SPID: Distilled Protein Backbone Generation

Anonymous authors
Paper under double-blind review

Abstract

Diffusion- and flow-based generative models have recently demonstrated strong performance in protein backbone generation tasks, offering unprecedented capabilities for *de novo* protein design. However, despite their generation quality, these models are constrained by slow sampling, often requiring hundreds of iterative steps. This computational bottleneck limits their practical utility in large-scale protein discovery, where thousands to millions of candidate structures are needed. To address this challenge, we explore the techniques of score distillation, which has shown great success in reducing the number of sampling steps in the vision domain while maintaining high generation quality. However, a straightforward adaptation of these methods results in unacceptably low designability. We introduce Score Protein identity Distillation (SPID), which resolves this incompatibility by combining few-step generation with inference-time noise scaling. SPID adapts the Score identity Distillation (SiD) framework to both diffusion- and flow-based models without requiring access to pretraining data. Applied to the Proteína flow-matching model, our 16-step generator achieves 94.4% designability — exceeding the 400-step teacher (94.2%) — while delivering more than a 20-fold reduction in effective sampling time and maintaining comparable diversity and novelty. SPID generalizes across unconditional generation, fold-class conditional generation, and motif scaffolding, and extends to equivariant diffusion architectures, achieving significant speedup with improved designability. The resulting reduction in inference cost facilitates large-scale *in silico* protein design, thereby advancing diffusion-based models toward real-world protein engineering applications.

1 Introduction

De novo protein design aims to generate novel proteins with user-specified biological functions, bypassing the domain expertise and laboratory experimentation required by traditional approaches such as directed evolution (Huang et al., 2016; Dougherty & Arnold, 2009). Deep generative models have shown great success in tasks like image generation, but adapting them from images to proteins requires addressing fundamental structural constraints. Proteins have no fixed canonical orientation, motivating SE(3)-equivariant architectures. Models must respect physical constraints and track pairwise inter-residue distances. Crucially, unlike images where local errors yield merely unrealistic outputs, errors in local or global backbone geometry, such as inconsistencies in pairwise distances or torsion angles, can cascade into loss of key biochemical features and render structures completely undesignable (Eguchi et al., 2022; Anand & Achim, 2022). Architectural innovations from AlphaFold2 (Jumper et al., 2021), including invariant point attention (IPA) and triangle attention layers, addressed the first two challenges, enabling models such as RFDiffusion (Watson et al., 2023), FrameDiff (Yim et al., 2023b), and Genie (Lin & AlQuraishi, 2023) to generate high-quality, designable backbone structures encoding genuinely novel folds. Meanwhile, Chroma (Ingraham et al., 2023) took a different architectural approach, achieving similar goals via a sub-quadratic random graph neural network. More recent work has shown that even without equivariant architecture, flow-based models such as Proteína (Geffner et al., 2025b) can achieve state-of-the-art designability.

Sampling speed as a bottleneck. A critical obstacle limits the practical utility of these models: **sampling speed**. Generating a single backbone requires hundreds or even 1000 iterative steps, with each step often involving expensive IPA and triangle computations. This is a fundamental barrier to large-scale protein

design, where biologists must screen tens of thousands of candidate structures to evaluate stability, binding affinity, or other functional properties. For example, using Proteína's reported sampling times, with one A6000-48GB GPU, a 100,000 bakcbone-design campaign would cost around 150 GPU-hours. Slow generation becomes a bottleneck in iterative generate–evaluate–test workflows, limiting throughput and delaying discovery. Accelerating backbone generation is therefore not merely a matter of computational efficiency but a prerequisite for practical deployment. Prior work has made partial progress: flow-based FrameFlow (Yim et al., 2023a) improved sampling speed over its diffusion predecessor FrameDiff (Yim et al., 2023b), and Proteína (Geffner et al., 2025b) demonstrated that an architecture without IPA and triangle layers can still achieve competitive designability. Nonetheless, Proteína still requires 400 sampling steps, leaving significant room for acceleration.

Distillation in the image domain. Score distillation methods have dramatically reduced sampling costs in image generation, training one- or few-step generators that match or exceed the quality of hundred-step models (Fan et al., 2025; Ma et al., 2024; Shen et al., 2025). Score identity Distillation (SiD) (Zhou et al., 2024) and its extensions (Zhou et al., 2025b;a) achieve state-of-the-art one- and few-step image generation by matching the generator's score function to that of a pretrained teacher, with the objective of minimizing a Fisher divergence-based loss. However, as we discuss next, a property unique to protein backbone generation prevents direct application of these methods.

The noise rescaling challenge. A further complication specific to protein backbone generation compounds the difficulty of acceleration. Because small geometric errors readily produce undesignable structures, protein backbone generators universally apply *inference-time noise rescaling*: lowering the sampling temperature to enforce structural precision at the cost of some diversity. While effective for standard multi-step sampling, this practice creates a fundamental mismatch for distillation: distillation methods are trained under the original noise schedule, and rescaling at inference breaks the alignment between the training and inference distributions. As a result, naively applying an off-the-shelf distillation method leads to near-zero designability.

This work. We introduce **Score Protein identity Distillation (SPID)**, a distillation framework that resolves this incompatibility through two key ingredients: few-step generation and inference-time noise scaling. SPID adapts the SiD framework (Zhou et al., 2024) to both diffusion- and flow-based backbone generators and supports data-free training. Our main contributions are:

- **Identifying the failure mode of one-step distillation for proteins.** We show that one-step generators trained under score distillation achieve near-zero designability (0.4%), and that guidance, alternative distillation objectives, and hyperparameter tuning all fail to recover performance. The root cause is the incompatibility between inference-time noise rescaling and the single-step generation paradigm.

- **SPID: few-step generation with inference-time noise scaling.** We propose combining few-step generation with the noise rescaling practice standard in protein backbone generation. Noise rescaling is applied only at inference, preserving the training distribution while enabling low-temperature sampling. Applied to Proteína (Geffner et al., 2025b), our 16-step generator achieves 94.4% designability—surpassing the 400-step teacher model (94.2%)—with a more than 20-fold reduction in effective sampling time and maintained diversity and novelty.

- **Generality across tasks and architectures.** We demonstrate that SPID generalizes across unconditional generation, fold-class conditional generation, and motif scaffolding tasks. It also applies to equivariant diffusion architectures, distilling the 1000-step Genie model (Lin & AlQuraishi, 2023) into a 10-step generator with improved designability.

To our knowledge, SPID is the first successful distillation of a protein backbone generator. By reducing sampling from hundreds of steps to as few as 16, it makes large-scale *in silico* screening of thousands to millions of candidate structures tractable, enabling tighter integration of generative models with practical protein engineering workflows.

2 Background and Related Work

Protein structure representation. A protein backbone is typically represented by the N-C_α-C-O atoms of each residue. The most expressive representation uses rigid-body *frames* centered on each C_α atom, capturing both the residue position and local orientation via a rotation matrix derived from the N-C_α-C atom positions. Because proteins have no canonical orientation, frame-based representations are typically modeled by SE(3)-equivariant networks with a diffusion process that handles both translations and rotations. An alternative, simpler representation uses only the C_α coordinates, expressing the backbone as an $N \times 3$ matrix of Cartesian coordinates. Although this discards orientation information, works such as Genie (Lin & AlQuraishi, 2023) and Proteína (Geffner et al., 2025b) have demonstrated that models whose diffusion process operates on C_α coordinates can achieve strong designability, while enabling simpler, more scalable training.

Flow matching. We take rectified flow (Liu et al., 2023) as an example, as it is the basis of our teacher model Proteína (Geffner et al., 2025b). It defines a linear interpolant between noise and data: given real data $\boldsymbol{x}_d$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, the noised sample at time $t \in [0, 1]$ is

$$\boldsymbol{x}_t = t\boldsymbol{x}_d + (1 - t)\boldsymbol{\epsilon}, \tag{1}$$

and the network is trained to regress the target velocity

$$\boldsymbol{v}_t^\phi(\boldsymbol{x}_t, t) \approx \boldsymbol{x}_d - \boldsymbol{\epsilon}. \tag{2}$$

This is equivalent to a Gaussian diffusion process with $a_t = t$ and $\sigma_t = 1 - t$, and as shown in Albergo et al. (2023), flow matching and diffusion models with Gaussian noise are equivalent upon reparametrization. This connection is key: the score-based distillation ideas of SiD transfer to flow matching models with appropriate adjustments, as we derive in the Method section.

Diffusion distillation with SiD. In SiD (Zhou et al., 2024), the teacher model f_ϕ provides a score estimate $S_\phi(\boldsymbol{x}_t) = \sigma_t^{-2}(a_t f_\phi(\boldsymbol{x}_t, t) - \boldsymbol{x}_t)$, where $f_\phi(\boldsymbol{x}_t, t) \approx \mathbb{E}(\boldsymbol{x}_d|\boldsymbol{x}_t)$ is the teacher's clean-data prediction. This score is used as a proxy for the true data score $\nabla_{\boldsymbol{x}_t} \ln p_{\text{data}}(\boldsymbol{x}_t)$. A generator G_θ is trained so that its induced distribution matches the data distribution, by training a secondary *fake score network* f_ψ to track the generator's score and minimizing the Fisher divergence between S_ϕ and S_ψ. A key advantage of SiD is that it supports data-free training, as it requires only the pretrained teacher and fresh noise samples to calculate the Fisher divergence. On the other hand, when training data are available, they can be incorporated via adversarial components as in Zhou et al. (2025b).

Related work in protein backbone generation. RFDiffusion (Watson et al., 2023) fine-tunes the RoseTTAFold structure-prediction network as a DDPM, adding Gaussian noise to translations and Brownian motion to rotations in the SE(3) frame representation. Chroma (Ingraham et al., 2023) uses a correlated polymer diffusion process with a sub-quadratic random graph neural network, enabling generation of very long proteins and complexes. FrameDiff (Yim et al., 2023b) applies a theoretically grounded SE(3) diffusion directly over backbone frames using IPA. FrameFlow (Yim et al., 2023a) and FoldFlow (Bose et al., 2024) adopt flow matching over SE(3) frames as a faster alternative. Proteus (Wang et al., 2024) improves efficiency by replacing dense triangle layers with a graph-based approximation. SMCDiff (Trippe et al., 2023) pioneered the use of a C_α-coordinate diffusion process for backbone generation. Genie (Lin & AlQuraishi, 2023), Genie2 (Lin et al., 2024), and Proteína (Geffner et al., 2025b) subsequently demonstrated that such models can match or exceed frame-based methods in designability, with the added benefit of simpler training and better scalability.

Related work in diffusion distillation. Progressive distillation (Salimans & Ho, 2022) was among the first methods to reduce sampling steps, halving them at each distillation iteration. Consistency models (Song et al., 2023) learn the full ODE trajectory, enabling direct denoising from any point along it. Consistency Trajectory Models (Kim et al., 2024) and Flow Map Matching (Boffi et al., 2025) extend this to model flow maps between arbitrary trajectory points, and Align-Your-Flow (Sabour et al., 2025) demonstrates strong few-step image generation results in this family. Distribution-matching approaches, beginning with Diffusion-GAN (Wang et al., 2023a), instead minimize a divergence between the generator distribution $p_\theta(\boldsymbol{x}_t)$ and the data distribution at every noise level t. Wang et al. (2023b) and Luo et al. (2024) minimize the KL

divergence, while SiD (Zhou et al., 2024) uses the Fisher divergence for data-free distillation driven entirely by the teacher's score function; its adversarial extension SiDA (Zhou et al., 2025b) incorporates real training data to further improve quality. SiD-FewStep (Zhou et al., 2025a) extends SiD with a theoretically grounded multi-step training framework. To our knowledge, none of these approaches have been successfully applied to protein backbone generation, a gap we attribute to the structural sensitivity of protein backbones to geometric error and the resulting incompatibility between noise rescaling and one-step generation.

3 Method

SPID adapts SiD (Zhou et al., 2024) to protein backbone generation through two key ingredients: few-step generation and inference-time noise scaling. In Section 3.1 we derive the SiD distillation objective for rectified flow. Section 3.2 extends this to few-step generators. Section 3.3 introduces inference-time noise scaling, the ingredient that makes designability recoverable. Section 3.4 covers additional training details. We focus on the rectified flow formulation (Liu et al., 2023) and use the $\mathcal{M}_{\text{FS}}^{\text{no-tri}}$ model, the fastest Proteína variant, as the teacher.

3.1 SiD for Flow Matching

As established in Albergo et al. (2023) and Geffner et al. (2025b), flow matching and diffusion models are equivalent upon reparametrization and the key quantity linking the two classes of models is the score function, which is also the core of SiD (Zhou et al., 2024). SiD estimates the score of the data distribution as $\nabla_{\boldsymbol{x}_t} \ln p_{\text{data}}(\boldsymbol{x}_t) \approx S_\phi(\boldsymbol{x}_t) = \sigma_t^{-2}(f_\phi(\boldsymbol{x}_t, t) - \boldsymbol{x}_t)$, where $f_\phi(\boldsymbol{x}_t, t) \approx \mathbb{E}(\boldsymbol{x}_d|\boldsymbol{x}_t)$ is the predicted true data $\boldsymbol{x}_d$ by the pretrained model conditioned on the noised data $\boldsymbol{x}_t$ at timestep t. The distillation objective minimizes the score difference between the pretrained network f_ϕ and the generator network G_θ for all t:

$$\min_\theta \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}[\|S_\phi(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \ln p_\theta(\boldsymbol{x}_t)\|_2^2] \tag{3}$$

In order to apply SiD to a flow matching model, we first derive the score function for rectified flow as $\nabla_{\boldsymbol{x}_t} \ln p_{\text{data}}(\boldsymbol{x}_t) \approx (1-t)^{-2}(t f_\phi(\boldsymbol{x}_t, t) - \boldsymbol{x}_t)$ where $f_\phi(\boldsymbol{x}_t, t) = \boldsymbol{x}_t + (1-t)v_t^\phi(\boldsymbol{x}_t, t)$. To estimate the score of the generator in Eq. 3, we train another neural network f_ψ using the usual flow matching loss. We treat the generator output $\boldsymbol{x}_g$ as clean data and define the fake score training objective:

$$\min_\psi \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_g,t)p_\theta(\boldsymbol{x}_g)}[\frac{1}{N(1-t)^2}\|f_\psi(\boldsymbol{x}_t, t) - \boldsymbol{x}_g\|_2^2] \tag{4}$$

where N is the number of residues.

Following Zhou et al. (2024), we derive the initial approximation to the target score difference in Eq. 3 as:

$$\begin{aligned} \mathcal{L}_\theta^{(1)} =& \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}\Big[\|\frac{t}{(1-t)^2}\delta_{\phi,\psi}(\boldsymbol{x}_t, t)\|_2^2\Big], \\ & \delta_{\phi,\psi}(\boldsymbol{x}_t, t) = f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t) \end{aligned} \tag{5}$$

and the projected score matching loss as:

$$\begin{aligned} \mathcal{L}_\theta^{(2)} =& \frac{t^2}{(1-t)^4}\mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)\boldsymbol{x}_t \sim q(\boldsymbol{x}_t \mid \boldsymbol{x}_g, t)}\Big[\\ & (f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t))^T(f_\phi(\boldsymbol{x}_t, t) - \boldsymbol{x}_g)\Big] \end{aligned} \tag{6}$$

As discussed in Zhou et al. (2024), minimizing $\mathcal{L}_\theta^{(1)}$ alone is insufficient, and a fused loss of both $\mathcal{L}_\theta^{(1)}$ and $\mathcal{L}_\theta^{(2)}$ in the form of $\mathcal{L}_\theta = \mathcal{L}_\theta^{(2)} - \alpha\mathcal{L}_\theta^{(1)}$ needs to be considered. We set $\alpha = 1.0$ in all experiments, consistent with the range $\alpha \in [0.75, 1.2]$ reported in Zhou et al. (2024) (see Appendix E).

With derivations detailed in Appendix F, the final generator loss takes the form:

$$\begin{aligned} \mathcal{L}_\theta =& \frac{\omega(t)t^2}{(1-t)^4}\Big[(1-\alpha)\|f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t)\|_2^2 + \\ & (f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t))^T(f_\psi(\boldsymbol{x}_t, t) - \boldsymbol{x}_g)\Big] \end{aligned} \tag{7}$$

And the generator loss reweighting function $\omega(t)$ is defined as:

$$\omega(t) = \frac{(1-t)^4}{Nt^2 \|\boldsymbol{x}_g - f_\phi(\boldsymbol{x}_t, t)\|_{1,\text{sg}}} \tag{8}$$

We defer the detailed derivation relating the velocity $\boldsymbol{v}_t^\phi(\boldsymbol{x}_t, t)$ to the neural network output $f_\phi(\boldsymbol{x}_t, t)$ to Appendix F.

3.2 Few-Step Generation

Although one-step SiD generators perform well in image generation, our one-step protein generator produces almost no designable samples (0.4%, Tab. 1). Adding guidance, switching to other distillation strategies such as Diff-Instruct (Luo et al., 2024), trying different loss reweighting, and extensive hyperparameter tuning all fail to recover performance. The root cause is that one-step generators have no analog of inference-time noise rescaling: the only related operation—scaling down the initial Gaussian noise—is ineffective in this setting. To allow low temperature sampling, SPID modifies the SiD Few-Step framework (Zhou et al., 2025a) to distill few-step generators. Samples are generated by chaining K steps:

$$\boldsymbol{x}_g^{(k)} = G_\theta(t_k \text{sg}(\boldsymbol{x}_g^{(k-1)}) + (1-t_k)\boldsymbol{\epsilon}_k),\ \boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \mathbf{I}) \tag{9}$$

where t_k is the kth step in a predefined time path t_{steps}, $\boldsymbol{x}_g^{(k)}$ is the generated sample at step k, and sg stands for the stop gradient function, which is important during training since we use the uniform-step matching approach specified in Zhou et al. (2025a).

During training, we uniformly sample a step $k \in \{1, ..., K\}$ and generate $\boldsymbol{x}_g^{(k-1)}$ using Eq. 9. Only the final step from $\boldsymbol{x}_g^{(k-1)}$ to $\boldsymbol{x}_g^{(k)}$ is differentiated, avoiding backpropagation through the entire generation chain.

3.3 Inference Time Noise Scaling

Our few-step generators distilled from the strategy above still only achieve underwheling designability (41.4%, Tab. 1) using the standard sampling procedure. Now that our few-step generators take multiple steps, we can apply the same noise rescaling at each step. Critically, rescaling is applied *only at inference*, leaving the training distribution unaffected. We modify the sampling step of Eq. 9 as:

$$\boldsymbol{x}_g^{(k)} = G_\theta(t_k \boldsymbol{x}_g^{(k-1)} + \gamma(1-t_k)\boldsymbol{\epsilon}_k) \tag{10}$$

where γ is the noise scaling factor and $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \mathbf{I})$. The choice of the noise scale γ greatly affects the generation quality as discussed in Appendix B. We set $\gamma = 0.45$, matching Proteína , which gives the optimal balance between designability and diversity (see Appendix B).

Eq. 9 (without noise rescaling) is used when generating samples during distillation training. On the other hand, Eq. 10 (with noise rescaling) is used at infernce when producing actual protein structures for evaluation.

3.4 Additional Training Details

Number of Residues. Reweighting both the generator loss and the fake score loss by N, the number of residues in the protein structure, is crucial so that the generator learns to generate proteins of different lengths. For unconditional distillation, we do not require any existing databases like the PDB (Berman et al., 2000) or AFDB (Varadi et al., 2021). We uniformly sample the lengths within the supported range (in all our main experiments the range is set to 50-256) in each training batch and use the mask for loss reweighting. While we are data-free, there is potential to further enhance our method by utilizing these high quality data by adding the adversarial components, we leave that for future study.

Time schedule. During training, we follow the log schedule in Proteína when sampling the timestep t. Choosing $p = 2$ and $N_{steps} = 400$, we first sample $s \sim \text{Unif}[0, N_{steps}]$, then define t as:

$$t = 1 - 10^{-\frac{s}{N_{step}}p} \tag{11}$$

Generator time steps are equally spaced points between T_{init} and N_{steps}, before being converted to continuous time using Eq. 11. T_{init} is set to 30 which corresponds to a signal-to-noise-ratio of around 1/2.5, matching the σ_{init} value of 2.5 in Zhou et al. (2024).

Motif Scaffolding. The distillation scheme extends naturally for motif scaffolding tasks. Following Proteína (Geffner et al., 2025b), we randomly sample contiguous blocks from PDB structures (Berman et al., 2000) as fixed motifs. The rest of the distillation process is identical to unconditional distillation, except that all three networks perform conditional generation on the same motif, and the generator loss is reweighted by scaffold length.

Fold Class Conditioning. For evaluation, we also test our distilled generators' ability to generate structures conditioned on the fold class labels based on the CATH structural hierarchy (Dawson et al., 2016), following the fold class labels defined in Proteína (Geffner et al., 2025b).

Parameters for SiD. The ablation study for α and other parameter settings are discussed in Appendix E. The detailed training procedure is presented in Algo. 1 and sampling procedure in Algo. 2.

Computing resources. Distillation training is performed on 8 Nvidia A6000-48GB GPUs and takes roughly 4 hours to converge.

Limitations. SPID is designed for diffusion processes where all noise is Gaussian, as in rectified flow and Gaussian DDPM on coordinates. Models such as RFDiffusion (Watson et al., 2023) and FrameDiff (Yim et al., 2023b) additionally apply Brownian motion on SO(3) to the rotational component of backbone frames; adapting SPID to these models would require extending the score identities to the non-Euclidean geometry of the rotation group. Additionally, models that incorporate discrete or non-differentiable components, such as the one-hot sequence encoding in Genie2 (Lin et al., 2024), block the backpropagation required by the SiD generator loss which differentiates with the respect to the output of the generator model. Adapting SPID to such architectures would alternative distillation objectives.

4 Experiments

We evaluate SPID by distilling the $\mathcal{M}_{\text{FS}}^{\text{no-tri}}$ Proteína model and comparing distilled generators of varying step counts against the teacher across unconditional generation, fold-class conditional generation, motif scaffolding tasks. We also validate generalization to an equivariant diffusion architecture by distilling Genie (Lin & AlQuraishi, 2023). Finally, we perform biological plausibility analyses to verify that the generated protein structures are indeed biologically meaningful.

4.1 Evaluation

We adopt the evaluation pipeline from Proteína , which assesses designability, diversity, and novelty through a self-consistency test involving ProteinMPNN (Dauparas et al., 2022) and ESMFold (Lin et al., 2023). A structure is considered designable if the best self-consistency root mean square deviation (scRMSD) is less than 2Å. Designability is the percentage of all generated samples that meet this threshold. Full metric definitions follow Geffner et al. (2025b). For each distilled generator, we generate 100 samples for each length in {50, 100, 150, 200, 250} and report the designability, diversity, and novelty, as well as the metrics proposed in Proteína (FPSD, fS, and fJSD). We also record the total generation across all 500 samples for sampling time analysis. See Appendix G for implementation details.

4.2 Unconditional Generation in One Step

We first applied our flow-matching-adjusted score distillation to train a one-step generator. As expected, the it produced few designable structures (Tab. 1). Adding autoguidance and classifier-free guidance (CFG), modifying the SiD losses, and tuning training parameters all failed to produce any meaningful improvements, confirming that one-step distillation is fundamentally incompatible with the noise rescaling required for protein backbone generation.

Table 1: Unconditional backbone generation performance for our distilled generators with different sampling steps compared to the Proteína pretrained model $\mathcal{M}_{\text{FS}}^{\text{no-tri}}$ and the distilled generators using Diff-Instruct. Optimal performance values are bolded.

Number of Steps K	Design-ability (%)↑	Diversity Cluster↑	Diversity TM↓	Novelty PDB↓	Novelty AFDB↓	FPSD PDB↓	FPSD AFDB↓	fS (C / A / T)↑	fJSD PDB↓	fJSD AFDB↓	Sec. Struct. % (α/β)	Sampling Time (s)
1	0.4	1.00 (2)	0.63	0.78	0.80	2470.17	2403.07	2.31 / 2.37 / 4.45	2.74	2.27	75.1 / 0.1	5.687
5	47.4	0.64 (151)	0.39	0.81	0.81	457.29	423.92	1.04 / 2.05 / 11.74	3.48	2.54	71.5 / 0.0	0.172
8	75.6	0.77 (292)	0.35	**0.76**	**0.77**	525.78	524.53	1.20 / 2.43 / 14.75	3.06	2.25	66.9 / 0.7	0.167
10	92.0	0.60 (277)	0.36	0.78	0.79	586.74	586.73	1.20 / 2.74 / 10.69	3.16	2.38	71.2 / 0.8	0.169
12	92.8	0.65 (300)	0.35	0.79	0.79	558.26	547.21	1.24 / 2.87 / 10.96	3.05	2.27	69.6 / 1.1	0.200
16	94.4	0.61 (287)	0.35	0.80	0.81	404.04	394.20	1.86 / 4.27 / 16.18	1.94	1.54	63.2 / 4.6	0.261
20	**95.4**	0.60 (286)	0.35	0.80	0.81	515.35	510.83	1.44 / 3.29 / 13.79	2.66	2.01	68.2 / 2.0	0.320
16 ($\gamma = 1$)	41.4	**0.79 (164)**	**0.33**	0.79	0.80	**142.2**	**146.0**	**2.51 / 5.33 / 35.46**	**0.62**	**0.48**	44.9 / 9.0	0.710
10-step Diff-Instruct	74	0.92 (34)	0.35	**0.76**	0.78	729.93	678.33	1.26 / 1.96 / 10.42	3.21	2.44	64.0 / 1.0	0.213
$\mathcal{M}_{\text{FS}}^{\text{no-tri}}$	94.2	0.63 (297)	0.34	0.83	0.84	316.73	290.66	1.92 / 4.78 / 20.57	1.80	1.24	64.4 / 4.6	6.395

Table 2: "Failure" rate within each fold class at the "C" level. The labels are specific to the Fold Classifier provided by Proteína (Geffner et al., 2025b), not the actual "C" codes. The number in the parenthesis represents the number of samples predicted to be within each class. Optimal performance values are bolded.

Fold class label	0	1	2	3	4
Distilled	0.039 (358)	0.200 (5)	0.098 (132)	NA (0)	**0 (5)**
Pretrained	**0.036 (366)**	**0.181 (11)**	**0.087 (115)**	0 (1)	0.571 (7)

4.3 Unconditional Generation in Multiple Steps

A critical intermediate finding is that few-step training alone does not recover designability: even with $K > 1$ steps, generators trained via Section 3.2 result in underwhelming designability under standard sampling (with $\gamma = 1$) as shown in Tab. 1. Only when inference-time noise scaling ($\gamma = 0.45$, Section 3.3) is applied does quality improve substantially, confirming that both ingredients of SPID are necessary.

We trained generators with $K \in \{5, 8, 10, 12, 16, 20\}$ steps and reran the pretrained $\mathcal{M}_{\text{FS}}^{\text{no-tri}}$ model for a fair comparison. As shown in Tab. 1 and Fig. 5 in Appendix H, both designability and diversity improve steadily with step count. The teacher's designability is matched once $K \geq 10$, while all SPID generators produce slightly less diverse samples than the teacher.

Tab. 1 shows evaluation metrics for our generators of various numbers of steps, compared with a 10-step generator trained with Diff-Instruct (with similar modification to the distillation training and low temperature sampling) and the pretrained model. Our 16- and 20-step generators can achieve higher designability than the pretrained model, while maintaining a comparable diversity. Our distilled generators are also better at generating novel structures distinct from those in the reference databases. Secondary structure content (α/β ratio) is similar between the 16-step generator and the teacher. We provide a more detailed analysis on the distribution match in Appendix D.

For the Proteína distributional metrics (FPSD, fS, fJSD), our generators score worse than the teacher. One potential explanation for the drop in fold-class-related metrics would be that SPID systematically fails on certain fold classes, skewing the distribution.

Tab. 2 rules this out: our distilled model does not under- or over-represent any fold class. The "failure" rate, defined as the ratio between the number of undesignable structures and the total number of structures belonging to each class, did not demonstrate any notable difference compared to the pretrained model. This confirms that SPID distillation process is fair and did not favor a particular type of proteins.

We hypothesize the drop in these metrics is a consequence of the slight diversity reduction and novelty improvement: a less diverse sample set spans fewer distinct fold classes, lowering fS; and the higher novelty of our samples increases distributional distance from the reference databases, inflating FPSD and fJSD.

Table 3: Fold class-conditional generation performance for our distilled generator compared to the Proteína pretrained model $\mathcal{M}_{\text{FS}}^{\text{no-tri}}$. Optimal performance values are bolded.

Model	Design-ability (%)↑	Diversity Cluster↑	Diversity TM↓	Novelty PDB↓	Novelty AFDB↓	FPSD PDB↓	FPSD AFDB↓	fS (C / A / T)↑	fJSD PDB↓	fJSD AFDB↓	Sec. Struct. % (α/β)
16-step distilled $\mathcal{M}_{\text{FS}}^{\text{no-tri}}$	95.0	0.78 (74)	0.35	**0.82**	**0.83**	473.21	456.62	**1.96 / 4.87 / 13.01**	**1.94**	1.69	61.8 / 5.2
pretrained $\mathcal{M}_{\text{FS}}^{\text{no-tri}}$	**96.0**	**0.82 (79)**	**0.34**	0.83	0.84	**428.82**	**395.73**	1.88 / 4.29 / **19.01**	2.01	**1.43**	65.0 / 4.2

Table 4: Average rank of each method across 24 motif scaffolding targets, where rank 1 indicates the highest number of unique successes. Ties receive the average of the tied ranks. Also reported: median rank and number of targets on which each method is best (including ties), and average pairwise win rate computed as $(N - \bar{r})/(N - 1)$ where $\bar{r}$ is the average rank and $N = 5$.

Method	**Avg. Rank ↓**	**Median Rank ↓**	**# Rank-1 (w/ ties) ↑**	**Win Rate ↑**
SPID	**2.04**	**2.00**	**14**	**0.740**
Proteina	2.62	2.75	9	0.594
Genie2	2.90	3.00	6	0.526
FrameFlow	3.33	3.75	6	0.417
RFDiffusion	4.10	4.25	4	0.224

4.4 Sampling Time

Although sampling time strictly decreases with fewer generation steps, it is crucial to account for the quality of the generated structures. Therefore, we measure the effective sampling time by dividing the total time taken to generate all 500 samples, in the unconditional generation experiment, by the number of designable samples. The last column of Tab. 1 shows the results for the effective sampling time using an A6000-48GB GPU in batches of size 10. The one-step generator has a significantly larger effective sampling time due to its near-zero designability. All other generators displayed shorter effective sampling times from the pretrained model by more than 20 times. Fig. 6 in Appendix H provides an illustration of the impact of the number of steps on the effective sampling time. It confirms that although fewer number of steps generally lead to reduced effective sampling time, lowering the number of steps to fewer than 10 is not beneficial as the designability of the generated samplings decrease drastically.

4.5 Fold Class Conditional Generation

SPID generalizes to fold-class conditional generation without additional conditional distillation training. Using the unconditionally distilled 16-step generator, we provide fold class labels sampled from the empirical joint distribution of length and CATH code (Dawson et al., 2016) from AFDB (Varadi et al., 2021) at inference. As shown in Tab. 3, our distilled generator can achieve comparable metrics as the pretrained model in fold class-conditional generation, although a similar drop in the fold-class-related metrics is observed.

4.6 Motif Scaffolding Benchmark

We evaluate our 16-step distilled motif scaffolding generator on the RFDiffusion benchmark (Watson et al., 2023); baseline results for Proteína, Genie2, RFDiffusion, and FrameFlow are taken from Geffner et al. (2025b).

As shown in Tab. 4, SPID outperforms the pretrained Proteína teacher on 13 of 24 targets and ties on 4. It achieves the highest unique success count among all baselines on 10 targets, the most among the models. Both the mean and median ranks of our distilled generator are the best among all compared models. Among all pairwise comparisons, our distilled model has the highest win rate. On several targets the gains are dramatic: 1QJG (433 vs. 3 for Proteína), 7MRX_128 (253 vs. 51), and 5TRV_long (294 vs. 179). Notably, SPID is the only method to produce any successful designs for 4JHW (17 unique successes vs. 0 for all baselines),

Table 5: Unconditional generation performance for our distilled generator compared to the Genie pretrained model. Optimal performance values are bolded.

Genie Model	Designability (%)		Diversity		Novelty	
	scTM ↑	scRMSD ↑	Cluster↑	TM↓	PDB↓	AFDB↓
10-step distilled	**87.0**	**57.0**	0.66 (45)	0.38	0.79	**0.79**
1000-step pretrained	85.0	39.0	**0.85 (33)**	**0.35**	**0.78**	0.80

suggesting that SPID's distilled generator explores a structurally distinct sampling regime that is inaccessible to the teacher. The performance is lower than Proteína on the 6E6R family and 1YCR; understanding what makes these targets harder for the distilled model is left for future investigation. Full per-target results are in Appendix C, Tab. 6.

4.7 Application to Equivariant Diffusion Architecture

SPID is not limited to flow-based or non-equivariant models. To verify generalization to diffusion-based equivariant architectures with IPA and triangle layers, we distill a 10-step generator from the pretrained 1000-step Genie model (Lin & AlQuraishi, 2023). The SiD generator loss requires backpropagation through $\boldsymbol{x}_g$ into both the fake score network and the teacher; Genie2 (Lin et al., 2024) includes a one-hot encoding step that blocks this gradient, so we apply SPID to the original Genie instead. We additionally report scTM-based designability (TM-score threshold 0.5) as defined in Lin & AlQuraishi (2023).

As shown in Tab. 5, the 10-step distilled Genie model achieves a 100× reduction in sampling steps while improving designability over the 1000-step teacher. Diversity is modestly reduced, likely due to a suboptimal noise scale; we expect that tuning γ for Genie with the same care applied to Proteína would recover diversity, and leave this for future work.

4.8 Biological Plausibility Analyses

To evaluate the structural and functional plausibility of our designed proteins, we conducted a case study on a representative design with high composite confidence (pLDDT = 0.936, pTM = 0.896, RMSD = 0.545), with high novelty score (seqid = 0.14), confirming that the generated sequence is distinct from any known proteins. Fig. 1a shows the protein backbone, which features a stable helical bundle with well-packed secondary structure. Importantly, automated cavity detection (Fpocket) revealed the presence of two adjacent surface pockets (Fig. 1b): one predominantly polar and the other largely hydrophobic.

Pocket 1 (red) has a druggability score of 0.758, a volume of 596 $Å^3$, and a polarity score of 7 (polar-leaning), while Pocket 2 (orange) has a druggability score of 0.796, a volume of 507 $Å^3$, and a polarity score of 3 (hydrophobic-leaning). These scores suggests both pockets are potential pharmaceutically relevant binding sites, with sufficient space to accommodate small molecules and complementary chemical environments. The combination of adjacent chemically distinct pockets supports two plausible binding modes: 1) dual-ligand occupancy, with a polar ligand in Pocket 1 and a hydrophobic ligand in Pocket 2, potentially allowing cooperative binding; 2) single bifunctional ligand, where a fragment-linked molecule bridges both pockets, exploiting complementary polar and nonpolar interactions.

The novel structure of chemically distinct binding sites within a compact fold highlights the potential of our generative pipeline to produce proteins with functionally relevant ligand-binding opportunities. Importantly, this observation underscores not only the structural realism of the designs, but also their direct usability in downstream tasks such as molecular recognition and drug discovery.

5 Conclusion

We propose SPID, a distillation framework for protein backbone generative models that enables few-step generation with low-temperature sampling, a capability critical for protein design but absent from existing

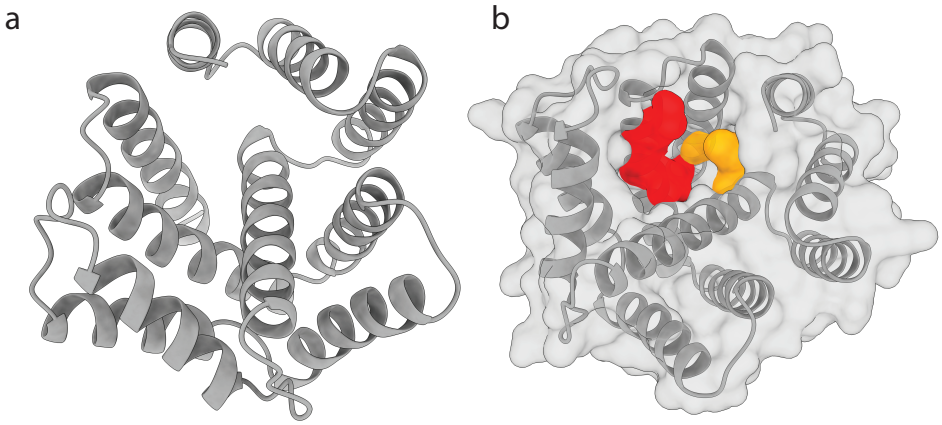

Figure 1: Designed protein backbone and adjacent polar–hydrophobic pocket pair. (a) Front view of the designed protein backbone. (b) Cavity view highlighting two adjacent but chemically distinct pockets: Pocket 1 (red, polar) and Pocket 2 (orange, hydrophobic). The complementary polarities enable two use modes: simultaneous binding of a polar and a hydrophobic ligand, or one bifunctional molecule spanning both pockets.

approaches. SPID applies to both diffusion- and flow-based backbone generators and supports data-free training, making it broadly applicable to large pretrained models. Distilling the flow-based Proteína model without access to training data, SPID generators achieve more than 20× reduction in effective sampling time while matching the teacher in designability, diversity, and novelty across unconditional and fold-class conditional generation. A motif scaffolding distillation using real PDB motifs further demonstrates practical utility in targeted design tasks. SPID also exposes a flexible speed–quality trade-off: increasing step count monotonically improves designability, as shown by our 20-step generator. We observe a modest degradation in fold-class distributional metrics (FPSD, fS, fJSD), which we hypothesize arises from the data-free training setting and unrealistic distributions of length and fold class. Concretely, a way to mitigate this would involve sampling protein lengths and fold-class labels from the AFDB empirical distribution during distillation training, rather than sampling lengths uniformly as in our current unconditional setup. A biological plausibility case study confirms that SPID-generated structures exhibit realistic binding geometry. Generalization to the equivariant Genie architecture demonstrates that SPID is not limited to a single model class. Overall, this work advances the scalability of protein backbone generation and brings us closer to large-scale protein design, while motivating future acceleration of more recent all-atom generation models (Geffner et al., 2025a; Qu et al., 2025; Chu et al., 2024) and other downstream components such as folding models including AlphaFold3 (Abramson et al., 2024). Incorporating real protein structure databases via an adversarial extension (Zhou et al., 2025b) is also natural next step that could further improve designability as well as the fold-class distributional metrics.

Broader Impact Statement

This work introduces a distillation method to accelerate protein backbone generation. The primary societal benefit is the democratization of high-throughput protein design for therapeutic and enzymatic applications, enabling rapid screening with substantially lower energy consumption. However, because our method relies on distilling an existing teacher model, it inherits the biases and potential risks of that teacher. While accelerating generation could theoretically speed up malicious design cycles, the generated backbones still require sequence design and wet-lab synthesis, maintaining existing physical bottlenecks for misuse. We emphasize that our method improves the efficiency of protein design rather than expanding the capability for harmful design beyond current baselines.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, June 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions, November 2023. URL http://arxiv.org/abs/2303.08797. arXiv:2303.08797 [cs].

Namrata Anand and Tudor Achim. Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models, May 2022. URL http://arxiv.org/abs/2205.15019. arXiv:2205.15019 [q-bio].

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, January 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235. URL https://doi.org/10.1093/nar/28.1.235. _eprint: https://academic.oup.com/nar/article-pdf/28/1/235/9895144/280235.pdf.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models, 2025. URL https://arxiv.org/abs/2406.07507.

Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. SE(3)-Stochastic Flow Matching for Protein Backbone Generation, April 2024. URL http://arxiv.org/abs/2310.02391. arXiv:2310.02391 [cs].

Alexander E. Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W. Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences of the United States of America*, 121(27):e2311500121, July 2024. ISSN 1091-6490. doi: 10.1073/pnas.2311500121.

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. doi: 10.1126/science.add2187. URL https://www.science.org/doi/abs/10.1126/science.add2187.

Natalie L. Dawson, Tony E. Lewis, Sayoni Das, Jonathan G. Lees, David Lee, Paul Ashford, Christine A. Orengo, and Ian Sillitoe. CATH: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Research*, 45(D1):D289–D295, November 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw1098. URL https://doi.org/10.1093/nar/gkw1098. _eprint: https://academic.oup.com/nar/article-pdf/45/D1/D289/8846836/gkw1098.pdf.

Michael J Dougherty and Frances H Arnold. Directed evolution: new parts and optimized function. *Current Opinion in Biotechnology*, 20(4):486–491, August 2009. ISSN 0958-1669. doi: 10.1016/j.copbio.2009.08.005. URL https://www.sciencedirect.com/science/article/pii/S0958166909000986.

Raphael R. Eguchi, Christian A. Choe, and Po-Ssu Huang. Ig-VAE: Generative modeling of protein structure by direct 3D coordinate generation. *PLOS Computational Biology*, 18(6):e1010271, June 2022. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1010271. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1010271. Publisher: Public Library of Science.

Xuhui Fan, Zhangkai Wu, and Hongyu Wu. A survey on pre-trained diffusion model distillations, 2025. URL https://arxiv.org/abs/2502.08364.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-proteina: Atomistic protein generation via partially latent flow matching, 2025a. URL https://arxiv.org/abs/2507.09466.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations (ICLR)*, 2025b.

Po-Ssu Huang, Scott E. Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537 (7620):320–327, September 2016. ISSN 1476-4687. doi: 10.1038/nature19946. URL https://www.nature.com/articles/nature19946. Publisher: Nature Publishing Group.

John B. Ingraham, Max Baranov, Zak Costello, Karl W. Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M. Lord, Christopher Ng-Thow-Hing, Erik R. Van Vlack, Shan Tie, Vincent Xue, Sarah C. Cowles, Alan Leung, João V. Rodrigues, Claudio L. Morales-Perez, Alex M. Ayoub, Robin Green, Katherine Puentes, Frank Oplinger, Nishant V. Panwar, Fritz Obermeyer, Adam R. Root, Andrew L. Beam, Frank J. Poelwijk, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, November 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06728-8. URL https://www.nature.com/articles/s41586-023-06728-8. Publisher: Nature Publishing Group.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2. Publisher: Nature Publishing Group.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion, 2024. URL https://arxiv.org/abs/2310.02279.

G. Labesse, N. Colloc'h, J. Pothier, and J. P. Mornon. P-SEA: a new efficient assignment of secondary structure from C alpha trace of proteins. *Computer applications in the biosciences: CABIOS*, 13(3):291–295, June 1997. ISSN 0266-7061. doi: 10.1093/bioinformatics/13.3.291.

Yeqing Lin and Mohammed AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2, 2024. URL https://arxiv.org/abs/2405.15489.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/abs/10.1126/science.ade2574.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhiyuan Ma, Yuzhu Zhang, Guoli Jia, Liangliang Zhao, Yichao Ma, Mingjie Ma, Gaofeng Liu, Kaiyan Zhang, Jianjun Li, and Bowen Zhou. Efficient diffusion models: A comprehensive survey from principles to practices, 2024. URL https://arxiv.org/abs/2410.11795.

Wei Qu, Jiawei Guan, Rui Ma, Ke Zhai, Weikun Wu, and Haobo Wang. P(all-atom) Is Unlocking New Path For Protein Design, May 2025. URL https://www.biorxiv.org/content/10.1101/2024.08.16.608235v4.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation, 2025. URL https://arxiv.org/abs/2506.14603.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022. URL https://arxiv.org/abs/2202.00512.

Hui Shen, Jingxuan Zhang, Boning Xiong, Rui Hu, Shoufa Chen, Zhongwei Wan, Xin Wang, Yu Zhang, Zixuan Gong, Guangyin Bao, Chaofan Tao, Yongfeng Huang, Ye Yuan, and Mi Zhang. Efficient diffusion models: A survey. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=wHECkBOwyt. Survey Certification.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models, May 2023. URL http://arxiv.org/abs/2303.01469. arXiv:2303.01469 [cs].

Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem, 2023. URL https://arxiv.org/abs/2206.04119.

Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42(2):243–246, February 2024. ISSN 1546-1696. doi: 10.1038/s41587-023-01773-0. URL https://www.nature.com/articles/s41587-023-01773-0. Publisher: Nature Publishing Group.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, Augustin Žídek, Tim Green, Kathryn Tunyasuvunakool, Stig Petersen, John Jumper, Ellen Clancy, Richard Green, Ankur Vora, Mira Lutfi, Michael Figurnov, Andrew Cowie, Nicole Hobbs, Pushmeet Kohli, Gerard Kleywegt, Ewan Birney, Demis Hassabis, and Sameer 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, November 2021. ISSN 0305-1048. doi: 10.1093/nar/gkab1061. URL https://doi.org/10.1093/nar/gkab1061. _eprint: https://academic.oup.com/nar/article-pdf/50/D1/D439/43502749/gkab1061.pdf.

Chentong Wang, Yannan Qu, Zhangzhi Peng, Yukai Wang, Hongli Zhu, Dachuan Chen, and Longxing Cao. Proteus: Exploring protein structure generation for enhanced designability and efficiency. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=IckJCzsGVS.

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=HZf7UbpWHuA.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=ppJuFSOAnM.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington,

Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8. URL https://www.nature.com/articles/s41586-023-06415-8. Publisher: Nature Publishing Group.

Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, Tommi Jaakkola, and Frank Noé. Fast protein backbone generation with SE(3) flow matching, October 2023a. URL http://arxiv.org/abs/2310.05297. arXiv:2310.05297 [q-bio].

Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation, May 2023b. URL http://arxiv.org/abs/2302.02277. arXiv:2302.02277 [cs].

Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, December 2004. ISSN 0887-3585, 1097-0134. doi: 10.1002/prot.20264.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=QhqQJqe0Wq.

Mingyuan Zhou, Yi Gu, and Zhendong Wang. Few-step diffusion via score identity distillation, 2025a. URL https://arxiv.org/abs/2505.12674.

Mingyuan Zhou, Huangjie Zheng, Yi Gu, Zhendong Wang, and Hai Huang. Adversarial score identity distillation: Rapidly surpassing the teacher in one step. In *International Conference on Learning Representations*, 2025b.

Appendix

A Algorithms

Algorithm 1 details the training procedure for SPID.

Algorithm 2 specifies the inference-time sampling procedure. Note that at inference, Eq. 10 with noise scale is used instead of the standard Eq. 9 used during training.

B Effect of Sampling Noise Scale

As discussed by prior works, the noise scale during sampling is crucial for protein structure generation tasks. We empirically verify the impact of noise scaling and determine our optimal noise scale.

As shown in Fig. 2, the designability is the highest when the noise scale is around 0.45. It drastically drops as the noise scale increases beyond 0.5. As for the diversity, it worsens significantly as the noise scale decreases below 0.4. Therefore, it seems the optimal noise scale for our 10-step generator is between 0.4 and 0.5. We chose 0.45 eventually, which coincides with the value picked for the $\mathcal{M}_{\text{FS}}^{\text{no-tri}}$ model in Proteína .

C Motif Scaffolding Results

Motif scaffolding results are generated in the same way as described in the Geffner et al. (2025b) paper. We generated 1000 designs per target in the RFDiffusion benchmark set (Watson et al., 2023). For each backbone design, 8 sequences are generated by ProteinMPNN and fed to ESMFold. The scRMSD is then computed using the predicted backbones. The motifRMSD is the RMSD between the predicted backbone and the

Algorithm 1 SPID training loop for protein backbone generation

Input: Pretrained score network f_ϕ, generator G_θ, generator score network f_ψ, $t_{init} = 0.37$, $t_{min} = 0.02$, $t_{max} = 0.98$, number of generation steps $K = 16$, $\alpha = 1.0$, optimizer learning rates $\eta_\psi = 1e^{-4}$ and $\eta_\theta = 5e^{-5}$.
Initialization: $\theta \leftarrow \phi, \psi \leftarrow \phi, t_{steps} \leftarrow K$ equally spaced points between 30 and 400
repeat
  Sample target protein length $N \in \{50, \cdots, 256\}$.
  if The number of generation steps $K = 1$ **then**
    Sample $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$ and set $\boldsymbol{x}_g = G_\theta(t_{\text{init}}\boldsymbol{z})$
  else
    Sample k from $\{1, ..., K\}$, then sample $\boldsymbol{x}_g$ in k steps recursively using Eq. (9)
  end if
  Sample t using Eq. (11), and clamp t to be between t_{min} and t_{max}.
  Sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$. Set $\boldsymbol{x}_t = t\text{sg}(\boldsymbol{x}_g) + (1-t)\epsilon_t$
  Update ψ using 4:
    $\mathcal{L}_\psi = \frac{1}{N(1-t)^2}\|f_\psi(x_t, t) - \text{sg}(\boldsymbol{x}_g)\|_2^2$, where N is the number of residues.
    $\psi \leftarrow \psi - \eta_\psi \nabla_\psi \mathcal{L}_\psi$
  if The number of generation steps $K = 1$ **then**
    Sample $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$ and set $\boldsymbol{x}_g = G_\theta(t_{\text{init}}\boldsymbol{z})$
  else
    Sample k from $\{1, ..., K\}$ sample $\boldsymbol{x}_g$ in k steps using Eq. 9
  end if
  Sample t using Eq. (11), and clamp t to be between t_{min} and t_{max}. Compute $\omega(t)$ as defined in Eq. (8).

  Sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$. Set $\boldsymbol{x}_t = t\boldsymbol{x}_g + (1-t)\boldsymbol{\epsilon}_t$
  Update θ using Eq. (7)
    $\mathcal{L}_\theta = \frac{\omega(t)t^2}{(1-t)^4}[(1-\alpha)\|f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t)\|_2^2 + (f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t))^T(f_\psi(\boldsymbol{x}_t, t) - \boldsymbol{x}_g)]$
    $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_\theta$
until The average scRMSD plateaus or the training budget is exhausted
Output: G_θ

Algorithm 2 SPID inference for protein backbone generation

Input: Generator G_θ, $t_{init} = 0.37$, $\gamma = 0.45$, number of generation steps $K = 16$, target protein length N.
Initialization: $t_{steps} \leftarrow K$ equally spaced points between 30 and 400
if The number of generation steps $K = 1$ **then**
  Sample $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$ and set $\boldsymbol{x}_g = G_\theta(t_{\text{init}}\boldsymbol{z})$
else
  Sample $\boldsymbol{x}_0 \sim \mathcal{N}(0, \mathbf{I})$
  for $k \in \{1, \cdots, K\}$ **do**
    Sample $\boldsymbol{x}_{k+1}$ from $\boldsymbol{x}_k$ using Eq. (10) with noise scale γ and time path t_{steps}
  end for
end if
Output: $\boldsymbol{x}_K$

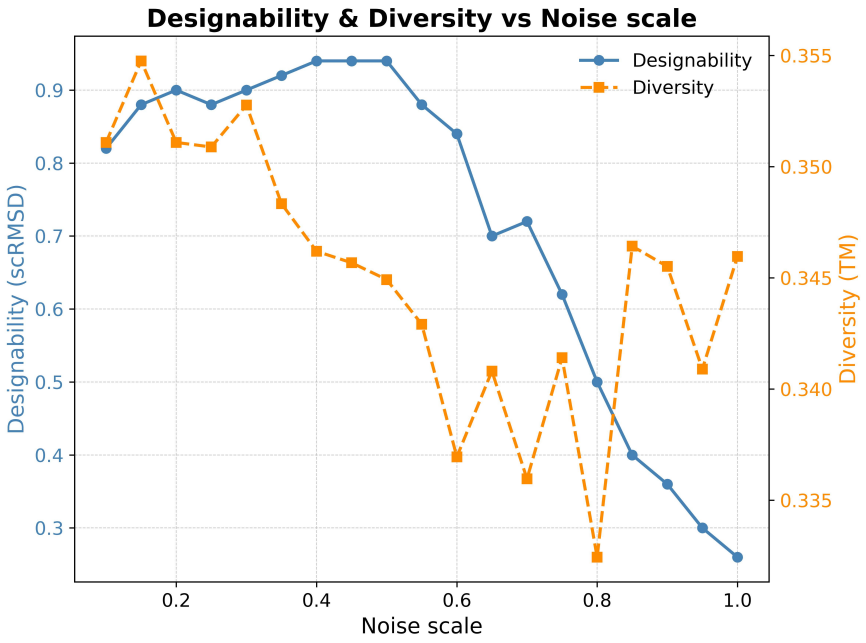

Figure 2: An illustration of the effects of the noise scale on the designability (scRMSD) and diversity (TM) of our 10-step generator. For designability, it is the fraction of samples in the batch with scRMSD < 2Å. The higher the designability the better. For the diversity metric, it is measuring the average similarity (TM-score) within the batch. Lower values indicate more diversity. Setting the noise scale to 1 is the standard way of sampling in diffusion and flow matching models for images, which results in close-to-0 designability for protein structures. The designability is the highest for noise scales around 0.45 while the best diversity is reached at 0.8.

Table 6: Motif scaffolding RFDiffusion benchmark: number of unique successes for each model out of 1,000 designs per target. Baselines are Proteina (Geffner et al., 2025b), Genie2 (Lin et al., 2024), RFDiffusion (Watson et al., 2023), and FrameFlow (Yim et al., 2023a). *vs Proteina* and *vs Best* report the absolute difference between our method and Proteina, and between our method and the best baseline, respectively.

Target	**Proteina**	**Genie2**	**RFDiff**	**FrameFlow**	**Ours (SPID)**	**vs Proteina**	**vs Best**
1QJG	3	5	1	18	**433**	+430	+415
7MRX_128	51	27	66	35	**253**	+202	+187
5TRV_long	179	97	23	77	**294**	+115	+115
6EXZ_long	290	326	167	**403**	371	+81	−32
6EXZ_medium	43	54	25	**110**	102	+59	−8
5YUI	5	3	1	1	**52**	+47	+47
5TPN	4	8	5	6	**42**	+38	+34
5TRV_medium	22	23	10	21	**53**	+31	+30
6EXZ_short	3	2	1	3	**22**	+19	+19
4JHW	0	0	0	0	**17**	+17	+17
3IXT	8	**14**	3	8	13	+5	−1
5TRV_short	1	3	1	1	**6**	+5	+3
7MRX_85	31	23	13	22	**34**	+3	+3
2KL8	**1**	**1**	**1**	**1**	**1**	0	0
1BCF	**1**	**1**	**1**	**1**	**1**	0	0
5IUS	**1**	**1**	**1**	0	**1**	0	0
1PRW	**1**	**1**	**1**	**1**	**1**	0	0
7MRX_60	2	**5**	1	1	1	−1	−4
5WN9	2	1	0	**3**	1	−1	−2
4ZYP	**11**	3	6	4	5	−6	−6
6E6R_short	**56**	26	23	25	32	−24	−24
6E6R_long	**713**	415	381	110	533	−180	−180
1YCR	**249**	134	7	149	55	−194	−194
6E6R_medium	**417**	272	151	99	217	−200	−200

ground truth motif. Successful designs are filtered out using the criteria: scRMSD $\leq$ 2Å, motifRMSD $\leq$ 1Å, pLDDT $\geq$ 70, and pAE $\leq$ 5. The successful designs are then clustered via hierarchical clustering with single linkage and a TM-score threshold of 0.6 to reach the final number of unique successes, which are reported in 6.

D Secondary Structure Distributions

We evaluated the structural fidelity of the generated backbones by quantifying their secondary structure content, specifically the percentages of α-helices and β-strands. As illustrated in Fig. 3, the distributions produced by the distilled model closely align with those of the pretrained baseline. This confirms that SPID faithfully reproduces the structural statistics of the teacher distribution.

E Ablation Study and Parameter Settings

Impact of α. Similar to the original SiD paper (Zhou et al., 2024), we conduct an ablation study for the impact of α on the distillation training process. Designability metrics including the average scTM, the average scRMSD, and the designability based on the proportions of structures with scRMSD $<$ 2Å in a batch are plotted in Fig. 4 as it evolves from 0 to 102.4 thousand samples processed during training. Across the α values of [0, 0.5, 0.8, 1.0, 1.2, 1.5], we see the best designability performance with α being set to 0.8 or 1.0. With other α values, we do not observe a meaningful convergence towards an improved designability. Between

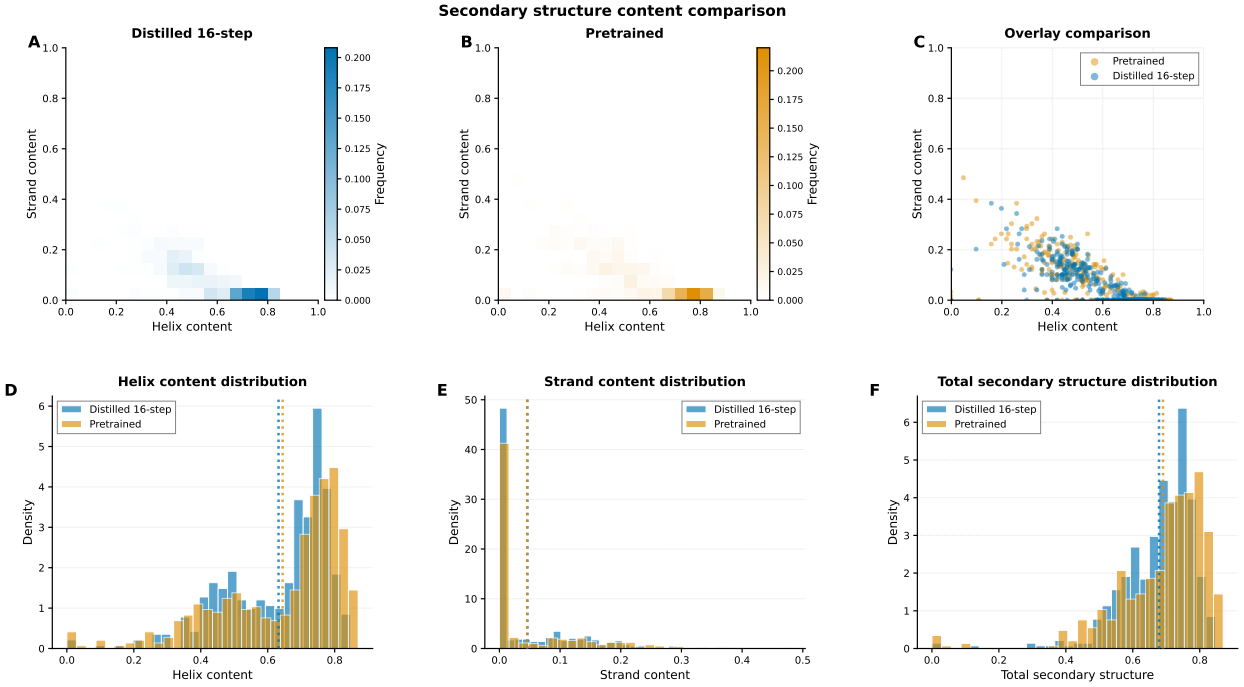

Figure 3: Distribution of secondary structures in the generated protein backbones. Panels A and B show heatmaps for the helix and strand percentages for the 16-step distilled model and the pretrained model respectively. Panel C shows and overlay of each generated backbone. Panels D, E, and F show histograms of the helix percentage, strand percentage, and total secondary structure percentage respectively.

the values 0.8 and 1.0, it seems setting $\alpha = 1.0$ results in a slightly smoother and more stable convergence. Hence, we select $\alpha = 1$ for all our experiments.

Impact of β_1. We follow a similar experimental process as in the SiD paper (Zhou et al., 2024) to investigate the impact of the β_1 parameter for the Adam optimizer. We compare the performance of setting β_1 to 0 and 0.9 respectively for the optimizer for the generator G_θ. We did not observe any notable difference in either the convergence speed or the final designability. The β_1 parameter for the optimizer for the fake score network, f_ψ was kept at 0. As reported in Zhou et al. (2024), setting $\beta_1 = 0.9$ often does not result in convergence.

Impact of batch size and learning rate. We initially set the overall batch size to 256 and failed to observe any noticeable improvement in designability. As we increase the batch size to 1024, 2048 and eventually 4096, we observe more and more stable convergence to the level of designability of the pretrained model, with the combination of batch size set to 4096 and learning rate set to 1e-4 and 5e-5 for f_ψ and G_θ respectively resulting in our best performance. Although it is tempting to conclude that the larger the batch size the better, it is also possible that the unstable performance was due to the learning rate not being optimized for the smaller batch sizes. A more comprehensive grid search might be needed to uncover the optimal setting for the batch size and learning rate. It is also worth noting that the batch size per GPU was set to 2 because the model can be memory-intensive for longer proteins. This results in a large number of gradient accumulation rounds.

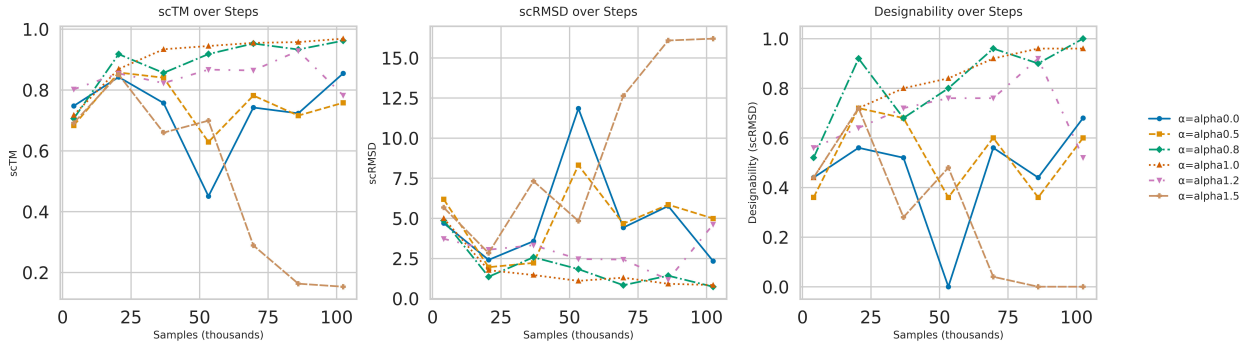

Figure 4: Ablation study of α. Each plot illustrates the generation quality, measured in the average scTM, average scRMSD, and scRMSD-based designability within each generated batch against the number of training samples processed during distillation across different α values. The ablation study showcases the effect of α on distillation training and highlights the reason why α is set to 1.0 for subsequent experiments.

F Proofs

As the score function is the key quantity for SiD, we derive it under the flow matching formulation in Proteína . For a pretrained network f_ϕ with original data $\boldsymbol{x}_d$ and noised sample $\boldsymbol{x}_t$ at timestep t, the score can be

written as:

$$S_\phi(\boldsymbol{x}_t) = \frac{t\boldsymbol{v}_t^\phi(\boldsymbol{x}_t,t) - \boldsymbol{x}_t}{1-t} \tag{12}$$
$$= \frac{t\mathbb{E}(\boldsymbol{x}_d - \boldsymbol{\epsilon}_t|\boldsymbol{x}_t) - \boldsymbol{x}_t}{1-t} \tag{13}$$
$$= \frac{\mathbb{E}(t\boldsymbol{x}_d - t\boldsymbol{\epsilon}_t - (t\boldsymbol{x}_d + (1-t)\boldsymbol{\epsilon}_t)|\boldsymbol{x}_t)}{1-t} \tag{14}$$
$$= -\frac{\mathbb{E}(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t)}{1-t} \tag{15}$$
$$= -\frac{\mathbb{E}(\frac{1}{1-t}(\boldsymbol{x}_t - t\boldsymbol{x}_d)\,|\,\boldsymbol{x}_t)}{1-t} \tag{16}$$
$$= (1-t)^{-2}\mathbb{E}((t\boldsymbol{x}_d - \boldsymbol{x}_t)\,|\,\boldsymbol{x}_t) \tag{17}$$
$$= (1-t)^{-2}(t\mathbb{E}(\boldsymbol{x}_d\,|\,\boldsymbol{x}_t) - \boldsymbol{x}_t) \tag{18}$$
$$= (1-t)^{-2}(tf_\phi(\boldsymbol{x}_t,t) - \boldsymbol{x}_t) \tag{19}$$

where $f_\phi(\boldsymbol{x}_t,t) = \boldsymbol{x}_t + (1-t)v_t^\phi(\boldsymbol{x}_t,t)$ and v_t^ϕ is the velocity function for f_ϕ.

The score difference objective in Eq. 3 also involves the score of the generator, $\nabla_{\boldsymbol{x}_t}\ln p_\theta(\boldsymbol{x}_t)$. To estimate the generator score, we use another neural network f_ψ and train it using the same training objective as Proteína .

$$\min_\psi \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_g,t)p_\theta(\boldsymbol{x}_g)}[\frac{1}{N}\|\boldsymbol{v}_t^\psi(\boldsymbol{x}_t,t) - (\boldsymbol{x}_g - \boldsymbol{\epsilon})\|_2^2] \tag{20}$$

, where N denotes the number of residues in the desired protein structure.

We can derive the relationship between the network estimation $f_\phi(\boldsymbol{x}_t,t)$ and the velocity function $v_t^\phi(\boldsymbol{x}_t,t)$:

$$f_\phi(\boldsymbol{x}_t,t) = \mathbb{E}(\boldsymbol{x}_d|\boldsymbol{x}_t) \tag{21}$$
$$= \mathbb{E}(\frac{1}{t}(\boldsymbol{x}_t - (1-t)\boldsymbol{\epsilon}_t|\boldsymbol{x}_t) \tag{22}$$
$$= \frac{1}{t}(\boldsymbol{x}_t - (1-t)\mathbb{E}(\boldsymbol{x}_d - \boldsymbol{v}_t^\phi(\boldsymbol{x}_t,t)|\boldsymbol{x}_t) \tag{23}$$
$$\Rightarrow tf_\phi(\boldsymbol{x}_t,t) = \boldsymbol{x}_t - (1-t)f_\phi(\boldsymbol{x}_t,t) + (1-t)v_t^\theta(\boldsymbol{x}_t,t) \tag{24}$$
$$\Rightarrow f_\phi(\boldsymbol{x}_t,t) = \boldsymbol{x}_t + (1-t)\boldsymbol{v}_t^\phi(\boldsymbol{x}_t,t) \tag{25}$$
$$\Rightarrow \boldsymbol{v}_t^\phi(\boldsymbol{x}_t,t) = \frac{1}{1-t}(f_\phi(\boldsymbol{x}_t,t) - \boldsymbol{x}_t) \tag{26}$$

and therefore,

$$\boldsymbol{v}_t^\phi(\boldsymbol{x}_t,t) - (\boldsymbol{x}_d - \boldsymbol{\epsilon}) = \frac{1}{1-t}(f_\phi(\boldsymbol{x}_t,t) - \boldsymbol{x}_t) - \boldsymbol{x}_d + \boldsymbol{\epsilon} \tag{27}$$
$$= \frac{1}{1-t}f_\phi(\boldsymbol{x}_t,t) - \frac{1}{1-t}(t\boldsymbol{x}_d + (1-t)\boldsymbol{\epsilon}) - \boldsymbol{x}_d + \boldsymbol{\epsilon} \tag{28}$$
$$= \frac{1}{1-t}f_\phi(\boldsymbol{x}_t,t) - (\frac{t}{1-t} + 1)\boldsymbol{x}_d \tag{29}$$
$$= \frac{1}{1-t}(f_\phi(\boldsymbol{x}_t,t) - \boldsymbol{x}_d) \tag{30}$$

Note that the network f_ψ is essentially trained by treating the samples $\boldsymbol{x}_g$ produced by the generator G_θ as the true clean data. Thus, the above relationship can be rewritten for f_ψ as $\boldsymbol{v}_t^\psi(\boldsymbol{x}_t,t) - (\boldsymbol{x}_g - \boldsymbol{\epsilon}) = \frac{1}{1-t}(f_\psi(\boldsymbol{x}_t,t) - \boldsymbol{x}_g)$ and the score of f_ψ is $S_\psi(\boldsymbol{x}_t) = (1-t)^{-2}(t\mathbb{E}(\boldsymbol{x}_g\,|\,\boldsymbol{x}_t) - \boldsymbol{x}_t)$.

Hence, Eq. 20 is equivalent to

$$\min_\psi \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_g,t)p_\theta(\boldsymbol{x}_g)}[\frac{1}{N(1-t)^2}\|f_\psi(\boldsymbol{x}_t,t) - \boldsymbol{x}_g\|_2^2] \tag{31}$$

We can then approximate the loss in Eq. 3 as the score difference between f_ϕ and f_ψ, and define:

$$\mathcal{L}_\theta^{(1)} = \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}[\|S_\phi(\boldsymbol{x}_t) - S_\psi(\boldsymbol{x}_t)\|_2^2] \tag{32}$$

$$= \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}[\|\frac{t}{(1-t)^2}(f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t)\|_2^2] \tag{33}$$

Expanding the L_2 norm of the target score difference in Eq. 3, we have:

$$\mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}[\|S_\phi(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \ln\, p_\theta(\boldsymbol{x}_t)\|_2^2] \tag{34}$$

$$\approx \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}[(S_\phi(\boldsymbol{x}_t) - S_\psi(\boldsymbol{x}_t))^T(S_\phi(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \ln\, p_\theta(\boldsymbol{x}_t))] \tag{35}$$

$$= \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}[(S_\phi(\boldsymbol{x}_t) - S_\psi(\boldsymbol{x}_t))^T(\frac{1}{(1-t)^2}(t\mathbb{E}(\boldsymbol{x}_d|\boldsymbol{x}_t) - \boldsymbol{x}_t))] \tag{36}$$

$$- \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}\mathbb{E}_{\boldsymbol{x}_t \sim q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_g, t)}[(S_\phi(\boldsymbol{x}_t) - S_\psi(\boldsymbol{x}_t))^T\nabla_{\boldsymbol{x}_t} \ln\, q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_g)] \tag{37}$$

$$= \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}[(S_\phi(\boldsymbol{x}_t) - S_\psi(\boldsymbol{x}_t))^T(\frac{1}{(1-t)^2}(t\mathbb{E}(\boldsymbol{x}_d|\boldsymbol{x}_t) - \boldsymbol{x}_t))] \tag{38}$$

$$- \mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}\mathbb{E}_{\boldsymbol{x}_t \sim q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_g, t)}[(S_\phi(\boldsymbol{x}_t) - S_\psi(\boldsymbol{x}_t))^T(\frac{1}{(1-t)^2}(t\boldsymbol{x}_g - \boldsymbol{x}_t))] \tag{39}$$

$$= \frac{t}{(1-t)^2}\mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t)}\mathbb{E}_{\boldsymbol{x}_t \sim q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_g, t)}[(S_\phi(\boldsymbol{x}_t) - S_\psi(\boldsymbol{x}_t))^T(\mathbb{E}(\boldsymbol{x}_d|\boldsymbol{x}_t) - \boldsymbol{x}_g)] \tag{40}$$

Therefore, we define

$$\mathcal{L}_\theta^{(2)} = \frac{t}{(1-t)^2}\mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t), \boldsymbol{x}_t \sim q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_g, t)}[(S_\phi(\boldsymbol{x}_t) - S_\psi(\boldsymbol{x}_t))^T(f_\phi(\boldsymbol{x}_t, t) - \boldsymbol{x}_g)] \tag{41}$$

$$= \frac{t}{(1-t)^2}\mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t), \boldsymbol{x}_t \sim q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_g, t)}[\frac{t}{(1-t)^2}(f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t))^T(f_\phi(\boldsymbol{x}_t, t) - \boldsymbol{x}_g)] \tag{42}$$

$$= \frac{t^2}{(1-t)^4}\mathbb{E}_{\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t), \boldsymbol{x}_t \sim q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_g, t)}[(f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t))^T(f_\phi(\boldsymbol{x}_t, t) - \boldsymbol{x}_g)] \tag{43}$$

and thus the final SiD loss, $\mathcal{L}_\theta = \mathcal{L}_\theta^{(2)} - \alpha\mathcal{L}_\theta^{(1)}$ as:

$$\mathcal{L}_\theta = \frac{\omega(t)t^2}{(1-t)^4}[(1-\alpha)\|f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t)\|_2^2 + (f_\phi(\boldsymbol{x}_t, t) - f_\psi(\boldsymbol{x}_t, t))^T(f_\psi(\boldsymbol{x}_t, t) - \boldsymbol{x}_g)] \tag{44}$$

G Evaluation Pipeline

Our evaluation pipeline largely follows that described in Geffner et al. (2025b), with the same metrics for designability, diversity, and novelty, as well as its proposed metrics of FPSD, fS, and fJSD. In the self-consistency pipeline for **designability**, each generated structure is passed to ProteinMPNN (Dauparas et al., 2022) to generate 8 candidate sequences. Each of the 8 sequences is fed to ESMFold (Lin et al., 2023) for predict the corresponding structure. Each ESMFold-predicted structure is then compared with the original generated structure in terms of RMSD. The best RMSD among the 8 predicted structures is reported as the scRMSD. A generated structure is considered designable if the scRMSD is less than 2Å. **Diversity** is measured by both the average pairwise TM-score (Zhang & Skolnick, 2004) among designable samples and the proportion of designable clusters among all designable samples. **Novelty** is calculated by first computing the highest TM-score between each generated structure and each structure in the reference dataset, either PDB (Berman et al., 2000) or AFDB (Varadi et al., 2021). The average of these maximum TM-scores is reported as the novelty metric. The **secondary structure content** is calculated based on the P-SEA algorithm (Labesse et al., 1997). The implementation of the evaluation pipeline, however, is based on both Proteína (Geffner et al., 2025b) and Genie (Lin & AlQuraishi, 2023). We used the code for the evaluation pipeline provided by Genie to calculate the scRMSD, diversity based on the TM-score, and the secondary structure contents. For

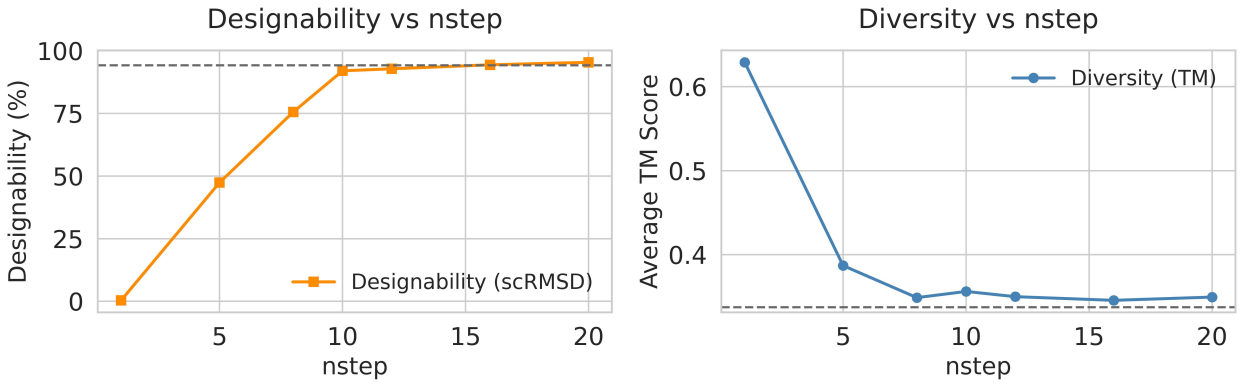

Figure 5: Plots of designability and diversity versus the number of generation steps. On the left, we show the designability as the percentage of generated samples that meet the designable threshold (scRMSD < 2). On the right, we show the diversity as the average pairwise TM-scores between designable samples. The lower the average TM-score is, the more diverse the generated samples are. 16 steps seems enough to beat the pretrained model in designability, while being slightly worse in diversity.

the cluster-based diversity and the novelty metrics, we followed the Foldseek commands specified in Proteína . In addition, we use the implementation and the fold class predictor network provided by Proteína for the FPSD, fS, and fJSD metrics. Since the implementation is slightly different, we reran the metrics on the pretrained model instead of copying over the reported results to ensure a fair comparison. For unconditional generation, we generate 100 samples with batch size 10 for protein lengths [50, 100, 150, 200, 250]. We use an Nvidia A6000-48GB GPU and track the total time for generating all samples for timing analyses.

H Impact of Number of Sampling Steps

Increasing the number of sampling steps for our few-step distilled generator strictly improves both designability and diversity. Comparable performance against the teacher is achieved with $K > 10$ steps. Designability exceeds the teacher for our 16-step generator.

Although the intuition is that the generation time strictly decreases as the number of sampling steps decrease, it is important to account for the quality of the generated structures in the evaluation of sampling time. As shown in Fig. 6, our one-step generator, on average, takes the longest time by far to generate one designable sample compared to other generators. The reason is that the drop in the designability performance outweighs the gain in generation speed for the one-step generator. We argue that the effective sampling time more faithfully reflects the speed of generation, compared to simple metrics such as the number of sampling steps or time taken to generate one sample.

I Long chain generation

One of the strengths of Proteína is its capability of long chain generation, maintaining high designability for proteins of up to 800 residues. To verify that SPID can distill this ability to our generator, we trained a 32-step unconditional generator and evaluated its long-chain generation performance.

We have found that the number of steps is very influential for these longer chains. We had to increase the number of steps to 32 to achieve comparable performance to the pretrained model. Yet we still see an abrupt drop in performance at length 500.

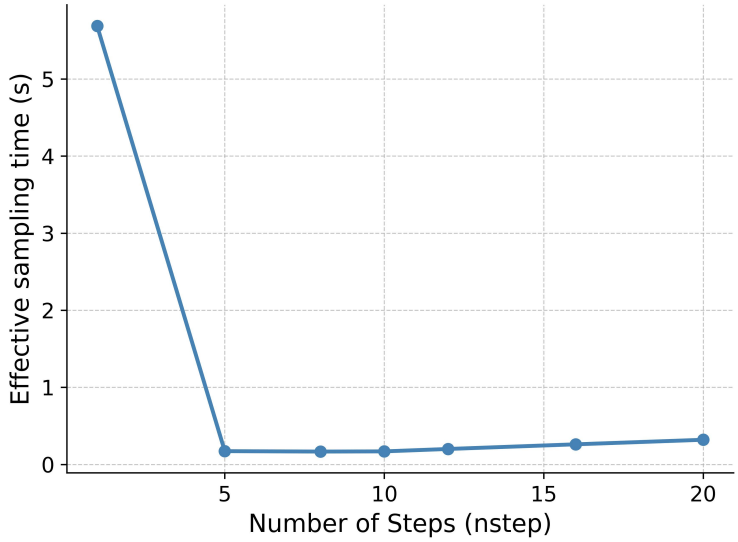

Figure 6: Effective Sampling Time against the number of generator steps. 500 batches generated in batches of 10 on an A6000-48GB GPU.

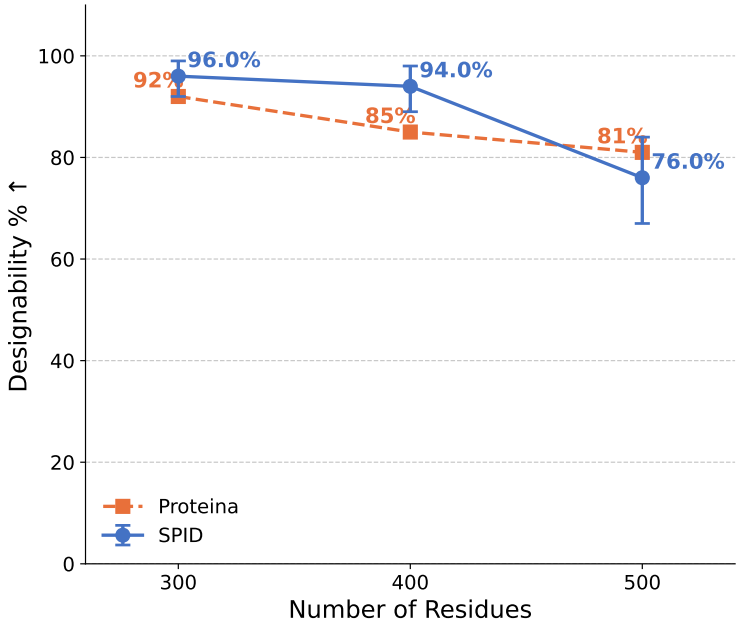

Figure 7: Designability for proteins of lengths in [300, 400, 500]. Orange dotted line represents the designability reported in Geffner et al. (2025b). Blue solid line represent our 32-step generator performance.

J Example of Generated Structures

Fig. 8 shows backbones generated unconditionally by our 16-step distilled generator model. Fig. 9 shows backbones generated while conditioned on fold-class labels and lengths sampled from the empirical distribution of AFDB. Both sets of examples are randomly selected from the generated structures that are considered designable.

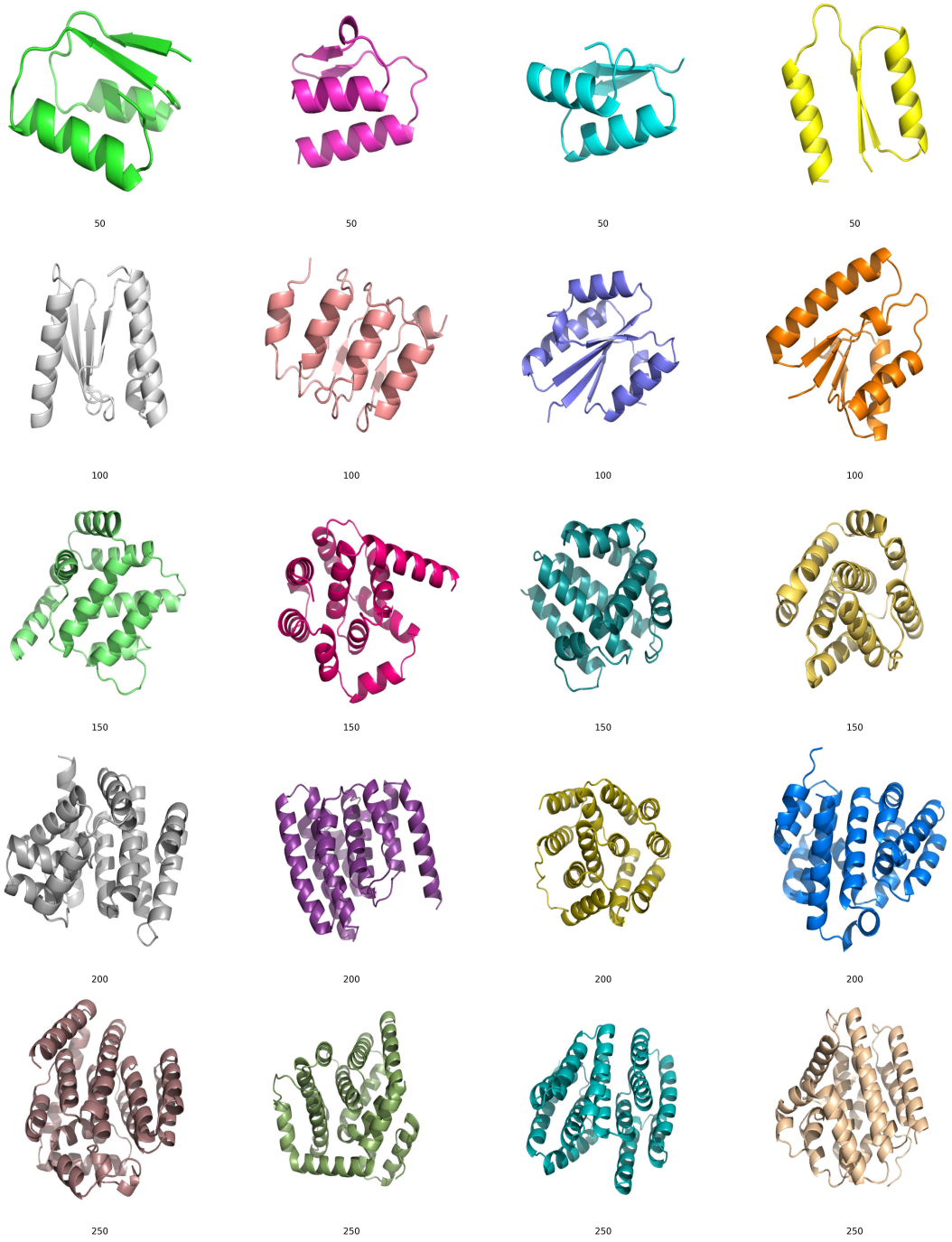

Figure 8: Example of structures generated unconditionally of varying lengths. The lengths are displayed below each structure. All structures shown here are designable.

Figure 9: Examples of generated structures conditioned on the fold class as specified by the CATH code. The length and the C.A.T fold class label are denoted below the structures. All structures shown here are designable.

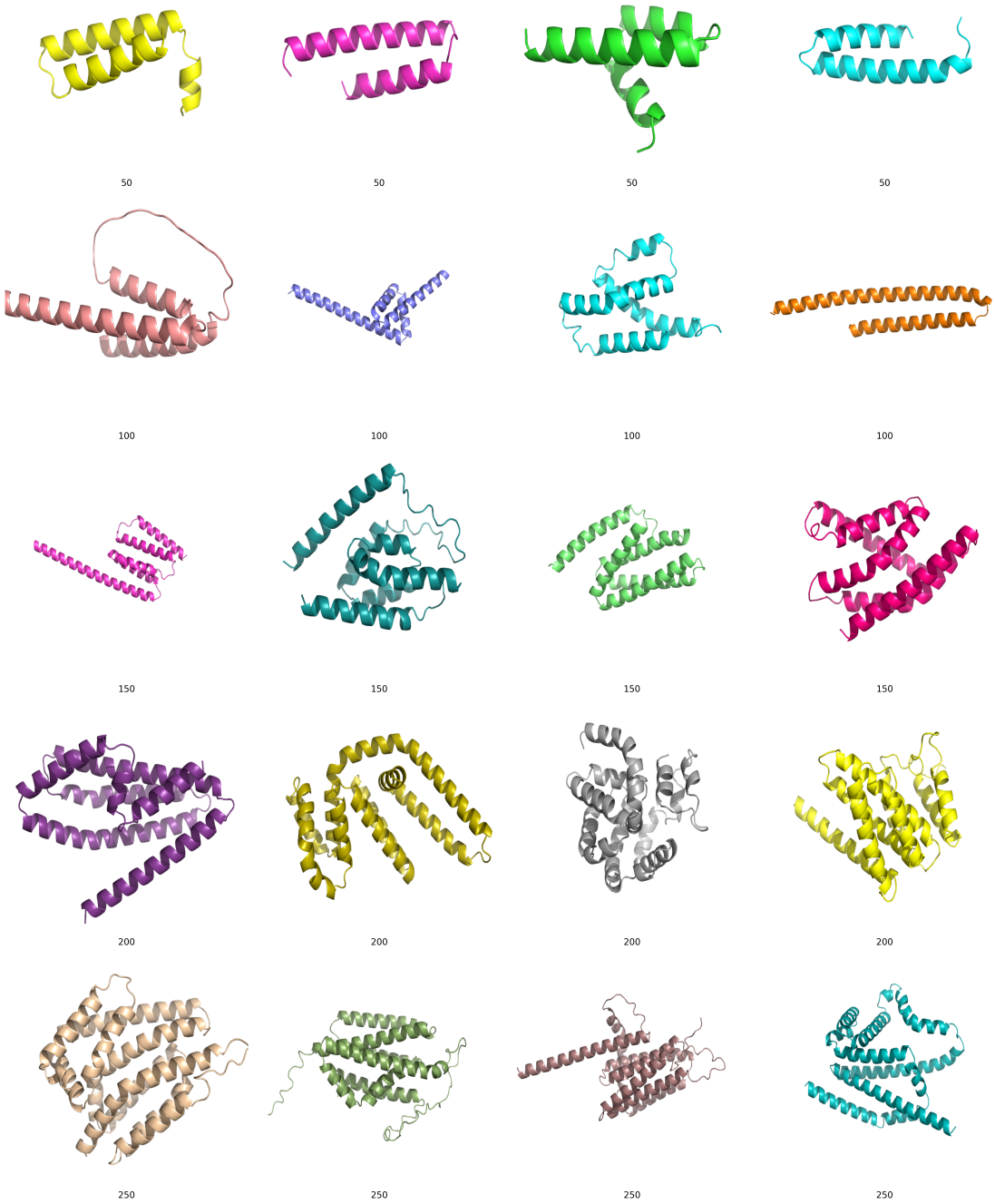

Figure 10: Examples of structures generated unconditionally by our one-step distilled generator that are considered undesignable. The lengths are displayed below each structure.

K Example of Failure Cases

Examining these undesignable structures carefully, it is evident that our one-step generator favors alpha helices. No beta sheet has been generated in these samples. Additionally, there exists some exceptionally long

alpha helices, such as the ones in the last example of the $N = 100$ row and the first sample in the $N = 150$ row. Furthermore, the alpha helices tend to appear in 180 degree angles to one another, which is again unusual in stable protein structures. Overall, the one-step generator was not able to model the geometric properties of protein structures.

L Reproducibility Statement

We ensure full reproducibility of our training, sampling, and evaluation pipeline. The model architecture and weights of the pretrained Proteína model (Geffner et al., 2025b) are publicly available at https://github.com/NVIDIA-Digital-Bio/proteina/tree/main under the Nvidia license for non-commercial or research purposes only. The distillation code is based on SiD (Zhou et al., 2024) (https://github.com/mingyuanzhou/SiD, Apache-2.0 license), and our evaluation pipeline integrates publicly available components from Genie (Lin & AlQuraishi, 2023) (https://github.com/aqlaboratory/genie/tree/main, Apache-2.0 license) and Proteína (Geffner et al., 2025b) (Nvidia license). Other models used for evaluation such as FoldSeek (van Kempen et al., 2024) (https://github.com/steineggerlab/foldseek, GPL-3.0 license), ProteinMPNN (Dauparas et al., 2022) (https://github.com/dauparas/ProteinMPNN, MIT license), ESMFold (Lin et al., 2023) (https://github.com/facebookresearch/esm, MIT license), and TM-score and TM-align (Zhang & Skolnick, 2004) (https://aideepmed.com/, permissive custom license) are publicly available as well. We used the newest version as of January 1st, 2026 for all aforementioned software. Therefore, our training, sampling, and evaluation pipeline are fully reproducible. Anonymized code for reproducing all main experimental results is provided in the supplemental material. The weights of the distilled models will be made publicly available, subject to the original license constraint of Proteína .