Sequence-Augmented $\mathrm{SE}(3)$-Flow Matching For Conditional Protein Backbone Generation

Guillaume Huguet[1,2,3,*] **James Vuckovic**[1,*] **Kilian Fatras**[1,†] **Eric Thibodeau-Laufer**[1,†]
Pablo Lemos[1], **Riashat Islam**[1], **Cheng-Hao Liu**[1,3,4], **Jarrid Rector-Brooks**[1,2,3],
Tara Akhound-Sadegh[1,2,4], **Michael Bronstein**[1,5,6], **Alexander Tong**[1,2,3,‡] **Avishek Joey Bose**[1,5,‡]

[1]Dreamfold, [2]Université de Montréal, [3]Mila, [4]McGill University, [5]University of Oxford, [6]Aithyra

Abstract

Proteins are essential for almost all biological processes and derive their diverse functions from complex 3D structures, which are in turn determined by their amino acid sequences. In this paper, we exploit the rich biological inductive bias of amino acid sequences and introduce FOLDFLOW-2[4], a novel sequence-conditioned $\mathrm{SE}(3)$-equivariant flow matching model for protein structure generation. FOLDFLOW-2 presents substantial new architectural features over the previous FOLDFLOW family of models including a protein large language model to encode sequence, a new multi-modal fusion trunk that combines structure and sequence representations, and a geometric transformer based decoder. To increase diversity and novelty of generated samples—crucial for de-novo drug design—we train FOLDFLOW-2 at scale on a new dataset that is an order of magnitude larger than PDB datasets of prior works, containing both known proteins in PDB and high-quality synthetic structures achieved through filtering. We further demonstrate the ability to align FOLDFLOW-2 to arbitrary rewards, e.g. increasing secondary structures diversity, by introducing a Reinforced Finetuning (ReFT) objective. We empirically observe that FOLDFLOW-2 outperforms previous state-of-the-art protein structure-based generative models, improving over RFDiffusion in terms of unconditional generation across all metrics including designability, diversity, and novelty across all protein lengths, as well as exhibiting generalization on the task of equilibrium conformation sampling. Finally, we demonstrate that a fine-tuned FOLDFLOW-2 makes progress on challenging conditional design tasks such as designing scaffolds for the VHH nanobody.

1 Introduction

Rational design of novel protein structures via generative modeling holds significant promise for accelerating computational drug discovery [Chevalier et al., 2017, Ebrahimi and Samanta, 2023]. In particular, the ability to design proteins with a pre-specified functional property is arguably one of the principal tools in addressing global health challenges such as COVID-19 [Cao et al., 2020, Gainza et al., 2023], influenza [Strauch et al., 2017], and cancer [Silva et al., 2019]. In many instances, designing function involves the design of both the 3D geometric structure of the protein as well as its specific chemical interactions. In proteins, the amino-acid sequences determine the interaction between protein backbones and side chains, which fold into a distribution of protein structures. Consequently, the functional properties of protein structures can be *inferred* from its sequence.

---

[*]Co-first and corresponding authors: `{guillaume.huguet,james}@dreamfold.ai`

[†]Core contributor

[‡]Equal advising

[4]Our code can be found at `https://github.com/DreamFold/FoldFlow`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

The representation of proteins plays a key aspect in any computational approach to protein engineering. The 3D structure of proteins can be mathematically represented on the space of rotation and translation invariant $\mathrm{SE}(3)^N$. Several unconditional protein generative models have been developed recently to generate new protein backbones [Yim et al., 2023b, Bose et al., 2024]. While these models demonstrate the ability to design new proteins, they are insufficiently tailored for downstream drug discovery applications, where the primary challenge lies in generating proteins that are specifically tailored to interact effectively with a given target. In real-world drug design problems, one often knows the target protein (its amino acid sequence and often an experimentally verified 3D structure). In machine learning terms, the design of new proteins ("*de novo*" design) that can drug the given target can be framed as a *conditional* generation problem. This raises the following research question:

How can we leverage the structure and sequence of a target to inform de novo *protein design?*

Current work. In this paper, we introduce FOLDFLOW-2 a novel protein structure generative model that is additionally conditioned on protein sequences. FOLDFLOW-2 is built on the foundations of FOLDFLOW [Bose et al., 2024] and is an $\mathrm{SE}(3)^N$-invariant generative model for protein backbone generation that handles multi-modal data by design. Specifically, FOLDFLOW-2 introduces several new architectural components over previous protein structure generative models that enable it to process both 3D structure and discrete sequences. These include (1) a joint structure and sequence encoder; (2) a multi-modal fusion trunk that combines the representations from each modality in a shared representation space; and (3) a transformer-based geometric decoder. In contrast to prior efforts to incorporate sequences in structure-based generative models [Campbell et al., 2024], FOLDFLOW-2 leverages the representational power of a large pre-trained protein language model in ESM [Lin et al., 2022] enabling it to make use of the rich biological inductive bias found in sequences but at a scale far beyond ground-truth experimental 3D structures found in the Protein Data Bank (PDB).

As a sequence-conditioned model, FOLDFLOW-2 is able to tackle a suite of new tasks beyond simple unconditional generation. Specifically, our model can additionally be used for protein folding by simply generating structures conditioned on sequence as well as hard, biologically motivated conditional design problems. For instance, our model can perform partial structure generation by conditioning on a masked sequence, i.e., structure in-painting. This enables FOLDFLOW-2 to be better equipped than prior structure-only generative models to tackle the key challenges in de novo drug design. For example, in settings where we aim to engineer a structure that binds and neutralizes a desired target protein structure and sequence pair; this is precisely a structure and sequence in-painting problem.

As diversity and quantity of training samples play a crucial role in downstream generative modeling performance on conditional design tasks, we construct a new large dataset—an order of magnitude larger than PDB—of high-quality synthetic structures filtered from SwissProt [Jumper et al., 2021, Varadi et al., 2021]. We further investigate the impact of fine-tuning FOLDFLOW-2 using Reinforced Fine-Tuning (ReFT), a new approach that aligns flow-matching generative models to arbitrary rewards. In the context of protein backbone generation, we apply fine-tuning to improve the properties of generated backbones, such as optimizing for the diversity of secondary structures, as well as improving the performance on conditional generation tasks like generating scaffolds around a target motif.

Main results. We summarize the main empirical results obtained using FOLDFLOW-2 below:

- We empirically demonstrate that FOLDFLOW-2 achieves state-of-the art performance for unconditional generation and leads to the most *designable, novel, and diverse* proteins. In particular, FOLDFLOW-2 improves over RFDiffusion [Watson et al., 2023] and FoldFlow [Bose et al., 2024].
- We find FOLDFLOW-2 closes the gap in performance with purpose-built folding models like ESMFold and improves by a factor of $\approx 4\times$ compared to MultiFlow [Campbell et al., 2024], the most comparable protein structure generative model that also leverages sequences.
- We use FOLDFLOW-2 to solve a biologically relevant conditional design problem in motif scaffolding. We find that a fine-tuned FOLDFLOW-2 is able to solve all $24/24$ scaffolds in the benchmark dataset from Watson et al. [2023]. On challenging VHH nanobodies, it solves $9/25$ refoldable motifs in comparison to $5/25$ for the previous best approach RFDiffusion.
- We hypothesize that FOLDFLOW-2 is able to perform zero-shot equilibrium conformation sampling on unseen proteins in the ATLAS molecular dynamics (MD) dataset [Vander Meersche et al., 2024] based on conformation variation seen within the protein data bank. We observe that FOLDFLOW-2 is able to capture different modes of the equilibrium conformation comparably to ESMFlow-MD [Jing et al., 2024], a model fine-tuned on MD data, but lags behind AlphaFlow-MD [Jing et al., 2024].

2

2 Background and preliminaries

2.1 Protein backbone and sequence parametrizations

Sequence representation. Protein sequences correspond to the chain of amino acids, which for a protein of length N is identified by a discrete token $a^i \in \{1, \ldots, 20\} =: \mathcal{A}$. As is customary in protein language models [Lin et al., 2022], these discrete tokens are encoded using a one-hot representation. We denote the entire amino acid sequence associated with a protein as $A \in \mathbb{R}^{N \times 20}$.

Structure representation. The 3D structure of protein backbones can be represented as rigid frames associated with each residue in an amino acid sequence [Jumper et al., 2021]. Each residue, i, within a protein backbone of length N consists of idealized coordinates of their 4 heavy atoms $N^*, C_\alpha^*, C^*, O^* \in \mathbb{R}^3$, with $C_\alpha^* = (0,0,0)$. The defining property of rigid frames is that they can be viewed as elements of the special Euclidean group $SE(3)$ and as such each frame $x = (r, s) \in SE(3)$ contains a rotation r and translation s component. Applying a rigid transformation x^i to the idealized coordinates of the heavy atoms allows us to represent the rigid frame of a given residue, $[N, C_\alpha, C, O]^i = x^i \circ [N^*, C_\alpha^*, C^*, O^*]$, where $\circ$ is the binary operator associated to the group, which for $SE(3)$ is simply matrix multiplication. This leads to a structure representation of the complete 3D coordinates associated with all heavy atoms of a protein as the tensor $X \in \mathbb{R}^{N \times 4 \times 3}$.

$SE(3)$**: the group of rigid motions.** The special Euclidean group $SE(3)$ contains rotations and translations in three dimensions and can be thought of in several ways. It is a Lie group, i.e., a differentiable manifold endowed with a group structure. $SE(3)$ can be seen as the group of rigid frames, representing 3D rotations and translations. As a Lie group, $SE(3)$ can be uniquely identified with its Lie algebra, the tangent space at the identity element of the group. $SE(3)$ is also a matrix Lie group, meaning that its elements can be represented with matrices. It can formally be written as the semidirect product of the rotation and the translation groups, $SE(3) \cong SO(3) \ltimes (\mathbb{R}^3, +)$. A more detailed introduction to Riemannian manifolds and Lie theory, with an emphasis on $SE(3)$ is provided in §A.

2.2 Flow matching on the $SE(3)$ group

As Lie groups are smooth manifolds, they can also be equipped with a Riemannian metric, which can be used to define distances and geodesics on the manifold. On $SE(3)$, a natural choice of the metric decomposes into the metrics on its constituent subgroups, $SO(3)$ and $\mathbb{R}^3$ [Bose et al., 2024, Yim et al., 2023b]. This allows us to build independent flows on the group of rotations and translations and induce a flow directly on $SE(3)$. As flow matching on Euclidean spaces is well-studied [Albergo and Vanden-Eijnden, 2023, Lipman et al., 2023, Liu et al., 2023], we restrict our focus on reviewing flows, conditional probability paths, and vector fields over the group $SO(3)$.

Probability paths on $SO(3)$**.** Given two densities $\rho_0, \rho_1 \in SO(3)$, a probability path $\rho_t : [0,1] \to \mathbb{P}(SO(3))$ is an interpolation, parametrized by time, t, between the two densities in probability space. Without loss of generality, we may consider ρ_0 to be the target data distribution and ρ_1 an easy-to-sample source distribution. A *flow* is a one-parameter diffeomorphism in t, $\psi_t : SO(3) \to SO(3)$. It is the solution to the ordinary differential equation (ODE): $\frac{d}{dt}\psi_t(r) = u_t(\psi_t(r))$, with initial condition $\psi_0(r) = r$, where u_t is the time-dependent smooth vector $u_t : [0,1] \times SO(3) \to SO(3)$. It is said that ψ_t *generates* ρ_t if it induces a pushforward map $\rho_t = [\psi_t]_\#(\rho_0)$.

Matching vector fields on $SO(3)$**.** The framework of Riemannian flow matching [Chen and Lipman, 2024] can also accommodate Lie groups such as $SO(3)$. Consequently, to learn a continuous normalizing flow (CNF) that pushes forward samples $r_0 \sim \rho_0$ to $r_1 \sim \rho_1$ we must regress a parametric vector field $v_\theta \in \mathfrak{X}(SO(3))$ in the tangent space of the manifold to the target conditional vector field $u_t(r_t|r_0, r_1)$, for all $t \in [0,1]$. Conveniently, the target $u_t(r_t|r_0, r_1)$ is the time derivative of a point r_t along the shortest path between r_0 and r_1—i.e., the geodesic interpolant $r_t = \exp_{r_0}(t \log_{r_0}(r_1))$. Furthermore, for $SO(3)$ the target conditional vector field admits a closed-form expression $u_t(r_t|r_0, r_1) = \log_{r_t}(r_0)/t$ as the exponential and logarithmic maps are numerically computable using the axis-angle representation of the group elements [Bose et al., 2024]. Given these ingredients, we can formulate the flow matching objective for $SO(3)$ as:

$$\mathcal{L}_{SO(3)}(\theta) = \mathbb{E}_{t \sim \mathcal{U}(0,1), q(r_0, r_1), \rho_t(r_t|r_0, r_1)} \left\| v_\theta(t, r_t) - \log_{r_t}(r_0)/t \right\|_{SO(3)}^2. \quad (1)$$

In eq. (1), $q(r_0, r_1)$ is any coupling between samples from the source and target distributions. An optimal choice is to set $q(r_0, r_1) = \pi(r_0, r_1)$ which is the coupling, π, that solves the Riemannian

3

optimal transport problem using minibatches [Bose et al., 2024, Tong et al., 2023, Fatras et al., 2020]. Finally, the generation of samples is done by first drawing from a source sample $r_1 \sim \rho_1$ and integrating the ODE backward in time using the learned vectorfield v_θ.

3 FOLDFLOW-2

We now present FOLDFLOW-2, our sequence-conditioned structure generative model. FOLDFLOW-2 operates on protein backbones $x_0 \sim \rho_0$ which are parametrized as N rigid frames as well as their corresponding sequence a. As protein backbones contain symmetries from $\mathrm{SE}(3)$, we design FOLDFLOW-2 as an $\mathrm{SE}(3)^N$-invariant density using a flow-matching objective. We achieve translation invariance by constructing the flow on the subspace $\mathrm{SE}(3)^N_0$, where the center of mass of the inputs is removed. Additionally, we can focus on building flows on the group of rotations $\mathrm{SO}(3)$

Table 1: Overview of the conditioning capability of unconditional ($\emptyset$), folding (A), and inpainting (A, X) of various protein backbone generation models.

Method	$\emptyset$	A	(A, X)
AlphaFold [Jumper et al., 2021]	✗	✓	✗
RFDiffusion [Watson et al., 2023]	✓	✗	✗
Chroma [Ingraham et al., 2023]	✓	✗	✗
FrameDiff [Yim et al., 2023b]	✓	✗	✗
FOLDFLOW [Bose et al., 2024]	✓	✗	✗
FrameFlow [Yim et al., 2023a]	✓	✗	✓
Motif RFDiffusion [Watson et al., 2023]	✗	✗	✓
Multiflow [Campbell et al., 2024]	✓	✓	✓
FOLDFLOW-2 (Ours)	✓	✓	✓

and translations $\mathbb{R}^3$, for each of the N residues independently, as $\mathrm{SE}(3)^N_0$ can be viewed as a product manifold consisting of N copies of $\mathrm{SE}(3)_0$. The overall loss function for the model decomposes into per residue rotation and translation losses $\mathcal{L} = \mathcal{L}_{\mathrm{SO}(3)} + \mathcal{L}_{\mathbb{R}^3}$,

$$\mathcal{L} = \mathbb{E}_{t \sim \mathcal{U}(0,1), \rho_t(x_t | x_0, x_1, \bar{a})} \left[\left\| v_\theta(t, r_t, \bar{a}) - \log_{r_t} \frac{r_0}{t} \right\|^2_{\mathrm{SO}(3)} + \left\| v_\theta(t, s_t, \bar{a}) - \frac{(s_t - s_0)}{t} \right\|^2_2 \right], \quad (2)$$

where the pair $(x_0, x_1) \sim \pi(x_0, x_1)$ is sampled from the optimal transport plan π. In addition, the sequence $\bar{a} = a \odot m$ corresponds to x_0 and is masked completely, with a mask m, with a probability $\mathrm{Bern}(0.5)$. Operationally, this means 50% of the time the model is trained unconditionally with no sequence information, i.e., $\bar{a} = [\emptyset]^N$, while the other 50% the model has access to the full sequence $\bar{a} = a$. Optimizing the loss in eq. (2) is equivalent to maximizing the conditional log-likelihood of observing protein structures given their sequences $\log p(\mathrm{X}|\mathrm{A})$ when the sequence is not masked and maximizing the unconditional log-likelihood $\log p(\mathrm{X})$ when the sequence is fully masked. Due to the ability to mask sequences, FOLDFLOW-2 enables new modeling capabilities in comparison to existing models as outlined in table 1. More precisely, FOLDFLOW-2 trained using masked sequences can perform a diverse set of tasks, outlined in table 2, beyond simple unconditional backbone generation which aids in tackling more biologically relevant problems that require conditional generation such as mimicking a protein folding model and designing the 3D scaffolds around a target motif.

With the breadth of tasks **T1–T3** (table 2) FOLDFLOW-2 unlocks new structural design capabilities beyond the simple unconditional generation ability of FOLDFLOW. We next outline the architectural components of FOLDFLOW-2 in §3.1 before detailing the training procedure in §3.2, which includes key design decisions regarding the construction of our new scaled dataset of ground truth PDBs

Table 2: By manipulating the input modalities, FOLDFLOW-2 is able to perform a diverse set of conditional and unconditional generation tasks including biologically relevant tasks such as designing scaffolds.

	Task Name	Sequence Inputs	Structure Inputs
(T1)	Unconditional	Fully-masked	Noise
(T2)	Folding	Unmasked	Noise
(T3)	In-Painting	Partially Masked	Partially Masked

and filtered AlphaFold2 synthetic structures. We also outline the inference procedure for sampling in §B.4. We conclude by discussing various techniques to fine-tune FOLDFLOW-2, including methods based on filtering with auxiliary rewards for supervised fine-tuning §3.3 to align protein structures.

3.1 FOLDFLOW-2 Architecture

The FOLDFLOW-2 architecture is comprised of three core components: (1) **Structure and sequence encoder:** An encoder which encodes both structures and sequences; (2) **Multi-modal fusion trunk:** the trunk which combines the multi-modal representations of the encoded structure and sequences; and (3) **Geometric Decoder:** a decoder that consumes the fused representation from the trunk and outputs a generated structure. The overall architecture of FOLDFLOW-2 is depicted in fig. 1.

4

Figure 1: FOLDFLOW-2 architecture which processes sequence and structure and outputs $\mathrm{SE}(3)_0^N$ vectorfields.

Structure and Sequence Encoder. We leverage existing state-of-the-art architectures to encode the structure and sequence modalities separately. For structure encoding, we rely on the invariant point attention (IPA) transformer architecture [Jumper et al., 2021], which is $\mathrm{SE}(3)$-equivariant. The benefit of the IPA architecture is that it is highly flexible and can both consume and produce a structure—i.e., N rigid frames—*and* also output single and pair representations of the input structure.

To encode amino-acid sequences, we use a large pre-trained protein language model: the $650\mathrm{M}$ variant of the ESM2 sequence model [Lin et al., 2022]. Large protein language models have a strong inductive bias on atomic-level predictions of protein structures while exhibiting strong generalization properties beyond any known experimental structures—which we argue is highly correlated with goals of *de novo* structure design. Moreover, the ESM2 architecture also produces single and pair representations for an encoded sequence of amino acids, which conceptually correspond to the single and pair representations from the structure encoder. Consequently, the output space of each modality prescribes a natural fusion of representations into a joint single and pair latent space for a given input protein.

Multi-Modal fusion trunk. After encoding both input structure and sequence, we construct a joint representation for the single and pair representation using a "project and concatenate" combiner module with simple MLPs, see fig. 1. We use LayerNorm [Ba et al., 2016] throughout the architecture as it is essential to accommodate differently-scaled inputs. The joint representations are further processed by a series of Folding blocks [Lin et al., 2023], which refines the single and pair representations via triangular self-attention updates.

Geometric decoder. To decode the joint representations of the inputs into $\mathrm{SE}(3)_0^N$ vector fields, we once again leverage the IPA Transformer architecture. The decoder takes as input the single, pair outputs of the trunk *and* the rigid representations from the structure encoder. One of our major findings is that including a skip-connection between the structure encoder and the decoder is essential for good performance as the temporal information is only given to the structure encoder.

Given each component, we stack $2-2-2$ blocks for the encoder, trunk, and decoder components.

3.2 Training

We train FOLDFLOW-2 by alternating between both folding and unconditional generation tasks using a novel sequence-and-structure flow matching procedure, described below.

Dataset construction. The generalization ability of generative models trained using maximum likelihood is determined by the quality and diversity of curated training data [Kadkhodaie et al.,

5

2024]. Due to the limited size of ground truth structures in the Protein Data Bank (PDB) we aim to improve training set diversity by additionally curating a dataset of filtered AlphaFold2 structures from SwissProt [Jumper et al., 2021, Varadi et al., 2021]. To ensure FOLDFLOW-2 is trained on high-quality synthetic structures, we employ a set of stringent filtering techniques that remove many undesignable proteins from SwissProt. After filtering, our final dataset consists of $160K$ structures and constitutes approximately an $8\times$ fold increase compared to prior works [Yim et al., 2023b, Bose et al., 2024]. Our exact layered filtering strategy for synthetic structures in SwissProt is outlined by the following steps:

(Step 1) **Filtering low-confidence structures.** We use per-residue local confidence metrics like the average pLDDT to filter out low-confidence structures from the initial SwissProt dataset.

(Step 2) **Masking low-confidence residues.** Globally high-confident structures may include low-confidence residues with disordered regions that can impede training. We use a per-residue pLDDT threshold of 70 to mask such "low-quality" residues during training.

(Step 3) **Filter high-confidence, low-quality structures.** The nature of synthetic data means that even following steps 1 and 2 low-quality data persists in a curated dataset. To combat this we further filter structures by learning a light-weight structure prediction model trained on structural features predictive of protein quality.

We report a detailed analysis of each step in the filtration process in §B.1 which includes examples of low-quality structures that were filtered as illustrative examples. The impact of these findings is empirically corroborated in by analyzing generated samples from FOLDFLOW-2 in §C.2.

During training, we set the fraction of synthetic samples that may be seen during an epoch to $2/3$ of the epoch. This prevents the model from overfitting to the remaining noise in the synthetic data, and is also common practice when training with synthetic data [Hsu et al., 2022, Lin et al., 2023]. Anecdotally, we did not notice an improvement from using a smaller proportion of synthetic structures. Finally, in the FOLDFLOW-2 architecture, we keep the ESM pre-trained language model fixed during training and train all other components (encoder, trunk, and decoder) from scratch. The results presented in table 3 and §4.4 use PDB data only, as this displayed the best performance for designability scores.

3.3 Fine-Tuning FOLDFLOW-2

We explore the efficacy of fine-tuning FOLDFLOW-2 with preferential alignment. We take a supervised fine-tuning approach [Wei et al., 2022] that uses an additional fine-tuning dataset which is filtered using pre-specified auxiliary rewards r_{aux} to create a preferential dataset $\mathcal{D}_{\text{pref}}$. We term this Reinforced FineTuning (ReFT) since fine-tuning in this manner can be considered aligning FOLDFLOW-2 generations to the auxiliary reward. Summarizing this in three steps: (1) We take a curated dataset of proteins with desirable metrics; (2) We use r_{aux} to score the samples from step 1 and filter them to get a subset of high-scoring samples; (3) We then improve FOLDFLOW-2 by SFT on the filtered subset. Finetuning with ReFT optimizes the following optimization objective $\mathcal{L}_{\text{REFT}}(\theta)$,

$$\max_{p_\theta} \mathcal{L}_{\text{REFT}}(\theta) = \mathbb{E}_{(x,a)\sim\mathcal{D}_{\text{pref}}} \left[r_{\text{aux}}(x) \log p_\theta(x|a) \right]. \qquad (3)$$

Compared to recent alignment methods based on reward models, as in RLHF [Bai et al., 2022], ReFT uses a filtered structure dataset to fine-tune FOLDFLOW-2. Standard RL approaches seek to fine-tune generative model-based model-generated data and assume access to evaluating the reward function. Our approach maximizing $\mathcal{L}_{\text{REFT}}(\theta)$ requires constructing $\mathcal{D}_{\text{pref}}$ with auxiliary reward r_{aux}, demonstrated by the improvement in secondary structure diversity in §4.2.

4 Experiments

We evaluate FOLDFLOW-2 on multiple protein design tasks including unconditional generation, motif scaffolding, folding, fine-tuning to improve secondary structure diversity, and equilibrium conformation sampling from molecular dynamics trajectories. We provide implementation details in §B.

Baselines. As our main baselines for the unconditional generation task we rely on pre-trained versions of FrameDiff [Yim et al., 2023b], Chroma [Ingraham et al., 2023], Genie [Yeqing and Mohammed, 2023], MultiFlow [Campbell et al., 2024], and RFDiffusion which is the current gold standard [Watson et al., 2023]. In conditional generation tasks like motif scaffolding, we compare against a conditional variant of FrameFlow [Yim et al., 2023a] as well as RFDiffusion. For protein folding, we focus on comparing against ESMFold [Lin et al., 2022] and MultiFlow which also

6
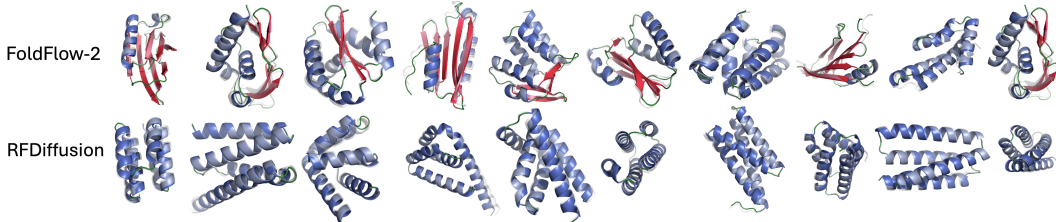
Figure 2: Uncurated designable (scRMSD < 2Å) length 100 structures with ESMFold refolded structure from FOLDFLOW-2 and RFDiffusion colored by secondary structure assignment. FOLDFLOW-2 is significantly more diverse in terms of secondary structure composition where we see RFDiffusion generates mostly α-helices.

leverages sequence information. Lastly, for conformational sampling the principal baselines are ESMFlow and AlphaFlow [Jing et al., 2024].

4.1 Unconditional protein backbone generation

We evaluate unconditional structure generation using metrics that assess the designability, novelty, and diversity of generated structures. For each method, we generate 50 proteins at lengths $\{100, 150, 200, 250, 300\}$ (c.f. FOLDFLOW-2 samples in fig. 12). Designability is computed by using the *self-consistency* metric which compares the refolded proteins (with ProteinMPNN [Dauparas et al., 2022] and ESMFold [Lin et al., 2022]) with the original one. Novelty is computed using: 1.) the fraction of designable proteins with TM-score < 0.3 and 2.) the average maximum TM-score of designable generated proteins to the training data. Finally, diversity uses the average pairwise TM-score designable samples averaged across lengths as well as the maximum number of clusters.

Table 3: Comparison of Designability (fraction with scRMSD < 2.0Å), Novelty (max. TM-score to PDB and fraction of proteins with averaged max. TM-score < 0.3 and scRMSD < 2.0Å), and Diversity (avg. pairwise TM-score and MaxCluster fraction). Designability and Novelty metrics include standard errors.

	Designability	Novelty		Diversity	
	Frac. < 2Å (↑)	Frac. TM < 0.3 (↑)	avg. max TM (↓)	pairwise TM (↓)	MaxCluster (↑)
RFDiffusion	0.969 ± 0.023	0.116 ± 0.020	0.449 ± 0.012	0.256	0.172
Chroma	0.636 ± 0.030	0.214 ± 0.033	0.412 ± 0.011	0.272	0.132
Genie	0.581 ± 0.064	0.120 ± 0.021	0.434 ± 0.016	0.228	0.274
FrameDiff	0.402 ± 0.062	0.020 ± 0.009	0.542 ± 0.046	0.237	0.310
FOLDFLOW	0.820 ± 0.037	0.188 ± 0.025	0.460 ± 0.020	0.247	0.228
FOLDFLOW-2	0.976 ± 0.010	0.368 ± 0.031	0.363 ± 0.009	0.205	0.348

Results. We see that FOLDFLOW-2 outperforms all other methods on all metrics—crucially without requiring a pretrained folding model as part of the architecture like RFDiffusion. In particular, we observe that FOLDFLOW-2 produces the most designable samples with 97.6% of samples being refolded by ESMFold to within < 2Å. We also find that FOLDFLOW-2 novelty improves over RFDiffusion by an absolute 25.2% in the fraction of designable samples with TM-score < 0.3. Furthermore, we observe 19.9% and 102.3% relative improvement in the diversity of FOLDFLOW-2 over RFDiffusion as measured by the pairwise TM-score and Max Cluster fraction. This places FOLDFLOW-2 as the current *most designable, novel, and diverse* protein structure generative model.

We present uncurated generated samples of FOLDFLOW-2 and RFDiffusion in fig. 2. We further visualize the distribution of secondary structures of all methods in fig. 3. We see a clear indication that FOLDFLOW-2 is able to produce the most diverse secondary structures—more closely matching the training distribution (see fig. 3e)—and improving over RFDiffusion. We further observe increased amounts of β-sheets and coils which are particularly challenging for models like FrameDiff and FOLDFLOW that primarily generate α-helices. We also include multiple ablations on architectural choices, inference annealing, and sequence conditioning in table 14.

4.2 Increasing secondary structure diversity with finetuning

We investigate ReFT based data filtering to improve diversity of secondary structures in generated samples. We use a diversity score based auxiliary reward for filtering, based on weighted

7
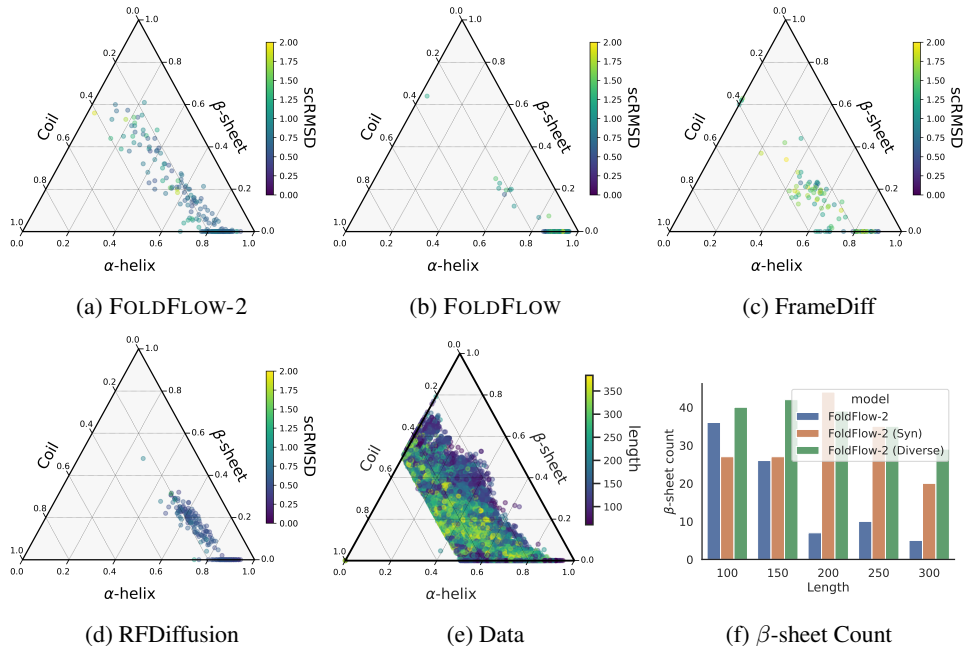
Figure 3: Distribution of secondary structure elements (α-helices, β-sheets, and coils) of designable (scRMSD < 2.0) proteins generated by various models. FOLDFLOW-2 generates more diverse designable backbones.

entropy on the proportions of each residue belonging to each type of secondary structure—i.e., alpha-helices (α), coils c, beta-sheets β in the set $\mathcal{S}$, that can be analytically written as $r_{\text{diversity}} = \left(\sum_{s \in \mathcal{S}} p_s w_s\right)\left(1 + \sum_{s \in \mathcal{S}} p_s \log p_s\right)$. Due to models producing increasing amounts of helices, we use $w_\alpha = 1$, $w_c = 0.5$ and $w_\beta = 2$, and take top 25% of samples according to the $r_{\text{diversity}}$. Experimental results in fig. 3f with generated samples in fig. 12 demonstrate that protein at all lengths benefit from training with ReFT as measured by diversity of generated samples, and produces most amount of β-sheets, and can surpass diversity improvement already obtained by training using synthetic structures as in fig. 3.

4.3 Folding sequences

Given that FOLDFLOW-2 is sequence conditioned, we can perform protein folding by providing a valid sequence during inference. During training, FOLDFLOW-2 tries to transform a $\mathrm{SE}(3)_0^N$ noise sample into the given sequence's 3D structure. Therefore, we aim to measure the generalization properties of our model to fold unseen sequences. We evaluate folding on a test set of 268 unseen proteins from the PDB dataset. We compare the folding capabilities of FOLDFLOW-2, ESMFold, and Multiflow. In table 4, we report the aligned RMSD between the predicted backbone and the ground truth backbone. We

Table 4: Folding model evaluation on a test set of 268 proteins from PDB.

Model	RMSD ($\downarrow$)
ESMFold	2.322 ± 4.270
MultiFlow	14.995 ± 3.977
FOLDFLOW-2	3.237 ± 4.145

find that FOLDFLOW-2, trained for structure generation, approaches the performance of ESMFold which is a purpose-built folding model. We contextualize this result by noting that FOLDFLOW-2 $\approx 4\times$ is better at folding than the most comparable model in MultiFlow [Campbell et al., 2024] which is a multi-modal flow matching model using sequences.

4.4 Motif Scaffolding

In motif scaffolding, we are tasked with designing a subset of residues, termed "scaffolds", around one or more subsections of a ("motif") protein structure that have known biologically-important functions through its interaction with a target. This enables the design of proteins with *a priori* functional sites using generative models [Wang et al., 2021, Watson et al., 2023]. The motifs can be

8

small and have non-specific shapes (e.g. a helix), and hence it is important for the generative model to understand the chemical information it carries on top of its geometry. We thus consider the task of motif scaffolding as an example of how our model can be fine-tuned for conditional generation tasks. We consider two datasets for evaluating motif scaffolding performance: the benchmark proposed in Watson et al. [2023] consisting of 24 single-chain motifs, and a new benchmark based on scaffolding the Complimentary Determining Regions (CDRs) of VHH nanobodies, as found in the Structural Antibody Database [Schneider et al., 2022].

Motif Scaffolding Benchmark. We use the scaffolding benchmark from Watson et al. [2023] and follow the pseudo-label fine-tuning procedure described in Yim et al. [2023b] by randomly generating motifs from proteins by training on both the motif structure *and* sequence. For inference, we sample the scaffold lengths for each motif and provide both the both partially masked structure and sequence to the model. We follow the same evaluation procedure used in RFDiffusion (c.f. §B.6 for details). Our results in table 5 show that both FOLDFLOW-2 and RFDiffusion solve all $24/24$ motifs.

CDR Scaffolding. VHH antibodies, also known as nanobodies, have shown significant promise in protein design and therapeutics due to their unique properties [Muyldermans, 2021]. They are composed of a single variable domain derived from camelid heavy-chain antibodies, featuring three complementarity-determining regions (CDRs) that confer specificity and variability in antigen binding. As a result, creating effective scaffolds for nanobodies is challenging due to the need to maintain the designability of the CDRs and especially because any scaffolding effort must avoid altering these characteristics to preserve binding functionality. We treat this as a conditional generative modeling problem and fix the motif atoms, and mask the scaffold sequence information. Exact training and experimental details along with additional metrics are provided in §B.6. Our results are found in table 5, where the average motif scRMSD is much higher than the average scaffold scRMSD. The result is a much lower number of solved motif scaffolding.

Table 5: Motif-scaffolding benchmarks. FrameFlow does not have public code for motif-scaffolding and thus cannot be evaluated on the VHH benchmark. "+FT" indicates "with fine-tuning". *Using reported numbers with AlphaFold2 instead of ESMFold used in our evaluation procedure; c.f. §B.6 for further discussion.

Benchmark	RFDiffusion	VHH			
Model	Solved /24 ↑	Diversity ↑	Motif ↓	Scaffold ↓	Solved /25 ↑
RFDiffusion	24	0.345	3.94 ± 1.54	2.40 ± 0.93	5
FrameFlow (+FT)*	21	–	–	–	–
FOLDFLOW-2 (+FT)	24	0.445	2.78 ± 1.01	1.67 ± 0.24	9

4.5 Zero-shot Equilibrium Conformation Sampling

We now test FOLDFLOW-2 on zero-shot equilibrium conformation sampling task. Starting from a sequence, we generate multiple conformations of the same proteins and compare the distribution of conformations with the ones from molecular dynamic simulations. We compare FOLDFLOW-2 with AlphaFlow-MD and ESMFlow-MD; two folding models fine-tuned on a molecular dynamic dataset, and non finetuned models Eigenfold [Jing et al.] and STR2STR [Lu et al., 2024]. In table 6, we report the pairwise and global RMSD, the root mean square fluctuation (RMSF), and the 2-Wasserstein on the top two principal components. For both RMSD and the RMSF metrics, we report the Pearson correlation

Figure 4: Protein conformation ensembles from the ATLAS dataset, ESMFlow-MD and FOLDFLOW-2. Proteins are colored by their secondary structure with α-helices in blue, β-sheets in red, and coils in green.

between the values from the generated ensemble and those of the ground truth ensemble (the procedure is detailed in full in §B.7).

We use the same test set as in Jing et al. [2024] restricted to proteins of length at most 400 amino acids. Notably FOLDFLOW-2 performs similarly or better than the comparable model ESMFlow-MD across

9

all metrics *without any fine-tuning* and *with significantly fewer parameters* on molecular dynamics data, indicating that the base model trained only on PDB already captures similar information about protein dynamics as models given explicit access to this data. Moreover, we observed that FOLDFLOW-2 requires $4.5\times$ less GPU hours for training and $33\times$ less trainable parameters while allowing for $6\times$ faster inference steps than ESMFlow-MD as reported in table 13, improving FOLDFLOW-2's prospects as a practical base model for future work on capturing protein dynamics.

Table 6: Zero-shot performance of the base FOLDFLOW-2 model on the ATLAS dataset of MD trajectories compared to ESMFlow and AlphaFlow models fine-tuned on ATLAS. FOLDFLOW-2 is competitive to the comparable model ESMFlow across all metrics. r denotes Pearson's correlation coefficient. Time is per sample (Time / sample (s)) on length 300 protein 7c45_A. Eigenfold only models $C\alpha$ atom so we compare on PCA $\mathcal{W}_2$-dist on the $C\alpha$ atom only.

Model	Pairwise RMSD r (↑)	Global RMSF r (↑)	Per-target RMSF r (↑)	PCA $\mathcal{W}_2$-dist (↓)	PCA $\mathcal{W}_2$-dist ($C\alpha$ only) (↓)	Time / sample (s)
AlphaFlow-MD	0.468 ± 0.005	0.415 ± 0.006	0.824 ± 0.000	10.67 ± 0.29	—	32.6 ± 0.1
ESMFlow-MD	0.293 ± 0.005	0.161 ± 0.005	0.737 ± 0.001	11.51 ± 0.13	—	11.2 ± 0.1
FOLDFLOW-2	0.297 ± 0.004	0.236 ± 0.004	0.658 ± 0.001	10.85 ± 0.15	5.273 ± 0.075	9.0 ± 0.0
STR2STR	0.279 ± 0.003	0.244 ± 0.005	0.586 ± 0.001	5.37 ± 0.05	2.678 ± 0.023	$10.6 \pm —$
Eigenfold	0.194 ± 0.004	0.156 ± 0.004	0.668 ± 0.001	—	5.296 ± 0.080	38.6 ± 0.5

5 Related work

Protein design. Physics-based protein structure design yielded the first de novo proteins [Huang et al., 2016]. For example, structure-based biophysics approaches have previously resulted in several drug candidates [Röthlisberger et al., 2008, Fleishman et al., 2011, Cao et al., 2020].This was followed by language models [Hie et al., 2022, Ferruz et al., 2022] and geometric deep learning [Gainza et al., 2020] for protein structure design. Recently, diffusion [Wu et al., 2024, Yim et al., 2023b, Watson et al., 2023, Ingraham et al., 2023, Wang et al., 2024, Frey et al., 2024] and flow-based models [Bose et al., 2024, Yim et al., 2023a, Jing et al., 2024] have risen to prominence. These methods employ a backbone-first approach with the notable exception of MultiFlow [Campbell et al., 2024] which uses sequence to perform co-generation.

RLHF and Supervised Fine-Tuning (SFT). Aligning the outputs of language models with RLHF has recently gained interest [Ouyang et al., 2022, Stiennon et al., 2020, Bai et al., 2022]. These methods learn a reward model for post-training alignment to desired behavior [Mishra et al., 2022], which can prove challenging for protein design [Zhou et al., 2024]. SFT on hand-crafted data has proven to be effective in enhancing performance but requires high-quality data [Rozière et al., 2023, Yuan et al., 2023]. Filtering real data using auxiliary rewards serve as a substitute for steering the desired properties of the generated samples.

6 Conclusion

In this paper, we introduce a new sequence-conditioned protein structure generative model called FOLDFLOW-2. FOLDFLOW-2 leverages a protein language model to condition Flow Matching-based protein generative models with sequences. Our model achieves state-of-the-art results on unconditional generation and generates diverse and novel proteins, especially when trained on our new dataset. Conditioning over sequences allows our model to perform novel tasks such as folding sequences and motif-scaffolding tasks and we show its competitiveness on those tasks. Regarding the limitations of our model, we note that it requires a competitive pre-trained language model to be sequence conditioned, which can be hard to acquire. We also note that ProteinMPNN, used in our evaluation pipeline, has been trained only on PDBs. Therefore, it is possible that our models trained on our new dataset generates designable proteins which are not correctly processed by ProteinMPNN.

Acknowledgements

We thank Alexandre Stein, Maksym Korablyov, and the entire DreamFold team for providing a vibrant workspace that enabled this research. The authors would like to acknowledge Anyscale, Amazon AWS, and Google GCP for providing computational resources for the protein experiments.

10

Contribution statement

Architecture design was led by G.H. Infrastructure development was led by J.V. The experiments were divided as follows: Unconditional (K.F., A.T.), diversity (E.T.L., R.I., C.L.), folding (G.H.), motif-scaffolding (G.H., J.V., E.T.L, C.L.), and equilibrium conformation (J.R.B, G.H., T.A.S.). Dataset preparation including synthetic structure filtering (J.V., P.L., K.F.). A.J.B. drove the writing of the paper with contributions from all other authors. A.J.B. and A.T. cosupervised the project with guidance from M.B.

References

M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *International Conference on Learning Representations (ICLR)*, 2023. (Cited on page 3)

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *Advances in neural information processing systems*, 2016. (Cited on page 5)

Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. E. Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. B. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, and J. Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv*, 2022. (Cited on pages 6 and 10)

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, Jan. 2000. (Cited on page 17)

A. J. Bose, T. Akhound-Sadegh, K. Fatras, G. Huguet, J. Rector-Brooks, C.-H. Liu, A. C. Nica, M. Korablyov, M. Bronstein, and A. Tong. Se(3)-stochastic flow matching for protein backbone generation. In *International Conference on Learning Representations (ICLR)*, 2024. (Cited on pages 2, 3, 4, 6, 10, 17, 19, 20, and 21)

A. Campbell, J. Yim, R. Barzilay, T. Rainforth, and T. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *International Conference on Machine Learning (ICML)*, 2024. (Cited on pages 2, 4, 6, 8, and 10)

L. Cao, I. Goreshnik, B. Coventry, J. B. Case, L. Miller, L. Kozodoy, R. E. Chen, L. Carter, A. C. Walls, Y.-J. Park, et al. De novo design of picomolar sars-cov-2 miniprotein inhibitors. *Science*, 370(6515):426–431, 2020. (Cited on pages 1 and 10)

S. R. Carter, S. Curtis, C. Emerson, J. Gray, I. C. Haydon, A. Hebbeler, C. Qureshi, N. Randolph, A. Rives, and L. Stuart. Community values, guiding principles, and commitments for the responsible development of ai for protein design, 2024. (Cited on page 16)

R. T. Q. Chen and Y. Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on page 3)

A. Chevalier, D.-A. Silva, G. J. Rocklin, D. R. Hicks, R. Vergara, P. Murapa, S. M. Bernard, L. Zhang, K.-H. Lam, G. Yao, et al. Massively parallel de novo protein design for targeted therapeutics. *Nature*, 550(7674):74–79, 2017. (Cited on page 1)

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. (Cited on pages 7, 20, and 21)

S. B. Ebrahimi and D. Samanta. Engineering protein-based therapeutics through structural and chemical design. *Nature Communications*, 14(1):2411, 2023. (Cited on page 1)

11

K. Fatras, Y. Zine, R. Flamary, R. Gribonval, and N. Courty. Learning with minibatch wasserstein : asymptotic and gradient properties. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020. (Cited on pages 4 and 17)

K. Fatras, Y. Zine, S. Majewski, R. Flamary, R. Gribonval, and N. Courty. Minibatch optimal transport distances; analysis and applications. *arXiv*, 2021. (Cited on page 17)

N. Ferruz, S. Schmidt, and B. Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022. (Cited on page 10)

S. J. Fleishman, T. A. Whitehead, D. C. Ekiert, C. Dreyfus, J. E. Corn, E.-M. Strauch, I. A. Wilson, and D. Baker. Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science*, 332(6031):816–821, 2011. (Cited on page 10)

N. C. Frey, D. Berenberg, K. Zadorozhny, J. Kleinhenz, J. Lafrance-Vanasse, I. Hotzel, Y. Wu, S. Ra, R. Bonneau, K. Cho, et al. Protein discovery with discrete walk-jump sampling. *International Conference on Learning Representations (ICLR)*, 2024. (Cited on page 10)

P. Gainza, F. Sverrisson, F. Monti, E. Rodola, D. Boscaini, M. M. Bronstein, and B. E. Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020. (Cited on page 10)

P. Gainza, S. Wehrle, A. Van Hall-Beauvais, A. Marchand, A. Scheck, Z. Harteveld, S. Buckley, D. Ni, S. Tan, F. Sverrisson, et al. De novo design of protein interactions with learned surface fingerprints. *Nature*, pages 1–9, 2023. (Cited on page 1)

B. C. Hall. *Lie groups, Lie algebras, and representations*. Springer, 2013. (Cited on page 16)

A. Herbert and M. Sternberg. Maxcluster: a tool for protein structure comparison and clustering, 2008. (Cited on page 21)

B. Hie, S. Candido, Z. Lin, O. Kabeli, R. Rao, N. Smetanin, T. Sercu, and A. Rives. A high-level programming language for generative protein design. *bioRxiv*, pages 2022–12, 2022. (Cited on page 10)

C. Hsu, R. Verkuil, J. Liu, Z. Lin, B. Hie, T. Sercu, A. Lerer, and A. Rives. Learning inverse folding from millions of predicted structures. *International conference on machine learning*, 2022. (Cited on page 6)

P.-S. Huang, S. E. Boyken, and D. Baker. The coming of age of de novo protein design. *Nature*, 537 (7620):320–327, 2016. (Cited on page 10)

J. B. Ingraham, M. Baranov, Z. Costello, K. W. Barber, W. Wang, A. Ismail, V. Frappier, D. M. Lord, C. Ng-Thow-Hing, E. R. Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023. (Cited on pages 4, 6, 10, and 21)

B. Jing, E. Erives, P. Pao-Huang, G. Corso, B. Berger, and T. S. Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. In *ICLR 2023-Machine Learning for Drug Discovery workshop*. (Cited on page 9)

B. Jing, B. Berger, and T. Jaakkola. Alphafold meets flow matching for generating protein ensembles. *International Conference on Machine Learning (ICML)*, 2024. (Cited on pages 2, 7, 9, 10, 23, and 24)

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. (Cited on pages 2, 3, 4, 5, 6, 18, and 19)

Z. Kadkhodaie, F. Guth, E. P. Simoncelli, and S. Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. In *International Conference on Learning Representations (ICLR)*, 2024. (Cited on page 5)

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (Cited on page 19)

12

Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022. (Cited on pages 2, 3, 5, 6, 7, 19, and 20)

Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. (Cited on pages 5, 6, and 21)

Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. (Cited on page 3)

X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *International Conference on Learning Representations (ICLR)*, 2023. (Cited on page 3)

J. Lu, B. Zhong, Z. Zhang, and J. Tang. Str2str: A score-based framework for zero-shot protein conformation sampling. *International Conference on Learning Representations (ICLR)*, 2024. (Cited on page 9)

S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3470–3487. Association for Computational Linguistics, 2022. (Cited on page 10)

L. S. Mitchell and L. J. Colwell. Comparative analysis of nanobody sequence and structure data. *Proteins: Structure, Function, and Bioinformatics*, 86(7):697–706, 2018. (Cited on page 23)

S. Muyldermans. Applications of nanobodies. *Annual review of animal biosciences*, 9:401–421, 2021. (Cited on page 9)

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS*, 2022. (Cited on page 10)

F. C. Park and R. W. Brockett. Kinematic dexterity of robotic mechanisms. *The International Journal of Robotics Research*, 13(1):1–15, 1994. (Cited on page 17)

A.-A. Pooladian, H. Ben-Hamu, C. Domingo-Enrich, B. Amos, Y. Lipman, and R. T. Chen. Multi-sample flow matching: Straightening flows with minibatch couplings. *International Conference on Learning Representations (ICLR)*, 2023. (Cited on page 27)

D. Röthlisberger, O. Khersonsky, A. M. Wollacott, L. Jiang, J. DeChancie, J. Betker, J. L. Gallaher, E. A. Althoff, A. Zanghellini, O. Dym, et al. Kemp elimination catalysts by computational enzyme design. *Nature*, 453(7192):190–195, 2008. (Cited on page 10)

B. Rozière, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin, A. Kozhevnikov, I. Evtimov, J. Bitton, M. Bhatt, C. Canton-Ferrer, A. Grattafiori, W. Xiong, A. Défossez, J. Copet, F. Azhar, H. Touvron, L. Martin, N. Usunier, T. Scialom, and G. Synnaeve. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950, 2023. (Cited on page 10)

C. Schneider, M. I. Raybould, and C. M. Deane. Sabdab in the age of biotherapeutics: updates including sabdab-nano, the nanobody structure tracker. *Nucleic acids research*, 50(D1):D1368–D1372, 2022. (Cited on page 9)

D.-A. Silva, S. Yu, U. Y. Ulge, J. B. Spangler, K. M. Jude, C. Labão-Almeida, L. R. Ali, A. Quijano-Rubio, M. Ruterbusch, I. Leung, et al. De novo design of potent and selective mimics of il-2 and il-15. *Nature*, 565(7738):186–191, 2019. (Cited on page 1)

13

N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020. (Cited on page 10)

E.-M. Strauch, S. M. Bernard, D. La, A. J. Bohn, P. S. Lee, C. E. Anderson, T. Nieusma, C. A. Holstein, N. K. Garcia, K. A. Hooper, et al. Computational design of trimeric influenza-neutralizing proteins targeting the hemagglutinin receptor binding site. *Nature biotechnology*, 35(7):667–671, 2017. (Cited on page 1)

A. Tong, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, K. Fatras, G. Wolf, and Y. Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint 2302.00482*, 2023. (Cited on pages 4 and 27)

Y. Vander Meersche, G. Cretin, A. Gheeraert, J.-C. Gelly, and T. Galochkina. ATLAS: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic Acids Research*, 52 (D1):D384–D392, 11 2023. (Cited on page 23)

Y. Vander Meersche, G. Cretin, A. Gheeraert, J.-C. Gelly, and T. Galochkina. Atlas: protein flexibility description from atomistic molecular dynamics simulations. *Nucleic Acids Research*, 52(D1): D384–D392, 2024. (Cited on page 2)

M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, and S. Velankar. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444, 11 2021. (Cited on pages 2 and 6)

C. Wang, Y. Qu, Z. Peng, Y. Wang, H. Zhu, D. Chen, and L. Cao. Proteus: exploring protein structure generation for enhanced designability and efficiency. *International Conference on Machine Learning (ICML)*, 2024. (Cited on page 10)

J. Wang, S. Lisanza, D. Juergens, D. Tischer, I. Anishchenko, M. Baek, J. L. Watson, J. H. Chun, L. F. Milles, J. Dauparas, et al. Deep learning methods for designing proteins scaffolding functional sites. *BioRxiv*, 2021. (Cited on page 8)

J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. (Cited on pages 2, 4, 6, 8, 9, 10, 19, 20, 21, 22, and 23)

J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *International Conference on Learning Representations (ICLR)*, 2022. (Cited on page 6)

K. E. Wu, K. K. Yang, R. van den Berg, S. Alamdari, J. Y. Zou, A. X. Lu, and A. P. Amini. Protein structure generation via folding diffusion. *Nature Communications*, 15(1):1059, 2024. (Cited on page 10)

L. Yeqing and A. Mohammed. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *International Conference on Machine Learning (ICML)*, 2023. (Cited on pages 6, 20, and 21)

J. Yim, A. Campbell, A. Y. Foong, M. Gastegger, J. Jiménez-Luna, S. Lewis, V. G. Satorras, B. S. Veeling, R. Barzilay, T. Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023a. (Cited on pages 4, 6, 10, and 19)

14

J. Yim, B. L. Trippe, V. De Bortoli, E. Mathieu, A. Doucet, R. Barzilay, and T. Jaakkola. Se (3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*, pages 40001–40039, 2023b. (Cited on pages 2, 3, 4, 6, 9, 10, 19, and 21)

J. Yim, A. Campbell, E. Mathieu, A. Y. Foong, M. Gastegger, J. Jiménez-Luna, S. Lewis, V. G. Satorras, B. S. Veeling, F. Noé, et al. Improved motif-scaffolding with se (3) flow matching. *Transactions on Machine Learning Research*, 2024. (Cited on page 21)

Z. Yuan, H. Yuan, C. Li, G. Dong, C. Tan, and C. Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv*, 2023. (Cited on page 10)

Y. Zhang, I. A. Hubner, A. K. Arakaki, E. Shakhnovich, and J. Skolnick. On the origin and highly likely completeness of single-domain protein structures. *Proceedings of the National Academy of Sciences*, 103(8):2605–2610, 2006. (Cited on page 21)

X. Zhou, D. Xue, R. Chen, Z. Zheng, L. Wang, and Q. Gu. Antigen-specific antibody design via direct energy-based preference optimization. *arXiv*, 2024. (Cited on page 10)

15

Broader impact

The development of generative AI for protein backbone design holds significant promise for the fields of biotechnology and medicine. By enabling the precise engineering of protein structures, these models can accelerate the discovery of novel therapeutics and vaccines, potentially leading to more effective treatments for a wide range of diseases. Multi-modal and conditional generative models in the space of biotechnology are seeing rapid improvements, from which we have yet to see the full impact on modern and future medicine. We also acknowledge the potential for dual use of our results but, as discussed in Carter et al. [2024], the benefits of public research into computational drug discovery outweigh the potential drawbacks.

A   A short review of Riemannian geometry, Lie groups, and optimal transport

A.1   Riemannian manifolds

Informally, a *topological manifold*, $\mathcal{M}$ is a topological space that is locally Euclidean (i.e. homeomorphic to a Euclidean space). The manifold is said to be *smooth* or *differentiable* if it additionally is C^p differential for all p. An important notion is the *tangent space*, $\mathcal{T}_x\mathcal{M}$, which is attached to every point on the manifold, $x \in \mathcal{M}$. The disjoint union of all the tangent spaces is called a *tangent bundle*, $\mathfrak{X}(\mathcal{M})$. If in addition, the manifold is equipped with a *Riemannian metric*, g_x, it is said to be a *Riemannian manifold*. The notion of a Riemannian metric is used to define inner products on the tangent space at each point of the manifold. This means that for $\mathfrak{r}_1, \mathfrak{r}_2 \in \mathcal{T}_x\mathcal{M}$, $g_x(\mathfrak{r}_1, \mathfrak{r}_2) := \langle \mathfrak{r}_1, \mathfrak{r}_2 \rangle$. Similar to how inner products can be used to define key geometric properties, such as length and distance, the Riemannian metric allows us to define such notions on an arbitrary Riemannian manifold: the length of $\mathfrak{r}$, a vector in the tangent space of the manifold is defined by $|\mathfrak{r}| = \langle \mathfrak{r}, \mathfrak{r} \rangle^{1/2}$. Finally, an important property on Riemannian manifolds is *geodesics*, which generalizes the notion of shortest paths in Euclidean spaces. While in Euclidean space the shortest path between two points is the length of a straight line between them, on a manifold, the idea is to find the shortest smooth curve between two points, which is called a geodesic.

A.2   Lie groups

Symmetries refer to transformations of an object that preserve a certain structure. A set of continuous symmetries, paired with a composition operation satisfying group axioms is a *Lie group* $(G, \circ)$. More precisely, a group is a set paired with a group operation, $\circ : G \times G \to G$, which is associative, has an identity element and there is an inverse element for every element of the set. In addition to this group structure, a Lie group is also a smooth manifold, where the group operations of multiplication, $(x, y) \to xy$ for $x, y \in G$, and inversion, $x \to x^{-1}$, are smooth maps.

Given $y \in G$, we can define a diffeomorphism, $L_y : G \to G$ defined by $x \mapsto yx$, known as left multiplication. Given a vector field X on the group, we say that it is *left invariant*, if, under this left multiplication, it is left-invariant, meaning that $L_y^* X = X, \forall y \in G$. Note that L_y^* is the differential of left action, naturally identifying the tangent spaces, $\mathcal{T}_y \to \mathcal{T}_{yx}$. Therefore, given the group multiplication, we can *uniquely* define a left-invariant vector field with its values on the tangent space at the identity element of the group, $\mathcal{T}_e$. We can additionally equip this vector space, V, with a bilinear operation known as the *Lie bracket*, $[\cdot, \cdot] : V \times V \to V$, that satisfies the Jacobi identity and is anticommutative. Such a vector space is called a *Lie algebra*. The tangent space of Lie groups form Lie algebras and are denoted with $\mathfrak{G}$. The elements of the Lie algebra can be mapped into group elements using an invertible map called the *exponential map*, $\exp : \mathfrak{G} \to G$. The inverse is called the *logarithmic map* allowing us to go from the group elements to their corresponding elements in the Lie algebra, $\log : G \to \mathfrak{G}$.

Finally, we note that the set of $n \times n$ non-singular matrices forms a Lie group. The group operation is matrix multiplication and it can be seen as a smooth manifold that is an open subset of $\mathbb{R}^{2n}$. This group is known as the *General Linear Group*, $GL(n)$. Any closed subgroup of $GL(n)$ is known as a *matrix Lie group*, which are perhaps some of the most important Lie groups that are studied. In the case of these matrix Lie groups, the exponential and logarithmic maps also coincide with the matrix exponential and logarithm. For a more detailed overview of this subject, we refer the reader to Hall [2013].

16

A.3 The Special Euclidean Group in 3 Dimensions

One of the closed subgroups of $GL(n)$ that has been studied extensively in various fields is the $3D$ Special Orthogonal group, $SO(3)$. The elements of this group are 3×3 rotation matrices, namely $SO(n) = \{r \in GL(n) : r^T r = I, det(r) = 1\}$. Additionally, the translations of an object in $3D$ space by a translation vector s can also be seen as a matrix Lie group by considering the translation matrix, $\begin{pmatrix} I & s \\ 0 & 1 \end{pmatrix}$, where I is an 3×3 identity matrix. With the group operation being translations, this group is also a matrix group, denoted as $(\mathbb{R}^3, +)$.

Combining these, we can represent the *rigid transformations* of objects in $3D$ space with a group that encompasses both rotations and translations. This group is known as the *Special Euclidean group in 3D* and is the semidirect product of the rotation and translation groups, $SE(3) \cong SO(3) \ltimes (\mathbb{R}^3, +)$. This group is also a matrix group and its elements can be written as $SE(3) = \left\{ (r, s) = \begin{pmatrix} r & s \\ 0 & 1 \end{pmatrix} : r \in SO(3), s \in (\mathbb{R}^3, +) \right\}$. Finally, we note that given a suitable choice of metric [Park and Brockett, 1994] the inner product on $SE(3)$ decomposes naturally into inner products on $SO(3)$ and $(\mathbb{R}^3, +)$, i.e. $\langle \mathfrak{r}_1, \mathfrak{r}_2 \rangle_{SE(3)} = \langle \mathfrak{r}_1, \mathfrak{r}_2 \rangle_{SO(3)} + \langle s_1, s_2 \rangle_{(\mathbb{R}^3, +)}$, where we have denoted the elements of tangent spaces of $SO(3)$ and $\mathbb{R}^3$ with $\mathfrak{r}$ and $\mathfrak{t}$ respectively and have omitted to do so for the translation group as the tangent space coincides with the space itself.

A.4 Optimal Transport

Optimal transport (OT) focuses on finding the most efficient way to move mass or resources from one distribution to another. It involves minimizing a cost function $c(x, y)$ that quantifies the expense of transporting mass from point x in one space to point y in another space. This problem is framed as an optimization problem, often using the Kantorovich formulation:

$$\min_{\gamma \in \Pi(\mu, \nu)} \int_{X \times Y} c(x, y) \, d\gamma(x, y)$$

where $\Pi(\mu, \nu)$ is the set of all joint distributions γ with marginals μ and ν. The goal is to find a transportation plan γ that minimizes the total cost. Optimal transport has significant applications in probability theory, where it is used to compare probability distributions. One common measure is the Wasserstein distance, defined as:

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Pi(\mu, \nu)} \int_{X \times Y} d(x, y)^p \, d\gamma(x, y) \right)^{1/p}$$

where $d(x, y)$ represents the distance between points x and y and $p \geq 1$ is a parameter that defines the type of Wasserstein distance.

In machine learning, optimal transport enhances algorithms in tasks such as domain adaptation, where it helps align different data distributions, and in generative modeling, where it aids in generating data samples that match a given distribution. In the case of FOLDFLOW-2 and as used in [Bose et al., 2024], OT is used to sample tuples of source noisy sample and target protein (x_0, x_1) to perform Flow Matching. It takes the form of using the OT plan for the joint distribution $q(x_0, x_1)$ between minibatches (X_0, X_1). The OT variant performs here corresponds to what is called minibatch OT as studied in [Fatras et al., 2021, 2020].

B Experimental Details

B.1 Dataset filtering

B.1.1 PDB Structures

We use a subset of PDB with resolution $< 5\text{Å}$ downloaded from the PDB [Berman et al., 2000] on July 20, 2023. We performed standard filtering to remove any proteins with $> 50\%$ loops. During

17

preprocessing, we also removed any non-organic residues at either end of the structure. In previous works, these residues are typically kept but masked during training, however they contributed to the total forward pass FLOPs and therefore decrease training efficiency. By removing these residues during preprocessing, we are able to increase the number of training examples per batch. Finally, we re-clustered the PDB dataset using `mmseqs2` at the 50% sequence identity threshold to obtain 6,593 clusters during training. The PDB is made available under a CC0 1.0 Universal (CC0 1.0) Public Domain Dedication.

B.1.2 Synthetic Structures

In this section, we provide more detail on the filtering procedure used in our curation of synthetic data for training. We began with a SwissProt data dump consistent of 532,003 structures predicted by AlphaFold2 [Jumper et al., 2021] accessed in February 2024. The AlphaFold2 predicted structure database is made available under a CC-BY-4.0 license for academic and commercial uses.

Global pLDDT Filtering. We first filtered out globally low-confident structures, as measured by average and standard deviation of pLDDT taken across all residues in a predicted structure. Despite SwissProt already being a curated set of high-confidence structures, we still found considerable variation in quality. See fig. 5a for an empirical analysis average pLDDT for a random sample of 500 proteins from SwissProt. Our final filtering criteria were (`avg pLDDT > 85`) `&&` (`std pLDDT < 15`) to keep only "consistently good" structures, see fig. 5b for a graphical representation.

(a)	**(b)**
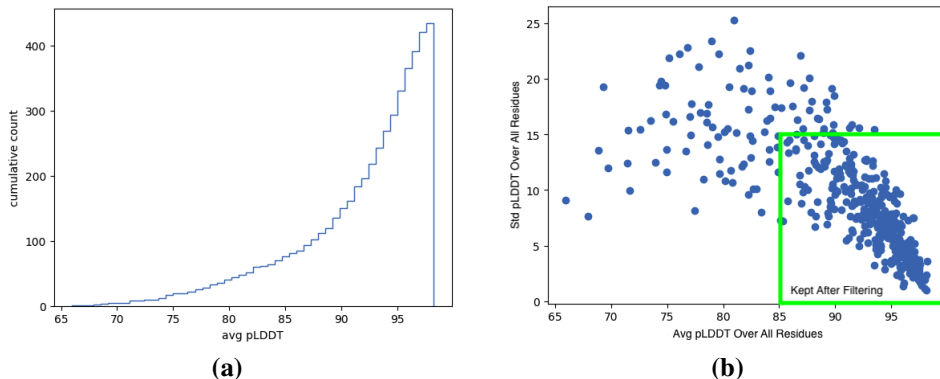
Figure 5: Analysis of global pLDDT distribution on a sample of 500 proteins from SwissProt.

High-Confidence Low-Quality Filtering. Despite the global pLDDT filtering, there were still low-quality structures in the training after global pLDDT filtering. Some examples of these structures can be seen in fig. 6. They are characterized by having high overall confidence and good local qualities but unrealistic global structures or sub-chain interactions. Our finding is that these structures easily corrupt the training data and cause a model to produce similarly "unfolded" generations.
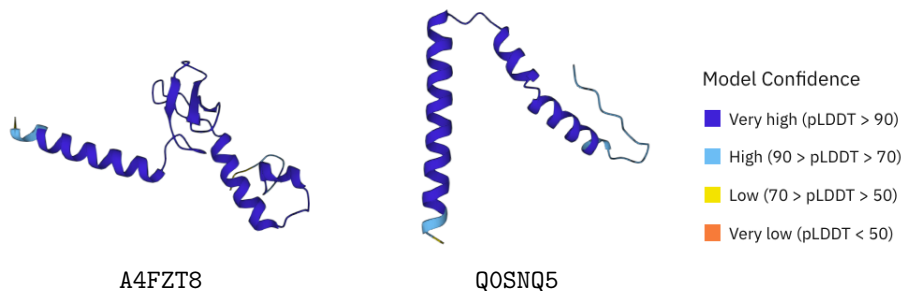
Figure 6: Examples of high-confidence, low-quality structures that were filtered out of the training set. Images are from the AlphaFold Protein Structure Database https://alphafold.ebi.ac.uk/ accessed in May 2024; identifiers are the UniProt IDs of each example.

18

Table 7: Overview of Training Setup

Training Parameter	Value
Optimizer	ADAM Kingma and Ba [2014]
Learning Rate	0.0001
β_1, β_2, ε	0.9, 0.999, 1e-8
Effective M (max squared residues per batch)	500k
% of experimental structures per epoch	33%
Minimum number of residues	60
Maximum number of residues	384
Sequence masking probability	50%

Table 8: An overview of training time. *RFDiffusion initializes from RoseTTAFold, and we include that training time in the estimates. **We recall that FOLDFLOW-2 uses frozen ESM2-650M which was trained on 512 GPUs for 8 days.

Model	# Steps	# GPUs	Total Time (days)
RFDiffusion* [Watson et al., 2023]	25k	64 + 8	28 + 3
FOLDFLOW [Bose et al., 2024]	600k	4	2.5
FOLDFLOW-2 **	500k	2	4
FOLDFLOW-2 ** w/o folding block	500k	2	2.5

B.2 Model architecture details

We provide details for the IPA blocks and Folding blocks. For the IPA blocks, we follow the setting developed in [Yim et al., 2023b] by adding to the original IPA [Jumper et al., 2021] a skip connection and transformer layer on the node representation. The IPA modules takes as inputs the single and pair representations and a structure. For the structure encoder, we initialize the single and pair representations with positional and time embedding passed to MLPs. In our experiment, we used a node embedding dimension of 256, an edge dimension of 128, and a hidden dimension of 256. These settings are the same for both the encoder and decoder. We have use a skip connection between the representations of the structure encoder and decoder.

We combine the single and pair representations of different modalities by projecting each with a linear layer of output dimensions 128 for the single representations and 64 for the pair representation. We then concatenate all modalities' representation to obtain a single representation of dimension 128 and pair representation of dimension 256.

The Folding blocks are taken from Lin et al. [2022]. They are composed of 2 Triangular Self-Attention Blocks with single and pair head width of size 32. Finally, the pair and single representation dimensions are of 128.

The structure decoder's single and pair representations inputs are an average between the refined ones from the Folding blocks and the initial ones. All other architectural details not specified here are set to the defaults of IPA or ESM, respectively.

B.3 Training Details

Hyperparameters. See table 7 for an overview of the experimental setup. We train with the "length batching" scheme described in Yim et al. [2023a] in which each batch consists of the same protein sampled at different times. The number of samples in a batch is variable and is approximately $\lceil \text{num\_residues}^2/M \rceil$ where M is a hyperparameter in table 7. Other training details such as the loss computation are the same as Bose et al. [2024].

Training hardware setup. FOLDFLOW-2 is coded in PyTorch and was trained on 2 A100 40GB NVIDIA GPUs for 4 days. Initial tests runs were trained in a similar setting.

19
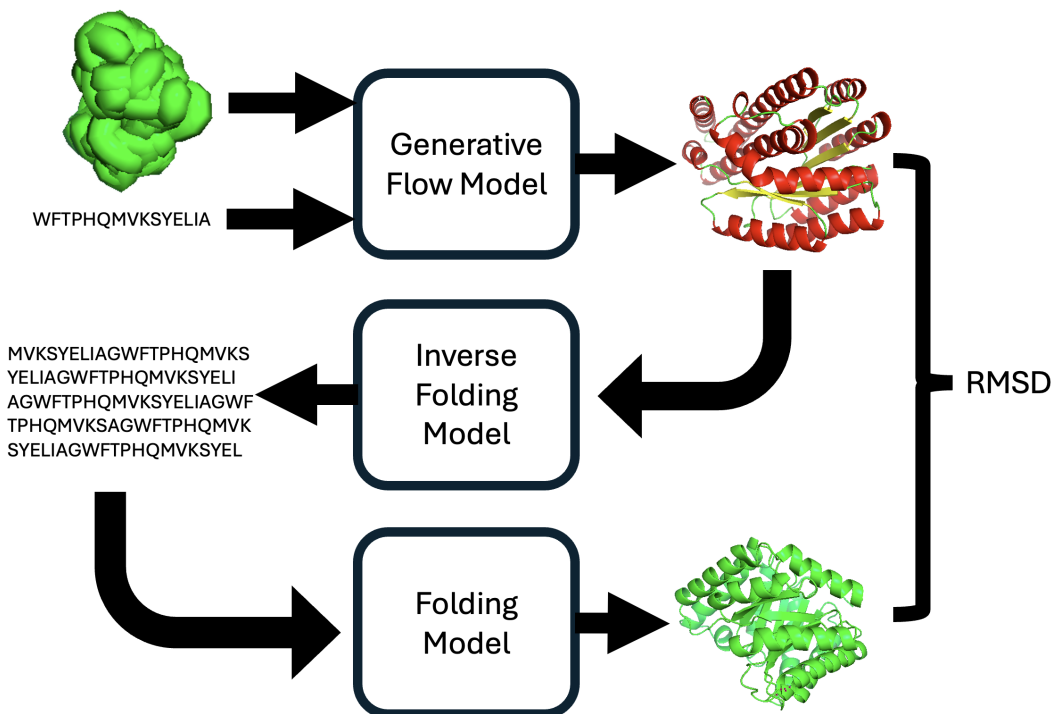
Figure 7: Schematic of designability calculation. First a generative flow model is used to generate a protein backbone from an initial structure (possibly noise) and (optionally) a protein sequence. This is then fed to an inverse folding model (ProteinMPNN Dauparas et al. [2022]) eight times to generate eight sequences for the structure. Then all eight sequences are fed back to ESMFold to produce a structure for each sequence. All eight structures are compared using scRMSD from the ESMFold "refolded" structure to the generated structure, and the minimum is taken as the "designability" of the generated structure with at least one structure with error < 2.0Å being classified as designable following Watson et al. [2023].

B.4 Inference details

The inference is performed with Euler integration steps. In our experiments, we found that 50 steps gave state-of-the-art results. We use the *Inference Annealing* trick from Bose et al. [2024] multiplying the rotation vector by some time dependent scaling function $i(t)$ where $t \in [0, 1]$. We tried different scaling parameters and as found in [Bose et al., 2024], $10t$ provided the best performance. In practice this means greatly speeding up the rotation components at the beginning of inference ($t \approx 1$) and slowing it down at the end of inference ($t \approx 0$).

B.5 Unconditional protein backbone generation

Metrics. We compute several quantity of interest to measure the performance of FOLDFLOW-2. i) We start with designability. We measure designability using the *self-consistency* metric with ProteinMPNN [Dauparas et al., 2022] and ESMFold [Lin et al., 2022], counting the fraction of proteins that refold (C_α-RMSD (scRMSD) < 2.0Å) over 50 proteins at lengths $\{100, 150, 200, 250, 300\}$. For our ablation study and sensitivity analysis in § C.2, we also provide the average self-consistency rmsd over all generated proteins.

ii) In order to use generative models for drug discovery applications, we want to measure how different and novel are the generated data compared to training data. We measure novelty using two metrics: 1.) the fraction of generated proteins that are both designable and are dissimilar to PDB structures (quantified by template-match score to PDB, *i.e.*, PDB-TM score) < 0.3, as used in Yeqing and Mohammed [2023], higher is better) and 2.) for designable proteins, the average closest similarity to training data (quantified by the maximum TM score, lower is better). We note that the threshold

20

for similarity has been studied previously, where the average TM-score on random structure pairs is ~ 0.3 [Zhang et al., 2006].

iii) Finally we want a model that generates diverse proteins and not just of the same type. Proteins are usually gathered into different clusters during training. So for diversity, we use the number of generated clusters with a TM-score threshold of 0.5 [Herbert and Sternberg, 2008](higher is better) as well as the average pairwise TM-score of the *designable* generated samples averaged across lengths as our diversity metric (lower is better). Note that in certain model, designability is inversely correlated with diversity as these models can produce unrealistic (e.g. unfolded) proteins that are "diverse" because they do not align well with each other.

Baselines. On the unconditional backbone generation task we compare to pre-trained versions of FrameDiff [Yim et al., 2023b], Chroma [Ingraham et al., 2023], Genie [Yeqing and Mohammed, 2023], FoldFlow [Bose et al., 2024], and RFDiffusion [Watson et al., 2023]. We use the default parameters for each model including # of Euler steps for inference and default noise levels (0.1 for RFDiffusion and FrameDiff). We use the OT version of FoldFlow as it is the most similar to our setup and achieved the highest designability.

In table 3 we use 30 Euler steps for inference to better match the diversity levels of the baseline models for more accurate comparison on these metrics. Results for the 50 step model can be seen in table 17 with slightly worse designability but improved novelty and diversity.

B.6 Motif scaffolding

Evaluation procedure. We follow the same evaluation procedure as in Watson et al. [2023], Yim et al. [2023b]. In particular, to evaluate the designability of a scaffold, we use ProteinMPNN [Dauparas et al., 2022] to decode 8 sequences and then re-fold those sequences, *fixing the motif sequence* which is known a priori. Given these re-folded structures, we compare three numbers:

1. **Global RMSD**: the overall aligned RMSD between the entire generated structure and the refolded structure.
2. **Motif RSMD**: the RMSD of the refolded motif residues aligned to the original motif residues.
3. **Scaffold RMSD**: the RMSD of the refolded scaffold residues aligned to the generated scaffold residues.

Following Watson et al. [2023], a scaffold is considered "designable" if the Global RMSD is < 2 AND the motif RMSD < 1 AND the scaffold RMSD < 2. A detailed breakdown of our results can be found in table 9, with some samples of designable motifs in fig. 8.

We note that the choice of folding model appears to have a nontrivial impact on this metric. In their original papers, Watson et al. [2023], Yim et al. [2023b] used AlphaFold2 with no MSA and 0 recycles to refold their structures; however ESMFold is known to be significantly more accurate when no MSAs are provided [Lin et al., 2023]. Given this, we generated new samples from RFDiffusion (FrameFlow doesn't have public code for generating scaffolds) and re-folded them with ESMFold. The result is that RFDiffusion is able to solve all 24 examples; an increase of 4 vs. their reported numbers. Moreover, the proportion of solved increases relative to their reported results, suggesting that the accuracy of the folding model significantly impacts the ability to measure scaffold quality *in silico*.

B.6.1 Pseudo-label training

We follow a similar data augmentation procedure as in [Yim et al., 2024, Watson et al., 2023], with the only modification of adding a minimum number of contiguous residues per motif. We use the same min length and adding an absolute minimum length of 2 for a motif. We continue training FOLDFLOW-2 for 330,000 steps on the same dataset, with a learning rate of 10e-5. This was done using 2 NVIDIA A100 80G, for 2.85 days.

B.6.2 VHH CDR

From the Structural Antibody Database we used 615 nanobody sequences, yielding 1831 chains from the PDB as training set, and for testing we used 40 sequences appearing in 106 PDB chains.

21

Table 9: A detailed breakdown of FOLDFLOW-2 motif scaffolding performance using ESMFold to refold all structures. All numbers are out of 100 samples.

	FOLDFLOW-2				RFDiffusion
Example	# Overall Valid	# Motif Valid	# Scaffold Valid	# Designable	# Designable
1BCF	100	100	100	**100**	**100**
2KL8	100	100	100	**100**	**100**
1PRW	100	100	98	**98**	91
1YCR	96	100	89	85	**91**
4ZYP	89	97	87	80	**85**
3IXT	97	101	75	73	**85**
7MRX_small	70	92	83	**66**	22
6EXZ_long	78	83	62	59	**91**
5TPN	62	95	78	57	**79**
6EXZ_med	67	70	55	54	**87**
6EXZ_small	56	63	55	**53**	28
6E6R_long	84	98	57	51	**82**
1QJG	54	82	76	49	**80**
7MRX_med	59	86	67	**49**	22
6E6R_med	78	96	54	43	**87**
5TRV_med	44	86	70	41	**80**
5TRV_long	40	85	69	38	**77**
6E6R_small	82	100	34	30	**50**
5TRV_small	28	78	47	24	**53**
7MRX_long	27	59	47	20	**22**
5YUI	14	54	48	**13**	8
5IUS	10	86	79	10	**11**
5WN9	6	12	10	**4**	1
4JHW	3	74	39	3	**8**

Figure 8: Samples of solved motif scaffolding problems from the benchmark of Watson et al. [2023]. The motif is in red, the designed scaffold is in blue, and the refolded structure from ESMFold is in gray.

7MRX_med   4JHW   5IUS

5WN9   1BCF   2KL8

22
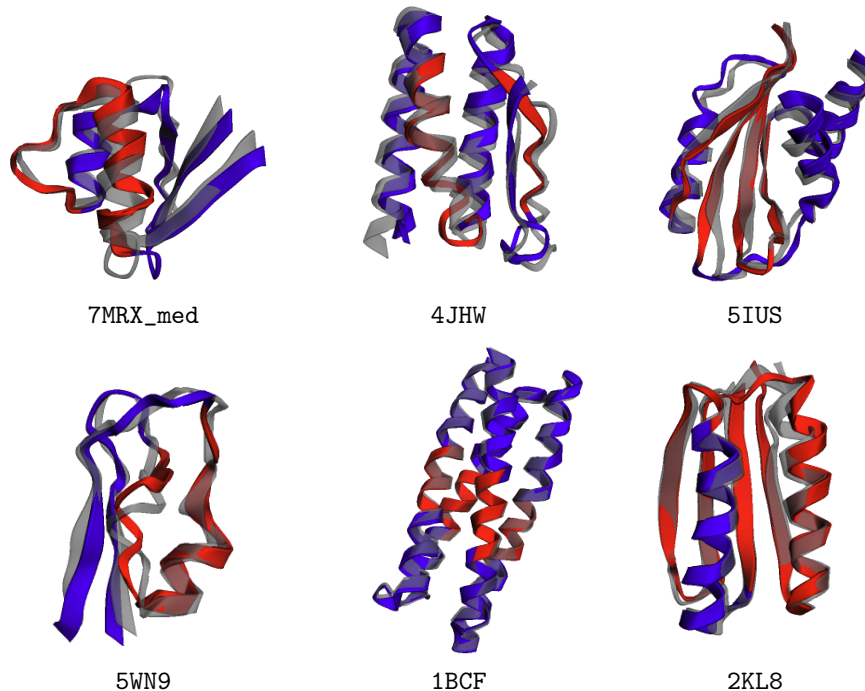
Table 10: A detailed breakdown of FOLDFLOW-2 motif scaffolding performance applied to refoldable VHH structures. The same comments as table 9 apply to evaluation. All numbers are out of 25 samples.

| Example | FOLDFLOW-2 | | | | RFDiffusion |
	# Overall Valid	# Motif Valid	# Scaffold Valid	# Designable	# Designable
6qtl-B	4	21	16	**4**	0
6qtl-G	4	19	18	**4**	0
6rpj-H	4	21	21	**4**	0
6rpj-D	3	23	19	**3**	0
6qtl-F	2	22	18	**2**	0
7epb-C	2	35	33	2	**4**
6rpj-F	3	24	21	**2**	0
6qtl-C	3	20	14	**2**	1
5l21-B	1	16	7	1	1
6oz6-G	0	16	2	0	0
6oz6-E	0	17	1	0	0
6oyz-G	0	12	0	0	0
6oyz-F	0	15	0	0	0
6oyh-H	0	17	0	0	0
6oyh-G	0	19	1	0	0
6gs7-H	1	19	9	0	**2**
1kxq-E	0	23	13	0	0
6rpj-B	0	21	19	0	0
7a50-C	0	23	18	0	0
7epb-D	0	22	19	0	**2**
7o31-X	0	14	12	0	0
7q3q-B	0	13	5	0	0
7tjc-B	0	17	9	0	0
8cxr-E	0	14	1	0	0
8cxr-F	0	18	1	0	0

With both the sequence and CDRs readily available, we can build the appropriate mask for training, which is used to fix the motif atoms, and mask the scaffold sequence information. For testing, we sampled the scaffold segment lengths based on the median value of each scaffold segment, $\pm$ 5. Empirically the scaffold segment lengths varied less than that amount; nanobodies are known to exhibit less variability across their framework regions, both in sequence and structure [Mitchell and Colwell, 2018]. We continue training FOLDFLOW-2 using this dataset, training for 10,000 steps with a learning rate of 10e-5. Using a single A100 80G this process takes 3.5 hours. We observed rapid increase but rapid tapering in performance using the VHH dataset; this is likely due to the lack of variability of the scaffolds themselves.

We note the designability metric used here and in other papers shows certain limitations when applied to this dataset and this task. Using the test set sequence and structures themselves, we compute the same scRMSD scores, in order to benchmark what should be, in theory, the best possible performance. We observe that out of the 106 testing chains, only 25 of them are "solved" according to Watson et al. [2023]'s criteria. This raises the question as to whether or not this particular criteria and setup is applicable to any motif scaffolding task. We provide the full set of results in table 12 on all 106 chains, and on a subset of size 25 in table 11, with generated samples in fig. 9. In addition, the particular number of solved samples are reported in table 10.

B.7 Zero-shot Molecular Dynamics

To evaluate FOLDFLOW-2's ability to capture protein dynamics we evaluated its performance on the test set of ATLAS [Vander Meersche et al., 2023] molecular dynamics used by Jing et al. [2024] but restricted to proteins at most 40 amino acids in length. To measure performance we used the following metrics. All metrics were computed exclusively over backbone atoms.

23

Table 11: VHH Motif Scaffolding results on the re-foldable examples only. Reported are the global, motif, and scaffold RMSD along with the number of solved tasks.

	Global scRMSD	Motif scRMSD	Scaffold scRMSD	Solved (out of 25)
RFDiffusion	3.1±1.23	3.94±1.54	2.4±0.93	5
FOLDFLOW-2 VHH	2.27±0.5	2.78±1.01	1.67±0.24	9
Test Set	0.55±0.19	0.65±0.18	0.48±0.19	25

Table 12: VHH Motif Scaffolding results on all samples, same numbers as table 11

	Global scRMSD	Motif scRMSD	Scaffold scRMSD	Solved (out of 106)
RFDiffusion	2.86±1.1	3.6±1.42	2.25±0.85	18
FOLDFLOW-2 VHH	2.3±0.47	2.99±0.86	1.66±0.31	15
Test Set	1.49±1.38	2.17±1.22	1.05±1.69	25

1. **Pairwise RMSD r.** To measure FOLDFLOW-2's ability to capture protein flexibility we first compute the average pairwise RMSD between every pair of conformations generated by each method. We then evaluate the average pairwise RMSD for the ground truth data and report the Pearson correlation r between the average pairwise RMSD per generated protein and ground truth data.

2. **Global and per-target RMSF r.** To further investigate flexibility, we measure the RMSF both globally and per target and compute the Pearson correlation r to ground truth data. Global RMSF is computed by, for each target calculating the backbone RMSF and taking its average, then measuring Pearson correlation between generated and ground truth samples over the sequence of averaged RMSFs. For per-target RMSF we instead compute the Pearson correlation between generated and ground truth backbone atom RMSFs and report the average taken over all targets.

3. **PCA $\mathcal{W}_2$.** Following Jing et al. [2024] we seek to measure the distributional accuracy of our generated samples by evaluating the 2-Wasserstein distance using the first two principal components given by ground truth data. We use a PCA as evaluating $\mathcal{W}_2$ on all atom coordinates would yield inaccurate measurements due to $\mathcal{W}_2$ needing samples exponential in dimensionality in order to obtain reasonable estimates. We compute the PCA $\mathcal{W}_2$ by, for each target, computing the first two principal components of backbone atom coordinates from ground truth MD data and projecting the backbone coordinates of each method's generated samples onto these coordinates. We then compute the $\mathcal{W}_2$ per target and report the averaged $\mathcal{W}_2$ over all targets.

As the test set contains 30,000 frames per protein computing test metrics using all ground truth conformations would be computationally infeasible. As such, following Jing et al. [2024] we randomly sample 300 conformations for each protein to be used as the test set. table 6 reports the mean and standard deviation over 5 resamplings of these test sets. Samples were generated from FOLDFLOW-2 using 50 inference steps and the inference annealing trick wherein the rotation vector is multiplied by $10t$.

We report in table 13 details on the resources required for training FOLDFLOW-2 compared to AlphaFlow-MD and ESMFlow-MD. We see that FOLDFLOW-2 requires an order of magnitude less time per inference step than ESMFlow-MD and AlphaFlow-MD while attaining results competitive with ESMFlow-MD while using 4.5X less GPU hours for training and 33X less trainable parameters. FOLDFLOW-2, ESMFlow-MD, and AlphaFlow-MD were all done on NVIDIA A100s. Inference time benchmarks were done on an NVIDIA A100, performing inference on a single protein of length 300 amino acids. Finally, in appendix B.7 we provide additional generated conformational ensembles from the test set, ESMFlow, and FOLDFLOW-2.

24

Figure 9: Samples of scaffolds for VHHs. Motif (i.e. CDR) is in red, scaffold is in blue, and refolded structure from ESMFold is in gray.
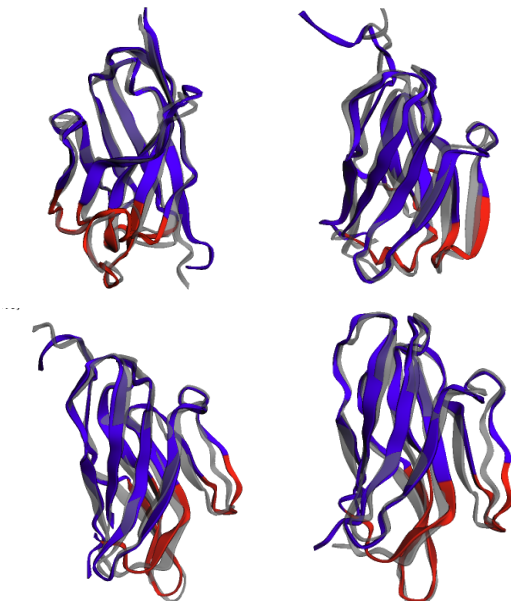
Table 13: Molecular dynamics experiment training details.

Model	# training GPU hours	# total parameters	# trainable parameters	Inference time / step (sec)
AlphaFlow-MD	2224	95M	95M	3.26 ± 0.01
ESMFlow-MD	872	3.5B	694M	1.12 ± 0.01
FOLDFLOW-2	192	672M	21M	0.18 ± 0.00

C Additional Results

C.1 Unconditional generation diversity and novelty exploration

We next provide several additional results on unconditional generation to give a better understanding of the behavior of FOLDFLOW-2 relative to the baselines. In fig. 11 we can see that FOLDFLOW-2 creates many more novel proteins at *all thresholds* of what is considered novel as compared to

Ground truth   ESMFlow-MD   FoldFlow-2
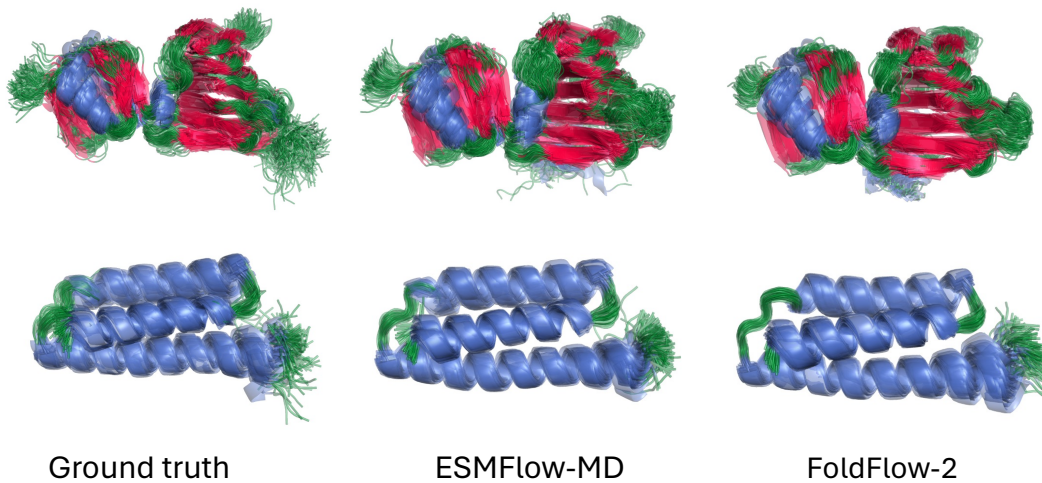
Figure 10: Additional conformation generation task samples. Proteins are colored by secondary structure with α-helices in blue, β-sheets in red, and loops in green.

25
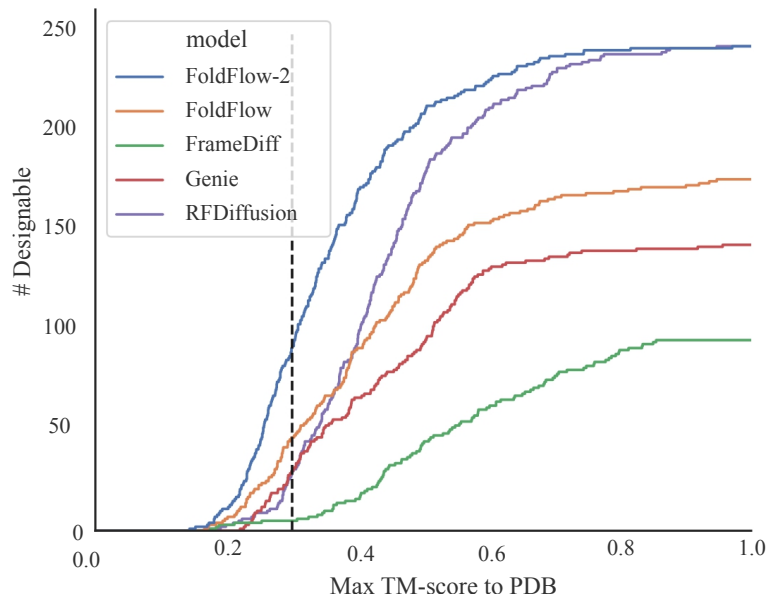
Figure 11: Curve showing the number of designable proteins that are at least some distance away from the PDB. FOLDFLOW-2 has many more novel and designable proteins than baselines. We report designability fraction at TM-score = 0.3 in table 3.

previous methods. We also depict more generated samples of all lengths in fig. 12. We can see that FOLDFLOW-2 creates more diversity of structures, especially at shorter lengths. With synthetic data or diversity fine-tuning this is expanded to all lengths 100-300.
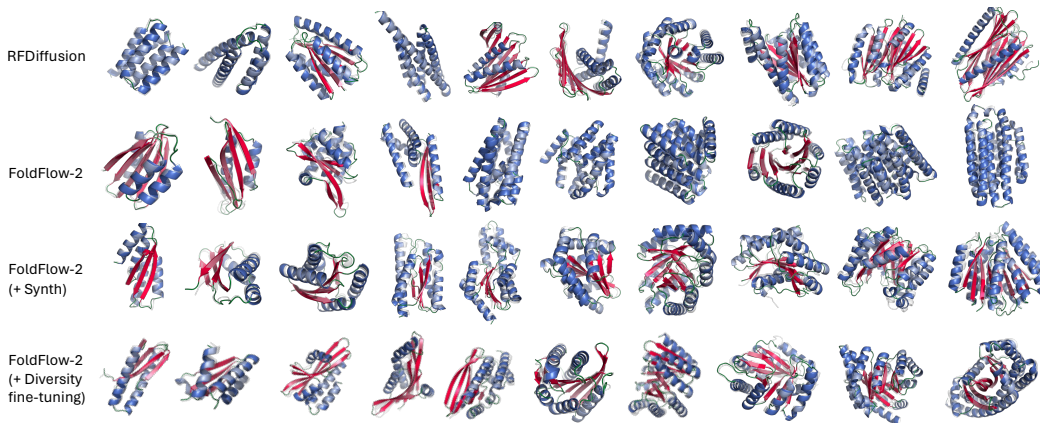
Figure 12: Designable samples from various methods. Overlayed in silver are refolded ESMFold structures. FOLDFLOW-2 exhibits significantly more diversity in secondary structure at shorter lengths than RFDiffusion with fine-tuned models able to produce diverse proteins across lengths.

C.2 Ablation study and sensitivity analysis

In this section, we provide several ablations of our FOLDFLOW-2 method. In table 14, we provide the unconditional generation performance for the different architecture components of our FOLDFLOW-2 method. In table 15 and table 16, we compare the performance achieved by our model FOLDFLOW-2 when we use different inference annealing t values for respectively unconditional generation and folding. In table 17 and in table 18, we compared the performance achieved by our model FOLDFLOW-

26

2 when using different numbers of Euler steps at inference for respectively unconditional generation and folding.

Architecture ablation. In table 14 we seek to understand the effects of architecture and dataset on the performance across our main designability, novelty, and diversity metrics for unconditional backbone generation. Starting from FOLDFLOW-2 we first investigate the effect of replacing the Folding Block with a simple MLP with FOLDFLOW-2 (- F. Block) and removing the sequence conditioning entirely (- ESM2). In this comparison we find that both the folding block and structure conditioning significantly improve the results. We find that the Folding block improves all metrics while the structure conditioning improves designability and diversity at the cost of novelty.

Dataset and Training. In table 14, we look at how adding synthetic data or stochastic flow matching affects designability, diversity, and novelty metrics. We find that both these additions actually hurt these metrics, although this is likely due to the change in composition of their generated structures. Overall, we find that these two models create more diverse proteins.

Number of inference steps & inference annealing scale. We also studied the influence of number of Euler steps on the generated proteins in table 17. We note that our model performs quite well at a relatively small number of steps, although performance starts to drop off under 25 steps. We attribute this to the optimal transport approach which is known to increase quality of generation especially with few inference steps Tong et al. [2023], Pooladian et al. [2023]. We find an interesting tradeoff between designability and diversity in table 17 and visually in fig. 13. Specifically that more steps increases the diversity of samples at the cost of designability.

We also studied the impact of the inference annealing scale factor in table 15. We see that small scaling (or none at all, corresponding to a value of 1) produce highly undesignable proteins, but designability improves quickly and beyond the optimal value of 10 it is somewhat stable. We notice the opposite effect in diversity as measured by the MaxCluster metric: no time scaling yields a score which is 64% larger than the same metric with scaling 10, and this trend is clearly anti-correlated with the scaling value.

Impact of synthetic augmented dataset on diversity. One of the interesting benefits of using our synthetic augmented dataset is that it increases diversity among designable generated data. Proteins have different secondary structures such as helices, beta-sheets and coil. While our model generates a lot of helices, it is important for drug discovery applications to generate other secondary structure such as beta-sheets. As shown in fig. 13, we can see that our synthetic augmented dataset leads to an increased of beta-sheets.

Impact of rotation time scaling & number of inference steps on folding. We conducted an analysis of inference parameters on FOLDFLOW-2's ability to fold proteins. In table 16 we sweep over the rotation time scaling parameter and measure its impact on folding RMSD. We see a similar trend to the unconditional case in table 15: very small scaling factors (e.g., 2) produce worse results, but the performance improves quickly as scaling increases and remains somewhat stable in a neighbourhood of 10.

We also conducted a sweep over the number of inference steps during generation on folding RMSD in table 18. We again see a similar trend to the counterpart for unconditional generation in table 17: a small number of Euler steps near 30-50 is best, with performance degrading significantly beyond 50 steps.

(a) 100 steps   (b) 50 steps   (c) 30 steps   (d) Data
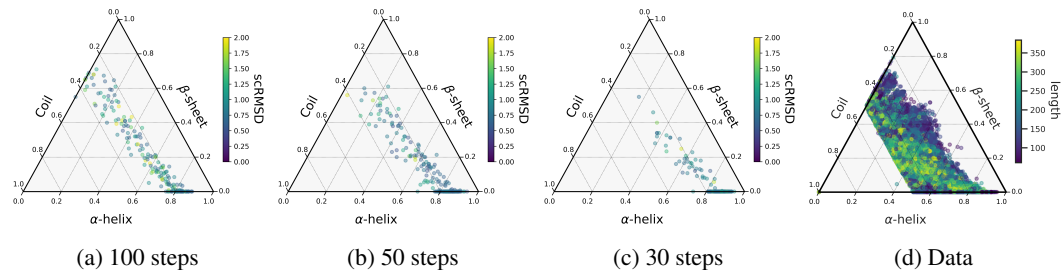
Figure 13: Secondary structure elements distributions (α-helices, β-sheets, and coils) of designable (scRMSD < 2.0) proteins, along with the data's distribution. We find more steps leads to more diverse designs with fewer α-helix only generations.

27

Table 14: Ablation study on FOLDFLOW-2 (FF-2) using: synthetic data, folding blocks, and stochastic flow matching (SFM). We generated 250 proteins (50 of length 100, 150, 200, 250) and compared Designability (fraction with scRMSD < 2.0Å), Novelty (max. TM-score to PDB and fraction of proteins with averaged max. TMscore < 0.3 and scRMSD < 2.0Å), and Diversity (avg. pairwise TMscore and MaxCluster fraction).

	Designability Frac. < 2Å (↑)	Novelty Frac. TM < 0.3 (↑)	Novelty avg. max TM (↓)	Diversity p.wise TM (↓)	Diversity MaxClust. (↑)	Seq. cond.	Folding blocks	SFM	iter/s	# train param.
FF-2 (- F. Block - ESM2)	0.716 ± 0.029	0.188 ± 0.025	0.419 ± 0.012	0.240	0.228	✗	✗	✗	2.7	17M
FF-2 (- F. Block)	0.852 ± 0.023	0.148 ± 0.023	0.438 ± 0.010	0.227	0.271	✓	✗	✗	2.1	18M
FF-2	**0.976 ± 0.010**	**0.368 ± 0.031**	**0.363 ± 0.009**	**0.205**	**0.348**	✓	✓	✗	1.6	21M
FF-2 (+Synthetic)	0.785 ± 0.027	0.047 ± 0.014	0.465 ± 0.008	0.226	0.264	✓	✓	✗	1.6	21M
FF-2 (+SFM)	0.935 ± 0.016	0.274 ± 0.029	0.386 ± 0.009	0.218	0.281	✓	✓	✓	1.5	21M

Table 15: Comparison of Designability (fraction with scRMSD < 2.0Å), Novelty (max. TM-score to PDB and fraction of proteins with averaged max. TMscore < 0.3 and scRMSD < 2.0Å), and Diversity (avg. pairwise TMscore and MaxCluster fraction) for different inference annealing functions $i(t)$.

	Designability	Novelty		Diversity	
	Frac. < 2Å (↑)	Frac. TM < 0.3 (↑)	avg. max TM (↓)	pairwise TM (↓)	MaxCluster (↑)
1	0.104 ± 0.019	0.012 ± 0.007	0.427 ± 0.022	0.197	**0.571**
2t	0.148 ± 0.023	0.040 ± 0.012	0.403 ± 0.024	0.198	0.549
5t	0.832 ± 0.024	0.312 ± 0.029	**0.375 ± 0.011**	**0.193**	0.387
10t	**0.976 ± 0.010**	**0.368 ± 0.031**	0.363 ± 0.009	0.205	0.348
15t	0.928 ± 0.016	0.308 ± 0.029	0.388 ± 0.010	0.199	0.358
20t	0.944 ± 0.015	0.304 ± 0.029	0.395 ± 0.010	0.199	0.347

Table 16: Speed of the integration on rotations. Integrating with a faster time for rotations compared to translation leads to more designable structures. Reporting the mean ± std. on 278 test samples.

Rotation time scaling	RMSD (↓)
2t	3.641 ± 4.457
5t	3.257 ± 4.113
10t	3.334 ± 4.325
15t	**3.237 ± 4.145**

Table 17: Comparison of Designability (fraction with scRMSD < 2.0Å), Novelty (max. TM-score to PDB and fraction of proteins with averaged max. TMscore < 0.3 and scRMSD < 2.0Å), and Diversity (avg. pairwise TMscore and MaxCluster fraction) for different number of Euler steps at inference.

	Designability	Novelty		Diversity	
	Frac. < 2Å (↑)	Frac. TM < 0.3 (↑)	avg. max TM (↓)	pairwise TM (↓)	MaxCluster (↑)
15 Euler steps	0.480 ± 0.032	0.136 ± 0.022	0.382 ± 0.012	0.196	0.430
20 Euler steps	0.876 ± 0.021	0.328 ± 0.030	0.358 ± 0.009	0.203	0.341
25 Euler steps	0.948 ± 0.014	0.376 ± 0.031	0.372 ± 0.010	0.207	0.336
30 Euler steps	0.976 ± 0.010	0.368 ± 0.031	0.363 ± 0.009	0.205	0.348
35 Euler steps	**0.980** ± 0.009	0.356 ± 0.030	0.370 ± 0.008	0.210	0.305
40 Euler steps	0.960 ± 0.012	0.304 ± 0.029	0.382 ± 0.009	0.201	0.349
45 Euler steps	0.940 ± 0.015	0.328 ± 0.030	0.382 ± 0.009	0.201	0.357
50 Euler steps	0.952 ± 0.014	**0.384 ± 0.031**	0.383 ± 0.011	0.194	0.387
75 Euler steps	0.800 ± 0.025	0.300 ± 0.029	0.380 ± 0.010	0.189	0.434
100 Euler steps	0.748 ± 0.028	0.340 ± 0.030	**0.366 ± 0.011**	0.188	0.415
150 Euler steps	0.620 ± 0.031	0.180 ± 0.024	0.408 ± 0.013	0.185	0.437
200 Euler steps	0.580 ± 0.031	0.208 ± 0.026	0.391 ± 0.012	**0.183**	**0.447**

Table 18: Effect of the number of integration steps on the aligned RMSD between the generated and ground truth backbone. Reporting the mean ± std. on 278 test samples.

# Euler steps	RMSD (↓)
30 Euler steps	3.384 ± 4.223
50 Euler steps	**3.334 ± 4.325**
100 Euler steps	3.374 ± 4.377
150 Euler steps	3.405 ± 4.409
200 Euler steps	3.481 ± 4.465

28

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction are supported by extensive experiments in §4 and in the Appendix §C.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mention limitations of our model and our evaluation in the conclusion §6. In §B.6.2 we also mention the difficulties of evaluating VHH with the current best folding models.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed information regarding experimental setup and model parameters is made available in the appendix. We introduce a new architecture which is described graphically in fig. 1, and is described in the main text with more details provided in the appendix §B.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes] and No

   Justification: Repreducible code is available under a CC-BY-NC 4.0 license at `https://github.com/DreamFold/FoldFlow`.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: High-level experimental details are discussed in the main text in §3.1,4 (architecture, training setup, etc) and extensive experimental details (hyperparameters, dataset details, training time, etc) are provided in §B.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: When applicable, in all tables we reported the standard deviation of the scores. We provide details on the metrics computation in §4.1, §B.7, and §B.6.

8. **Experiments Compute Resources**

29

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide details on the training resources and initial runs in §B.3, as well as resources for finetuning in §B.7, and §B.6.1.

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: This work uses publicly-available data and does not involve human or animal participants.

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The positive societal impacts of accelerated drug discovery are obvious and discussed in numerous places in our paper. This work represents an advance in developing tools for drug research and development, but does not have immediate impact to the broader public. This is discussed in §6.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This work investigates protein design techniques, which do not pose a high risk of misuse.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In §B we detail the creators and owners of the datasets used in this work. We mention the license and copyright information. We intend to repackage the AlphaFold2 structure database to curate structures which are useful for improved training of generative protein design models.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets at this time.

14. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

30