AlphaFold Meets Flow Matching for Generating Protein Ensembles

Bowen Jing [1] **Bonnie Berger** [1 2] **Tommi Jaakkola** [1]

Abstract

The biological functions of proteins often depend on dynamic structural ensembles. In this work, we develop a flow-based generative modeling approach for learning and sampling the conformational landscapes of proteins. We repurpose highly accurate single-state predictors such as AlphaFold and ESMFold and fine-tune them under a custom flow matching framework to obtain sequence-conditioned *generative* models of protein structure called AlphaFLOW and ESMFLOW. When trained and evaluated on the PDB, our method provides a superior combination of precision and diversity compared to AlphaFold with MSA subsampling. When further trained on ensembles from all-atom MD, our method accurately captures conformational flexibility, positional distributions, and higher-order ensemble observables for unseen proteins. Moreover, our method can diversify a static PDB structure with faster wall-clock convergence to certain equilibrium properties than replicate MD trajectories, demonstrating its potential as a proxy for expensive physics-based simulations. Code is available at https://github.com/bjing2016/alphaflow.

1. Introduction

Proteins adopt complex three-dimensional structures, often as members of structural *ensembles* with distinct states, collective motions, and disordered fluctuations, to carry out their biological functions. For example, conformational changes are critical in the function of transporters, channels, and enzymes, and the properties of equilibrium ensembles help govern the strength and selectivity of molecular interactions (Meller et al., 2023; Vögele et al., 2023). While deep learning methods such as AlphaFold (Jumper et al., 2021) have excelled in the single-state modeling of experimental protein structures, they fail to account for this conformational heterogeneity (Lane, 2023; Ourmazd et al., 2022). Hence, a method which builds upon the level of accuracy of single-structure predictors, but reveals underlying structural ensembles, would be of great value to structural biologists.

Existing machine learning approaches for generating structural ensembles have focused on inference-time interventions in AlphaFold that modify the multiple sequence alignment (MSA) input (Del Alamo et al., 2022; Stein & Mchaourab, 2022; Wayment-Steele et al., 2023), resulting in a different structure prediction for each version of the MSA. While these approaches have demonstrated some success, they suffer from two key limitations. First, by operating on the MSA, they cannot be generalized to structure predictors based on protein language models (PLMs) such as ESMFold (Lin et al., 2023) or OmegaFold (Wu et al., 2022), which have grown in popularity due to their fast runtime and ease of use. Secondly, these inference-time interventions do not provide the capability to train on protein ensembles from beyond the PDB—for example, ensembles from molecular dynamics, which are of significant scientific interest but can be extremely expensive to simulate (Shaw et al., 2010).

To address these limitations, in this work we combine AlphaFold and ESMFold with *flow matching*, a recent generative modeling framework (Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022), to propose a principled method for sampling the conformational landscape of proteins. While AlphaFold and ESMFold were originally developed and trained as *regression* models that predict a single best protein structure for a given MSA or sequence input, we develop a strategy for repurposing them as (sequence-conditioned) *generative* models of protein structure. This synthesis relies on the key insight that iterative denoising frameworks (such as diffusion and flow-matching) provide a general recipe for converting regression models to generative models with relatively little modification to the architecture and training objective. Unlike inference-time MSA ablation, this strategy applies equally well to PLM-based predictors and can be used to train or fine-tune on arbitrary ensembles.

While flow matching has been well established for images, its application to protein structures remains nascent (Bose et al., 2023). Hence, we develop a custom flow matching

[1]CSAIL, Massachusetts Institute of Technology [2]Department of Mathematics, Massachusetts Institute of Technology. Correspondence to: Bowen Jing <bjing@mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

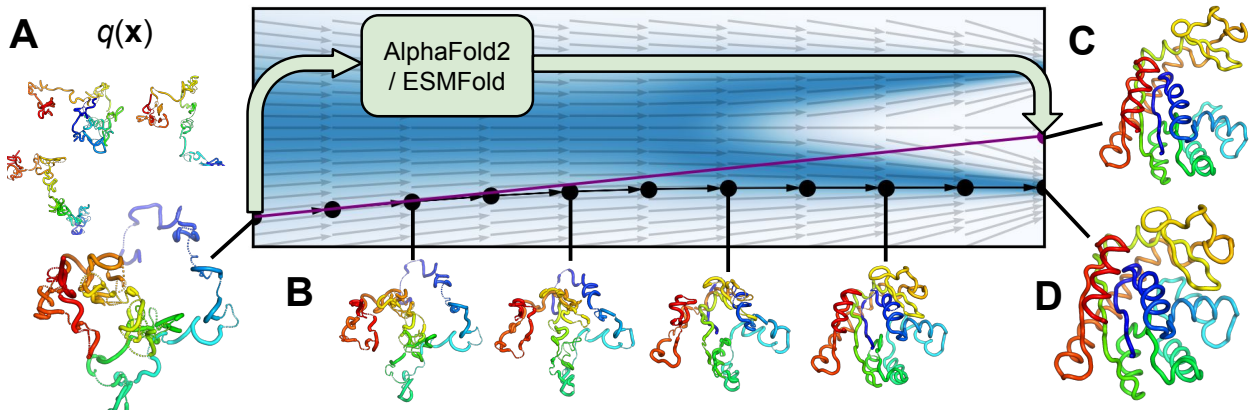

Figure 1. **Conceptual overview of AlphaFLOW / ESMFLOW.** (A) Samples are drawn from a harmonic (polymer-like) prior. (B) The sample is progressively refined or denoised under a flow field controlled by the structure prediction model (AlphaFold or ESMFold). (C) At each step, the denoised structure prediction parameterizes the direction of the flow and we interpolate the current sample towards it. (D) The final prediction is a sample from the learned distribution of structures.

framework tailored to the architecture and training practices of AlphaFold and ESMFold. Our framework leverages the polymer-structured prior distribution from harmonic diffusion (Jing et al., 2023), but improves over it by defining a scale-invariant noising process resilient to missing and cropped residues. These improvements directly result from the increased modeling flexibility offered by flow matching and contribute to the performance of our method.

We demonstrate the performance of our flow-matching variants of AlphaFold and ESMFold—named AlphaFLOW and ESMFLOW—in two distinct settings. First, after fine-tuning these models only on structures from the PDB, we substantially surpass the precision-diversity Pareto frontier of MSA ablation baselines on a test set of recently deposited conformationally heterogeneous proteins. Second, we showcase the ability to learn from ensembles beyond the PDB by further training on the ATLAS dataset (Vander Meersche et al., 2023) of molecular dynamics simulations. When evaluated on test proteins structurally dissimilar from the training set, AlphaFLOW substantially surpasses the MSA baselines in the prediction of conformational flexibility, distributional modeling of atomic positions, and replication of higher-order ensemble observables such as intermittent contacts and solvent exposure. Furthermore, when a static PDB structure is provided as a template, sampling from AlphaFLOW provides faster wall-clock convergence to many equilibrium properties than running molecular dynamics (MD) simulation starting from that structure. Thus, our method can be used in place of expensive simulations to diversify and obtain equilibrium ensembles of solved protein structures.

2. Background

Protein structure prediction. The modern approach for protein structure prediction was pioneered by AlphaFold (Jumper et al., 2021), which takes as input (1) the protein sequence, (2) a MSA of evolutionarily related sequences, and optionally (3) a template structure of a related protein, and predicts the all-atom 3D coordinates of single protein structure. AlphaFold was developed and trained in an end-to-end fashion under a regression-like FAPE loss with structures from the PDB. Later works, such as ESMFold (Lin et al., 2023) and OmegaFold (Wu et al., 2022), modified the pipeline by substituting the MSA with embeddings from a protein language model (PLM) and eschewing the template input, but otherwise kept the same architecture and training framework as AlphaFold.

Modeling protein ensembles. In the post-AlphaFold era, several works have emphasized diversifying highly accurate single-structure predictions to reflect underlying conformational heterogeneity (Lane, 2023; Chakravarty & Porter, 2022; Saldaño et al., 2022; Xie & Huang, 2023; Brotzakis et al., 2023; Bryant, 2023; Porter et al., 2023). Most prominently, Del Alamo et al. (2022) demonstrated that multiple functional states could be obtained by subsampling the MSA input to AlphaFold. Since then, MSA subsampling has become the *de-facto* standard methodology and has been employed to study conformational states of kinases (Faezov & Dunbrack Jr, 2023; Herrington et al., 2023; Casadevall et al., 2023), variant effects on conformational states (da Silva et al., 2023), and to seed molecular dynamics simulations (Vani et al., 2023). Alternative approaches have also been proposed in the form of point mutations to the MSA (Stein & Mchaourab, 2022; 2023) and MSA clustering (Wayment-Steele et al., 2023). .

An emerging line of work seeks to directly train sequence-to-structure generative models of protein ensembles. EigenFold (Jing et al., 2023) and Distributional Graphormer (Zheng et al., 2023) use harmonic diffusion and $SE(3)$ diffusion (Yim et al., 2023), respectively, to generate ensembles. SENS (Lu et al., 2023) is a local generative model that diver-

sifies single starting structures via local exploration of the conformational landscape. However, these models have yet to show convincing validations or comparisons with MSA subsampling methods on PDB test sets.

A related but separate line of work has focused on learning generative models of Boltzmann distributions as proxies for expensive molecular dynamics simulation. These models were initially conceived as normalizing flows that provided exact likelihoods and thus a means to train with energies and reweigh samples at inference time (Noé et al., 2019; Köhler et al., 2021; Midgley et al., 2022; Abdin & Kim, 2023; Felardos et al., 2023). However, these normalizing flows have proven difficult to scale beyond small molecules and toy systems. More recently, the proliferation of diffusion models has shifted the focus of this line of work towards scalability and generalization (Arts et al., 2023; Zheng et al., 2023) rather than exact likelihoods. Our method, when trained on MD ensembles, can be viewed as belonging to this new generation of Boltzmann-targeting generative models.

Flow matching (Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023; Liu et al., 2022) is a generative modeling paradigm that resembles and builds upon the significant success of diffusion models (Ho et al., 2020; Song et al., 2021) in image and molecule domains. The fundamental object in flow matching is a conditional probability path $p_t(\mathbf{x} \mid \mathbf{x}_1), t \in [0, 1]$: a family of densities conditioned on a data point $\mathbf{x}_1 \sim p_{\text{data}}$ which interpolates between a shared prior distribution $p_0(\mathbf{x} \mid \mathbf{x}_1) = q(\mathbf{x})$ and an approximate Dirac $p_1(\mathbf{x} \mid \mathbf{x}_1) \approx \delta(\mathbf{x} - \mathbf{x}_1)$. Given a conditional vector field $u_t(\mathbf{x} \mid \mathbf{x}_1)$ that generates the time evolution of $p_t(\mathbf{x} \mid \mathbf{x}_1)$, one then learns the *marginal vector field* with a neural network:

$$\hat{v}(\mathbf{x}, t; \theta) \approx v(\mathbf{x}, t) := \mathbb{E}_{\mathbf{x}_1 \sim p_t(\mathbf{x}_1 \mid \mathbf{x})}[u_t(\mathbf{x} \mid \mathbf{x}_1)] \quad (1)$$

At convergence, the learned vector field $\hat{v}(\mathbf{x}, t; \theta)$ is a neural ODE that evolves the prior distribution $q(\mathbf{x})$ to the data distribution $p_{\text{data}}(\mathbf{x})$. Score-matching in diffusion models can be seen as a special case of flow matching; however, as discussed in Section 3.3, flow matching circumvents certain difficulties that would otherwise arise with diffusion.

3. Method

3.1. AlphaFold as a Denoising Model

Given a protein sequence A of amino acid tokens, our objective is to model the distribution $p(\mathbf{x} \mid A)$ over 3D coordinates $\mathbf{x} \in \mathbb{R}^{3 \times N}$ which represents the structural ensemble of that protein sequence. Considering the enormous intellectual efforts that went into a *deterministic* sequence-to-structure model (i.e., AlphaFold), developing a distributional model of equivalent accuracy and generalization ability would appear to pose a considerable challenge. Our solution is to leverage recent conceptual advances in generative modeling in order to simply *repurpose* AlphaFold—nearly out of the box—as a generative model.

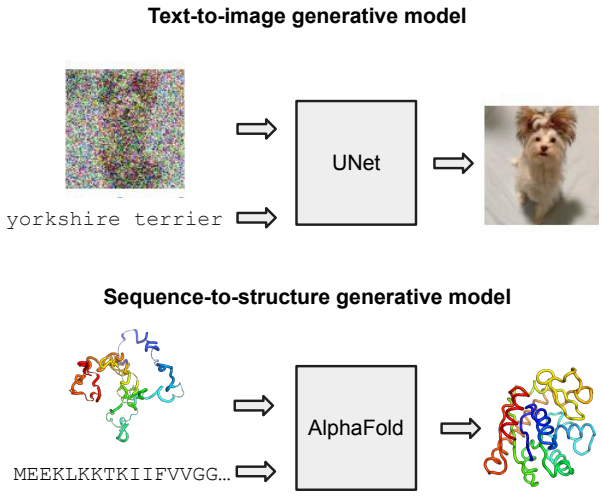

Figure 2. **AlphaFold as a denoising model.** Just as (diffusion-based) text-to-image generative models are simply neural networks that denoise images (with text input), a modified AlphaFold that ingests noisy structures and predicts clean structures (with sequence input) immediately provides a sequence-to-structure generative model—when trained under an appropriate framework.

Consider, for example, the (simplified) architecture of prototypical text-to-image diffusion models (Ho et al., 2020; Rombach et al., 2022), which aim to model conditional distributions $p(\mathbf{x} \mid s)$ of images $\mathbf{x}$ conditioned on text prompt s. At the heart of these models lies a *denoising neural network* (e.g., a UNet) which ingests a noisy image, along with a text prompt, to predict a clean image. Conditioned on these inputs, such models are otherwise are trained with simple, regression-like MSE objectives. Analogously, a protein structure predictor trained on a regression-like loss—like AlphaFold or ESMFold—can be converted to a denoising model simply by supplying an additional, *noisy* structure input (Figure 2). Not coincidentally, this is reminiscent of the idea of *template structures* employed by certain AlphaFold workflows. Thus, we develop an input embedding module very similar to AlphaFold's template embedding stack and prepend it to the pairwise folding trunks of AlphaFold and ESMFold (details in Appendix A.1). By doing so, we obtain structure denoising architectures that are thin wrappers around well-validated single-structure predictors.

With these architectural modifications, we are ready to plug AlphaFold and ESMFold into any iterative denoising-based generative modeling framework. Next, we will see how this concretely applies to flow matching for protein ensembles.

3.2. Flow Matching for Protein Ensembles

Designing a flow-matching generative framework amounts

to the choice of a conditional probability path $p_t(\mathbf{x} \mid \mathbf{x}_1)$ and its corresponding vector field $u_t(\mathbf{x} \mid \mathbf{x}_1)$. Inspired by the interpolant-based perspective on flow matching (Albergo & Vanden-Eijnden, 2022), we define the conditional probability path by sampling noise $\mathbf{x}_0$ from the prior $q(\mathbf{x}_0)$ and interpolating linearly with the data point $\mathbf{x}_1$:

$$\mathbf{x} \mid \mathbf{x}_1, t = (1-t) \cdot \mathbf{x}_0 + t \cdot \mathbf{x}_1, \quad \mathbf{x}_0 \sim q(\mathbf{x}_0) \tag{2}$$

This probability path is associated with the vector field

$$u_t(\mathbf{x} \mid \mathbf{x}_1) = (\mathbf{x}_1 - \mathbf{x})/(1-t) \tag{3}$$

which matches the CondOT path and field proposed in (for example) Pooladian et al. (2023). Customarily, we then learn a neural network to approximate the marginal vector field according to Equation 1. However, if we instead define a neural network $\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta)$ and reparameterize via

$$\hat{v}(\mathbf{x}, t; \theta) = (\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) - \mathbf{x})/(1-t) \tag{4}$$

then rearrangements of Equations 1 and 4 reveal that we can equivalently learn the expectation of $\mathbf{x}_1$:

$$\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) \approx \mathbb{E}_{\mathbf{x}_1 \sim p_t(\mathbf{x}_1 \mid \mathbf{x})}[\mathbf{x}_1] \tag{5}$$

This reparameterization is identical—up to the choice of probability path $p_t(\mathbf{x}_1 \mid \mathbf{x})$—to that employed for image diffusion models (Ho et al., 2020). In our setting, since $\mathbf{x}_1$ refers to samples from the data distribution (i.e., protein structures), this allows the AlphaFold-based architectures discussed previously to be immediately used as the the denoising model $\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta)$, with $\mathbf{x}$ as the noisy input and t as an additional time embedding.

To apply flow matching to protein structures, we describe a structure by the 3D coordinates of its β-carbons (α-carbon for glycine): $\mathbf{x} \in \mathbb{R}^{N \times 3}$. (We choose β-carbons because these are the inputs to the template embedding stack.) We then define the prior distribution $q(\mathbf{x})$ over the positions of these β-carbons to be a *harmonic prior* (Jing et al., 2023):

$$q(\mathbf{x}) \propto \exp\left[-\frac{\alpha}{2} \sum_{i=1}^{N-1} \|\mathbf{x}_i - \mathbf{x}_{i+1}\|^2\right] \tag{6}$$

This prior ensures that samples along the conditional probability path, and hence inputs to the neural network, always remain polymer-like, physically plausible 3D structures.

The parameterization of learning the conditional expectation of $\mathbf{x}_1$ (Equation 5) suggests that the neural network should be trained with an MSE loss. However, there are several issues with this direct approach. (1) The structure prediction networks not only predict β-carbon coordinates, but also all-atom coordinates and residue frames. (2) The input to the network is $SE(3)$-invariant by design, which makes training with MSE loss unsuitable without further correction (Appendix A.2). Finally, (3) the networks obtain best performance (and were orginally trained) with the $SE(3)$-invariant Frame Aligned Point Error (FAPE) loss. To reconcile these issues with the flow-matching framework, we redefine the space of protein structures to be the *quotient* space $\mathbb{R}^{3\times N}/SE(3)$, with the prior distribution projected to this space. We redefine the interpolation between two points in this space to be linear interpolation in $\mathbb{R}^3$ after RMSD-alignment. Further, because the quotient space is no longer a vector space, there is no longer a notion of "expectation" of a distribution; instead, we aim to learn the more general Fréchet mean of the conditional distribution $p(\mathbf{x}_1 \mid \mathbf{x})$:

$$\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) \approx \min_{\hat{\mathbf{x}}_1} \mathbb{E}_{\mathbf{x}_1 \sim p_t(\mathbf{x}_1 \mid \mathbf{x})}\left[\mathrm{FAPE}^2(\mathbf{x}_1, \hat{\mathbf{x}}_1)\right] \tag{7}$$

where we leverage the property that FAPE is a valid metric (Jumper et al., 2021) to define a Fréchet mean. To learn this target, we use a training loss identical to the original FAPE, except now *squared*. The final result for the training and inference procedures are provided in Algorithms 1 and 2. An important implication of this modified framework is that while our model is faithfully supervised on all-atom coordinates, it technically is learning the *distribution* only over β-carbon coordinates. These procedures and their subtleties are more fully discussed in Appendix A.2.

Algorithm 1 TRAINING

Input: Training examples of structures, sequences, and MSAs $\{(S_i, A_i, M_i)\}$
for all (S_i, A_i, M_i) **do**
  Extract $\mathbf{x}_1 \leftarrow \mathrm{BetaCarbons}(S_i)$
  Sample $\mathbf{x}_0 \sim \mathrm{HarmonicPrior}(\mathrm{length}(A_i))$
  Align $\mathbf{x}_0 \leftarrow \mathrm{RMSDAlign}(\mathbf{x}_0, \mathbf{x}_1)$
  Sample $t \sim \mathrm{Uniform}[0, 1]$
  Interpolate $\mathbf{x}_t \leftarrow t \cdot \mathbf{x}_1 + (1-t) \cdot \mathbf{x}_0$
  Predict $\hat{S}_i \leftarrow \mathrm{AlphaFold}(A_i, M_i, \mathbf{x}_t, t)$
  Optimize loss $\mathcal{L} = \mathrm{FAPE}^2(\hat{S}_i, S_i)$
end for

Algorithm 2 INFERENCE

Input: Sequence and MSA (A, M)
Output: Sampled all-atom structure $\hat{S}$
Sample $\mathbf{x}_0 \sim \mathrm{HarmonicPrior}(\mathrm{length}(A))$
for $n \leftarrow 0$ to $N-1$ **do**
  Let $t \leftarrow n/N$ and $s \leftarrow t + 1/N$
  Predict $\hat{S} \leftarrow \mathrm{AlphaFold}(A, M, \mathbf{x}_t, t)$
  if $n = N-1$ **then**
    return $\hat{S}$
  end if
  Extract $\hat{\mathbf{x}}_1 \leftarrow \mathrm{BetaCarbons}(\hat{S})$
  Align $\mathbf{x}_t \leftarrow \mathrm{RMSDAlign}(\mathbf{x}_t, \hat{\mathbf{x}}_1)$
  Interpolate $\mathbf{x}_s \leftarrow \frac{s-t}{1-t} \cdot \hat{\mathbf{x}}_1 + \frac{1-s}{1-t} \cdot \mathbf{x}_t$
end for

3.3. Comparison with Diffusion

Since our flow matching framework involves defining and reversing a noising process, it bears a number of similarities with harmonic diffusion for protein structures (Jing et al., 2023), which converges to the same prior distribution. However, as a more general framework, flow matching offers two key advantages. First, harmonic diffusion converges to the prior distribution only in the infinite-time limit, and at a rate that *depends on the data dimensionality*, i.e., protein size. This causes inference-time distributional shifts when training only on crops of relatively small size, as is the case with AlphaFold and ESMFold. On the other hand, in flow matching, the prior distribution is imposed as a boundary condition at time $t = 0$ for all dimensionalities. Second, flow matching provides an easy means to deal with missing (gap) residues—which are very common in the PDB—by simply omitting them in the interpolation. In contrast, harmonic diffusion induces dependencies across atomic positions and hence requires data imputation for missing residues. We discuss these aspects (with additional theoretical results) further in Appendix A.3.

4. Experiments

4.1. Training Regimen

We fine-tune all weights of AlphaFold and ESMFold on the PDB with our flow matching framework, starting from their publicly available pretrained weights. We use OpenFold (Ahdritz et al., 2022) for the architecture implementation and training pipeline and OpenProteinSet (Ahdritz et al., 2023) for training MSAs. Adhering to the original works, we use a training cutoff of May 1, 2018 and May 1, 2020 for AlphaFold and ESMFold, respectively. At the conclusion of this stage of training (1.28M and 720k examples, respectively), we obtain flow-matching variants of AlphaFold and ESMFold which we call **AlphaFLOW** and **ESMFLOW**.

Next, to demonstrate and assess the ability of our method to learn from MD ensembles, we continue fine-tuning both models on the ATLAS dataset of all-atom MD simulations (Vander Meersche et al., 2023), which consists of 1390 proteins chosen for structural diversity by ECOD domain classification (Schaeffer et al., 2017). Using training and validation cutoffs of May 1, 2018 and May 1, 2019, we obtain train/val/test splits of 1265/39/82 ensembles (2 excluded due to length). After 43k and 27k additional training examples, respectively, we obtain MD-specialized variants of our model which we call **AlphaFLOW-MD** and **ESMFLOW-MD**. We also train variants of these models (**+Templates**) which accept the PDB structure that initialized the simulation as input using a copy of the input embedding module.

Because flow matching is an iterative generative process, sampling a single structure requires many forward passes of AlphaFold and ESMFold (up to 10 in our experiments), which can be somewhat expensive. To accelerate this process, we explore variants of all models where the generative process is *distilled* into a single forward pass (details in Appendix B.1). While distillation has been explored for diffusion models (Salimans & Ho, 2022; Song et al., 2023; Yin et al., 2023), this is (to our knowledge) the first demonstration of distillation in a protein or flow-matching setting.

4.2. PDB Ensembles

We first examine the ability of AlphaFLOW and ESMFLOW to sample diverse conformations of proteins deposited in the Protein Data Bank (PDB). To do, we construct a test set of 100 proteins deposited after the AlphaFold training cutoff (May 1, 2018) with multiple chains and evidence of conformational heterogeneity (details in Appendix B.2). For each protein, we sample 50 predictions with (1) unmodified AlphaFold/ESMFold (2) AlphaFold with varying degrees of MSA subsampling and (3) AlphaFLOW/ESMFLOW, with varying degrees of flow truncation in order to tune the amount of diversity (Appendix B.1). Each set of predictions is evaluated on three metrics: **precision**—the average lDDT$_{C\alpha}$ from each prediction to the closest crystal structure; **recall**—the average lDDT$_{C\alpha}$ from each crystal structure to the closest prediction; and **diversity**—the average dissimilarity (1-lDDT$_{C\alpha}$) between pairs of predicted structures.

The median results across the 100 test targets are shown in Figure 3. AlphaFLOW, similar to MSA subsampling, increases the prediction diversity relative to the unmodified AlphaFold at the cost of reduced precision. However, the variants of AlphaFLOW trace a substantially superior Pareto frontier relative to MSA subsampling. In some cases, PCA of the ground truth and predicted ensembles (Appendix C.1) offers an explanation for this result: in MSA subsampling, the ensembles drift away from the true structures as the input signal is ablated, whereas the AlphaFLOW predictions remain clustered around the ground truth conformations while reaching the same or greater levels of diversity. In terms of precision and recall, AlphaFLOW exhibits very similar behavior to MSA subsampling. Somewhat surprisingly, neither method is able to meaningfully improve aggregate recall relative to baseline AlphaFold, showing that they generally do not succeed in increasing the coverage of experimentally determined PDB structures, or (more optimistically) that the predicted conformational changes have yet to be experimentally observed. Selected cases of conformational changes successfully modeled by AlphaFLOW are visualized in Appendix C.1; Figure 8.

Overall (as expected), ESMFold and ESMFLOW exhibit reduced precision relative to AlphaFold-family methods. However, ESMFLOW is able to inject substantial diversity relative to baseline ESMFold—which, unlike AlphaFold, is

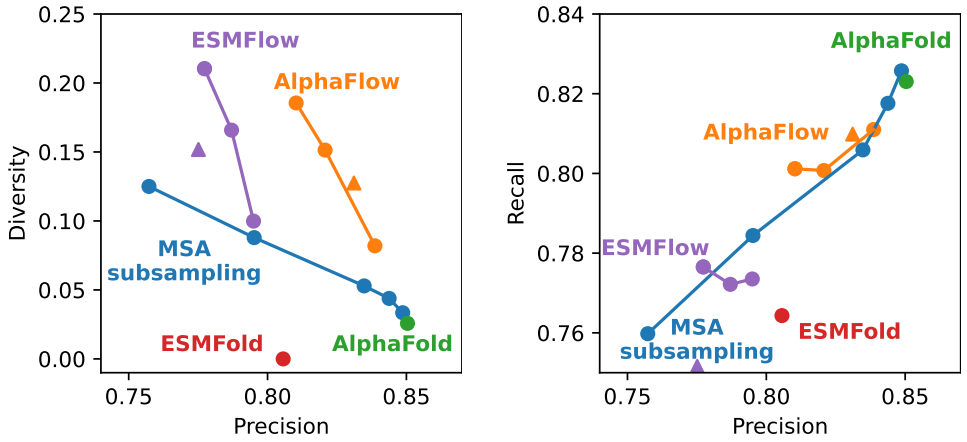

Figure 3. **Evaluation on PDB ensembles**—precision-diversity (*left*) and precision-recall (*right*) curves for all benchmarked methods (median taken over 100 test targets). The MSA subsampling curve is traced by reducing MSA depth (max 512, min 48) and joins to AlphaFold as they share the same weights (AlphaFold by default subsamples MSAs to a maximum depth of 1024 and thus has nonzero diveristy, unlike ESMFold). The AlphaFLOW / ESMFLOW curves are traced by truncating the initial steps of flow matching (described in Appendix B.1). Distilled models are marked by ▲. Tabular data is shown in Appendix C.1, Table 3

completely deterministic—and increase the recall at little to no cost in precision. Note that this test set includes some proteins deposited before the ESMFold cutoff; results on a later sub-split are similar (Appendix C.1; Table 3).

4.3. Molecular Dynamics Ensembles

We next evaluate the ability of AlphaFLOW and ESMFLOW to generate proxy MD ensembles for the 82 test proteins in the ATLAS database. These test proteins have minimal structural overlap with the training ensembles, providing a stringent test of generalization. For each target, we sample 250 predictions with each method and probe their similarity to the MD ensembles via a series of assessments, grouped under three broad categories of increasing difficulty: (1) predicting flexibility, (2) distributional accuracy, and (3) ensemble observables. Unless otherwise noted, we focus on AlphaFLOW ensembles generated with MSA input alone (i.e., no PDB templates). Main results are presented in Table 1 and Figure 5; further results (e.g. ESMFLOW, comparisons with normal mode analysis, and ablations) and ensemble visualizations can be found in Appendix C.2. We note that our evaluations are inherently limited to phenomena accessible within the ATLAS simulation timescales; we do not assess if our model captures slower conformational changes, which remain a key area for future work.

Q1: Is ensemble flexibility predictive of true protein flexibility? For each ensemble, we quantify the protein flexibility as the average Cα-RMSD between any pair of conformations. By this metric, the AlphaFLOW ensembles have the strongest Pearson correlation with the ground truth and matches the aggregate level of diversity in the MD ensembles. In contrast, MSA subsampling is unable to reach the same level of diversity while retaining any predictive power. Similar results hold when considering atomic-level flexibility in terms of root mean square fluctuation (RMSF), both when pooled globally and pooled per-target. Remarkably, AlphaFLOW attains a median Pearson correlation of 0.85 between modeled and predicted RMSFs within a target, while no level of MSA subsampling is able to meaningfully exceed baseline AlphaFold on this metric.

Q2: Are the atomic positions distributionally accurate? To generalize the all-atom RMSD metric to ensembles, define the *root mean Wasserstein distance* (RMWD) between ensembles $\mathcal{X}, \mathcal{Y}$ as

$$\mathrm{RMWD}(\mathcal{X}, \mathcal{Y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \mathcal{W}_2^2\left(\mathcal{N}[\mathcal{X}_i], \mathcal{N}[\mathcal{Y}_i]\right)} \quad (8)$$

where $\mathcal{N}[\mathcal{X}_i]$ are 3D-Gaussians fit to the positional distribution of the ith atom in ensemble $\mathcal{X}$ (this reduces to RMSD with a single structure). By this metric, AlphaFLOW ensembles are more accurate than any level of MSA subsampling. Decomposition of the RMWD into a translation contribution and variance contribution (Appendix B.3) reveals that AlphaFLOW slightly improves on AlphaFold in predicting the mean position of atoms, and substantially outperforms MSA subsampling in modeling the variance.

The joint distribution of Cα positions reveals collective motions and provides a more stringent test of distributional accuracy. We project this joint distribution onto the first two principal components from PCA—computed from the MD ensemble alone or from equally weighting the MD and predicted ensembles—and compute the $\mathcal{W}_2$-distance (in units of Å RMSD) between the predicted and true ensembles in this space. We also compute the (unsigned) *cosine similarity* between the top principal components of the predicted and true ensembles and consider the dominant motion to be successfully modeled if this similarity > 0.5. By all of

Table 1. **Evaluation on MD ensembles**. For each method, we compare the predicted ensemble with the ground truth MD ensemble according to various metrics, detailed in the main text. For protein flexibility and RMSF, the ground truth values (from the MD ensembles) are in parenthesis. When applicable, the median across the 82 test ensembles is reported. See Appendix C.2 for ESMFLOW results. r: Pearson correlation; ρ: Spearman correlation; J: Jaccard similarity; $\mathcal{W}_2$: 2-Wasserstein distance.

		AlphaFLOW-MD		MSA subsampling					AFMD+Templates	
		Full	Distilled	32	48	64	256	AlphaFold	Full	Distilled
Predicting flexibility	Pairwise RMSD (= 2.90)	**2.89**	1.94	4.40	2.34	1.67	0.72	0.58	2.18	1.73
	Pairwise RMSD r ↑	**0.48**	**0.48**	0.03	0.12	0.22	0.15	0.10	0.94	0.92
	All-atom RMSF (=1.70)	**1.68**	1.28	5.38	2.29	1.17	0.49	0.31	1.31	1.00
	Global RMSF r	**0.60**	0.54	0.13	0.23	0.29	0.26	0.21	0.91	0.89
	Per-target RMSF r	**0.85**	0.81	0.51	0.52	0.51	0.55	0.52	0.90	0.88
Distributional accuracy	Root mean $\mathcal{W}_2$-dist. ↓	**2.61**	3.70	6.15	5.32	4.28	3.62	3.58	1.95	2.18
	↪ Translation contrib. ↓	**2.28**	3.10	5.22	3.92	3.33	2.87	2.86	1.64	1.74
	↪ Variance contrib. ↓	**1.30**	1.52	3.55	2.49	2.24	2.24	2.27	1.01	1.25
	MD PCA $\mathcal{W}_2$-dist. ↓	**1.52**	1.73	2.44	2.30	2.23	1.88	1.99	1.25	1.41
	Joint PCA $\mathcal{W}_2$-dist. ↓	**2.25**	3.05	5.51	4.51	3.57	3.02	2.86	1.58	1.68
	% PC-sim > 0.5 ↑	**44**	34	15	18	21	21	23	44	43
Ensemble observables	Weak contacts J ↑	**0.62**	0.52	0.40	0.40	0.37	0.30	0.27	0.62	0.51
	Transient contacts J ↑	**0.41**	0.28	0.23	0.26	0.27	0.27	0.28	0.47	0.42
	Exposed residue J ↑	**0.50**	0.48	0.34	0.37	0.37	0.33	0.32	0.50	0.47
	Exposed MI matrix ρ ↑	**0.25**	0.14	0.14	0.11	0.10	0.06	0.02	0.25	0.18

these metrics, AlphaFLOW markedly improves over MSA subsampling, and in particular nearly doubles the success rate for obtaining > 0.5 cosine similarity.

Q3: Are complex ensemble observables faithfully reproduced? MD ensembles are often intended for downstream analysis of observables such as intermittent contacts and solvent exposure, often associated with thermal fluctuations around the low-energy crystal structure (Vögele et al., 2022). To probe if we model these properties accurately, for each ensemble we identify the set of *weak contacts* and *transient contacts*, defined as those Cα pairs which are in contact (respectively, not in contact) in the crystal structure but which dissociate (respectively, associate) in $> 10\%$ of ensemble structures, with a 8 Å threshold. We then compute the the Jaccard similarity of the sets produced by each method with the ground truth sets. We repeat the same analysis with the set of *cryptically exposed residues*—those whose sidechains are buried in the crystal structure but exposed to solvent in $> 10\%$ of ensemble structures—which are a key feature in the identification of cryptic pockets in drug discovery (Meller et al., 2023). Going further, for each pair of residues we compute the *mutual information* (MI) between their (binary) exposure states, yielding a MI matrix for each ensemble. Such matrices are an important in the so-called *exposon analysis* of protein dynamics, e.g., for collective motions and allostery (Porter et al., 2019). We then compute the Spearman correlation between the values of MI matrices from the MD and generated ensembles. Impressively, for all of these analyses, AlphaFLOW substantially outperforms MSA subsampling; we emphasize that these are complex properties to emulate involving sidechains and different parts of the protein (Figure 5 and Appendix C.2).

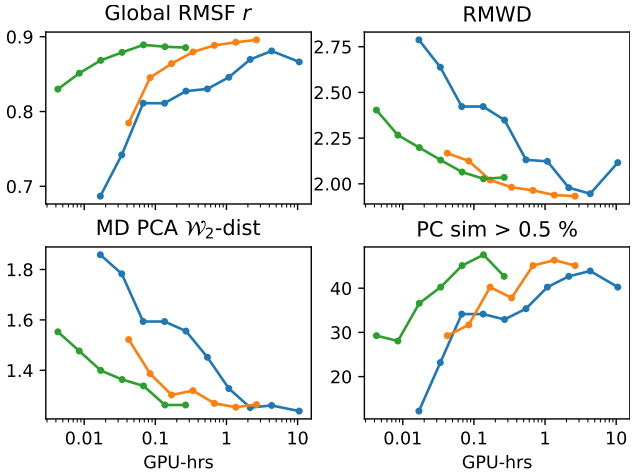

Figure 4. **Efficiency of AlphaFLOW vs replicate MD simulations.** AlphaFLOW+Templates with varying number of samples with distillation (green) and without distillation (orange); MD with varying trajectory lengths in blue. See Appendix B.3 for further experimental details and Appendix C.2 for further results.

Diversifying solved structures. Although we have so far focused on generating protein ensembles without the use of experimental structures, there is substantial scientific interest in obtaining ensembles for specific *solved* structures, often via molecular dynamics simulation (Hollingsworth & Dror, 2018). To investigate the utility of our method in this application setting, we repeat all experiments by providing the structure which initialized the ATLAS simulations to the +Templates version of AlphaFLOW-MD. As expected, the resulting ensembles improve—sometimes substantially—in their similarity to the ground truth MD ensemble, vastly surpasssing the performance of MSA subsampling. How-

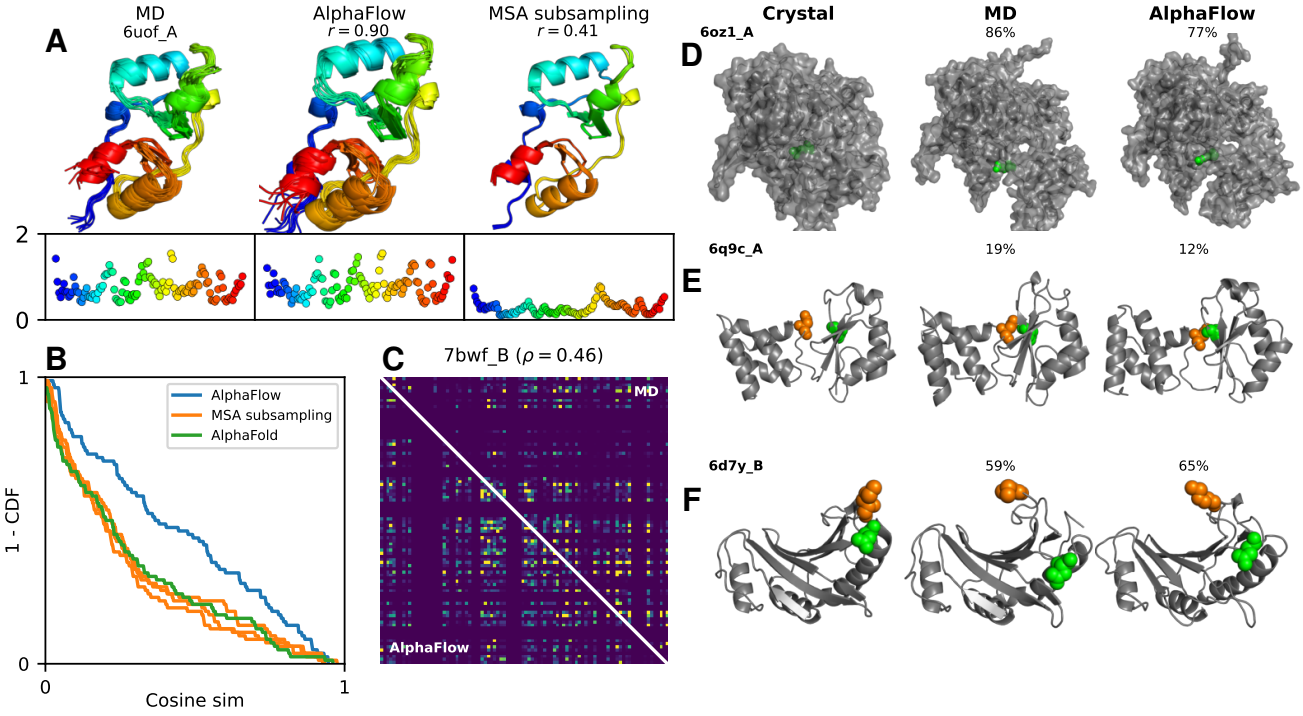

Figure 5. **MD evaluations visualized.** (**A**) Ensembles of PDB ID `6uof_A` (transcriptional regulator from *Streptococcus pneumoniae*) from MD, AlphaFLOW, and MSA subsampling (depth 48), with Cα RMSF by residue index shown in insets. (**B**) $1 - CDF$ of the distribution of (unsigned) cosine similarities between the top principal components of the predicted ensemble versus the MD ensemble. (**C**) Solvent exposure mutual information matrices computed from the ground truth MD ensemble and AlphaFLOW ensemble for target PDB ID `7bwf_B` (antitoxin from *Staphylococcus aureus*). (**D, E, F**) Deviations from the crystal structure in the MD simulation, corresponding to ensemble observables, which are correctly sampled by AlphaFLOW. The probability of occurence in each ensembles is shown. (**D**) solvent exposure of a buried residue in PDB ID `6oz1_A` (carboxylate reductase from *M. chelonae*). (**E**) association of a transient residue contact in PDB ID `6q9c_A` (NADH-quinone oxidoreductase subunit E from *Aquifex aeolicus*). (**F**) dissociation of a weak residue contact in PDB ID `6d7y_B` (immune protein from *Enterobacter cloacae*). Additional examples in Appendix C.2 r: Pearson correlation; ρ: Spearman correlation.

ever, in these settings, the appropriate baseline is *replicate simulations* (provided in ATLAS) starting from the same structure rather than MSA subsampling. Since MD is taken to be the ground truth but is expensive to run to convergence, we investigate if AlphaFLOW provides better results for an equivalent *limited computational budget*, e.g., in terms of GPU-hrs. To emulate these budgets, we reduce the number of samples drawn from AlphaFLOW (from 250 to as few as 4) and the length of the MD trajectory (100 ns–160 ps). As shown in Figure 4, the AlphaFLOW ensembles retain much of their quality with up to a 10x reduction in samples, while the MD trajectories require much longer to converge to or surpass the same quality. The distilled AlphaFLOW model, despite converging to a lower level of performance, provides an even greater improvement for short timescales by providing 10x as many samples for the same runtime. Thus, as measured by these metrics, AlphaFLOW provides a more efficient means to study the thermodynamic fluctuations of existing solved structures than short MD simulations and holds promise for large-scale diversification of the PDB.

5. Conclusion

We have presented AlphaFLOW and ESMFLOW, which combine AlphaFold and ESMFold with flow-matching towards the goal of sampling protein ensembles. Compared to existing approaches for obtaining multiple structure predictions, our method goes beyond *inference-time* input modifications and develops a more principled *training-time* approach to modeling structural diversity. Comprehensive experimental results demonstrate the utility and performance of our method in predicting precise and diverse PDB structures and replicating distributions and properties of MD ensembles, both with and without initial experimental structures. We anticipate these capabilities to have broad and exciting applications for structure biology. Further, with the increasing availability of high-resolution cryo-EM data (Kühlbrandt, 2014) and algorithms for resolving their structural heterogeneity (Zhong et al., 2021), we anticipate the paradigm of generative training of AlphaFold and ESMFold to have further applications beyond the settings considered here.

Acknowledgements

We thank Ruochi Zhang, Hannes Stärk, Samuel Sledzieski, Soojung Yang, Jason Yim, Rachel Wu, Hannah Wayment-Steele, Umesh Padia, Sergey Ovchinnikov, Andrew Campbell, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Martin Vögele, Daniel Richman, Jessica Karaguesian, Alex Powers, and anonymous ICML reviewers for helpful feedback and discussions.

This work was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number 1R35GM141861-01 and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0022158. We acknowledge support from NSF Expeditions grant (award 1918839: Collaborative Research: Understanding the World Through Code), the Machine Learning for Pharmaceutical Discovery and Synthesis (MLPDS) consortium, the Abdul Latif Jameel Clinic for Machine Learning in Health, the DTRA Discovery of Medical Countermeasures Against New and Emerging (DOMANE) threats program, and the DARPA Accelerated Molecular Discovery program. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC awards ASCR-ERCAP0027302 and ASCR-ERCAP0027818.

Impact Statement

Our work focuses on modeling the structures of proteins and could forseeably impact many biotechnology applications, such as drug discovery and protein design, in addition to advancing basic scientific research. The most likely outcomes of these applications are positive ones in human health, energy, and economic development. As with any biotechnology, however, negative applications are also possible and merit prudent discussion as appropriate.

References

Abdin, O. and Kim, P. M. Pepflow: direct conformational sampling from peptide energy landscapes through hypernetwork-conditioned diffusion. *bioRxiv*, pp. 2023–06, 2023.

Ahdritz, G., Bouatta, N., Kadyan, S., Xia, Q., Gerecke, W., O'Donnell, T. J., Berenberg, D., Fisk, I., Zanichelli, N., Zhang, B., et al. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *bioRxiv*, pp. 2022–11, 2022.

Ahdritz, G., Bouatta, N., Kadyan, S., Jarosch, L., Berenberg, D., Fisk, I., Watkins, A. M., Ra, S., Bonneau, R., and AlQuraishi, M. Openproteinset: Training data for structural biology at scale. *arXiv preprint arXiv:2308.05326*, 2023.

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2022.

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Arts, M., Garcia Satorras, V., Huang, C.-W., Zugner, D., Federici, M., Clementi, C., Noé, F., Pinsler, R., and van den Berg, R. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023.

Atilgan, A. R., Durell, S., Jernigan, R. L., Demirel, M. C., Keskin, O., and Bahar, I. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophysical journal*, 80(1):505–515, 2001.

Bahar, I., Atilgan, A. R., and Erman, B. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design*, 2(3):173–181, 1997.

Bakan, A., Meireles, L. M., and Bahar, I. Prody: protein dynamics inferred from theory and experiments. *Bioinformatics*, 27(11):1575–1577, 2011.

Bose, A. J., Akhound-Sadegh, T., Fatras, K., Huguet, G., Rector-Brooks, J., Liu, C.-H., Nica, A. C., Korablyov, M., Bronstein, M., and Tong, A. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

Brotzakis, Z. F., Zhang, S., and Vendruscolo, M. Alphafold prediction of structural ensembles of disordered proteins. *bioRxiv*, pp. 2023–01, 2023.

Bryant, P. Structure prediction of alternative protein conformations. *bioRxiv*, pp. 2023–09, 2023.

Casadevall, G., Duran, C., and Osuna, S. Alphafold2 and deep learning for elucidating enzyme conformational flexibility and its application for design. *JACS Au*, 3(6):1554–1562, 2023.

Chakravarty, D. and Porter, L. L. Alphafold2 fails to predict protein fold switching. *Protein Science*, 31(6):e4353, 2022.

Chen, R. T. and Lipman, Y. Riemannian flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

Chen, T., Zhang, R., and Hinton, G. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

da Silva, G. M., Cui, J. Y., Dalgarno, D. C., Lisi, G. P., and Rubenstein, B. M. Predicting relative populations of protein conformations without a physics engine using alphafold2. *bioRxiv*, 2023.

Dana, J. M., Gutmanas, A., Tyagi, N., Qi, G., O'Donovan, C., Martin, M., and Velankar, S. Sifts: updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic acids research*, 47(D1):D482–D489, 2019.

Del Alamo, D., Sala, D., Mchaourab, H. S., and Meiler, J. Sampling alternative conformational states of transporters and receptors with alphafold2. *Elife*, 11:e75751, 2022.

Diepeveen, W., Esteve-Yagüe, C., Lellmann, J., Öktem, O., and Schönlieb, C.-B. Riemannian geometry for efficient analysis of protein dynamics data. *arXiv preprint arXiv:2308.07818*, 2023.

Ellaway, J. I., Anyango, S., Nair, S., Zaki, H. A., Nadzirin, N., Powell, H. R., Gutmanas, A., Varadi, M., and Velankar, S. Identifying protein conformational states in the pdb and comparison to alphafold2 predictions. *bioRxiv*, pp. 2023–07, 2023.

Faezov, B. and Dunbrack Jr, R. L. Alphafold2 models of the active form of all 437 catalytically-competent typical human kinase domains. *bioRxiv*, pp. 2023–07, 2023.

Felardos, L., Hénin, J., and Charpiat, G. Designing losses for data-free training of normalizing flows on boltzmann distributions. *arXiv preprint arXiv:2301.05475*, 2023.

Herrington, N. B., Stein, D., Li, Y. C., Pandey, G., and Schlessinger, A. Exploring the druggable conformational space of protein kinases using ai-generated structures. *bioRxiv*, pp. 2023–08, 2023.

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

Hollingsworth, S. A. and Dror, R. O. Molecular dynamics simulation for all. *Neuron*, 99(6):1129–1143, 2018.

Jing, B., Erives, E., Pao-Huang, P., Corso, G., Berger, B., and Jaakkola, T. S. Eigenfold: Generative protein structure prediction with diffusion models. In *ICLR 2023-Machine Learning for Drug Discovery workshop*, 2023.

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.

Köhler, J., Krämer, A., and Noé, F. Smooth normalizing flows. *Advances in Neural Information Processing Systems*, 34:2796–2809, 2021.

Kühlbrandt, W. The resolution revolution. *Science*, 343 (6178):1443–1444, 2014.

Lane, T. J. Protein structure prediction has reached the single-structure frontier. *Nature Methods*, 20(2):170–173, 2023.

Larkin, M. A., Blackshields, G., Brown, N. P., Chenna, R., McGettigan, P. A., McWilliam, H., Valentin, F., Wallace, I. M., Wilm, A., Lopez, R., et al. Clustal w and clustal x version 2.0. *bioinformatics*, 23(21):2947–2948, 2007.

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

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

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.

Lu, J., Zhong, B., and Tang, J. Score-based enhanced sampling for protein molecular dynamics. *arXiv preprint arXiv:2306.03117*, 2023.

McGibbon, R. T., Beauchamp, K. A., Harrigan, M. P., Klein, C., Swails, J. M., Hernández, C. X., Schwantes, C. R., Wang, L.-P., Lane, T. J., and Pande, V. S. Mdtraj: a modern open library for the analysis of molecular dynamics trajectories. *Biophysical journal*, 109(8):1528–1532, 2015.

Meller, A., Ward, M., Borowsky, J., Kshirsagar, M., Lotthammer, J. M., Oviedo, F., Ferres, J. L., and Bowman, G. R. Predicting locations of cryptic pockets from single protein structures using the pocketminer graph neural network. *Nature Communications*, 14(1):1177, 2023.

Midgley, L. I., Stimper, V., Simm, G. N., Schölkopf, B., and Hernández-Lobato, J. M. Flow annealed importance sampling bootstrap. *arXiv preprint arXiv:2208.01893*, 2022.

Noé, F., Olsson, S., Köhler, J., and Wu, H. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

Ourmazd, A., Moffat, K., and Lattman, E. E. Structural biology is solved—now what? *Nature methods*, 19(1): 24–26, 2022.

Pooladian, A.-A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., and Chen, R. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

Porter, J. R., Moeder, K. E., Sibbald, C. A., Zimmerman, M. I., Hart, K. M., Greenberg, M. J., and Bowman, G. R. Cooperative changes in solvent exposure identify cryptic pockets, switches, and allosteric coupling. *Biophysical Journal*, 116(5):818–830, 2019.

Porter, L. L., Chakravarty, D., Schafer, J. W., and Chen, E. A. Colabfold predicts alternative protein structures from single sequences, coevolution unnecessary for afcluster. *bioRxiv*, pp. 2023–11, 2023.

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.

Saldaño, T., Escobedo, N., Marchetti, J., Zea, D. J., Mac Donagh, J., Velez Rueda, A. J., Gonik, E., García Melani, A., Novomisky Nechcoff, J., Salas, M. N., et al. Impact of protein conformational diversity on alphafold predictions. *Bioinformatics*, 38(10):2742–2748, 2022.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Schaeffer, R. D., Liao, Y., Cheng, H., and Grishin, N. V. Ecod: new developments in the evolutionary classification of domains. *Nucleic acids research*, 45(D1):D296–D302, 2017.

Shaw, D. E., Maragakis, P., Lindorff-Larsen, K., Piana, S., Dror, R. O., Eastwood, M. P., Bank, J. A., Jumper, J. M., Salmon, J. K., Shan, Y., et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330 (6002):341–346, 2010.

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.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

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.

Stein, R. A. and Mchaourab, H. S. Speach_af: Sampling protein ensembles and conformational heterogeneity with alphafold2. *PLOS Computational Biology*, 18(8):e1010483, 2022.

Stein, R. A. and Mchaourab, H. S. Rosetta energy analysis of alphafold2 models: Point mutations and conformational ensembles. *bioRxiv*, pp. 2023–09, 2023.

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33: 7537–7547, 2020.

Vander Meersche, Y., Cretin, G., Gheeraert, A., Gelly, J.-C., and Galochkina, T. Atlas: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic Acids Research*, pp. gkad1084, 2023.

Vani, B. P., Aranganathan, A., Wang, D., and Tiwary, P. Alphafold2-rave: From sequence to boltzmann ranking. *Journal of Chemical Theory and Computation*, 2023.

Vögele, M., Thomson, N. J., Truong, S. T., McAvity, J., Zachariae, U., and Dror, R. O. Systematic analysis of biomolecular conformational ensembles with pensa. *arXiv preprint arXiv:2212.02714*, 2022.

Vögele, M., Zhang, B. W., Kaindl, J., and Wang, L. Is the functional response of a receptor determined by the thermodynamics of ligand binding? *Journal of Chemical Theory and Computation*, 2023.

Wayment-Steele, H. K., Ojoawo, A., Otten, R., Apitz, J. M., Pitsawong, W., Hömberger, M., Ovchinnikov, S., Colwell, L., and Kern, D. Predicting multiple conformations via sequence clustering and alphafold2. *Nature*, pp. 1–3, 2023.

Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B., et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pp. 2022–07, 2022.

Xie, T. and Huang, J. Can protein structure prediction methods capture alternative conformations of membrane proteins? *bioRxiv*, pp. 2023–08, 2023.

Yim, J., Trippe, B. L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W. T., and Park, T. One-step diffusion with distribution matching distillation. *arXiv preprint arXiv:2311.18828*, 2023.

Zheng, S., He, J., Liu, C., Shi, Y., Lu, Z., Feng, W., Ju, F., Wang, J., Zhu, J., Min, Y., et al. Towards predicting equilibrium distributions for molecular systems with deep learning. *arXiv preprint arXiv:2306.05445*, 2023.

Zhong, E. D., Bepler, T., Berger, B., and Davis, J. H. Cryodrgn: reconstruction of heterogeneous cryo-em structures using neural networks. *Nature methods*, 18(2):176–185, 2021.

A. Method Details

A.1. Input Embedding Module

Algorithm 3 outlines the architecture of the input embedding module which we attach to AlphaFold and ESMFold to form AlphaFLOW and ESMFLOW, respectively. The output of the module is added to the input to the Evoformer or folding trunk. The various subroutines are as defined in AlphaFold (Jumper et al., 2021), whereas the Gaussian Fourier time embeddings are as previously used in Song et al. (2021); Tancik et al. (2020). For brevity, we have omitted droupout layers.

Algorithm 3 INPUTEMBEDDING

Input: Beta carbon coordinates $\mathbf{x} \in \mathbb{R}^{N\times 3}$, time $t \in [0, 1]$
Output: Input pair embedding $\mathbf{z} \in \mathbb{R}^{N\times N\times 64}$
$\mathbf{z}_{ij} \leftarrow \|\mathbf{x}_i - \mathbf{x}_j\|$
$\mathbf{z}_{ij} \leftarrow \mathrm{Bin}(\mathbf{z}_{ij}, \mathrm{min} = 3.25\ \text{Å}, \mathrm{max} = 50.75\ \text{Å}, N_{\text{bins}} = 39)$
$\mathbf{z}_{ij} \leftarrow \mathrm{Linear}(\mathrm{OneHot}(\mathbf{z}_{ij}))$
for $l \leftarrow 1$ to $N_{\text{blocks}} = 4$ **do**
  $\{\mathbf{z}\}_{ij}$ += $\mathrm{TriangleAttentionStartingNode}(\mathbf{z}_{ij}, c = 64, N_{\text{head}} = 4)$
  $\{\mathbf{z}\}_{ij}$ += $\mathrm{TriangleAttentionEndingNode}(\mathbf{z}_{ij}, c = 64, N_{\text{head}} = 4))$
  $\{\mathbf{z}\}_{ij}$ += $\mathrm{TriangleMultiplicationOutgoing}(\mathbf{z}_{ij}, c = 64)$
  $\{\mathbf{z}\}_{ij}$ += $\mathrm{TriangleMultiplicationIncoming}(\mathbf{z}_{ij}, c = 64)$
  $\{\mathbf{z}\}_{ij}$ += $\mathrm{PairTransition}(\mathbf{z}_{ij}, n = 2)$
end for
$\mathbf{z}_{ij}$ += $\mathrm{Linear}(\mathrm{GaussianFourierEmbedding}(t, d = 256))$

A.2. Flow Matching on Protein Ensembles

In this subsection, we describe how the final training and inferences Algorithms 1 and 2 are obtained, starting from the Euclidean flow matching procedure from a harmonic prior provided in Section 3.2, Equations 2–6. We note that other diffusion or flow matching formulations are also possible and leave further exploration of this design space to future work.

Unsuitability of MSE Loss In standard flow matching over $\mathbb{R}^{3N}$, the denoising network $\hat{\mathbf{x}}(\mathbf{x}, t; \theta)$ is designed to approximate $\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) \approx \mathbb{E}_{\mathbf{x}_1\sim p_t(\mathbf{x}_1|\mathbf{x})}[\mathbf{x}_1]$, which gives rise to the MSE training objective

$$\begin{aligned}\mathcal{L}_t(\theta) &= \mathbb{E}_{\mathbf{x}_1\sim p_{\text{data}}, \mathbf{x}\sim p_t(\mathbf{x}|\mathbf{x}_1)}\left[\|\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) - \mathbf{x}_1\|^2\right] \\ &= \mathbb{E}_{\mathbf{x}_1\sim p_{\text{data}}, \mathbf{x}_0\sim q}\left[\|\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) - \mathbf{x}_1\|^2\right]\end{aligned} \tag{9}$$

where $\mathbf{x} = (1 - t) \cdot \mathbf{x}_0 + t \cdot \mathbf{x}_1$, for each time $t \in [0, 1]$. The harmomic prior density q and the data distribution p_{data} are $SE(3)$-invariant (technically $SO(3)$-invariant after centering; see for example Yim et al. (2023)). This means that, for each training pair $(\mathbf{x}_0, \mathbf{x}_1)$, there is a corresponding uniform density over $R \in SO(3)$ supplying examples $(R.\mathbf{x}_0, R.\mathbf{x}_1)$:

$$\mathcal{L}_t(\theta) = \mathbb{E}_{\mathbf{x}_1\sim p_{\text{data}}, \mathbf{x}_0\sim q}\left[\int_{SO(3)} \|\hat{\mathbf{x}}_1(R.\mathbf{x}, t; \theta) - R.\mathbf{x}_1\|^2 \, dR\right] \tag{10}$$

However, because the input embedding takes only a distogram of $\mathbf{x}$, the denoising model $\hat{\mathbf{x}}_1$, i.e., AlphaFold or ESMFold, is $SE(3)$-invariant, meaning that

$$\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) = \hat{\mathbf{x}}_1(R.\mathbf{x}, t; \theta) \tag{11}$$

for any $R \in SO(3) \subset SE(3)$. Hence, the denoising network is tasked with predicting $R.\mathbf{x}_1$ despite having no access to R. This is impossible and would lead to the network to degenerately predict $\hat{\mathbf{x}}_1 = \mathbf{0}$, showing that the MSE loss (or more broadly any non-$SE(3)$-invariant) loss is unsuitable with a $SE(3)$-invariant denoising network.

Flow Matching on the Quotient Space To resolve the issue that AlphaFold and ESMFold are insensitive to $SE(3)$ transformations of the input, we consider flow matching over the quotient space $\mathbb{R}^{3N}/SE(3)$, such that inputs related by $SE(3)$ transformations are now defined to be identical. This quotient space, when defined with suitable care (Diepeveen

et al., 2023), gives a non-Euclidean, Riemannian manifold. The harmonic prior and data distribution can be straightforwardly projected to this space by taking the $SE(3)$ equivalency classes of each data point. The theory of flow matching over Riemmanian manifolds was developed by Chen & Lipman (2023) and closely follows standard flow matching, except the conditional vector fields and the learned marginal vector fields are elements of the tangent space:

$$u_t(\mathbf{x} \mid \mathbf{x}_1) \in T_{\mathbf{x}}\mathcal{M}, \quad \hat{v}(\mathbf{x}, t; \theta) := \mathbb{E}_{\mathbf{x}_1 \sim p_t(\mathbf{x}_1 \mid \mathbf{x})}[u_t(\mathbf{x} \mid \mathbf{x}_1)] \in T_{\mathbf{x}}\mathcal{M} \tag{12}$$

As in the Euclidean case, to develop a flow matching process, we require a conditional probability path and a corresponding conditional vector field. Chen & Lipman (2023) propose to generalize the CondOT probability path by defining the interpolant $\psi_t(\mathbf{x}_0 \mid \mathbf{x}_1)$ to be the geodesic from $\mathbf{x}_0$ to $\mathbf{x}_1$, and then specifying $p_t(\mathbf{x} \mid \mathbf{x}_0)$ via

$$\mathbf{x} \mid \mathbf{x}_1 = \psi_t(\mathbf{x}_0 \mid \mathbf{x}_1), \quad \mathbf{x}_0 \sim q(\mathbf{x}_0) \tag{13}$$

and the associated conditional vector field as

$$u_t(\mathbf{x} \mid \mathbf{x}_1) = \frac{d}{dt}\psi_t(\mathbf{x}_0 \mid \mathbf{x}_1) \tag{14}$$

Once the marginal vector field is learned, inference is performed by integrating the corresponding ODE over the manifold. To use this framework with protein structures and AlphaFold or ESMFold as the flow model, we make the following tweaks:

(1) We construct the interpolation between two elements in the quotient space $\mathbb{R}^{3N}/SE(3)$ to be given by RMSD alignment in the ambient space $\mathbb{R}^{3N}$, followed by linear interpolation in ambient space. Thus, as employed in Algorithm 1, the conditional probability path is sampled via

$$\begin{aligned} \mathbf{x}_0 &\sim q(\mathbf{x}_0) \\ \mathbf{x}_0 &\leftarrow \mathrm{RMSDAlign}(\mathbf{x}_0, \mathbf{x}_1) \\ \mathbf{x} \mid \mathbf{x}_1 &= (1-t) \cdot \mathbf{x}_0 + t \cdot \mathbf{x}_1 \end{aligned} \tag{15}$$

(2) Similar to the Euclidean case, we consider a reparameterization (cf. Equation 4) which allows a denoising model $\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta)$ such as AlphaFold or ESMFold to give the direction of the learned marginal flow:

$$\hat{v}(\mathbf{x}, t; \theta) = \frac{\log_{\mathbf{x}} \hat{\mathbf{x}}_1(\mathbf{x}, t; \theta)}{1-t} \tag{16}$$

where the logarithmic map gives the direction of the interpolation connecting $\mathbf{x}$ to $\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta)$ (discussed next). Unlike the Euclidean case, however, this expression does not provide a simple training objective for $\mathbf{x}_1$ in terms of a denoising loss. This is because flow matching requires minimizing error in the tangent space, which may not be easily related to distances on the manifold. Nevertheless, we posit that a model which minimizes denoising error can do a good job of implicitly learning the vector field. Thus, for some choice of distance metric d over the manifold, we aim to learn the so-called Fréchet mean of the clean data distribution conditioned on noisy data:

$$\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) \approx \arg\min_{\hat{\mathbf{x}} \in \mathcal{M}} \mathbb{E}_{\mathbf{x}_1 \sim p_t(\mathbf{x}_1 \mid \mathbf{x})} \left[d^2(\mathbf{x}_1, \hat{\mathbf{x}}_1) \right] \tag{17}$$

As a sanity check, note that when $\mathcal{M}$ is a Euclidean space and d is Euclidean distance, d^2 reduces to the usual MSE denoising loss whose minimizer is the conditional expectation of $p_t(\mathbf{x}_1 \mid \mathbf{x})$, in agreement with Equation 5.

(3) At inference time, in lieu of repeatedly evaluating the logarithmic map and integrating the vector field with the exponential map, we observe that such a procedure amounts to moving along the interpolant connecting $\mathbf{x}$ to $\hat{\mathbf{x}}_1$:

$$\exp_{\mathbf{x}} \left[\hat{v}(\mathbf{x}, t; \theta)\, dt \right] = \exp_{\mathbf{x}} \left[\frac{dt}{1-t} \log_{\mathbf{x}} \hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) \right] \tag{18}$$

i.e., a fraction $dt/(1-t)$ towards $\hat{\mathbf{x}}_1$. Hence, we take an integration step at inference-time via RMSD alignment followed by linear interpolation in ambient space, as executed in Algorithm 2.

FAPE and All-Atom Structure As defined in Section 3.2, our flow matching framework operates over residue-level structures; specifically, over Cβ coordinates $\mathbf{x} \in \mathbb{R}^{3N}$. However, the FAPE loss is defined over structures also containing (1) all-atom positions and (2) residue frames, and indeed we continue to supervise these outputs to ensure that AlphaFLOW and ESMFLOW produce meaningful all-atom structures. To reconcile these views, let $\mathcal{S}$ denote an all-atom structure, let $[\cdot]_{\mathrm{C}\beta}$ be the operator that extracts the Cβ coordinates, and denote the denoising model as $\hat{\mathcal{S}}(\mathbf{x}, t; \theta)$. Most of training and inference proceeds as if all structures were passed through the $[\cdot]_{\mathrm{C}\beta}$ operator: training points are sampled via $\mathbf{x}_1 = [\mathcal{S}]_{\mathrm{C}\beta}$ before noisy interpolation; and inference proceeds by parameterizing the Cβ denoising model as

$$\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) = \left[\hat{\mathcal{S}}(\mathbf{x}, t; \theta)\right]_{\mathrm{C}\beta} \tag{19}$$

However, this extraction is *not* applied to compute the denoising loss—neither to the sampled data nor the prediction. Instead, the denoising model is trained to approximate (cf. Equation 17):

$$\hat{\mathcal{S}}(\mathbf{x}, t; \theta) \approx \arg\min_{\hat{\mathcal{S}}} \mathbb{E}_{\mathcal{S}|\mathbf{x}} \left[\mathrm{FAPE}^2(\mathcal{S}, \hat{\mathcal{S}})\right] \tag{20}$$

and thus the reparameterized Cβ denoising model becomes

$$\hat{\mathbf{x}}_1(\mathbf{x}, t; \theta) \approx \left[\arg\min_{\hat{\mathcal{S}}} \mathbb{E}_{\mathcal{S}|\mathbf{x}} \left[\mathrm{FAPE}^2(\mathcal{S}, \hat{\mathcal{S}})\right]\right]_{\mathrm{C}\beta} \tag{21}$$

Colloquially, this means that the denoised Cβ structure (towards which we interpolate at inference time) is the Cβ *part of the best all-atom prediction*, rather than the best Cβ prediction. In the final inference step, rather than extracting $\hat{\mathbf{x}}_1$ from $\hat{\mathcal{S}}$ and interpolating the rest of the way towards it, we simply return the all-atom structure $\hat{\mathcal{S}}$. However, the model is predicting the denoised all-atom structure from the Cβ structure alone, and there is no iterative refinement of the non Cβ components. Hence, our model is best thought of as a generative model over Cβ positions *only*, which additionally fills in the all-atom information to minimize the FAPE loss conditioned on the input Cβ positions.

A.3. Comparison with Harmonic Diffusion

In harmonic diffusion, as in flow matching, a conditional probability path $p(\mathbf{x}_t \mid \mathbf{x}_0)$ represents a noising process for the data point $\mathbf{x}_0$ ($t = 0$ for the data by diffusion convention). Unlike flow matching, the path is given by the transition (or perturbation) kernel of a (Markovian) diffusion process rather than interpolation with the noise. The stationary distribution of the diffusion is the noisy prior by construction; however, the probability path converges to this prior only in the infinite-time limit. Instead, the maximum time is chosen such that the KL-divergence between the $p_{t|0}$ and the stationary distribution is acceptably low. Unfortunately, in harmonic diffusion:

$$D_{\mathrm{KL}}(p_{t|0} || p_\infty) = \sum_{i=1}^{3n} \left[e^{-\lambda_i t}\left(E_i - \frac{1}{2}\right) - \frac{1}{2}\log\left(1 - e^{-\lambda_i t}\right)\right] \tag{22}$$

where E_i is the (roughly constant) amount of energy in the ith mode (Equation 3 in Jing et al. (2023)). That is, the rate of convergence not only depends on the number of dimensions, but—more problematically—the smallest eigenvalue λ_i of the diffusion drift matrix, which becomes smaller for larger proteins. Hence, it becomes tricky to train a time-conditioned denoising model for proteins of arbitrary size. In the case of AlphaFLOW and ESMFLOW trained on crops of 256, the model would not be able to denoise longer proteins from an intermediate state at which the crops have converged to noise, but the entire protein has not—such states have never been seen during training. Our flow matching framework instead imposes the noisy prior as a boundary condition at the same $t = 0$ *for all protein lengths and crops*, avoiding this issue.

While the fixed convergence time is a desirable quality, our flow matching framework—at least as defined in Equations 2–6—satisfies an even stronger property, which we call *crop invariance* (Proposition A.1). Colloquially, this means that the marginal distribution of a crop of length M at time t is the same as if it were noised independently as an intact sequence of length M. This property ensures the noisy distributions over isolated crops seen at training time are exactly the same as those seen at inference time, when those crops are embedded in full-size proteins.

Proposition A.1. *Let $\mathbf{x}_1 \in \mathbb{R}^N$ and $\mathbf{x}_1\texttt{[i:i+M]} \in \mathbb{R}^M$ be a crop of $\mathbf{x}_1$ of length $M \leq N$ and define $p_t^{(M)}, p_t^{(N)}$ to be the conditional probability paths in dimensionalities N, M. Then for any $t, \tilde{\mathbf{x}} \in \mathbb{R}^M$, $p_t^{(N)}(\mathbf{x}\texttt{[i:i+M]} = \tilde{\mathbf{x}} \mid \mathbf{x}_1)$ is equal to $p_t^{(M)}(\mathbf{x} = \tilde{\mathbf{x}} \mid \mathbf{x}_1\texttt{[i:i+M]})$.*

Proof. Our key claim is that for time $t = 0$, i.e. in the noise distribution, the density $q^{(N)}(\mathbf{x}\texttt{[i:i+M]} = \tilde{\mathbf{x}})$ is equivalent to $q^{(M)}(\mathbf{x} = \tilde{\mathbf{x}})$. The former amounts to marginalizing the density $q^{(N)}(\mathbf{x})$ over the non-crop variables. For simplicity, we proceed with $i = 0$; the more general case is very similar:

$$
\begin{aligned}
q^{(N)}\left(\mathbf{x}_{[0,M)} = \tilde{\mathbf{x}}\right) &= \int q^{(N)}\left(\mathbf{x}_{[0,M)} = \tilde{\mathbf{x}}, \mathbf{x}_{[M,N)}\right)\, d\mathbf{x}_{[M,N)} \\
&\propto \int \exp\left[-\frac{\alpha}{2}\left[\sum_{j=0}^{M-2}\|\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_{j+1}\|^2 + \|\tilde{\mathbf{x}}_{M-1} - \mathbf{x}_M\|^2 + \sum_{j=M}^{N-2}\|\mathbf{x}_j - \mathbf{x}_{j+1}\|^2\right]\right] d\mathbf{x}_{[M,N)} \\
&= \exp\left[-\frac{\alpha}{2}\sum_{j=0}^{M-2}\|\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_{j+1}\|^2\right] \underbrace{\int \exp\left[-\frac{\alpha}{2}\left[\|\tilde{\mathbf{x}}_{M-1} - \mathbf{x}_M\|^2 + \sum_{j=M}^{N-2}\|\mathbf{x}_j - \mathbf{x}_{j+1}\|^2\right]\right] d\mathbf{x}_{[M,N)}}_{\text{constant}} \\
&\propto q^{(M)}(\mathbf{x} = \tilde{\mathbf{x}})
\end{aligned}
$$

where the constant is an offset Gaussian integral. This equivalence means that—up to some global translation—sampling noise of dimension N and then cropping to length M is equivalent to sampling noise of dimension M. Then, notice that linear interpolation of full structures implies linear interpolations of crops:

$$
\mathbf{x} = (1-t)\cdot\mathbf{x}_0 + t\cdot\mathbf{x}_1 \quad \Longrightarrow \quad \mathbf{x}\texttt{[i:i+M]} = (1-t)\cdot\mathbf{x}_0\texttt{[i:i+M]} + t\cdot\mathbf{x}_1\texttt{[i:i+M]}
$$

Thus, the sampling procedure for $p_t^{(N)}(\mathbf{x}\texttt{[i:i+M]} = \tilde{\mathbf{x}} \mid \mathbf{x}_1)$—which is to interpolate N-dimensional noise and data and then crop to M dimensions—is the same as the sampling procedure for $p_t^{(M)}(\mathbf{x} = \tilde{\mathbf{x}} \mid \mathbf{x}_1\texttt{[i:i+M]})$—which is to first crop the data and noise to M dimensions and then interpolate. □

We note that crop invariance no longer holds in the final form of flow matching that we use in Algorithms 1 and 2 and describe in Appendix A.2 due to the RMSD alignment step. Nevertheless, we posit that initial preservation of distributional alignment helps with generalization to proteins of unseen large sizes at inference time.

The second advantage of our flow matching framework over harmonic diffusion is in the treatment of missing residues. In harmonic diffusion, the perturbation kernel $p(\mathbf{x}_t \mid \mathbf{x}_0)$ is a Gaussian whose mean is given by $\boldsymbol{\mu} = e^{-t\mathbf{H}/2}\mathbf{x}_0$, where $\mathbf{H}$ is the drift matrix. This matrix exponential is far from diagonal, meaning that each entry of $\boldsymbol{\mu}$ is dependent on all initial entries of $\mathbf{x}_0$. Hence, if there are missing coordinates in $\mathbf{x}_0$, they must be imputed in order to sample $p(\mathbf{x}_t \mid \mathbf{x}_0)$. In contrast, in our flow matching framework, each coordinate in $\mathbf{x}$ at time t is a linear combination of only the same-index coordinates in $\mathbf{x}_0$ and $\mathbf{x}_1$. Hence, we can simply omit the missing residues in the RMSD alignment and the subsequent interpolation.

B. Experimental Details

B.1. Training and Inference

Training We use OpenFold (Ahdritz et al., 2022) to train AlphaFLOW and ESMFLOW, as it closely follows the training best practices described in AlphaFold (Jumper et al., 2021). However, because the OpenFold weights for AlphaFold were trained with a much later cutoff date, we instead initialize with the original CASP14 weights from DeepMind (version 1). For PDB training data, we use a January 2023 snapshot of the PDB and apply 40% clustering with MMSeqs2 (Steinegger & Söding, 2017). We train with crops of size 256, batch size of 64, no recycling, and no templates. AlphaFLOW is trained on the full set of auxiliary losses, except the structural violation loss and with the FAPE loss squared. ESMFLOW is trained on the FAPE, pLDDT, distogram, and supervised χ losses. To maintain precision in the initial prediction, we set $t = 0$ and omit the noisy input in 20% of training examples. Training progress is monitored via the precision and diversity on a validation set of 183 CAMEO targets deposited Aug–Oct 2022, following Jing et al. (2023). To fine-tune on MD ensembles, we resume from the selected checkpoints from the PDB training. All the training settings remain unchanged, except the targets are sampled uniformly at random (with a random conformation), the batch size is set to 8, and $t = 0$ is set 10% of the time. Training progress is monitored via the loss on the validation split.

Training Cost All training is done on a machine with 8x NVIDIA A100 GPUs and 2x Intel Xeon(R) Gold 6258R processors, with the total training cost shown in Table 2.

Table 2. **AlphaFLOW and ESMFLOW training cost**

		Total hours	Train examples	Secs per training pass
AlphaFLOW	PDB	267	1.28M	5.8
	PDB distillation	105	160k	17.4
	MD	11	43k	6.2
	MD distillation	28	38k	17.4
	MD+Templates	9	38k	6.3
	MD+Templates distillation	39	51k	18.0
ESMFLOW	PDB	104	720k	4.2
	PDB distillation	37	64k	11.9
	MD	5	27k	4.6
	MD distillation	34	51k	12.0
	MD+Templates	9	51k	4.7
	MD+Templates distillation	23	38k	12.5

Inference We run AlphaFLOW and ESMFLOW with 10 steps by default, evenly spaced from $t = 0$ to $t = 1$, where the first prediction is performed with no noisy input. However, by merging the first $K > 1$ steps, we can reduce the variance of the sampled distribution and increase precision, analogous to increasing MSA depth. This is because—after the initial large step to $t = 0.1K$—we are effectively starting the flow from a modified intermediate marginal $p_t(\mathbf{x})$ which differs from the $p_t(\mathbf{x})$ that would arise from properly following the flow:

$$\mathbf{x} = (1-t)\cdot\mathbf{x}_0 + t\cdot\mathbf{x}_1, \quad \mathbf{x}_0 \sim q(\mathbf{x}_0), \mathbf{x}_1 \sim p_{\text{data}}(\mathbf{x}_1) \qquad \text{(original)} \tag{23}$$

$$\mathbf{x} = (1-t)\cdot\mathbf{x}_0 + t\cdot\mathbb{E}_{p_{\text{data}}}[\mathbf{x}_1], \quad \mathbf{x}_0 \sim q(\mathbf{x}_0) \qquad \text{(modified)} \tag{24}$$

i.e., by stepping directly to intermediate time t, we interpolate towards the *initial* $\hat{\mathbf{x}}_1$ prediction, which is a single point estimate of the unconditional expectation, rather than the full distribution $p_{\text{data}}(\mathbf{x}_1)$. We omit recycling for all methods following Del Alamo et al. (2022). Note that, by default, AlphaFold accepts a maximum MSA depth equivalent to subsampling with depth 1024, and exhibits a small level of diversity; on the other hand, ESMFold is completely deterministic. For AlphaFLOW PDB experiments, we resample the MSA (with depth 1024) for each new sample, but not for each inference step. At inference time, MSAs for all sequences are computed with the ColabFold MMSeqs pipeline (Porter et al., 2023).

Self-conditioning Although we do not use recycling *per se* for either our methods or the baselines, we employ *self-conditioning* (Chen et al., 2022; Stärk et al., 2023) in the PDB experiments to increase the precision of AlphaFLOW. In particular, at training time, 50% of supervised forward passes are provided the (gradient-detached) outputs from an initial forward pass of the model; we reuse the recycling embedder of AlphaFold to embed these outputs. At inference time, every forward pass after the first is provided the outputs of the previous forward pass. Note that unlike Stärk et al. (2023), we self-condition with the full set of model output states, i.e., including pair embeddings, rather than just the output $\hat{\mathbf{x}}_1$ prediction. Self-conditioning is omitted for distillation training and for MD training and inference. Finally, although we also trained ESMFLOW with self-conditioning, we did not observe any improvements and report results without it.

Distillation Because the inference process is deterministic except for the initial noisy sample, it defines a map from the noisy distribution to the data distribution. We can aim to learn this map via a model that ingests the noisy sample and predicts the corresponding fully-denoised output in a single forward pass. To train such a model, for each training example (still a crop of 256), we run the full inference pipeline with the original flow model and set the result as the training target. All other training settings are kept the same and training performance is monitored the same way, except the batch size is always set to 8, and the concepts of sampling t, interpolating, and self-conditioning no longer apply. For AlphaFLOW and ESMFLOW on the PDB, we train for 160k and 64k training examples, respectively. For distilling the MD models, we start from the weights of the original AlphaFLOW-MD and ESMFLOW-MD and fine tune for 38k and 51k training examples, respectively.

B.2. Datasets

PDB Test Set To construct the test set of structurally heterogeneous recent proteins from the PDB, we follow Ellaway et al. (2023) and identify chains as representing the same protein if they map to the same segment in the same UniProt reference sequence. We use the SIFTS annotations database (Dana et al., 2019) and its residue-level mappings from PDB chains to UniProt reference sequences to associate each deposited chain with a segment. Then, we cluster all segments with a Jaccard similarity threshold of 0.75 and complete linkage, with each resulting cluster regarded as a distinct protein, yielding 75k proteins. We collect all proteins which (1) are represented by 2–30 chains deposited after the AlphaFold training cutoff and no chains before the cutoff, (2) have lengths between 256–768 residues, (3) have at least two structural clusters when the chains are clustered with a threshold of 0.85 symmetrized lDDT-Cα and complete linkage. From the resulting 563 proteins (represented by 2843 chains), we subsample 100 proteins (represented by 500 chains) to form the test set. At inference time, we run all models using the sequence given by the UniProt segment. The distribution of sequences lengths is shown in Figure 6.

MD Dataset The ATLAS dataset (Vander Meersche et al., 2023) consists of all-atom, explicit solvent MD simulations for 1390 non-membrane proteins, chosen as representatives for all eligible ECOD structural classes (Schaeffer et al., 2017). For each protein, 3 replicate simulations of length 100 ns are provided, each with 10k frames. To train and validate on these trajectories, we first generate MSAs for all 1390 ATLAS entries using the provided sequence and the ColabFold MMSeqs2 pipeline (Porter et al., 2023). Then, for the train and validation sets, we extract 300 conformations to be randomly sampled in the training pipeline. The test split consists of all 84 targets whose corresponding PDB entries were deposited after May 1, 2019, minus the two targets with sequence length greater than 1024. The resulting distribution of sequences lengths is shown in Figure 6.

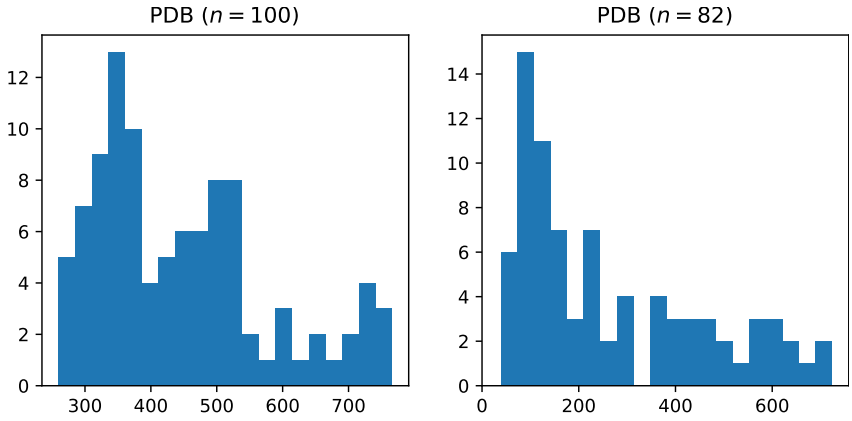

Figure 6. **Histogram of sequence lengths** in the PDB test set (*left*) and the ATLAS test set (*right*).

B.3. Evaluation Procedures

Symmetrized lDDT In the PDB experiments, we often need to compute the similarity (or dissimilarity) between two structures which may not share identical sequences, and which may differ significantly in length—for example, between two PDB chains or between a PDB chain and a structure predicted from the UniProt reference sequence. To do so, we define the *symmetrized lDDT* as a variant of lDDT-Cα which is (as the name suggests) symmetric and robust to these discrepancies. We perform a pairwise alignment of the two sequences, and tabulate the Cα pairs (identified by residue index only) which are within 15 Å of each other in *either structure*. Then, we score the fraction of these selected pairwise distances that are consistent within 0.5 Å, 1 Å, 2 Å, and 4 Å in the two structures. The symmetrized lDDT-Cα is the mean of these four scores.

MD Evaluations To compare a generated ensemble with the ground-truth MD ensemble, we first align both ensembles to the static all-atom structure that initialized the simulation (provided in the ATLAS download). We then perform all analyses using the Euclidean atomic coordinates in MDTraj (McGibbon et al., 2015). For most procedures, we subsample 1000 random MD frames to reduce the analysis runtime. To compute the RMWD, the Wasserstein distance between two 3D

Gaussians is given by

$$\mathcal{W}_2^2(\mathcal{N}(\mu_1,\Sigma_1),\mathcal{N}(\mu_2,\Sigma_2)) = \|\mu_1-\mu_2\|^2 + \mathrm{Tr}\left(\Sigma_1+\Sigma_2-2(\Sigma_1\Sigma_2)^{1/2}\right) \tag{25}$$

which reduces to Euclidean distance in the case of point masses. This squared distance decomposes into a translation term and a variance term; so the aggregate RMWD (Equation 8) also decomposes as

$$\mathrm{RMWD}^2(\mathcal{X}_1,\mathcal{X}_2) = \underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\mu_{1,i}-\mu_{2,i}\|^2}_{(\text{translation contribution})^2} + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\mathrm{Tr}\left(\Sigma_{1,i}+\Sigma_{2,i}-2(\Sigma_{1,i}\Sigma_{2,i})^{1/2}\right)}_{(\text{variance contribution})^2} \tag{26}$$

We report the translation contribution (which resembles RMSD) and variance contribution in Table 1. In the calculation of joint $\mathcal{W}_2$ distance, we first project to the PCA subspace because thermal fluctuations dominate in the full dimensional space and make the $\mathcal{W}_2$ metric unsuitable without an extremely large number of samples. While it is common to perform PCA using the MD reference ensemble alone, we note that doing so can obscure deviations of the predicted ensemble along the orthogonal degrees of freedom. Thus, we repeat the analysis using with the MD ensemble and the equally-weighted pooling of the MD and predicted ensembles. Finally, in the residue exposure analysis, we compute the solvent-accessible surface area (SASA) of each sidechain using the Shrake-Rupley algorithm and a probe radius of 2.8 Å. Following Porter et al. (2019), we use a SASA threshold of 2.0 Å^2 to distinguish buried and exposed residues.

Comparison with Replicate MD To compare the performance of our method with replicate MD simulations, we leverage the fact that ATLAS trajectories are provided in three replicates (100 ns and 10k frames each). In the main experiments, these three replicates are pooled to collectively represent the MD ensemble; however, such pooling would not be appropriate if one of these replicates is taken for comparison. Instead, we select the *first* replicate for comparison and pool the latter two to represent the ground truth MD ensemble. We emulate different computational budgets by truncating the first trajectory to its first 4096, 2048, 1024, 512, 256, 128, 64, 32, and 16 frames before analysis, respectively representing simulation lengths of 40.96 ns, 20.48 ns, 10.24 ns, 5.12 ns, 2.56 ns, 1.28 ns, 640 ps, 320 ps, and 160 ps. When necessary, we subsample or replicate by the appropriate power of 2 to ensure all analyses operate on 256 frames (important for finite-sample Wasserstein distances). The computational cost in GPU-hrs is estimated by running 1 minute of MD for each protein on a single NVIDIA A100 GPU and noting the average performance in hrs/ns. (The average GPU utilization is 62%, indicating efficient usage of resources.) For the AlphaFLOW and ESMFLOW ensembles, we first generate 250 samples as usual and also subsample 128, 64, 32, 16, 8, and 4 samples for analyses, duplicating by the appropriate power 2 to reach 256 ($\approx$ 250) samples. The runtime is provided as an average over all test proteins on a single NVIDIA A100 GPU.

Comparison with Normal Mode Analysis We also compare the performance of our method with normal mode analysis of the PDB protein structures using ProDy (Bakan et al., 2011). We construct Gaussian Network Models (GNM) (Bahar et al., 1997) and Anisotropic Network Models (ANM) (Atilgan et al., 2001) using the Cα coordinates and draw 250 samples from each model, keeping all nondegenerate eigenvectors. We use $\Gamma = 0.15$ (adjusted from default to match the average MD RMSF) and default 10 Å and 15 Å cutoffs for GNM and ANM, respectively.

C. Additional Results

C.1. PDB Ensembles

Table 3 provides precision, recall, and diversity results for the experiments on PDB ensembles, with a median taken over the 100 test set targets. For ESMFold and ESMFLOW, the second set of results corresponds to the subset of targets released after the training cutoff of May 1, 2020 ($n = 56$). Runtime measurements (per sample) are performed on a single A100 GPU and reported as a median over 100 targets. Figure 7 shows PCA of the true and generated ensembles for several selected targets to illustrate the degradation of the MSA subsampling ensembles. Figure 8 highlights conformational changes observed in the PDB ensembles and correctly sampled by AlphaFLOW. In both figures, the PCA is performed by first aligning all PDB sequences with the UniProt reference with ClustalW (Larkin et al., 2007) and taking the Cα positions of the common subset of aligned residues. The structures are then RMSD aligned to a randomly selected PDB structure and the PCA is performed on the resulting Euclidean coordinates. Sample weights are chosen so that the PDB structures account for half the loading, regardless of their number. Coordinates are converted to Å RMSD units.

Table 3. **Evaluation on PDB ensembles.**

		Precison	Recall	Diversity	Runtime
AlphaFLOW	Full	0.810	0.801	0.185	69.6
	5 steps	0.821	0.801	0.151	42.1
	2 steps	0.839	0.811	0.082	21.3
	Distilled	0.831	0.810	0.128	7.4
MSA subsampling	512	0.849	0.823	0.026	5.5
	256	0.844	0.818	0.044	4.2
	128	0.835	0.806	0.053	3.9
	64	0.795	0.784	0.088	3.6
	48	0.757	0.760	0.125	3.5
AlphaFold		0.850	0.823	0.026	7.7
ESMFLOW	Full	0.777 / 0.777	0.777 / 0.765	0.210 / 0.213	30.4
	5 steps	0.787 / 0.788	0.772 / 0.767	0.166 / 0.174	18.3
	2 steps	0.795 / 0.797	0.774 / 0.760	0.100 / 0.102	9.2
	Distilled	0.775 / 0.774	0.752 / 0.745	0.152 / 0.152	3.1
ESMFold		0.806 / 0.809	0.764 / 0.761	0.000	3.2

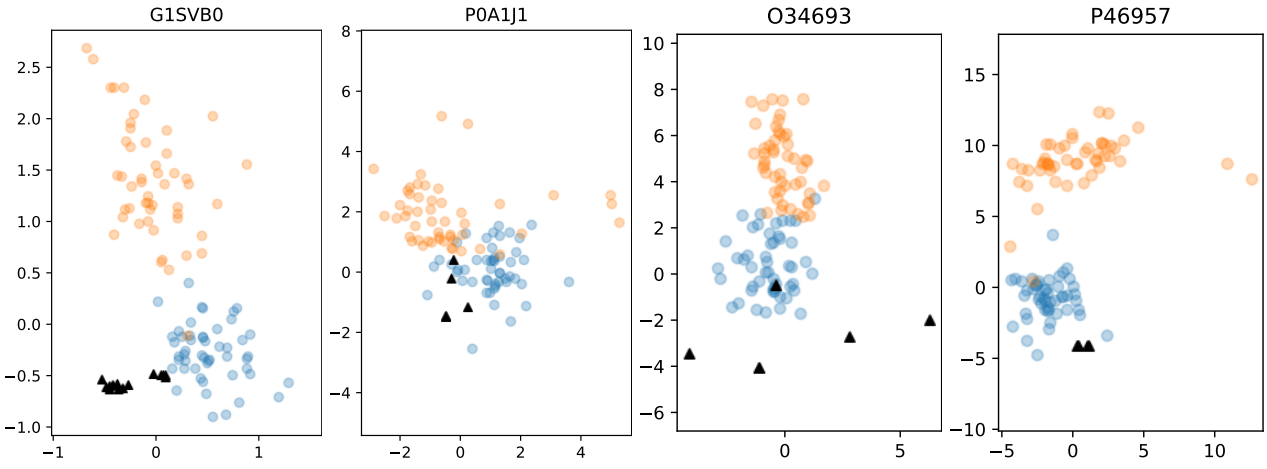

Figure 7. **PCA of PDB and predicted ensembles** from AlphaFLOW (blue) and MSA subsampling (depth 64) (orange), with PDB structures marked by ▲. The MSA subsampling ensembles have similar diversity as the AlphaFLOW ensembles but drift away from the true structures.

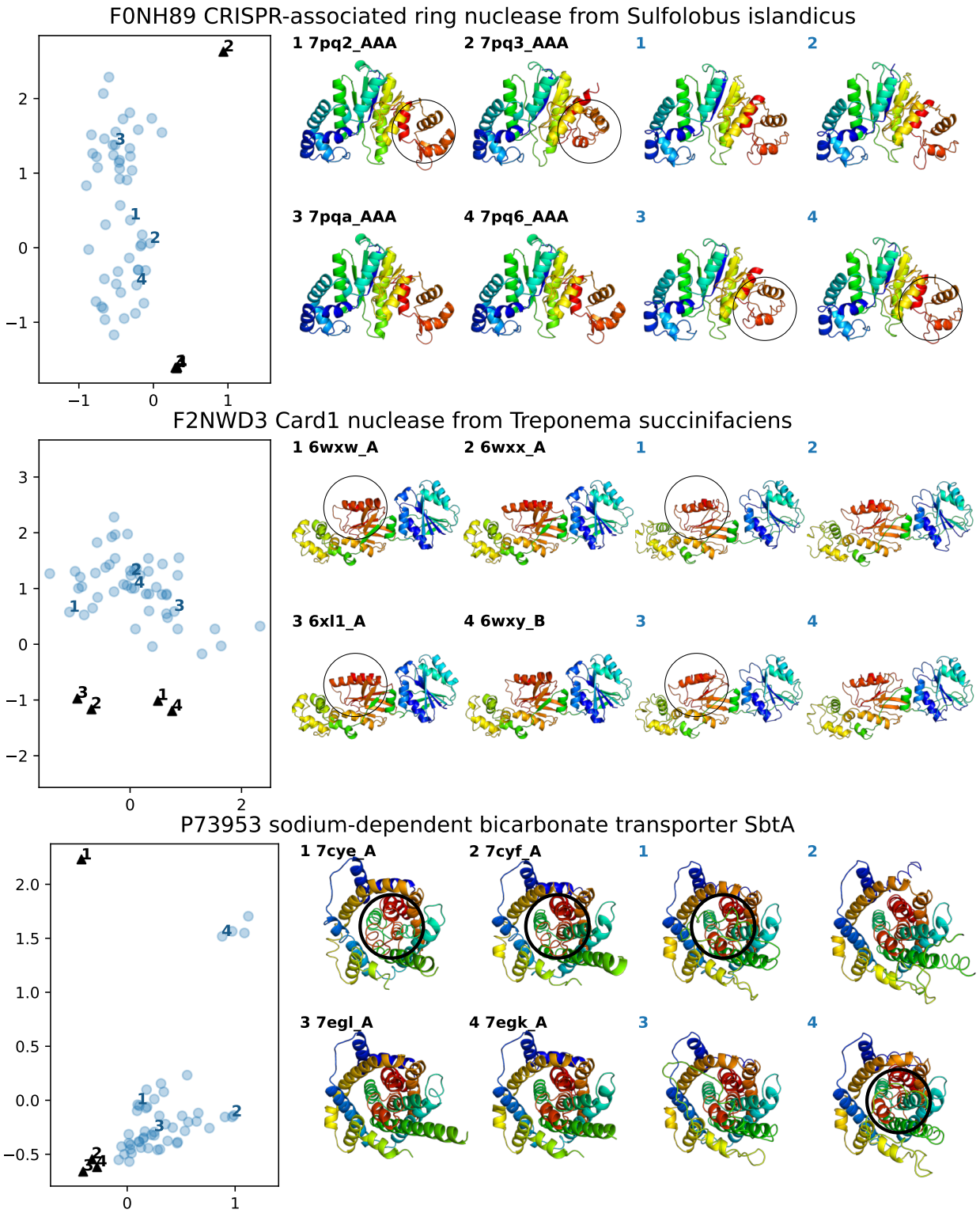

Figure 8. **PDB conformational changes** correctly sampled by AlphaFLOW. For each UniProt ID, the PCA plot shows the complete set of PDB structures (▲) and AlphaFLOW samples (blue). Four random PDB structures (*left*) and four random AlphaFLOW samples (*right*) are visualized, where the numbers label the positions of the selected structures.

C.2. MD Ensembles

Table 4 provides the evaluation of ESMFLOW on MD ensembles. In Table 5, we report the performance of AlphaFLOW-MD with ablated training procedures, and the comparison of AlphaFLOW with normal mode analysis conducted on the PDB structure. In Figures 9–12, we provide additional visualizations for the RMSF, transient contacts, weak contacts, and solvent exposure analyses of AlphaFLOW-MD ensembles. Finally, In Figure 13, we provide additional convergence results for AlphaFLOW-MD+Templates vs replicate MD simulations.

Table 4. **Evaluation of ESMFLOW on MD ensembles**

		ESMFLOW-MD			EFMD+Templates	
		Full	Distilled	ESMFold	Full	Distilled
Predicting flexibility	Pairwise RMSD (=2.90)	3.25	2.76	0.00	2.00	1.42
	Pairwise RMSD r ↑	0.19	0.19	—	0.85	0.80
	All-atom RMSF (=1.70)	2.16	2.12	0.00	1.07	0.80
	Global RMSF r ↑	0.31	0.33	—	0.84	0.79
	Per-target RMSF r ↑	0.76	0.74	—	0.90	0.87
Distributional accuracy	Root mean $\mathcal{W}_2$-dist. ↓	3.60	4.23	4.60	2.17	2.27
	↪ Translation contrib. ↓	3.13	3.75	3.65	1.66	1.70
	↪ Variance contrib. ↓	1.74	1.90	2.50	1.07	1.35
	MD PCA $\mathcal{W}_2$-dist. ↓	1.51	1.87	1.69	1.44	1.48
	Joint PCA $\mathcal{W}_2$-dist. ↓	3.19	3.79	3.87	1.70	1.81
	% PC-sim > 0.5 ↑	26	33	—	49	40
Ensemble observables	Weak contacts J ↑	0.55	0.48	0.22	0.59	0.48
	Transient contacts J ↑	0.34	0.30	0.15	0.47	0.41
	Exposed residue J ↑	0.49	0.43	0.28	0.50	0.44
	Exposure MI matrix ρ ↑	0.20	0.16	—	0.22	0.16

Table 5. **Ablations and normal mode analysis on MD ensembles**. The ablations verify the importance of the two-step training procedure for AlphaFLOW+MD. Normal mode analysis (NMA) often fails to outperform baseline AlphaFLOW+MD despite having access to the ground truth PDB structure, and significantly underperforms AlphaFLOW+MD+Templates when it is provided the same PDB template structure. ⋆Note that NMA results for RMSF and RMWD are Cα-only rather than all-atom, which likely overestimates the performance. GNM: Gaussian Network Model; ANM: Anisotropic Network Model.

			Ablations			NMA	
		Baseline	No ATLAS finetuning	No PDB pretraining	AFMD +Templates	GNM	ANM
Predicting flexibility	Pairwise RMSD (=2.90)	**2.89**	2.41	3.04	2.18	1.85	2.36
	Pairwise RMSD r ↑	**0.48**	0.34	0.29	**0.94**	0.71	0.65
	All-atom RMSF (=1.70)	**1.68**	1.25	1.81	1.31	1.22⋆	1.35⋆
	Global RMSF r ↑	**0.60**	0.48	0.45	**0.91**	0.64⋆	0.55⋆
	Per-target RMSF r ↑	**0.85**	0.82	0.83	**0.90**	0.72⋆	0.76⋆
Distributional accuracy	Root mean $\mathcal{W}_2$-dist. ↓	**2.61**	2.96	3.11	**1.95**	2.47⋆	2.54⋆
	↪ Translation contrib. ↓	**2.28**	2.52	2.71	**1.64**	2.07⋆	2.09⋆
	↪ Variance contrib. ↓	**1.30**	1.36	1.44	**1.01**	1.33⋆	1.27⋆
	MD PCA $\mathcal{W}_2$-dist. ↓	**1.52**	1.64	1.59	**1.25**	1.84	1.73
	Joint PCA $\mathcal{W}_2$-dist. ↓	**2.25**	2.60	2.67	**1.58**	2.44	2.35
	% PC-sim > 0.5 ↑	**44**	35	38	**44**	13	23
Ensemble observables	Weak contacts J ↑	**0.62**	0.48	0.60	**0.62**	0.45	0.40
	Transient contacts J ↑	**0.41**	0.36	0.39	**0.47**	0.25	0.25
	Exposed residue J ↑	**0.50**	0.40	**0.50**	0.50	—	—
	Exposure MI matrix ρ ↑	**0.25**	0.18	**0.25**	0.25	—	—

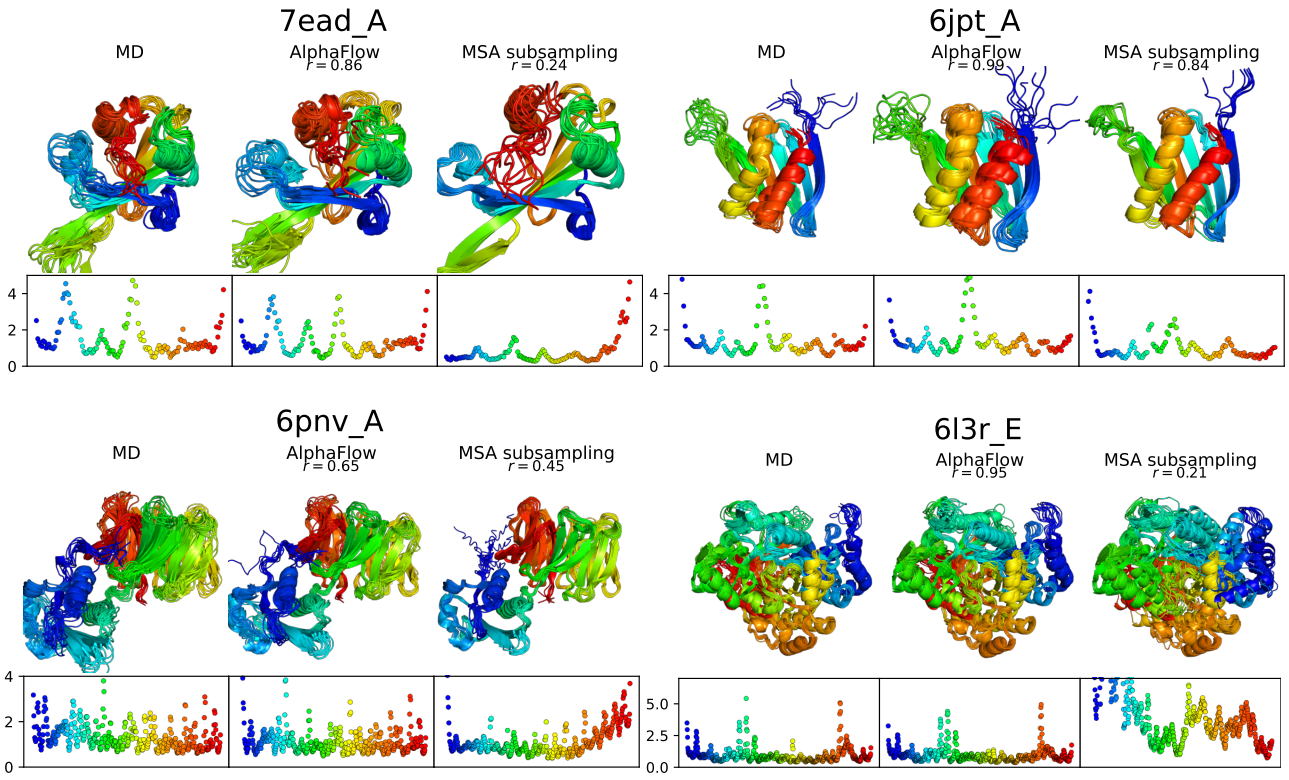

Figure 9. **Visualization of ensembles and their RMSF plots**. For each PDB ID, 10 samples from the MD, AlphaFLOW, and MSA subsampling (depth 48) ensembles are shown, with RMSF by residue index in insets. For the latter two, the Pearson correlation coefficient (r) with the MD RMSF is reported.

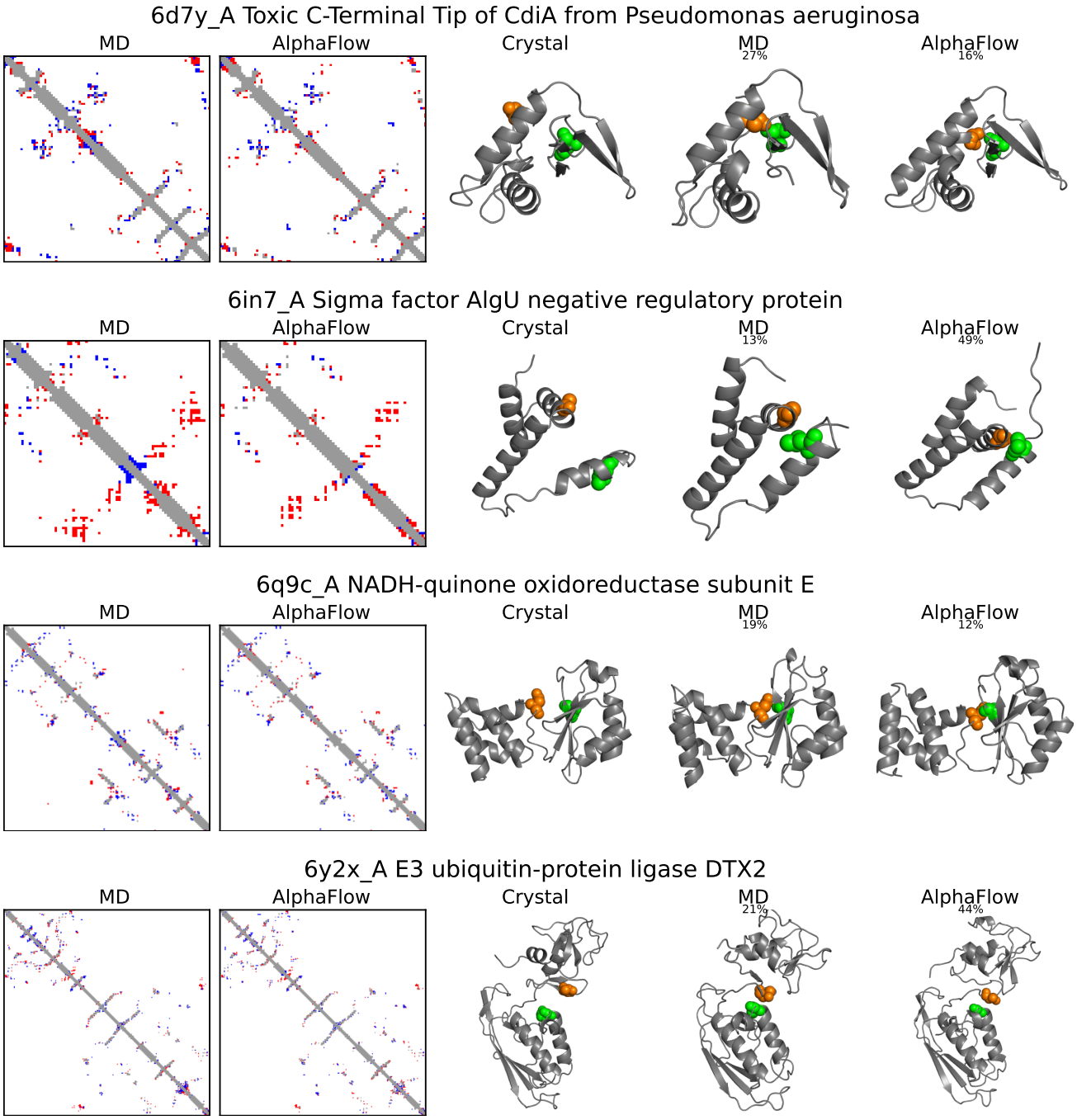

Figure 10. **Visualization of transient contacts.** For each PDB ID, the contact maps from MD simulation and AlphaFLOW are shown, with normal contacts in gray, weak contacts in blue, and transient contacts in red. Among the the transient contacts correctly identified by AlphaFLOW, one is selected for visualization: the two residues are highlighted in the crystal structure (*left*), a frame from the MD simulation (*middle*) where they are in contact, and an AlphaFLOW sample where they are in contact. The probability of occurence in each ensemble is shown.

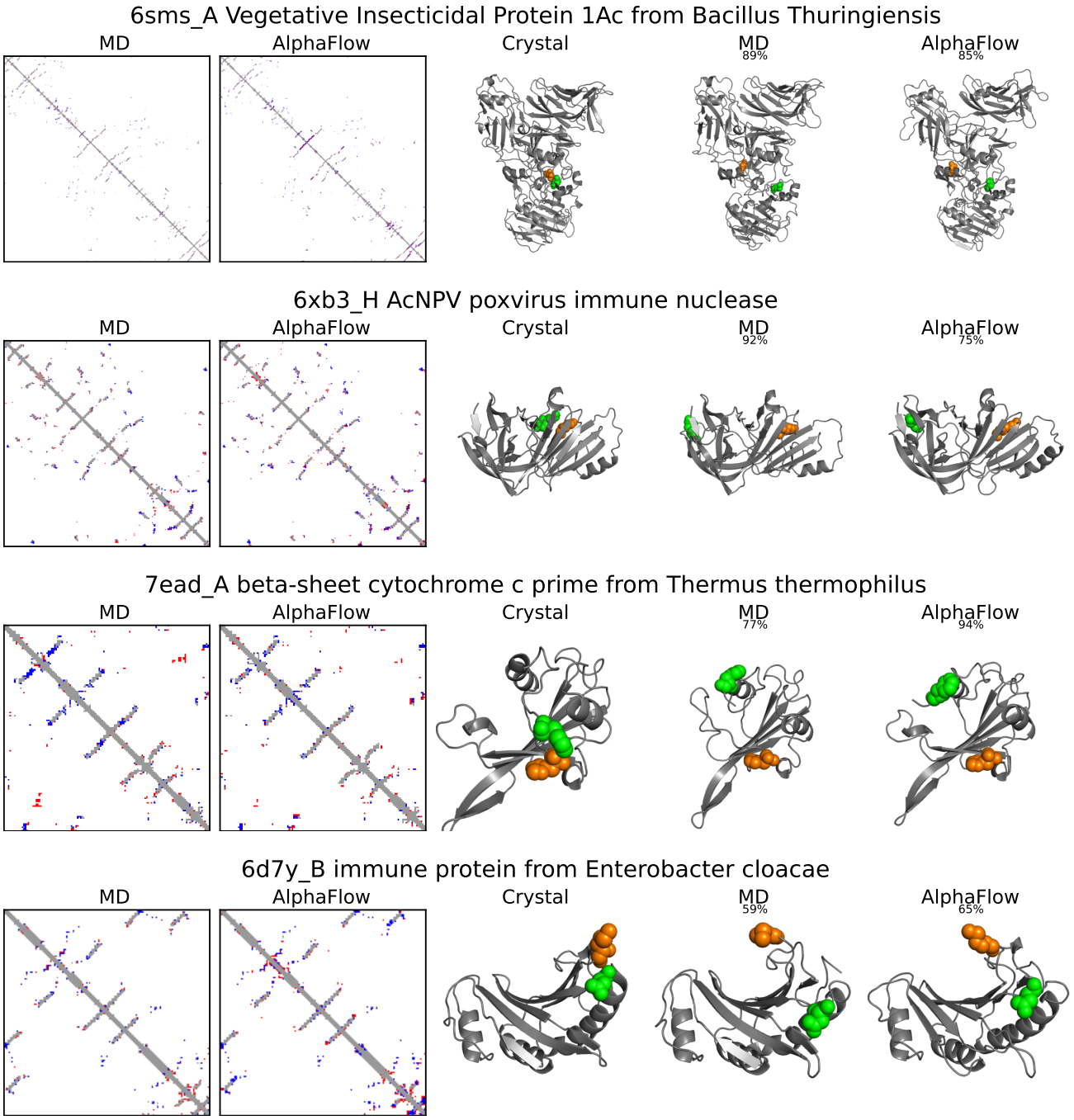

Figure 11. **Visualization of weak contacts.** For each PDB ID, the contact maps from MD simulation and AlphaFLOW are shown, with normal contacts in gray, weak contacts in blue, and transient contacts in red. Among the the weak contacts correctly identified by AlphaFLOW, one is selected for visualization: the two residues are highlighted in the crystal structure (*left*), a frame from the MD simulation (*middle*) where they are not in contact, and an AlphaFLOW sample where they are not in contact. The probability of occurence in each ensemble is shown.

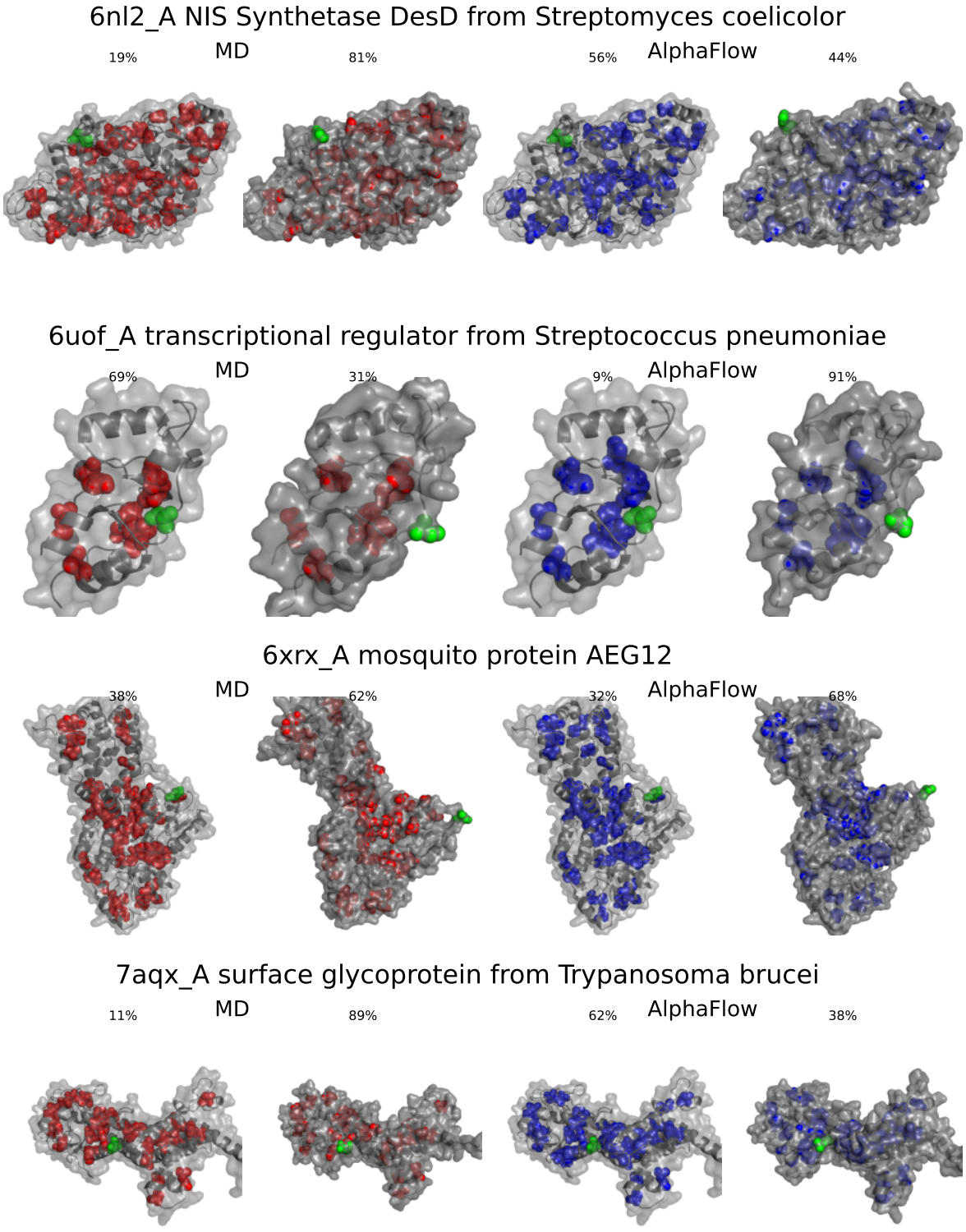

Figure 12. **Visualization of cryptic exposed residues.** For each PDB ID, in the *left* pair of structures, the set of true cryptic exposed residues (from MD) is colored red; in the *right* pair the set identified from AlphaFLOW ensembles is colored blue. A common identified residue is selected and highlighted in green. For each pair, the *left* structures shows the residues buried in the crystal structure whereas the *right* structure shows a frame (or sample) where the highlighted residue is exposed to the solvent. The probability of occurence in each ensemble is shown.

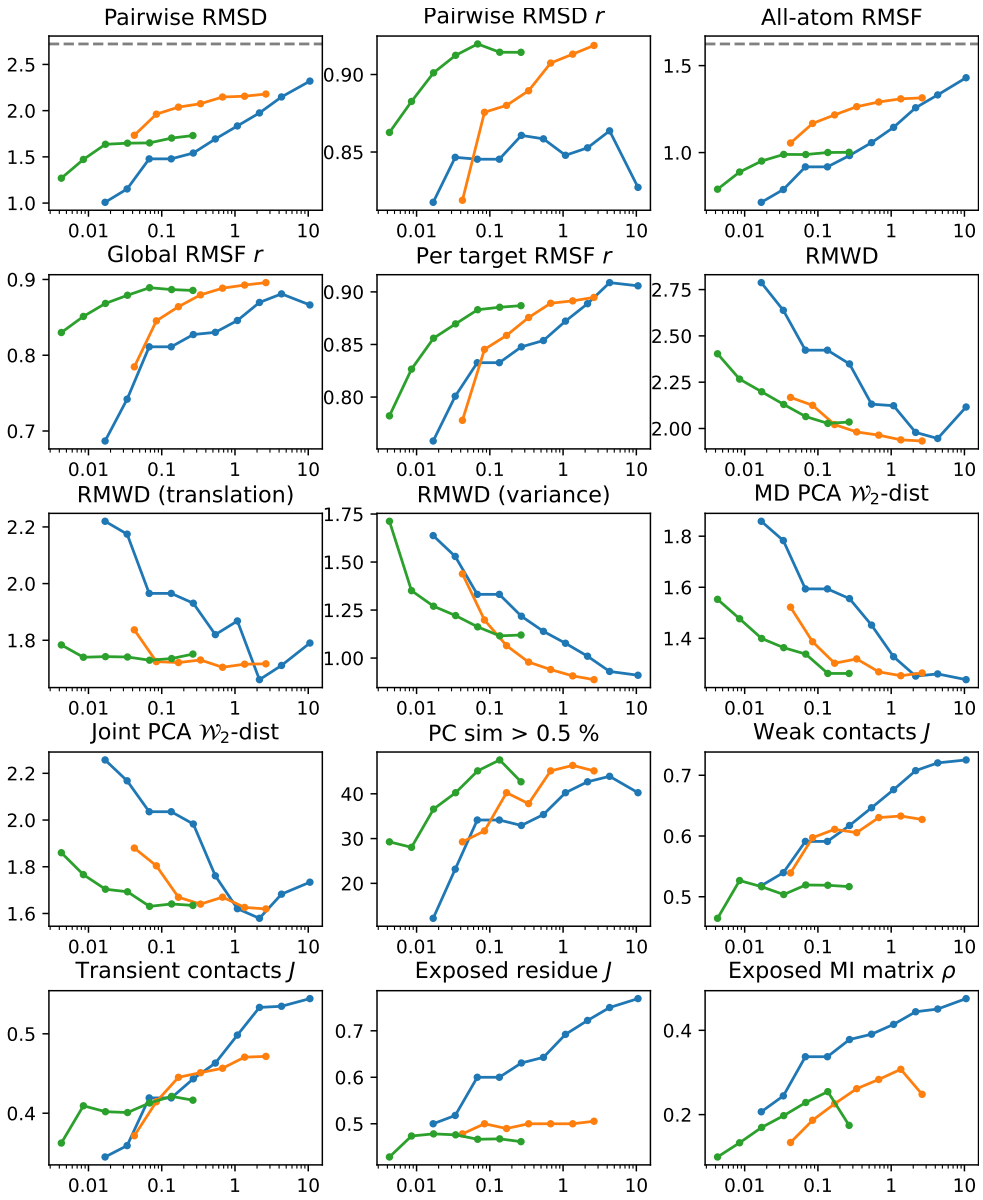

Figure 13. **Efficiency of AlphaFLOW vs replicate MD simulations.** AlphaFLOW (with templates) with varying number of samples in orange; AlphaFLOW distilled into a single forward pass in green; MD with varying trajectory lengths in blue. For Pairwise RMSD and RMSF, the values from the reference MD (i.e., pooling the remaining two replicates) are shown as horizontal dashed lines. The x-axis reports runtime in GPU-hrs averaged over targets. For MD, the average runtime is 6.3 mins / ns; for AlphaFLOW, the average runtime per sample is 38 s without distillation and 3.8 s with distillation. See Appendix B.3 for further benchmarking details.