Ambient Proteins
Training Diffusion Models on Noisy Structures

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present *Ambient Protein Diffusion*, a framework for training protein diffusion models that generates structures with unprecedented diversity and quality. State-of-the-art generative models are trained on computationally derived structures from AlphaFold2 (AF), as experimentally determined structures are relatively scarce. The resulting models are therefore limited by the quality of synthetic datasets. Since the accuracy of AF predictions degrades with increasing protein length and complexity, de novo generation of long, complex proteins remains challenging. Ambient Protein Diffusion overcomes this problem by treating low-confidence AF structures as corrupted data. Rather than simply filtering out low-quality AF structures, our method adjusts the diffusion objective for each structure based on its corruption level, allowing the model to learn from both high and low quality structures. Empirically, ambient protein diffusion yields major improvements: on proteins with 700 residues, diversity increases from 45% to 85% from the previous state-of-the-art, and designability improves from 70% to 88%.

1 Introduction

Proteins are the fundamental building block of life. They accelerate chemical reactions by many orders of magnitude, convert sunlight into food, and underpin the myriads of processes within cells and organisms with the level of accuracy and precision required to sustain life [6, 25]. Unlike computational protein engineering—which focuses on improving the developability or function of existing proteins through computationally guided mutations for practical biotechnological applications [17, 31, 20, 32, 8, 29, 16]—*de novo* protein design aims to create entirely new proteins with specified structures and functions, ultimately seeking to discover folds and activities not found in nature [10]. Since protein function is largely determined by tertiary and quaternary structure, generative machine learning frameworks for protein design focus on learning the sparse, evolutionarily sampled landscape of protein structures, with the goal of generating novel, functional backbone scaffolds beyond those observed in nature [40, 26, 27, 23, 19, 42, 18, 7, 44, 37].

Recent breakthroughs in machine learning–based structure prediction—most notably AlphaFold2 [24]—have made it possible to infer accurate protein structures directly from sequence [24, 9, 28]. This progress has enabled the creation of large-scale structural resources such as the AlphaFold Database (AFDB), which contains over 214M predicted structures from UniProtKB sequences [11, 36]. In parallel, high-throughput tools for sequence and structure comparison, such as MMSeqs2 and FoldSeek, have facilitated the curation of large, diverse training datasets from AFDB [5]. Among them, the 2.3M AFDB cluster dataset, has already been shown to improve the capabilities of generative models for protein structure design [27, 19].

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

The quality of a generative model depends on the size and fidelity of its training data. While AlphaFold2 (AF) has enabled large-scale protein structure prediction, its outputs often contain biological or computational inaccuracies [41]. To estimate the reliability of a predicted structure, AlphaFold provides a per-residue confidence score, the predicted Local Distance Difference Test (pLDDT), which is a proxy of local structural accuracy. In practice, researchers frequently filter predicted structures based on average pLDDT scores, training only on high-confidence subsets (typically using a cutoff of pLDDT > 80). Lower pLDDT scores are disproportionately associated with longer and more structurally complex proteins. As a result, filtering based on pLDDT introduces a bias toward smaller, simpler folds, reducing structural diversity in the training set and impairing the model's ability to generalize to more complex regions of structure space—including longer proteins. Notably, many low-pLDDT structures still contain well-folded domains that are misoriented with respect to each other, as reflected by low predicted alignment error (pAE). These structures can still offer valuable domain-level and coarse-grained information about the structure distribution, which is discarded by overly aggressive filtering.

To mitigate these issues, we depart from the standard paradigm of filtering low-confidence structures. Instead, we introduce *Ambient Protein Diffusion* —a framework for training diffusion models that incorporates proteins with noisy or incomplete structures directly into the training process. Ambient Protein Diffusion builds on recent advances in learning generative models from corrupted data [12, 14, 1, 13, 2, 33, 4, 39, 15, 30, 34], which have explored controlled corruption settings such as additive Gaussian noise [12, 14, 1, 15] and masking [13, 2]. Our framework generalizes these techniques to arbitrary, unknown corruption processes, enabling the training of generative models in scientific domains where the corruption mechanism is complex and non-parametric. In our setting, the AlphaFold prediction errors represent such a corruption: they are structured, not explicitly modeled, and vary across protein size and topology. Yet, our method effectively leverages these imperfect samples to significantly advance the capabilities of generative protein models. For example, on proteins with 700 residues, our 16.7M parameter model our method improves diversity from 45% to 85% and increases designability from 70% to 88% compared to the previous state-of-the-art, Proteína, a 200M parameter model. Below, we summarize our key contributions:

- We generalize recent approaches for training generative models on corrupted data to handle arbitrary, non-parametric, and unknown corruption processes, enabling their application to scientific domains.
- We construct a new training set from the AFDB cluster dataset optimized for geometric diversity, rather than evolutionary similarity, yielding a broader and more representative sampling of structural space for generative modeling.
- We demonstrate that Ambient Protein Diffusion effectively leverages low-pLDDT AlphaFold predictions, allowing the model to learn from all available samples without distorting the underlying structure distribution.
- We achieve state-of-the-art results in both diversity and designability for protein generation, improve diversity by 45% and designability by 18% on long proteins (up to 800 residues), and establish the Pareto frontier between these objectives on short proteins (< 256 residues).

2 Background and Related Work

De novo Protein Generation. Most *de novo* protein generation frameworks that operate in structure space follow a three-step pipeline: (1) a generative model samples a three-dimensional backbone structure; (2) an inverse folding model (e.g., ProteinMPNN) proposes amino acid sequences likely to fold into the generated backbone; and (3) these sequences are evaluated by a structure prediction model (e.g., ESMFold) to identify the ones that best recapitulate the target fold.

Pioneering methods such as RFDiffusion [40] and Chroma [23] have established strong baselines for backbone generation. More recent advances include Genie [26], which introduces a denoising diffusion model with an SE(3)-equivariant network that generates proteins as point clouds of reference frames; Genie2 [27], which scales Genie using synthetic AlphaFold structures to improve training data diversity; and Proteína [19], which replaces diffusion with flow matching and scales both model size and dataset scale by orders of magnitude to improve performance on longer and more complex monomeric proteins.

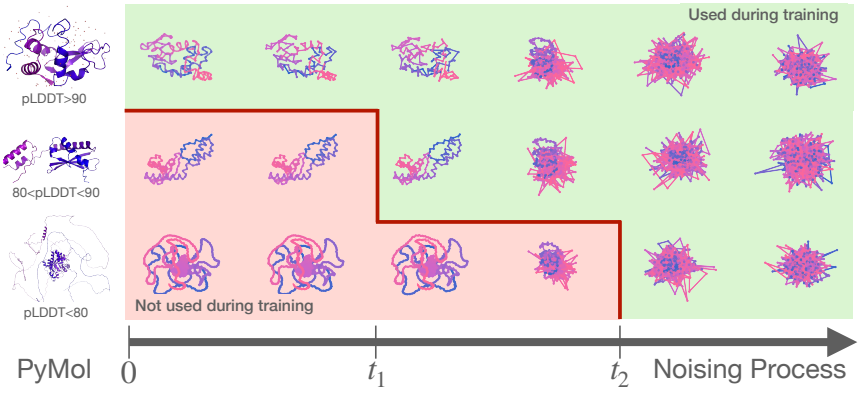

Figure 1: **Overview of Ambient Protein Diffusion on alphafold structures.** Rows 1-3 show the noising process (from left to the right) of three different alphafold proteins based on their average pLDDT (top: high, middle: medium, and bottom:low). These proteins are only used during training at the green diffusion times. At these noise levels, any initial AlphaFold prediction errors in low-pLDDT proteins have effectively been "erased" by the added noise, and the distributions of low- and high-pLDDT proteins have merged.

Ambient Protein Diffusion is built using the Genie architecture and makes use of ambient protein diffusion to achieve state-of-the-art results with substantially shorter training times, much fewer parameters (16.7M vs 200M), and significantly reduced computational requirements.

Training Datasets. Recent advances in structure prediction—most notably AlphaFold2 [24] and ESMFold [28]—have dramatically expanded the available structural data, enabling the prediction of ∼214M and ∼617M monomeric protein structures from UniProtKB (via the AlphaFold Protein Structure Database) [36] and metagenomic libraries (via the ESM Atlas) [28], respectively. While this explosion of computational structures presents unprecedented opportunities, it also poses significant challenges for downstream bioinformatic analysis and model training, particularly due to the scale, redundancy, and uneven quality of the predicted structures. To address this, prior work applied MMSeqs2 [22] and FoldSeek [35] to cluster the AlphaFold Database (AFDB), yielding ∼2.3M clusters designed to capture evolutionary relationships between predicted structures [5]. This AFDB cluster dataset has since served as the foundational dataset to train several generative protein structure models [19, 27].

In this work, we revisit the FoldSeek pipeline applied AFDB with a different objective: rather than optimizing for evolutionary insight, we recluster the AFDB cluster dataset with hyperparameters tuned to maximize geometric diversity. Our goal is to construct a dataset better suited for learning a generative model of structural space—one that emphasizes structural rather than evolutionary variation. Starting from the 2.3 million AFDB clusters, we use the cluster representatives with average pLDDT > 70 (∼1.29M structures) and apply our geometric clustering procedure. The resulting dataset comprises roughly ∼292K structurally diverse clusters.

Diffusion Models. The goal in diffusion modeling is to sample from an unknown density p_0 that we have sample access to. Formally, let $\mathcal{D} = \{x_0^i\}_{i=1}^N$ a dataset of N independent samples, where $X_0^i \sim p_0$. The unknown distribution p_0 is potentially complex, high-dimensional and multimodal. To make the sampling problem more tractable, in diffusion modeling we target smoothened densities p_t defined as the convolution with a Gaussian: $p_t = p_0 * \mathcal{N}(0, \sigma(t) I_d)$[1], where $\sigma(t)$ is an increasing function of t, with $\sigma(0) = 0$. In particular, the object of interest in diffusion modeling is the score-function of the smoothened densities, defined as $\nabla_{x_t} \log p_t(x_t)$. The latter is connected to the optimal denoiser (in the l_2 sense) through Tweedie's Formula: $\nabla \log p_t(x_t) = \frac{\mathbb{E}[X_0 | X_t = x_t] - x_t}{\sigma^2(t)}$.

[1]Alternative formulations of diffusion modeling, such as the Variance Preserving case, are equivalent to this case up to a simple reparametrization. For the ease of analysis, we focus our presentation on corruptions of the form $X_t = X_0 + \sigma_t Z, \quad Z \sim \mathcal{N}(0, I_d)$.

Given access to $\mathbb{E}[X_0|X_t = x_t]$ one can sample from the distribution p_0 of interest by running a reverse diffusion process [3]. Hence, the sampling problem becomes equivalent to the problem of approximating the set of functions $\{\mathbb{E}[X_0|X_t = \cdot]\}_{t=0}^{T}$. Given a sufficiently rich family of functions $\{h_\theta : \theta \in \Theta\}$, the conditional expectation at a particular time t can be learned by minimizing the objective:

$$J(\theta) = \mathbb{E}_{t\in\mathcal{U}[0,T]}\mathbb{E}_{x_0,x_t|t}\left[||h_\theta(x_t) - x_0||^2\right]. \tag{1}$$

Protein Diffusion. In protein diffusion models that target backbone generation, X_0 captures the 3-D co-ordinates for each one of the backbone residues of the protein. The length of a protein varies and the standard practice is to pad each protein to specified length (256 for Genie [26, 27]), with some special mask indicating the valid positions.

Learning from noisy data. Recent work has explored the problem of learning diffusion models from corrupted data. Typically, the corruption process is simple, e.g. it can be additive Gaussian noise as in [12, 14, 1], or masking as in [1, 2]. Even in works that the corruption process is more general, the degradation needs to be known and multiple diffusion trainings are required until an Expectation-Maximization algorithm converges [33, 4]. In this work, we deviate from this setting as the corruption process is unknown and complex, which may include AlphaFold learning and hallucination errors, and noise inherent to the structural biology technique used to solve the structure, etc. We also target a single diffusion training instead of performing multiple EM iterations. The method is detailed in Section 3.

Our work generalizes the techniques developed in [12, 14] for the additive Gaussian noise case. Particularly, in [14], the authors consider learning from a dataset $\mathcal{D} = \{(x^i_{t_i}, t_i)\}_{i=1}^N$ of samples noised with additive Gaussian noise of different variances $\{\sigma^2(t_i)\}_{i=1}^N$. Formally, let $X_{t_i} = X_0 + \sigma(t_i)Z$, where $X_0 \sim p_0, Z \sim \mathcal{N}(0, I_d)$. Each point X_{t_i} contributes to the learning only for $t \geq t_i$, using the objective:

$$\hat{J}(\theta) = \mathbb{E}_{t\in\mathcal{U}[0,T]} \sum_{x_{t_i}\in\mathcal{D}:\ t_i>t} \mathbb{E}_{x_t|x_{t_i},t_i}\left[||\alpha(t,t_i)h_\theta(x_t,t) + (1-\alpha(t,t_i))x_t - x_{t_i}||^2\right], \tag{2}$$

$\alpha(t,t_i) = \frac{\sigma^2(t)-\sigma^2(t_i)}{\sigma^2(t)}$. As the number of samples grows to infinity, Equation 2 also recovers the conditional expectation $\mathbb{E}[X_0|X_t = x_t]$, but it does so while being able to utilize noisy samples. This objective recovers the true minimizer because one can prove that the conditional expectation $\mathbb{E}[X_{t_i}|X_t = x_t]$, lies in the line that connects the current noisy point x_t and the prediction of the clean image, $\mathbb{E}[X_0|X_t = x_t]$.

3 Method

Formally, we are given access to samples from the AlphaFold distribution $\tilde{p}_0$ and aim to learn how to sample from the true distribution of experimentally solved structures, p_0, without an explicit degradation model mapping $p_0 \to \tilde{p}_0$. In the protein structure setting, it is not appropriate to model the structural deviations introduced by AlphaFold as additive Gaussian noise. Our key insight is that, regardless of how $\tilde{p}_0$ deviates from p_0, adding noise to both distributions causes them to contract toward one another. As the noise level increases, the distributions $\tilde{p}_t$ and p_t become progressively more aligned. This is because it is known that Gaussian noise contracts distribution distances (KL divergence) in the following sense:

$$\mathrm{D_{KL}}(p_t||\tilde{p}_t) \leq \mathrm{D_{KL}}(p_{t'} \,||\, \tilde{p}_{t'}), \quad \forall t \geq t'. \tag{3}$$

In fact, as $t \to \infty$, we have that: $\mathrm{D_{KL}}(p_t \,||\, \tilde{p}_t) \to 0$, as both distributions converge to the same Gaussian. We now define the concept of merging of two distributions towards the same measure.

Definition 3.1 (ϵ-merged) *We say that two distributions, p and $\tilde{p}$ are ϵ-merged, if the KL distance between the two is upper-bounded, by ϵ, i.e., if $\mathrm{D_{KL}}(p \,||\, \tilde{p}) \leq \epsilon$.*

Similarly, we define the merging time of two distributions as the minimal amount of noise we need to add such that the two distributions become ϵ-merged. Formally,

Definition 3.2 (ϵ-merging time) *Let two distributions $p, \tilde{p}$. We define their ϵ-merging time as follows:* $t_n(p, \tilde{p}, \epsilon) = \inf\{t : \mathrm{D_{KL}}(p * \mathcal{N}(0, \sigma(t)^2 I) \,||\, \tilde{p} * \mathcal{N}(0, \sigma(t)^2 I) \leq \epsilon\}$.

Assuming we can estimate the ϵ-merging time between two distributions p and $\tilde{p}$, our key idea is to treat samples from $\tilde{p}_t$ as approximate samples from p_t for all timesteps $t > t_n(p, \tilde{p}, \epsilon)$. This idea is illustrated in Figure 1. The intuition is that once the distributions have sufficiently merged under noise, the residual shift becomes negligible and samples from $\tilde{p}_t$ can be used for learning p_t. This holds because: (i) the learning algorithm may not be sensitive to small distributional discrepancies at high noise levels, and (ii) even if some bias is introduced, the remaining diffusion trajectory for times $t \leq t_n(p, \tilde{p}, \epsilon)$ is robust to small initial distributional mismatch due to its inherent stochasticity.

Sample dependent noise levels. At a high level, our objective is to determine the ϵ-merging time between the distribution of AlphaFold-predicted structures and that of experimentally resolved proteins. A key challenge arises from the fact that the AlphaFold distribution is highly heterogeneous in structural fidelity—that is, the accuracy with which AlphaFold predicts the true protein structure varies widely across samples. It is well established that short, structurally simple proteins are predicted with higher confidence, while longer and more complex proteins tend to yield lower-confidence predictions. This trend is illustrated in Figure 2B (Left). If we were to assign a single noise level across the entire AlphaFold dataset, we would need to select a relatively high noise level to accommodate the lowest-confidence predictions, particularly from long proteins. This would unnecessarily degrade the training signal for high-confidence structures—regardless of protein length—and limit the model's ability to learn from clean supervision. To address this, we treat the AlphaFold dataset as a mixture of K sub-distributions, $q_1, q_2, \ldots, q_K$, each representing a distinct confidence regime. We then assign each sub-distribution an appropriate noise level, sufficient to bring it ϵ-close to the distribution of high-confidence structures under the same noise schedule. This formulation allows the model to effectively learn from high-confidence AlphaFold predictions and incorporate low-confidence structures in a controlled manner, mitigating the degradation typically caused by noisy training data.

A natural way to decompose the AlphaFold distribution into a mixture of quality-specific sub-distributions is to leverage AlphaFold's self-reported confidence metric—the average predicted Local Distance Difference Test (pLDDT) score—as a proxy for predicted structural fidelity. In particular, given a dataset $\mathcal{D} = \{(x_0^{(i)}, \mathrm{pLDDT}^{(i)})\}_{i=1}^N$, we consider K distributions (where K is a hyperparameter to be chosen) with empirical observations for the j-th distribution being all the samples $\{(x_0^{(i)}, \mathrm{pLDDT}^{(i)}) : c_{\min}^{(j)} \leq \mathrm{pLDDT}^{(i)} \leq c_{\max}^{(j)}\}$, for some hyperparameters $c_{\min}^{(j)}, c_{\max}^{(j)}$.

Choice of sub-distribution boundaries. In this work, we adopt a deliberately simple and conservative strategy by partitioning the AlphaFold dataset into three discrete quality regimes based on the average pLDDT score: high-quality proteins (pLDDT > 90), medium-quality proteins (pLDDT in $[80, 90]$) and low-quality proteins (pLDDT in $[70, 80]$). We acknowledge that this discretization is coarse and that more principled alternatives may yield further improvements—for instance, by optimizing the bin boundaries or learning a continuous mapping from pLDDT to diffusion time. Despite the simplicity of our choices, our experimental results demonstrate that even a naive quality-aware decomposition can lead to important gains in performance across both short and long proteins. There are two sources of benefit over filtering methdos: 1) low-quality data (previously discarded) give diversity and 2) the distinction we do between medium-quality and high-quality data increases designability.

Ambient Protein Diffusion Algorithm. Our algorithm takes as input a dataset of protein structures together with their average pLDDT score, $\mathcal{D} = \{(x_0^{(i)}, \mathrm{pLDDT}^{(i)})\}_{i=1}^N$, a diffusion schedule, $\sigma(t)$, and a mapping function $f : [0, 100] \mapsto \mathbb{R}^+$ that translates the average pLDDT value of a protein to its estimated ϵ-merging time.

Annotation stage. The first step of the algorithm replaces each protein in the dataset with a noisy version of itself, where the noise level is determined by mapping function f. After this transformation, each protein can be treated as a sample from the target distribution convolved with a Gaussian at its assigned noise level. Importantly, this corruption step is only performed once during dataset preprocessing.

Loss function. After the annotation stage, we need to solve a training problem where we have data corrupted at different noise levels with additive Gaussian noise, as in [12, 14]. Hence, we can use the objective of Equation 2.

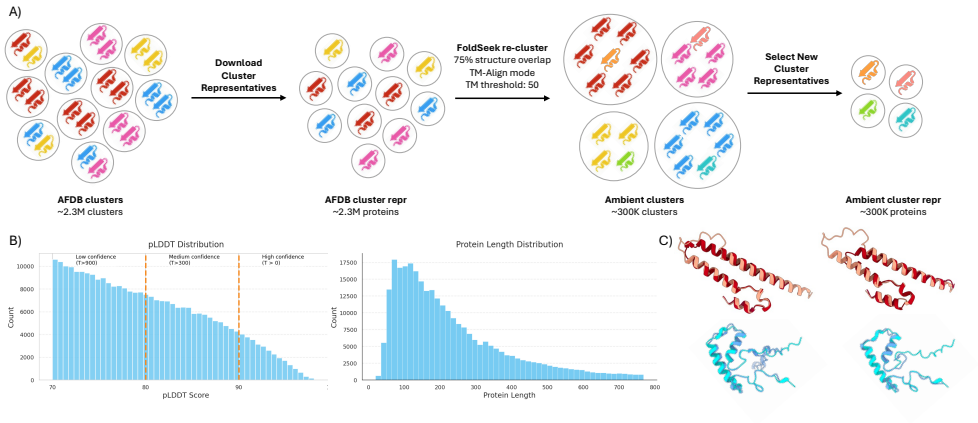

Figure 2: **Reclustering the AFDB cluster dataset for generative protein modeling.** (A) Starting from the 2.3M clusters in the AFDB, we cluster the representatives with FoldSeek with alignment-type set to TM-Align, TM threshold set to 0.5, and coverage set to 0.75. This results in 300K cluster (pLDDT > 70) from which we keep the representatives for training. (B) pLDDT and protein length statistics for our new training set. (C) Example of two protein clusters where two cluster members (red and blue) are superimposed with their cluster representative (beige and cyan).

Instead of directly applying the loss, we first need to rescale each time t to account for the vanishing gradient effect that is due to the multiplicative factor $a(t)$. Specifically, we need to rescale the loss at time t with: $w(t) = \frac{1}{a^2(t)} = \frac{\sigma^4(t)}{(\sigma^2(t) - \sigma^2(t_i))^2}$. We underline that this rescaling was not mentioned in the original paper of Daras et al. [14, 12], for training with noisy data. Yet, we find this rescaling critical for the success of our method. We hypothesize that the authors of [12, 14] did not encounter this issue because there were at most two noise levels considered, while in AF predicted protein structures there is a whole spectrum of assigned noise levels based on the predicted quality (measured by average pLDDT) of a protein structure. We provide further details about the loss implementation in the supplement and pseudocode for our Algorithm in the Appendix.

Uniform Protein Sampling in terms of diffusion times. To perform a training update for a diffusion model, we typically sample a point from the training distribution and then we uniformly sample the noise level t. However, since in our case we are dealing with noisy data, not all times t are allowed for a given protein, i.e. a protein with $\text{pLDDT}^{(i)}$ is only used for times $t \geq f(\text{pLDDT}^{(i)})$. To avoid spending most of the training updates on very noisy proteins, we opt for sampling first the diffusion time and then select from the eligible proteins that can be used in that diffusion time. This strategy ensures balanced coverage across the diffusion trajectory—from low to high noise—while still leveraging the diversity of low-confidence structures (pLDDT < 80) in our training dataset.

Reclustering AFDB clusters for generative modeling applications. The AFDB cluster dataset [5] has been used to train several generative protein models [27, 19]. However, the original intent behind the clustering was to study structure evolution across AFDB. Thus, the hyperparameters were chosen to obtain clusters of homologous structures, and the authors report that 97.4% of pairwise comparisons within clusters are conserved at the H-group (Homology) level of the ECOD hierarchical domain classification (median TM-score 0.71). While these FoldSeek hyperparameters are well-suited for evolutionary analysis of AFDB, we found that the AFDB cluster dataset has a significant degree of structural duplication and near-duplication between clusters that are more distantly evolutionarily related (see Figure 2C). This structural redundancy leads to an imbalanced training set, where structural motifs from the larger protein superfamilies are overrepresented.

Given this finding, we hypothesized that the datasets for generative modeling of protein structures—particularly for backbone-based models— benefit more from clusters defined purely by geometric similarity. To address this, we constructed a new clustering dataset derived from the AFDB cluster representatives, with an exclusive focus on structural topology. Specifically, these are the changes we made to the FoldSeek hyperparameters: switched the alignment-type from 3Di+AA to TM-Align to improve fidelity, used a TM-score threshold of 0.5, and relaxed the alignment coverage

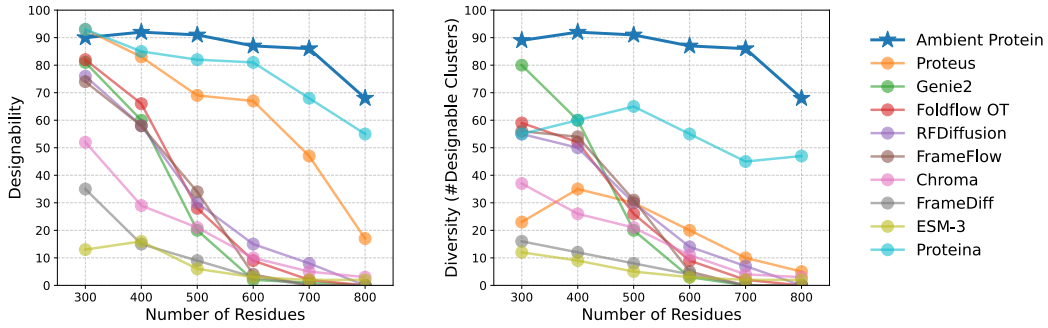

Figure 3: **Long protein generation performance.** We fine-tune Ambient Protein Diffusion on proteins up to 768 residues and sample sequences ranging from 300 to 800 residues. Ambient Protein Diffusion generates diverse and designable structures across this length range with consistent performance.

from 0.9 to 0.75. We relaxed the alignment coverage to improve clustering of AlphaFolded proteins with extended, unfolded N- or C-terminal regions (i.e., noodle tails) (Figure 2C). This approach produced a more balanced dataset that samples structural folds more uniformly, independent of their evolutionary relationships. Ablations showing the contribution of this reclustering versus our ambient training approach are given in Figure 4.

4 Experimental Results

We build on the Genie2 codebase [27]. Our model architecture follows the Genie2 architecture except that it is scaled larger, using 8 triangle layers as opposed to 5. We follow Genie2's schedule for inference time sampling and diffuse for 1000 timesteps.

We train Ambient Protein Diffusion in 3 stages with increasingly longer proteins. In first stage, we train on proteins from 50 to 256 residues for 200 epochs on our ambient clusters dataset using the representatives ($\sim$ 196,000 proteins). Since we increased the batch size to 384 items, we adopted a learning rate schedule to improve convergence [21]. We train with the AdamW optimizer with a maximal learning rate of 1.0×10^{-4}. During the second and third stage, we include additional cluster representatives of at most 512 and 712 residues, which scales our dataset to $\sim$269,000 and $\sim$291,000 proteins respectively. Training is performed on 48 GH200 GPUs and runs in 18, 48, and 48 hours for each stage respectively. More hyperparameters and details about our evaluation metrics can be found in the supplement. We underline that the computational cost of training our model, while significant, is still relatively low compared to the Proteína's estimated 14 days training on 128 A100 GPUs. This is due to the decreased size of our model ($<$ 17M vs 200M) and training set ($\sim$ 290K vs $\sim$ 780K). We further note that our goal is to develop models that perform well across a range of tasks, including long-protein generation, motif scaffolding, and more. To this end, we train only two models for the purposes of this paper: one model optimized for long-protein generation (Figure 3) and another optimized for short-protein generation (Figure 4).

Comparisons on unconditional generation of longer proteins.

In Figure 3, we compare Ambient Protein Diffusion performance on generating backbone for proteins with length ranging from 300 to 800 residues. To directly compare with Proteína on long-protein generation, we adopt its three-stage training and evaluation protocol. During training, the maximum sequence length is capped at 768 residues. For evaluation, we sample 100 protein backbones at each target length and evaluate them using the designability and diversity metrics. Since Ambient Protein Diffusion builds on Genie2, we use the same sampling procedure—running 1000 diffusion steps with a noise scale of $\gamma = 0.6$.

Ambient Protein Diffusion achieves designability and diversity scores exceeding 90% for proteins between 300 and 500 residues, and maintains scores above 85% for lengths up to 700 residues. For 800-residue proteins, both metrics decline to 68%. Compared to Proteína, Ambient Protein Diffusion outperforms by 26% in designability and 91% in diversity at length 700, and by 25% and 44%, respectively, at length 800. At every protein length, Ambient Protein Diffusion's diversity is equal to

its designability, indicating that every designable protein is unique. This is not the case for Proteína, where diversity scores consistently fall below designability, regardless of protein length.

Taken together, these results demonstrate the impact of ambient diffusion on backbone-based generative models and highlight the strength of Genie2's equivariant architecture. Our 17M parameter model trained on approximately 290K AlphaFold structures significantly outperforms a 200M-parameter transformer model trained on roughly 780K proteins. Our results show that smaller, more efficient models can surpass larger transformer baselines in both structural diversity and designability.

Comparisons on unconditional generation of shorter proteins. In this experiment, we evaluate the model on the unconditional generation of shorter proteins in Figure 4. The Ambient Protein Diffusion model used in this experiment was trained on a dataset filtered with a TM-Align threshold of 0.4 (as opposed to 0.5), resulting in a training set of approximately 90K cluster representative proteins.

Following the Genie2 protocol, we generate 5 structures for each sequence length from 50 to 256 residues, yielding a total of 1,035 structures. The generated structures are evaluated for both designability and diversity. In line with prior work, we sweep the noise scale γ to explore the tradeoff between designability and diversity. Ambient Protein Diffusion outperforms previous methods on both metrics, establishing a new Pareto frontier that achieves superior performance compared to all existing models, including Proteina. While it is well known that protein pairs with TM-scores above 0.5 typically share the same fold, and those below 0.5 generally do not, we find that the trade-off between designability and diversity is sensitive to the underlying structural heterogeneity of the dataset. Notably, clustering with a TM-align threshold of 0.4, which corresponds to less than a 1% chance of shared global topology, slightly outperforms the 0.5 threshold, which reflects a $\sim$38% probability of topological similarity [43].

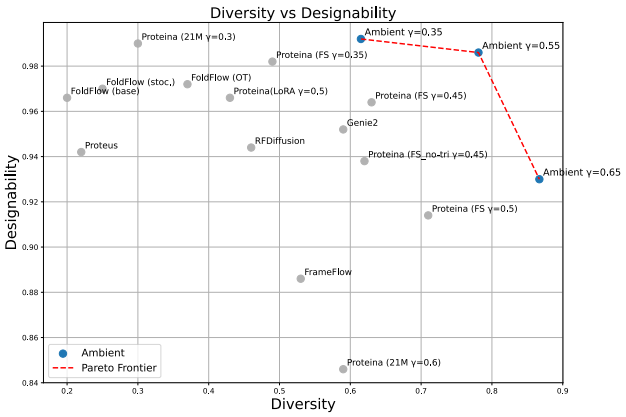

Figure 4: **Designability - diversity trade-off for short protein generation** (up to 256 residues). Ambient dominates completely the Pareto frontier between designability and diversity, while using a 12.88$\times$ smaller model. We further do so without using any higher-order sampler or (auto-) guidance method.

Ablating the Significance of Ambient Diffusion. We validate the effectiveness of Ambient Diffusion in Figure 5. For our ablation, we perform the identical three stages of training on a model variant that performs standard diffusion training as opposed to Ambient Diffusion. The Genie2 baseline is taken from the Proteína paper. We find that while our two models perform similarly on proteins of 300 residues, the designability and diversity of a vanilla diffusion model diminishes much more significantly with longer proteins. For proteins with 800 residues, the number of designable clusters drops from 68% to 25%.

Comparisons on Motif Scaffolding. We additionally compare our method to prior work on motif scaffolding in Figure 6, with full results provided in the supplement. Our evaluation follows the Genie2 benchmark, which comprises 24 single-motif and 6 multi-motif design tasks [27, 40]. For each task, we generate 1,000 scaffold samples using a noise scale of $\gamma = 0.45$. A design is considered successful if it (1) satisfies Genie2's motif designability criteria and (2) preserves the motif with an RMSD below 1Å. Among successful designs, a scaffold is counts as unique if its TM-score is at most 0.6 when compared to any other successful scaffold. A task is considered solved if at least one successful scaffold is generated.

With $\gamma = 0.45$, Ambient Protein Diffusion generates 1,923 unique successful scaffolds for single-motif tasks, a significant improvement over Genie2's 1,445 [27] and performs comparably to Proteína's 2,094 [19]. Notably, all methods solve a similar number of motifs – RFDiffusion solves 22 of

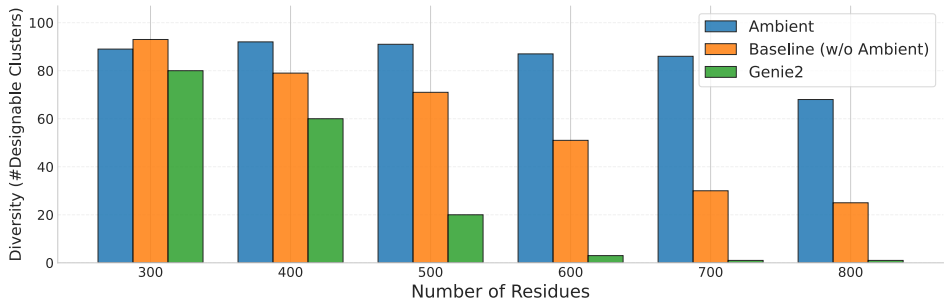

Figure 5: **Effect of ambient diffusion on long protein generation.** We sample protein sequences ranging from 300 to 800 residues. The baseline model (without Ambient Diffusion) shares the same architecture and training dataset, differing only in the diffusion loss and sampling procedure. At a sequence length of 300, the baseline yields four additional designable clusters. However, Ambient Protein Diffusion consistently outperforms both the baseline and Genie2, with increasingly significant improvements as sequence length grows.

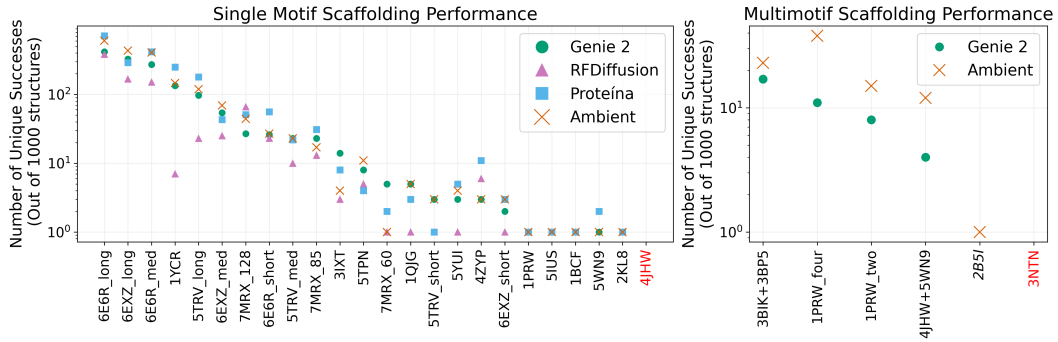

Figure 6: **Performance on Motif Scaffolding Tasks.** We compare Ambient Protein Diffusion to state-of-the-art models for motif scaffolding. The graphs show the number of unique successful scaffolds generated for each single- and multi-motif task. No model produced successful scaffolds for 4JHW and 3NTN. Only Ambient Protein Diffusion produced a valid solution for multi-motif scaffolding of *2B5I*.

the 24 tasks, while Ambient Protein Diffusion, Genie2, and Proteína each solve the same 23 tasks. For multi-motif scaffolding, Ambient Protein Diffusion generates 89 unique successful structures across 5 of the 6 benchmark problems, outperforming Genie2, which produces 40 and solves 4. Ambient Protein Diffusion performs particularly well on the 1PRW_four motif (38 vs. 11 successful structures) in which a scaffold is generated surrounding a calcium binding motif [38]. Overall, Ambient Protein Diffusion outperforms existing methods such as Genie2 and RFDiffusion on single-motif tasks and matches the performance of a Proteína model optimized specifically for motif-scaffolding.

5 Conclusion

We introduced *Ambient Protein Diffusion*, a diffusion-based model for protein structure generation that leverages low-confidence AlphaFold structures as a source of noisy training data. Ambient Protein Diffusion enables the generation of long protein structures with unprecedented levels of designability and diversity. Diversity increases as it can use low-confidence Alphafold structures that are typically discarded and designability increases as we separate the pristine quality proteins structures from the medium quality AlphaFold predictions. Ambient Protein Diffusion represents a foundational step toward robust de novo protein design at more natural, biologically relevant lengths. Despite progress, Ambient Protein Diffusion still favors generating alpha-helical structures and developing techniques that address this bias is a crucial direction for future work.

References

[1] Asad Aali, Marius Arvinte, Sidharth Kumar, and Jonathan I Tamir. Solving inverse problems with score-based generative priors learned from noisy data. *arXiv preprint arXiv:2305.01166*, 2023.

[2] Asad Aali, Giannis Daras, Brett Levac, Sidharth Kumar, Alex Dimakis, and Jon Tamir. Ambient diffusion posterior sampling: Solving inverse problems with diffusion models trained on corrupted data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=qeXcMutEZY.

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

[4] Weimin Bai, Yifei Wang, Wenzheng Chen, and He Sun. An expectation-maximization algorithm for training clean diffusion models from corrupted observations. *arXiv preprint arXiv:2407.01014*, 2024.

[5] Inigo Barrio-Hernandez, Jingi Yeo, Jürgen Jänes, Milot Mirdita, Cameron LM Gilchrist, Tanita Wein, Mihaly Varadi, Sameer Velankar, Pedro Beltrao, and Martin Steinegger. Clustering predicted structures at the scale of the known protein universe. *Nature*, 622(7983):637–645, 2023.

[6] Jeremy M Berg, John L Tymoczko, and Lubert Stryer. *Biochemistry (loose-leaf)*. Macmillan, 2007.

[7] 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. *arXiv preprint arXiv:2310.02391*, 2023.

[8] Jose M Carceller, Bhumika Jayee, Claire G Page, Daniel G Oblinsky, Gustavo Mondragón-Solórzano, Nithin Chintala, Jingzhe Cao, Zayed Alassad, Zheyu Zhang, Nathaniel White, et al. Engineering a photoenzyme to use red light. *Chem*, 11(2), 2025.

[9] Xinshi Chen, Yuxuan Zhang, Chan Lu, Wenzhi Ma, Jiaqi Guan, Chengyue Gong, Jincai Yang, Hanyu Zhang, Ke Zhang, Shenghao Wu, Kuangqi Zhou, Yanping Yang, Zhenyu Liu, Lan Wang, Bo Shi, Shaochen Shi, and Wenzhi Xiao. Protenix - advancing structure prediction through a comprehensive alphafold3 reproduction. *bioRxiv*, 2025. doi: 10.1101/2025.01.08.631967.

[10] Alexander E Chu, Tianyu Lu, and Po-Ssu Huang. Sparks of function by de novo protein design. *Nature biotechnology*, 42(2):203–215, 2024.

[11] UniProt Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1): D204–D212, 2015.

[12] Giannis Daras, Yuval Dagan, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *arXiv preprint arXiv:2302.09057*, 2023.

[13] Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alex Dimakis, and Adam Klivans. Ambient diffusion: Learning clean distributions from corrupted data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=wBJBLy9kBY.

[14] Giannis Daras, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. *arXiv preprint arXiv:2404.10177*, 2024.

[15] Giannis Daras, Yeshwanth Cherapanamjeri, and Constantinos Costis Daskalakis. How much is a noisy image worth? data scaling laws for ambient diffusion. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=qZwtPEw2qN.

[16] Daniel J Diaz, Anastasiya V Kulikova, Andrew D Ellington, and Claus O Wilke. Using machine learning to predict the effects and consequences of mutations in proteins. *Current opinion in structural biology*, 78:102518, 2023.

[17] Daniel J Diaz, Chengyue Gong, Jeffrey Ouyang-Zhang, James M Loy, Jordan Wells, David Yang, Andrew D Ellington, Alexandros G Dimakis, and Adam R Klivans. Stability oracle: a structure-based graph-transformer framework for identifying stabilizing mutations. *Nature Communications*, 15(1):6170, 2024.

[18] Cong Fu, Keqiang Yan, Limei Wang, Wing Yee Au, Michael Curtis McThrow, Tao Komikado, Koji Maruhashi, Kanji Uchino, Xiaoning Qian, and Shuiwang Ji. A latent diffusion model for protein structure generation. In *Learning on Graphs Conference*, pages 29–1. PMLR, 2024.

[19] Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. *arXiv preprint arXiv:2503.00710*, 2025.

[20] Chengyue Gong, Adam Klivans, James Madigan Loy, Tianlong Chen, Daniel Jesus Diaz, et al. Evolution-inspired loss functions for protein representation learning. In *Forty-first International Conference on Machine Learning*, 2024.

[21] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[22] Maria Hauser, Martin Steinegger, and Johannes Söding. Mmseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics*, 32(9):1323–1330, 2016.

[23] 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, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

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

[25] Albert L Lehninger, David L Nelson, and Michael M Cox. *Lehninger principles of biochemistry*. Macmillan, 2005.

[26] Yeqing Lin and Mohammed AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *arXiv preprint arXiv:2301.12485*, 2023.

[27] 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. *arXiv preprint arXiv:2405.15489*, 2024.

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

[29] Yi Liu, Sophie G Bender, Damien Sorigue, Daniel J Diaz, Andrew D Ellington, Greg Mann, Simon Allmendinger, and Todd K Hyster. Asymmetric synthesis of α-chloroamides via photoenzymatic hydroalkylation of olefins. *Journal of the American Chemical Society*, 146(11): 7191–7197, 2024.

[30] Haoye Lu, Qifan Wu, and Yaoliang Yu. SFBD: A method for training diffusion models with noisy data. In *Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025. URL https://openreview.net/forum?id=6HN14zuHRb.

[31] Jeffrey Ouyang-Zhang, Daniel Diaz, Adam Klivans, and Philipp Krähenbühl. Predicting a protein's stability under a million mutations. *Advances in Neural Information Processing Systems*, 36:76229–76247, 2023.

[32] Jeffrey Ouyang-Zhang, Chengyue Gong, Yue Zhao, Philipp Krähenbühl, Adam R Klivans, and Daniel J Diaz. Distilling structural representations into protein sequence models. *bioRxiv*, pages 2024–11, 2024.

[33] François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. *arXiv preprint arXiv:2405.13712*, 2024.

[34] Kulin Shah, Alkis Kalavasis, Adam R. Klivans, and Giannis Daras. Does generation require memorization? creative diffusion models using ambient diffusion, 2025.

[35] Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02, 2022.

[36] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

[37] 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. *bioRxiv*, pages 2024–02, 2024.

[38] Jue Wang, Sidney Lisanza, David Juergens, Doug Tischer, Joseph L Watson, Karla M Castro, Robert Ragotte, Amijai Saragovi, Lukas F Milles, Minkyung Baek, et al. Scaffolding protein functional sites using deep learning. *Science*, 377(6604):387–394, 2022.

[39] Yifei Wang, Weimin Bai, Weijian Luo, Wenzheng Chen, and He Sun. Integrating amortized inference with diffusion models for learning clean distribution from corrupted images. *arXiv preprint arXiv:2407.11162*, 2024.

[40] 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, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[41] Carter J Wilson, Wing-Yiu Choy, and Mikko Karttunen. Alphafold2: a role for disordered protein/region prediction? *International Journal of Molecular Sciences*, 23(9):4591, 2022.

[42] Kevin E Wu, Kevin K Yang, Rianne van den Berg, Sarah Alamdari, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *Nature communications*, 15(1):1059, 2024.

[43] Jinrui Xu and Yang Zhang. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

[44] Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

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 validated in the experimental results

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: See conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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]

Justification:

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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: We provide thorough implementation details in the paper and supplement.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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]

Justification: Code and data will be released upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We provide thorough implementation details in the paper and supplement.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

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: [No]

Justification: Our evaluation follows the standard deterministic benchmarking setup commonly used in the protein design literature.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   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 detail the compute requirements in the implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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: We have reviewed the ethics guidelines and find our paper conforms.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: See conclusion section

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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: Paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: All existing models and datasets are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: New datasets, code, and models will be released upon publication.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **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: No human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLM are not part of the core method.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Ambient Proteins
Training Diffusion Models on Noisy Structures
Supplement

Anonymous Author(s)
Affiliation
Address
email

A Evaluation Metrics

Evaluation of a protein generative model is challenging and there have been a few metrics that have been proposed. In what follows, we explain standard metrics in the protein-generative modeling literature that we will use in our Experimental Results section. Our experiments report using Proteína's definitions of the metrics when possible.

Designability (also referred to as refoldability) assesses the structural plausibility of generated proteins. Given a generated backbone, ProteinMPNN [1] generates eight plausible amino acid sequences for that backbone. ESMFold then folds each sequence and the resulting eight structures are compared to the original backbone. The self-consistency RMSD (scRMSD) is defined as the smallest root mean squared deviation between the generated backbone and each of the eight refolded structures. A backbone is considered *designable* if scRMSD < 2 Å and designability is defined as the percentage of generated backbones that meet this criterion.

Diversity quantifies the structural variability among the generated proteins. Designable backbones are clustered using Foldseek with a TM-score threshold of 0.5. Diversity is then defined as:

$$\text{Diversity} = \frac{\text{Number of Designable Clusters}}{\text{Number of Designable Samples}}.$$

This metric reflects the proportion of structurally distinct (i.e., non-redundant) designable backbones among all designable samples.

B Secondary Structure conditioning

Previous work [2] has explored conditioning protein structure generation on CATH labels, a form of hierarchical classification derived from the orientation and spatial organization of protein secondary structures [5]. In this setting, every residue in a protein sequence is typically assigned the same CATH label. In contrast, we propose a more fine-grained approach. Rather than relying on the manually curated and coarse-grained CATH classification, we condition our model directly on secondary structure annotations at the residue level. Each residue is assigned a label corresponding to its local secondary structure (e.g., helix, strand, coil), allowing the model to leverage localized structural context during generation.

We train a variant of our model with partial conditioning, in which the model is conditioned on the secondary structure sequence, without introducing any additional modifications to the input or architecture. We show designable samples conditioned on the secondary structure extracted from real

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

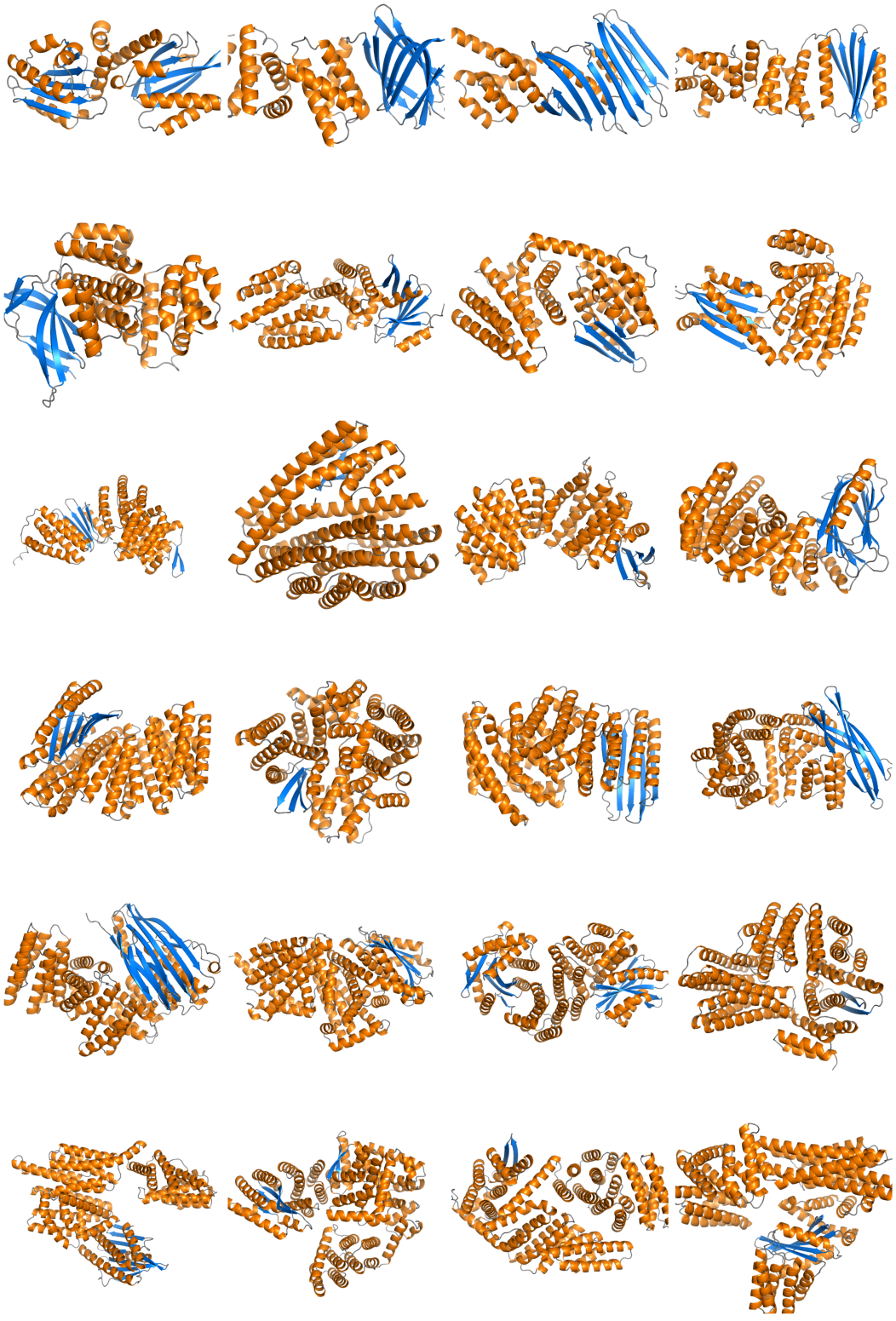

Figure 1: **Qualitative visualizations of unconditional generations.** Our model is capable of producing diverse, multi-domain long proteins.

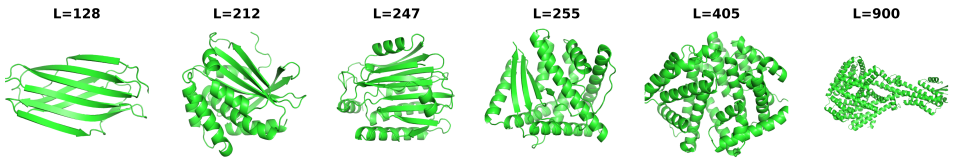

Figure 2: **Secondary Structure Conditioned Samples**. We generate proteins using a model variant trained with secondary structure conditioning. To guide generation, we extract secondary structure strings from existing proteins and use this coarse-grained structural representation as input. This conditioning enables the model to produce diverse and designable protein structures.

proteins in Figure 2. These results demonstrate that, even with coarse-grained secondary structure conditioning, our model can generate long, diverse proteins exhibiting a wide range of folds.

C Model and Training Hyperparameters

Table 1 includes a more thorough list of the hyperparameters used for our experiments.

D Full Motif Scaffolding Results

Table 2 and table 3 represent the numbers of unique successful scaffolds generated by Genie2 [3], RFDiffusion [6], Proteína [2] and *Ambient Protein Diffusion* for each motif in the benchmark dataset in Genie2.

E Full Training Algorithm and Implementation Details

E.1 Additional Implentation Details

Loss buffer. The loss rescaling introduced in the main paper ensures balanced weighting across noise levels. At the same time, it also introduces a potential instability: the loss explodes as $\sigma(t)$ approaches $\sigma(t_i)$. To mitigate this instability, we define a buffer zone around each protein's assigned noise level. Specifically, given a protein's assigned noise level t_i, it is only used during training at timesteps $t+\tau$, where τ is a buffer hyperparameter that controls the exclusion margin. This constraint prevents the model from encountering degenerate gradient behavior near the rescaling boundaries and is only applied to medium and low confidence structures (pLDDT < 90). We underline that is similar to how in normal diffusion there is a buffer time zone around $t = 0$ that is never sampled.

Ambient in high-noise regime. As explained in the main paper, each protein is only used for a subset of diffusion times according to its average pLDDT value. The proteins that have super high PLDDT (> 90) are considered clean data and can be used with the normal training objective. However, as found in [4], using the Ambient training objective for high-noise might be useful even if clean data is available. Intuitively, this objective prevents memorization and promotes diversity in the outputs. We ablated this design choice, and we found a slight increase in diversity for the same designability by using this. Hence, we used this tool from [4] for all our Ambient Protein Diffusion trainings.

E.2 Algorithm

We provide the full algorithm in Algorithm 1. We commit to open-sourcing our code and models to facilitate the broader adoption of our method from the community.

Hyperparameter	Genie2	*Ambient* (Stage 1)	Stage 2	Stage 3
Diffusion				
Number of timesteps	1,000	-	-	-
Noise schedule	Cosine	-	-	-
Ambient walls	-	(600,900)	(600,900)	(600,900)
Model Architecture				
Single feature dimension	384	-	-	-
Pair feature dimension	128	-	-	-
Pair transform layers	5	8	8	8
Triangle dropout	0.25	-	-	-
Structure layers	8	-	-	-
Training				
Optimizer	AdamW	-	-	-
Number of training proteins	586k	196k	269k	291k
Number epochs	40	200	50	20
Warmup iterations	10,000	1,000	500	100
Total batch size	384	384	96	48
Learning rate	1.0×10^{-4}	1.0×10^{-4}	1.0×10^{-5}	1.0×10^{-5}
Weight decay	0.05	-	-	-
Minimum protein length	20	20	50	50
Maximum protein length	256	256	512	768
Minimum mean pLDDT	80	70	70	70
Compute Resources				
Number of GPUs	48	48	48	48
Training time	18 hr	18hr	48hr	48hr

Table 1: **Hyperparameters of the diffusion protein model.** Dashes (-) indicate that the value is the same as the previous column. The Ambient walls correspond to the assigned diffusion times based on the protein's PLDDT (times are from 1 to 1000). Proteins with PLDDT > 90 are used everywhere. Proteins with PLDDT > 80 are used for times in $[600, 1000]$ and proteins with PLDDT > 70 are used for times in $[900, 1000]$. We underline that these hyperparameters were not particularly optimized, and even more benefits might be observed by properly tuning these values.

Motif Name	Genie 2	RFDiffusion	Proteína	*Ambient Protein Diffusion*
6E6R_long	415	381	**713**	601
6EXZ_long	326	167	290	**432**
6E6R_med	272	151	**417**	406
1YCR	134	7	**249**	146
5TRV_long	97	23	**179**	119
6EXZ_med	54	25	43	**69**
7MRX_128	27	66	**51**	44
6E6R_short	26	23	**56**	27
5TRV_med	**23**	10	22	**23**
7MRX_85	23	13	**31**	17
3IXT	**14**	3	8	4
5TPN	8	5	4	**11**
7MRX_60	**5**	1	2	1
1QJG	**5**	1	3	**5**
5TRV_short	**3**	1	1	**3**
5YUI	3	1	**5**	4
4ZYP	3	6	**11**	3
6EXZ_short	2	1	**3**	**3**
1PRW	**1**	**1**	**1**	**1**
5IUS	**1**	**1**	**1**	**1**
1BCF	**1**	**1**	1	**1**
5WN9	1	0	**2**	1
2KL8	**1**	**1**	1	1
4JHW	0	0	0	0
Total	1445	889	**2094**	1923

Table 2: **Detailed single motif scaffolding results.** Ambient Protein Diffusion achieves superior results to Genie 2 and RFDiffusion and performs on par with Proteina. Crucially, our model achieves these results zero-shot, i.e., unlike Proteina, it is not optimized for motif scaffolding and still achieves comparable performance while being an order of magnitude smaller.

Motif Name	Genie 2	*Ambient Protein Diffusion*
3BIK+3BP5	17	**23**
1PRW_four	11	**38**
1PRW_two	8	**15**
4JHW+5WN9	4	**12**
2B5I	0	**1**
3NTN	0	0
Total	40	**89**

Table 3: **Multi-motif scaffolding results.** Ambient Protein Diffusion achieves consistently superior results to the predecessor Genie-2 model, despite using the same architecture, i.e. the benefit comes from better use of the data. The motif 2B5I is only solved by Ambient Protein Diffusion.

Algorithm 1 Ambient Protein Diffusion: Training Algorithm.

Require: untrained network h_θ, dataset $\mathcal{D} = \{(x_0^{(i)}, \text{pLDDT}^{(i)})\}_{i=1}^N$, pLDDT to diffusion time mapping function $f : [0, 100] \mapsto \mathbb{R}^+$, noise scheduling $\sigma(t)$, batch size B, diffusion time T, buffer τ.

$\tilde{\mathcal{D}} \leftarrow \left\{ \left(x_0^{(i)} + f(\text{pLDDT}^{(i)})\epsilon^{(i)}, f(\text{pLDDT}^{(i)}) \right) | (x_0^{(i)}, \text{pLDDT}^{(i)}) \in \mathcal{D}, \epsilon^{(i)} \sim \mathcal{N}(0, I_d) \right\}$ ▷ Noise each point in the training set according to its pLDDT and get (noisy, noise level) pairs.

while not converged **do**

  $t_s^{(1)}, ..., t_s^{(B)} \leftarrow$ Sample uniformly B times in $[0, T]$ ▷ Sample diffusion times for this batch.

  $\tilde{\mathcal{D}}_p \leftarrow \text{shuffle}(\tilde{\mathcal{D}})$ ▷ Shuffle dataset.

  loss $\leftarrow 0$ ▷ Initialize loss.

  pos $\leftarrow 0$ ▷ Initialize index at shuffled dataset.

  for $i \in [1, B]$ **do**

    while True **do** ▷ find the first eligible point

      $y, t_y \leftarrow \tilde{\mathcal{D}}_p[\text{pos}]$

      if $t_y \geq t_s^i + \tau$ **then**

        break

      else

        pos $\leftarrow$ pos $+ 1$ ▷ Move to the next point in the dataset.

      end if

    end while

    $\epsilon \sim \mathcal{N}(0, I)$ ▷ Sample noise.

    $t \leftarrow t_s^{(i)}$ ▷ Time to be used in this training update.

    $t_i \leftarrow t_y$ ▷ Assigned time based on the PLDDT value

    $x_{t_i} \leftarrow y$ ▷ Noised point to the assigned time.

    $x_t \leftarrow x_{t_i} + \sqrt{\sigma^2(t) - \sigma^2(t_i)}\epsilon$ ▷ Add additional noise.

    $\alpha(t, t_i) \leftarrow \frac{\sigma^2(t) - \sigma^2(t_i)}{\sigma^2(t)}$.

    $w(t, t_i) \leftarrow \frac{\sigma^4(t)}{(\sigma^2(t) - \sigma^2(t_i))^2}$. ▷ Loss reweighting.

    loss $\leftarrow$ loss + $w(t, t_i) \left|\left| \alpha(t, t_i) h_\theta(x_t, t) + (1 - \alpha(t, t_i)) x_t - x_{t_i} \right|\right|^2$ ▷ Ambient loss

  end for

  loss $\leftarrow \frac{\text{loss}}{B}$ ▷ Compute average loss.

  $\theta \leftarrow \theta - \eta \nabla_\theta$loss ▷ Update network parameters via backpropagation.

end while

References

[1] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[2] Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. *arXiv preprint arXiv:2503.00710*, 2025.

[3] 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. *arXiv preprint arXiv:2405.15489*, 2024.

[4] Kulin Shah, Alkis Kalavasis, Adam R. Klivans, and Giannis Daras. Does generation require memorization? creative diffusion models using ambient diffusion, 2025.

[5] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla SM Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, et al. Cath: increased structural coverage of functional space. *Nucleic acids research*, 49(D1):D266–D273, 2021.

[6] 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, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.