from $t = 0$ to $t = 1$. There are infinitely many choices to define ψ_t and thus p_t, but the simplest formulation is to construct a set of linear conditional paths $p_t(X|X_1)$ for every training data $X_1 \sim q$, resulting in the following marginal:

$$p_t(X) = \int p_{t|X_1}(X|X_1)q(X_1)dX_1, \text{where } p_{t|X_1}(X|X_1) = \mathcal{N}(X|tX_1, (1-t)^2\mathbb{1}) \tag{2}$$

which results in a simple expression for X_t that takes a linear combination of X_1 and X_0 dependent on t:

$$X_t = tX_1 + (1-t)X_0 \tag{3}$$

and a regression target of u_t as follows:

$$u_t(X|X_1) = \frac{X_1 - X}{1-t} \tag{4}$$

Rather than conditioning on just X_1, [28] reports a more flexible approach to define the probability path for arbitrary source distributions by independent coupling of source and target points (referred to as I-CFM), resulting in the following expressions:

$$p_t(X|X_0, X_1) = \mathcal{N}(X|tX_1 + (1-t)X_0, \sigma^2) \tag{5}$$

$$u_t(X|X_0, X_1) = X_1 - X_0 \tag{6}$$

Equipped with this, we can define an objective function to train a neural network parameterized by θ to approximate u_t:

$$\mathcal{L}_\theta = \mathbb{E}_{t,X_t,X_1,X_0} \left\| u_t^\theta(X_t) - (X_1 - X_0) \right\|^2 \tag{7}$$

where $t \sim U[0,1], X_0 \sim p, X_1 \sim q$.

For sampling, we initalize the trajectory with $X_0 \sim p$ and apply a numerical ODE solver such as Euler's method defined below:

$$X_{t+h} = X_t + hu_t^\theta(X_t) \tag{8}$$

where h is a stepsize hyperparameter indicating fixed timesteps $t \in [0,1]$. We use 10 timesteps for all cases, and found no performance improvements with increased timesteps. Since we do not use an equivariant architecture (see *Model Details*), we use data augmentation at every iteration by randomly rotating and translating all atoms with a standard deviation of 1A across all stages of training. We also use exponential moving average of model parameters with a decay of 0.999 per iteration, and gradient clipping with norm 1.0. All models are trained on 4 NVIDIA A100s (40GB) with DistributedDataParallel.

1.2.2 Stage 1: small molecule pretraining

The initial pretraining stage on PubChem3D is rather straightforward, as the model simply learns to generate the full conformer given the atom types and bond connectivity graph. The conformers are centered at mean zero, and the model is trained with effective batch size of 32 and learning rate at $2e-4$. The model was trained for approx. 7 days, corresponding to 10 epochs.

1.2.3 Stage 2: protein-ligand finetuning

For protein-ligand data, the model takes in conditional pocket information defined by the 128 atoms closest to the ligand's centroid. Only the ligand is noised, and the loss is propagated from just the ligand atoms to train the model. The pocket and ligand coordinates are centered on the ligand's centroid to provide coarse information on the ligand coordinates - we note that this is not ideal if the desired task is protein-ligand docking or small molecule design. However, NCFlow is trained ultimately for single-residue ncAA structure prediction for which some conditional information (ex. backbone atom coordinates) exist, and therefore allowed such biases to be present in this training stage. The model was trained with effective batch size of 16 and learning rate $2e-5$. The model was trained for approx. 7 days, corresponding to 15 epochs.

1.2.4 Stage 3: ncAA-protein finetuning

To train on ncAA-protein environments, the pocket definition was expanded to 200 atoms to allow greater context when predicting the ncAA. The prior distribution is set as a Gaussian distribution with mean on the residue's CA coordinate. We also introduce local side-chain flexibility of the pocket residues by introducing Gaussian noise centered on the ground truth coordinates with standard deviation 0.2A, which then the model learns to locally repack neighboring side-chains while predicting the ncAA structure. Following [29], residues for local side-chain repacking were selected if any atom of the residue lies within 3.5A of any ncAA side-chain atom, which may increase the number of pocket atoms - we cropped this to a maximum of 256 atoms for training efficiency. Backbone atoms (including the carbonyl oxygen) remain fixed throughout training. At every training iteration, we sample a random structure corresponding to a ncAA to limit biases of certain ncAAs in the PDB. The model is trained with effective batch size of 4 and learning rate $2e-6$ for approx. 7 days.

Since the model does not explicitly enforce equivariance, we observe that chirality issues arise in some samples. We resolve this post hoc by discarding generated samples that do not exactly match the configuration of the chiral centers denoted in the input ncAA SMILES string (using RDKit's FINDMOLCHIRALCENTERS function). A more direct approach to resolve chirality issues using inference-time approaches such as Feynman-Kac steering [30] used in Boltz [15] may be beneficial, but the low memory and runtime requirements of NCFlow (<4GB VRAM and 5 seconds per sample for on a NVIDIA RTX3060) does not necessitate this. Moreover, we observe that 8 samples per ncAA is sufficient for obtaining ncAAs with correct chirality in almost all cases.

1.3 Model architecture and featurization

The model is based on AlphaFold3's Pairformer and Atom Transformer modules implemented in Boltz. We remove the tokenization scheme used in AlphaFold3, and rather use an atom-level representation throughout the model. The node features consist of the atom types, timestep, noised coordinates, and a binary mask indicating the target atoms to predict (ex. ligand atoms for PubChem/Plinder, ncAA for PDB). For the final finetuning stage on ncAA-protein environments, we also include a mask to indicate the sidechains for repacking. The pair features are the pairwise Euclidean distances and one-hot-encoded bond connectivities, which contains five labels indicating single, double, triple, aromatic, or no bonds between two atoms. The model contains 16 Pairformer layers and 4 Atom Transformer layers with single and pair hidden dimensionality of 128 and 64, respectively, resulting in 8.7 million trainable parameters. The confidence model is a down-sized version of the main model with 6 Pairformer layers and 1 Atom Transformer layer. The confidence model is trained to predict per-atom LDDT scores using a similar 'diffusion rollout' scheme adopted by AlphaFold3, where the main model is frozen and candidates are sampled using 10 timesteps which are then scored using the confidence model.

1.4 Design pipeline

To design peptides with ncAAs, we use *in silico* deep mutational scanning in conjunction with binding affinity estimators to identify variants that increase binding affinity to a given target. The details are described below in their respective sections.

1.4.1 Mutational scanning

Deep mutational scanning is a standard experimental technique for protein engineering, where proteins are mutated at every position with all 20 canonical amino acids to produce a library of variants, which then undergo a selection process to filter high fitness variants and sequencing for identification. We follow a similar approach here, where all positions of the peptide are mutated with a pool of ncAAs, which are assessed through *in silico* binding affinity estimators to identify variants that increase binding affinity. The pool of ncAAs are obtained by curating synthesizable ncAAs from WuXi AppTec, a peptide synthesis CDMO company. Specifically, we download all ncAAs linked to Fmoc, have a related parent canonical amino acid, and are parseable with RDKit, resulting in a total of 939 ncAAs. To reduce combinatorial complexity, we only mutate each residue to a related ncAA according to its parent amino acid, which can range from 3-379 variants (except isoleucine, which does not have any variants after these filters). The exact number of ncAAs per parent amino acid is listed in Table S1. Using the curated pool of ncAAs, we generate a large pool of structural variants at each position for each protein-peptide complex test case.

1.4.2 Binding affinity estimation

The presence of ncAAs makes it difficult to use any existing protein-peptide binding affinity prediction tools as they typically can only encode canonical residue identities. Thus, we explore using protein-ligand binding affinity prediction methods which are more flexible in the way that the ligands are encoded (ex. atom types and bond graph). Specifically, we use a deep learning-based tool called AEV-PLIG [21], which is trained on PDBBind2020 [31] and other augmented datasets to predict protein-ligand binding affinities closer to FEP+ calculations. We also note that PDBBind contains peptide ligands, so we reasoned that the model can generalize to predict ncAA-containing protein-peptide binding affinities. To protonate the peptides we use the hydride package within Biotite [32], which can then be processed by AEV-PLIG to predict pK_d values.

We also use an alchemical relative binding free energy prediction tool called Alchemical Transfer Method (ATM) [22], which runs MD ensembles with an alchemical coordinate transformation that translates one ligand from the binding pocket to the solvent, and another ligand from the solvent to the binding pocket. By calculating the free energy differences between the alchemical states, we can measure relative binding affinity of two ligands of interest. To prepare the system for ATM, we solvate the system with 10nm padding at pH 7.0, and use Antechamber to prepare the ligands with the GAFF2 force field and AM1-BCC charge model [33]. The wild-type peptide is left inside the binding pocket, and the peptide ncAA variant is displaced from the binding site by a coordinate translation of magnitude 25A along the vector from the target protein's centroid and the ligand binding site. We use the first three CA atoms for aligning the peptide reference atoms and positionally restrain all CA atoms of the protein target. The binding site restraints are set to all CA atoms for both the peptide and receptor. The system is simulated with 2fs timesteps using 8000 steps per sample for 200 samples across 22 replicas (alchemical states), corresponding to a 71ns MD ensemble per variant, or an average of 12 hours of runtime on a single NVIDIA A100 (40GB). We use the Acellera HTMD package [34] for system preparation and the Quantumbind-RBFE [35] pipeline to run ATM, using the Amber ff14sb force field [36] to parameterize the target protein. We experimented with using the AceFF 1.0 neural network potential to run NNP/MM simulations as in QuantumBind-RBFE, but the computational cost of using the NNP on longer ligands such as peptides made it infeasible to run in high-throughput. We tested using NNP/MM with ATM on three variants on a protein-peptide complex and observed comparable values with the GAFF2 force field at 10X computational cost - while inconclusive, this requires further investigation that we leave for future work. We also observe that some simulation runs fail due to exploding energy or coordinate issues, which can be a consequence of improperly defined constraints, initial positions, or simulation parameters (timestep, etc.). We observed that this issue is exacerbated with increased timesteps at 4fs with hydrogen mass repartitioning to 4 amu. Since molecular dynamics simulations often require system-specific optimization for accurate modeling, it may be possible that the current set of hyperparameters causes some simulations to explode. Thus, an optimized ATM protocol applicable to all protein-peptide system is a direction of future work.

We first tested AEV-PLIG on long 'peptides' in the PDBBind-PP dataset not used for training, and observed a moderate Spearman correlation of 0.39 on absolute binding affinity measurements. This suggests that AEV-PLIG can indeed be used for assessing protein-peptide complexes despite being trained on smaller chemical ligands and short peptides. On a test run on 28 peptide variants containing a relatively smaller pool of ncAAs from CycPeptMPDB [37] on the target PDB 1SSC, we observe a negative Spearman correlation of -0.62 between AEV-PLIG predicted ΔpK_d and ATM $\Delta\Delta G$ (Supplementary Figure 2), suggesting that the two deep learning-based and alchemical methods are moderately aligned on variants that constitute strong and poor binders.

To further evaluate the proposed scoring method on known protein-peptide complexes, we curated multiple experimental binding affinity datasets from literature. We obtained four sets of protein-peptide complexes containing multiple mutations to canonical amino acids and corresponding binding affinities [38, 39, 40], which allows us to assess the effectiveness of the AEV-PLIG/ATM scoring method on experimentally validated affinities. We generated 3D structure for each variant with either PyMOL's mutagenesis tool (which simply finds the lowest-energy rotamer), or AlphaFold2 using high-confidence (pLDDT > 90) and low-RMSD (<1.5A) predicted structures. We predicted the binding affinity of each variant using AEV-PLIG, and relative $\Delta\Delta G$ measurements using the ATM protocol compared to a given reference peptide (by selecting the wild-type or main peptide scaffold used by each paper), and report the results in Supplementary Figure 5 and Supplementary Table 5. Interestingly, we observe that correlations with binding affinity were often weak or insignificant in most cases, with the exception of ATM on Mdm2-p53 which exhibits a remarkable 0.76 correlation with its experimental binding affinity. Moreover, AEV-PLIG exhibits a moderate correlation of -0.43 and -0.48 with Htra1-PDZ and cetuximab-meditope complexes, though with $p > 0.05$ due to insufficient number of samples. However, if we apply the cutoff of AEV-PLIG $\Delta pK_d > 0$ and ATM $\Delta\Delta G < -1.0$ kcal/mol and analyze the confusion matrices, both methods show strong performance in distinguish stronger vs weaker binders compared to the reference peptide, with ATM often exhibiting higher accuracy and precision than AEV-PLIG across the four test complexes. Remarkably, combining both methods with their respective cutoffs results in a precision of 1.00 across all samples, suggesting that applying the two orthogonal methods for binding affinity prediction is a promising approach for identifying peptide variants with increased binding affinity. We then tested the protocol on a deep mutational scanning dataset of two protein-peptide complexes with 41 canonical and non-canonical amino acids reported in [41], and report the results in Supplementary Figure 6. The PUMA peptide showed a promising yet weak negative correlation of -0.19 with precision of 0.36 and recall 0.79 using the $\Delta pK_d > 0$ cutoff (given a 29-71 positive-negative imbalanced dataset), but running ATM on PUMA almost always failed due to the length of the peptide (35aa) causing simulation errors. On the other hand, the CP2 peptide failed to exhibit any notable correlation with neither AEV-PLIG or ATM on 150 selected variants, suggesting that the scoring method works on a system-specific basis and reiterates the need for more robust scoring methods beyond existing deep learning-based and alchemical approaches. We also note that the aforementioned dataset calculates $\Delta\Delta G$ using relative enrichment of DNA sequencing reads calibrated to known binding affinities and therefore may not capture true $\Delta\Delta G$ upon binding.

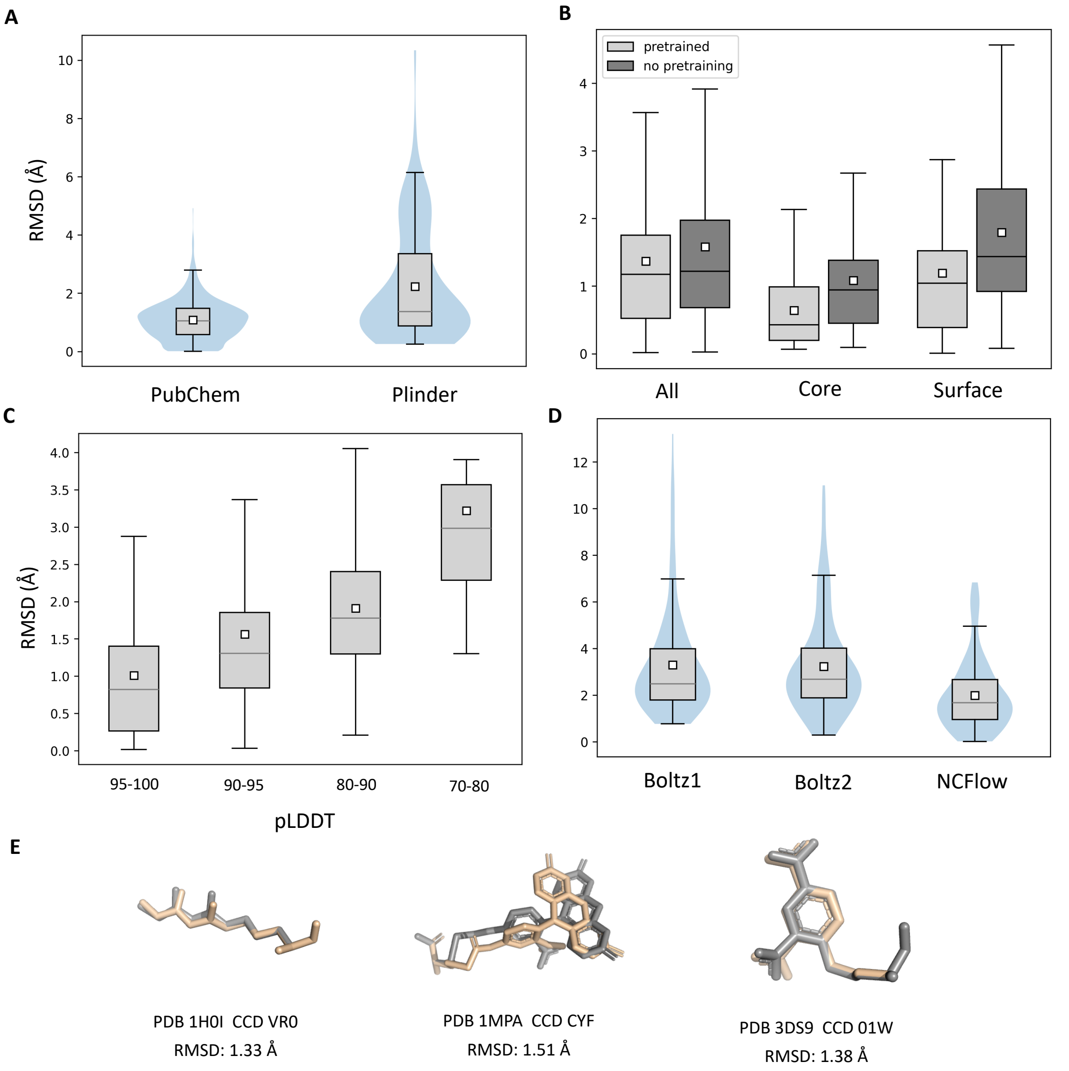

Figure S1: Performance evaluation of NCFlow. A) RMSD plots of PubChem and Plinder pretrained models. B) RMSD evaluation of NCFlow trained from scratch and finetuned from the PubChem/Plinder pretrained model. Across all, core (> 35 CA atoms within 12Å), and surface (< 20 CA atoms within 12Å) residues, the finetuned model outperforms the model trained from scratch. C) Binned pLDDT and RMSD distributions show that high-confidence samples can be used to filter out poor predicted conformers. D) NCFlow outperforms Boltz1 and Boltz2 in single-residue ncAA prediction with an mean RMSD difference of 1.86Å and 1.81Å, respectively. E) Randomly selected ncAA predictions by NCFlow, with ground truth in gray and generated conformers in beige. Note that the white square box in all boxplots represent the mean value of the distribution.

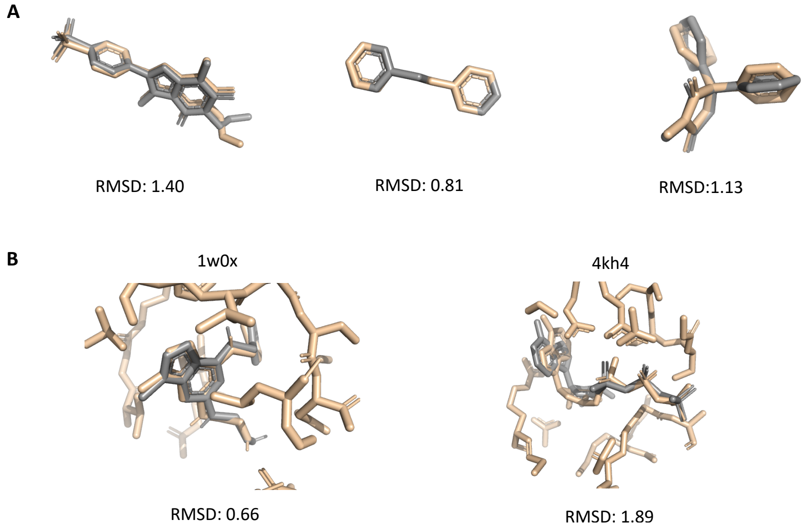

Figure S2: Randomly selected samples from pretrained models on (A) Pubchem and (B) Plinder datasets.

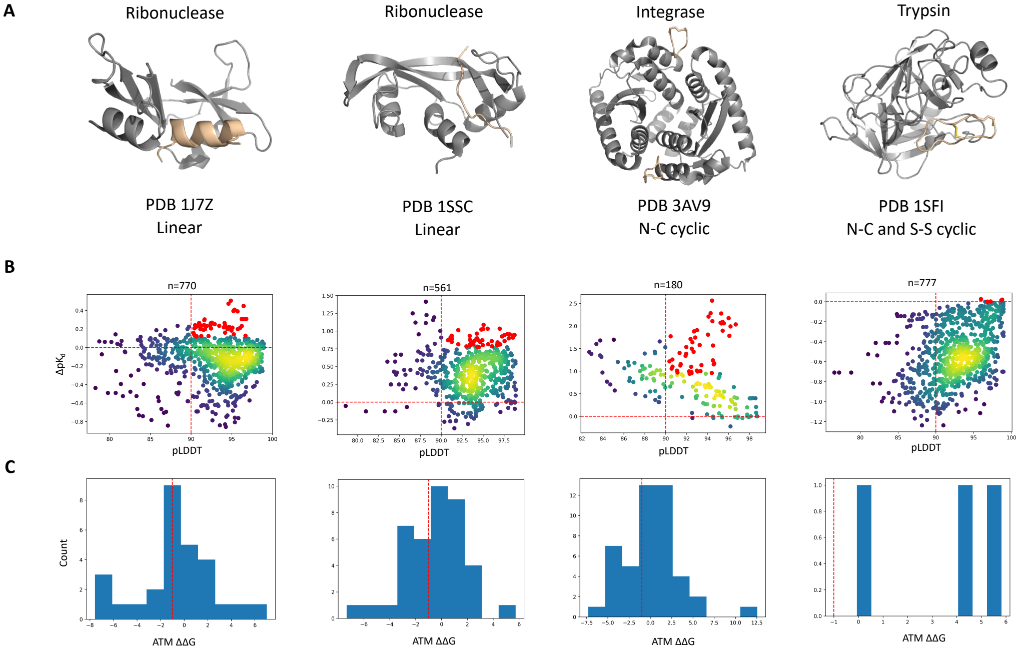

Figure S3: Design of peptide-ncAA variants using NCFlow. A) Selected protein-peptide test cases, ranging from helical linear peptides (1J7Z) to head-to-tail and disulfide-cyclized peptides (1SFI). B) pLDDT and AEV-PLIG ΔpK_d distributions of all ncAA variants in each test case. Yellow and blue indicate high and low density regions, respectively, and red indicate the top 50 selected variants by ΔpK_d. Vertical and horizontal red dashed lines indicate pLDDT = 90 and ΔpK_d = 0, respectively. C) ATM $\Delta\Delta G$ measurements of top 50 variants for each test case.

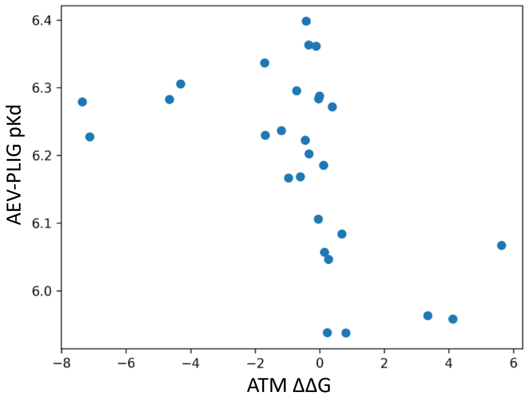

Figure S4: ATM $\Delta\Delta G$ and AEV-PLIG pK_d on PDB 1SSC with a smaller set of ncAAs from CycPeptMPDB [37], exhibiting a Spearman correlation of -0.62 with p < 0.01.

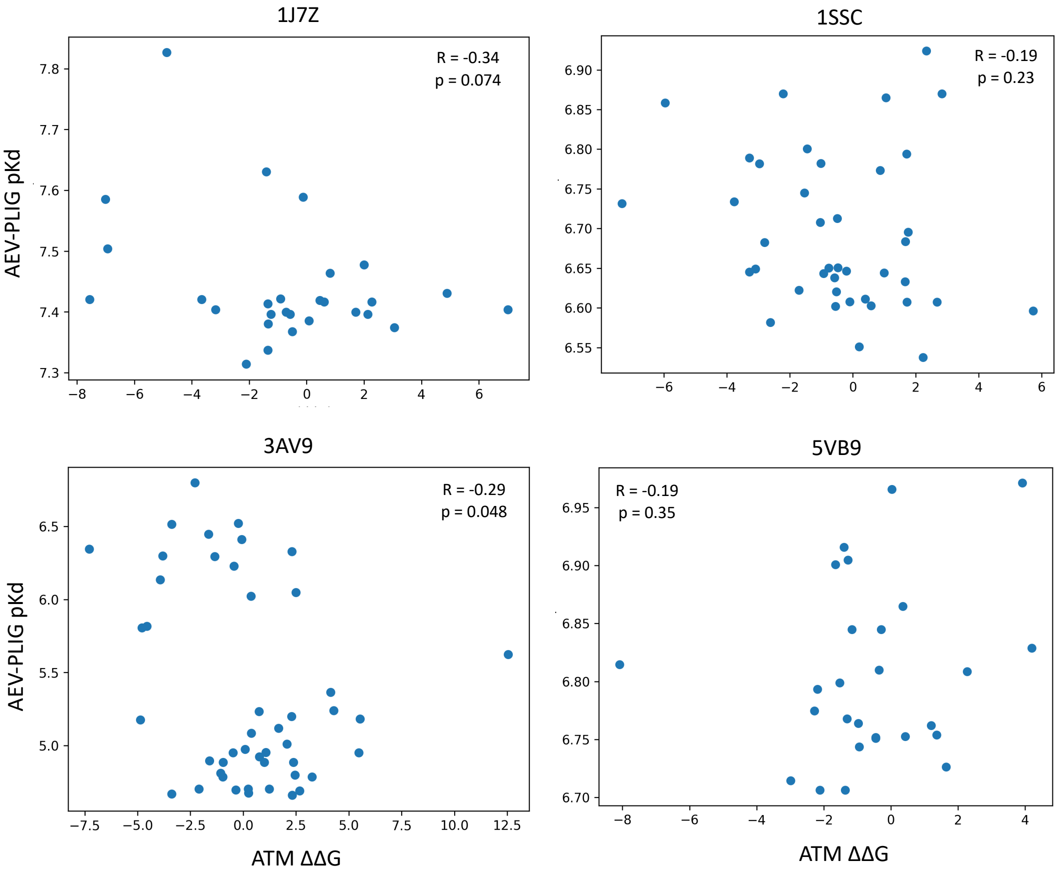

Figure S5: ATM $\Delta\Delta G$ and AEV-PLIG pK_d correlation on four test cases.

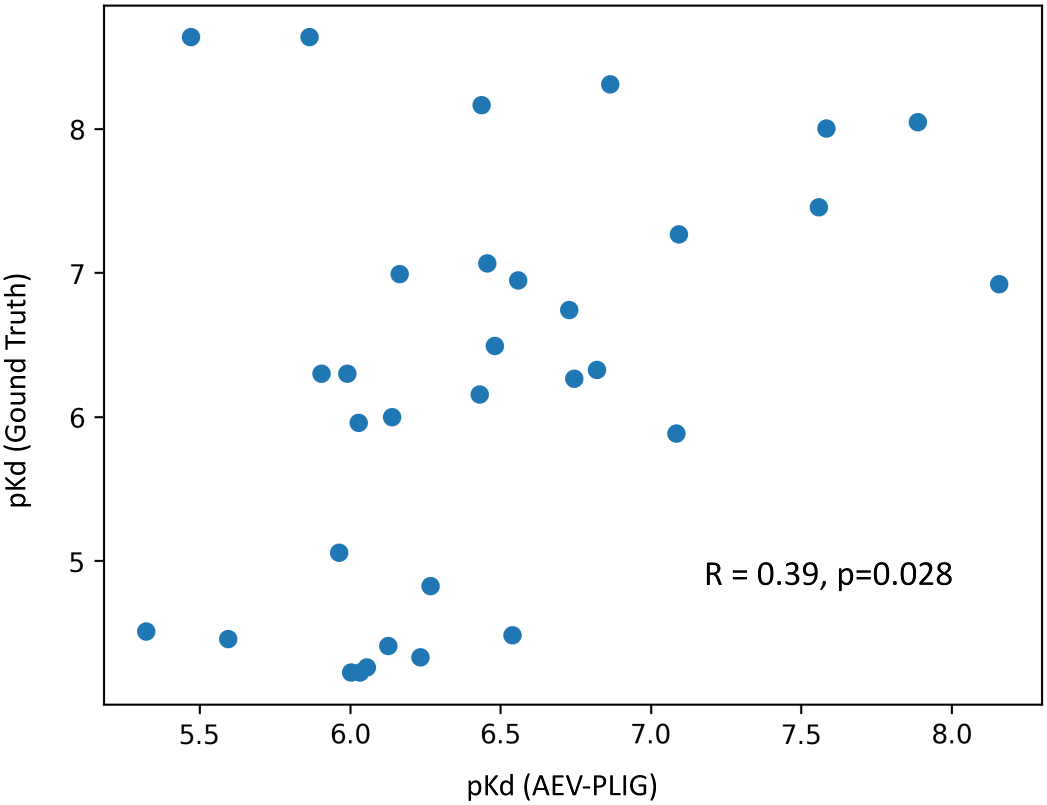

Figure S6: AEV-PLIG predictions vs experimental Kd values on long peptides (20-30aa) in PDBBind2020. These peptides are found in the PDBBind-PP dataset and are not in the training set of AEV-PLIG.

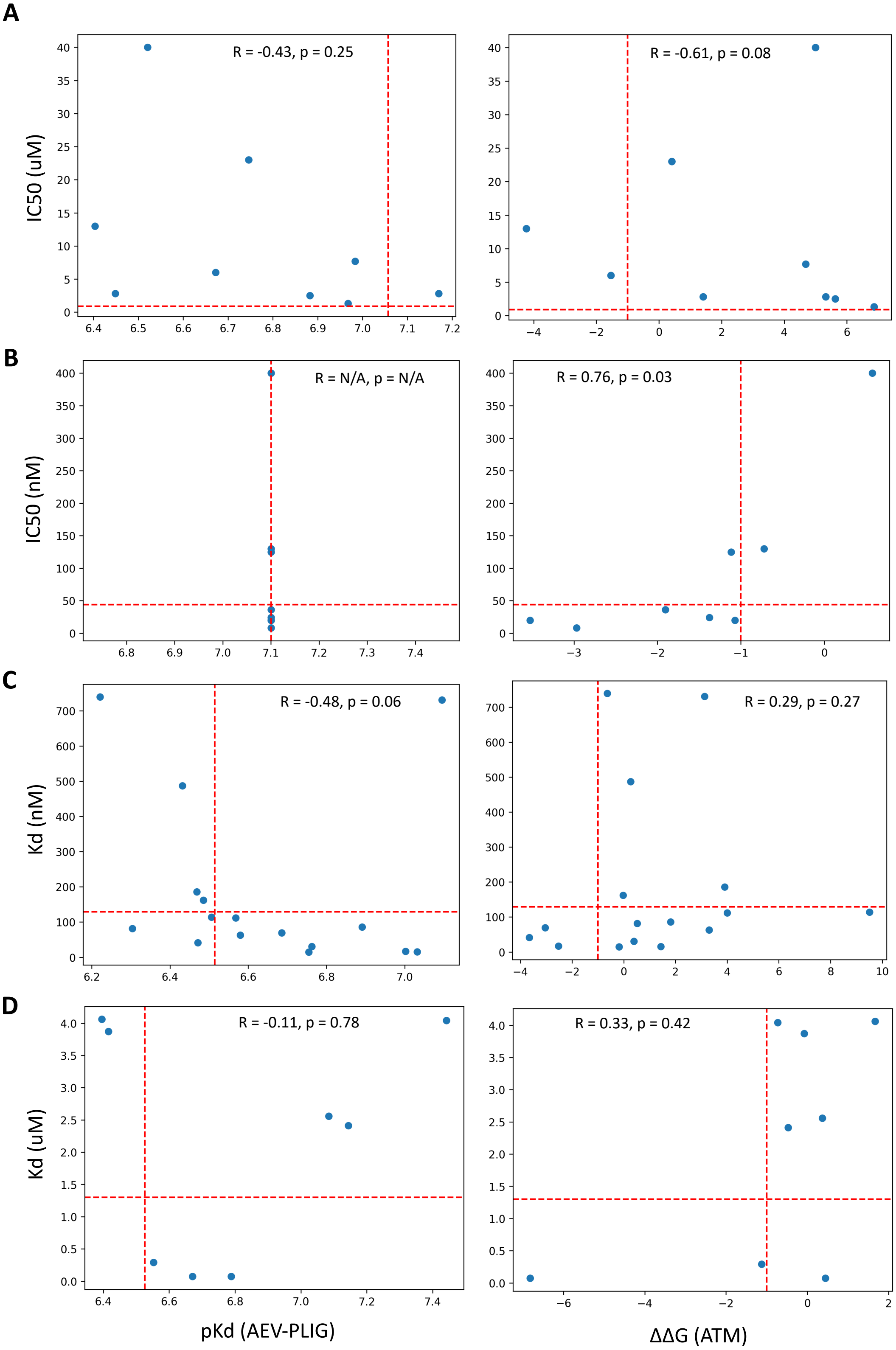

Figure S7: AEV-PLIG (left) and ATM (right) predictions of binding affinity plotted against experimentally measured binding affinities across the four experimental datasets in Table S5, corresponding to (A) Htra1-PDZ, (B) Mdm2-p53, (C) cetuximab-meditope, and (D) Undisclosed. Note that AEV-PLIG appears to predict the same affinity across all variants of Mdm2, perhaps due to

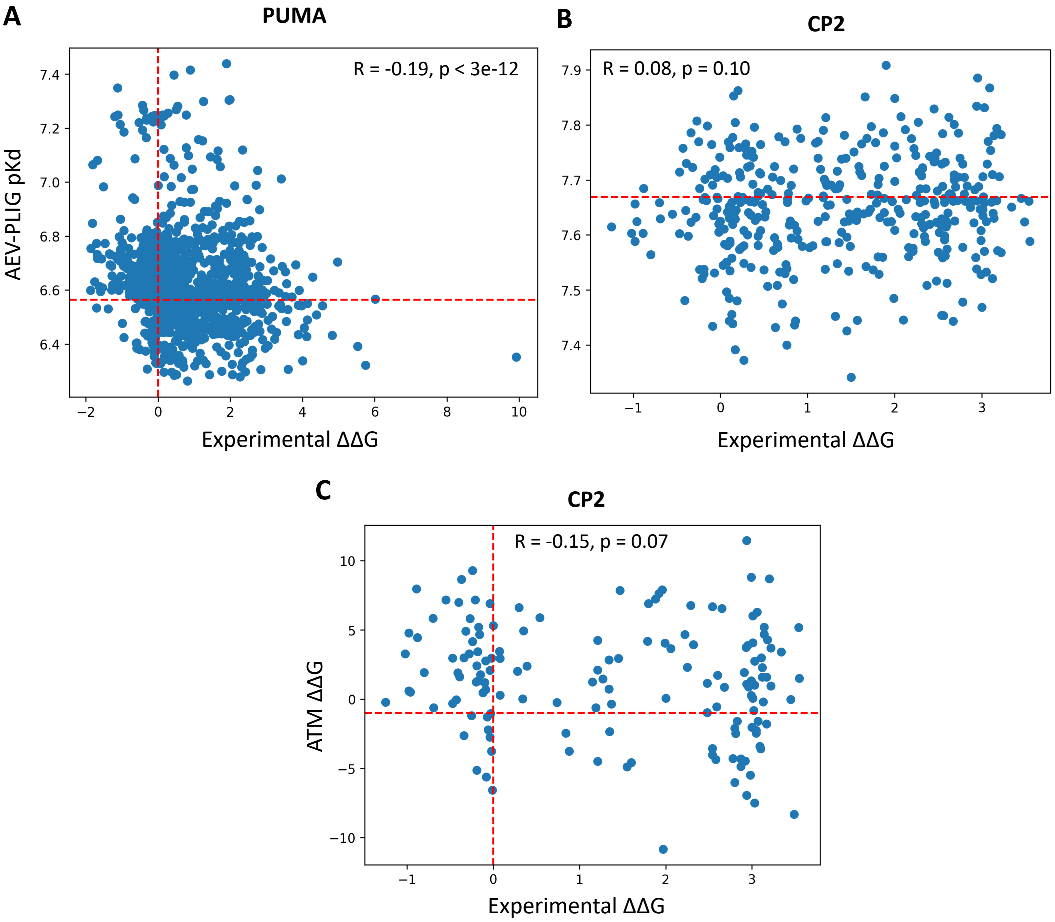

Figure S8: AEV-PLIG and ATM predictions of deep mutational scanning datasets [41] on the peptides PUMA (A) and CP2 (B,C). ATM on the PUMA peptide was unsuccessful due to its length (35aa).

Table S1: Number of ncAAs per parent amino acid.

Parent AA	ncAA Count
LYS	111
PHE	379
PRO	35
ASN	9
MET	7
TRP	67
ARG	36
SER	23
TYR	63
CYS	49
GLY	57
GLU	17
THR	17
GLN	7
ALA	19
ASP	10
HIS	25
LEU	5
VAL	3
ILE	0
Total	979

Table S2: List of variants and corresponding binding affinities for HtrA1 PDZ domain [38]. Bold indicates the reference peptide used for relative binding affinity calculations.

ID	Sequence	IC50 (uM)
H1C1	**DSRIWWV**	**0.9**
H1C2	GWKTWIL	7.7
H1C3	DIETWLL	23
H1C4	WDKIWHV	2.8
H1C1c	DSRIWAV	6
H1C1d	DSRIAWV	40
H1C1e	DSRAWWV	13
H1C1f	DSAIWWV	2.5
H1C1g	DARIWWV	1.3
H1C1h	ASRIWWV	2.8

Table S3: List of variants and corresponding binding affinities for Mdm2-p53 [39]. Bold indicates the reference peptide used for relative binding affinity calculations.

ID	Sequence	IC50 (nM)
PDI	**LTFEHYWAQLTS**	**44**
PDIQ	ETFEHWWSQLLS	8
6W11L	LTFEHWWAQLLS	20
1E6W	ETFEHWWAQLTS	20
6W8S	LTFEHWWSQLTS	24
6W	LTFEHWWAQLTS	36
6W9S	LTFEHWWASLTS	125
6W8S9S	LTFEHWWSSLTS	130
6N	LTFEHNWAQLTS	400
p53	ETFSDLWKLLPE	2000

Table S4: List of variants and corresponding binding affinities for cetuximab-meditope [40]. Bold indicates the reference peptide used for relative binding affinity calculations.

ID	Sequence	Kd (nM)
Md1	**CQFDLSTRRLKC**	**129.4**
Md3	CQYNLSSRALKC	739
Md2	CVWQRWQKSYVC	731
Q1V	CVFDLSTRRLKC	111.5
S5G	CQFDLGTRRLKC	113.9
K10R	CQFDLSTRRLRC	81.3
Q1V_S5G	CVFDLGTRRLKC	69.4
Q1V_K10R	CVFDLSTRRLRC	63.2
S5G_K10R	CQFDLGTRRLRC	41.1
Q1V_S5G_K10R	CVFDLGTRRLRC	30.3
Q1V_D3N_S5G_K10R	CVFNLGTRRLRC	14.5
Q1V_S5G_T6I_K10R	CVFDLGIRRLRC	16.8
Q1V_D3N_S5G_T6I_K10R	CVFNLGIRRLRC	15.8
Q1V_S5Y_K10R	CVFDLYTRRLRC	86.3
Q1V_S5G_T6M_K10R	CVFDLGMRRLRC	162
Q1V_S5Y_T6M_K10R	CVFDLYMRRLRC	487
Q1V_D3R_S5G_K10R	CVFRLGTRRLRC	186

Table S5: Performance evaluation of AEV-PLIG and ATM binding affinity predictions on four protein-peptide variant datasets.

Target	Method	Accuracy	Precision	Recall
HtrA1 [38]	AEV-PLIG	0.89	N/A	N/A
	ATM	0.78	N/A	N/A
	Both	**1.00**	N/A	N/A
Mdm2 [39]	AEV-PLIG	N/A	N/A	N/A
	ATM	**0.67**	**1.00**	**0.67**
	Both	**0.67**	**1.00**	**0.67**
Cetuximab [40]	AEV-PLIG	0.75	0.89	0.73
	ATM	**0.80**	0.50	**1.00**
	Both	0.44	**1.00**	0.18
Undisclosed[1]	AEV-PLIG	0.63	0.50	**1.00**
	ATM	**0.88**	**1.00**	0.67
	Both	**0.88**	**1.00**	0.67

[1] Due to conflict of interest.