Uncertainty-Aware Discrete Diffusion Improves Protein Design

Sazan Mahbub [1 2 *] **Christoph Feinauer** [1] **Caleb N. Ellington** [1] **Le Song** [1 3] **Eric P. Xing** [1 2 3]

Abstract

Protein inverse folding involves generating amino acid sequences that adopt a specified 3D structure—a key challenge in structural biology and molecular engineering. While discrete diffusion models have demonstrated strong performance, existing methods often apply uniform denoising across residues, overlooking position-specific uncertainty. We propose an uncertainty-aware discrete denoising diffusion model that employs a prior-posterior signaling mechanism to dynamically guide the denoising process. Our approach further integrates learned priors from a pretrained protein large language model and a structure encoder within a modular framework, jointly optimized through multi-objective training. Across multiple benchmarks, our method achieves substantial improvements over state-of-the-art baselines, offering a principled framework for structure-conditioned sequence generation in proteins and beyond.

1. Introduction

Designing protein sequences that fold into desired three-dimensional structures is a central challenge in computational biology, with broad applications in therapeutics, synthetic biology, and molecular engineering (Dauparas et al., 2022; Wang et al., 2024; Gao et al., 2022; Sun et al., 2024; Li et al., 2014; Hsu et al., 2022). This inverse folding problem—mapping from a fixed structural scaffold to a viable amino acid sequence—presents a fundamentally multimodal and ill-posed task, where small structural variations can permit diverse valid sequences (Dauparas et al., 2022). Recent advances in deep generative modeling have shown promise in addressing this challenge, particularly through autoregressive, masked, and diffusion-based frameworks (Sun et al., 2024; Dauparas et al., 2022; Zheng et al., 2023b; Wang et al., 2024).

Among these, discrete denoising diffusion models have gained traction due to their ability to generate high-quality and realistic molecules for biomolecular design tasks (Wang et al., 2024; Ellington et al., 2024; Zou et al., 2024; Sun et al., 2024). These models simulate a corruption-recovery process over sequence space, learning to denoise structure-conditioned sequences through repeated transitions (Wang et al., 2024; Sun et al., 2024; Austin et al., 2021). However, existing formulations typically apply uniform denoising updates across all sequence positions, overlooking the fact that sequence uncertainty—often due to structural constraints—vary widely across residues and time steps. This assumption of homogeneity in uncertainty can lead to premature or unreliable updates, especially in ambiguous regions.

In this work, we introduce a novel uncertainty-aware discrete denoising diffusion model for structure-conditioned protein sequence generation. We design a prior-posterior uncertainty signaling mechanism that enables our framework to dynamically decide where and when to denoise, focusing computational effort on positions where confident updates are possible and deferring those with high residual ambiguity. This formulation enables a probabilistic and interpretable denoising trajectory that adapts to both spatial and temporal uncertainty profiles in the sequence. We also propose an approach to parameterize this formulation with a set of learnable modules that can leverage the learned prior in pretrained large language models (LLMs) and structure encoders for proteins. Furthermore, we jointly train these modules through multi-objective optimization to further enhance inverse folding performance. Our proposed approach offers a promising way to partially reduce the overhead of hyperparameter search.

Through quantitative evaluation on three widely used benchmarks, we demonstrate that our framework significantly improves upon the current state-of-the-art in structure-conditioned protein design. Beyond protein design, our uncertainty-aware diffusion framework provides a general approach for structure-conditioned discrete sequence generation and holds potential for broader applications in RNA, DNA, and other domain-specific generative modeling tasks.

2. Method

In this section, we derive the probabilistic model underlying our inverse folding method, starting with the simplest for-

[*]Work done during internship at GenBio AI.
[1]GenBio AI [2]CMU [3]MBZUAI. Correspondence to: Le Song <le.song@genbio.ai>, Eric P. Xing <eric.xing@genbio.ai>.
ICML 2025 Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences, Vancouver, Canada.

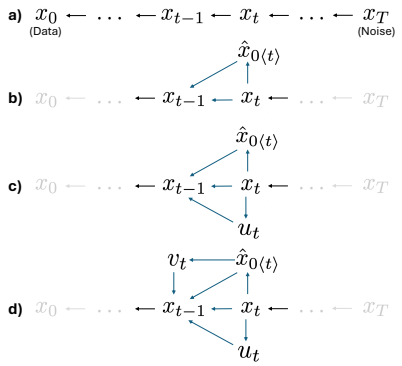

Figure 1. Probabilistic graphical model for the denoising process in discrete diffusion. See Section 2.1 for details.

mulation for the denoising steps in diffusion. The problem setup is detailed in Appendix B.1.

2.1. Probabilistic Model for Discrete Denoising Diffusion

We seek a transition probability $P(x_0|x_T)$, which allows sampling the discrete protein sequence x_0 given its noisy counterpart x_T. While single-step denoising has been widely studied in biological contexts (Sumi et al., 2024; Sevgen et al., 2023; Karimi et al., 2020), recent advances in iterative denoising via discrete diffusion language modeling (Ho et al., 2020; Austin et al., 2021) have introduced new directions for bio-sequence generation (Ellington et al., 2024; Sun et al., 2024; Zou et al., 2024; Wang et al., 2024). These models estimate the marginal $P(x_0|x_T)$ through intermediate variables x_t for $t \in [1, T-1]$ (Ho et al., 2020; Austin et al., 2021), with the joint probability expressed as Equation 1 and demonstrated in Figure 1a.

$$P(x_0, \ldots, x_{t-1}, x_t, \ldots, x_{T-1}|x_T) = \prod_{t\in[1,T]} P(x_{t-1}|x_t). \tag{1}$$

Here, $P(x_{t-1}|x_t)$ is the simplest form of a denoiser that assumes a fully Markovian reverse transition for a single reverse diffusion step from time t to $t-1$, where a discrete sequence $x_t \in \mathcal{A}^L$ is denoised into a cleaner version x_{t-1} (here $\mathcal{A}$ is a set of 20 standard amino-acids). As the noise depends on the time-step t, directly learning an effective transition function for $P(x_{t-1}|x_t)$ is challenging (Ho et al., 2020; Austin et al., 2021). To address this, recent works (Austin et al., 2021; Zheng et al., 2023a; Sahoo et al., 2024) introduce an intermediate variable to factorize the model. Following this strategy, we define $\hat{x}_{0\langle t\rangle}$ as a crude estimate of the clean data x_0 given the noisy input x_t (Figure 1b), leading to a joint probability,

$$P(x_{t-1}, \hat{x}_{0\langle t\rangle} \mid x_t) = P(\hat{x}_{0\langle t\rangle} \mid x_t) \cdot P(x_{t-1} \mid \hat{x}_{0\langle t\rangle}, x_t). \tag{2}$$

Such factorization provides us with two conditionals– $P(\hat{x}_{0\langle t\rangle} \mid x_t)$ and $P(x_{t-1} \mid \hat{x}_{0\langle t\rangle}, x_t)$. For the remaining discussion of this article, we call the former one as the denoising probability and the later as the refinement probability distribution, respectively. While it is reasonable to assume $x_{t-1} \perp\!\!\!\perp x_t \mid \hat{x}_{0\langle t\rangle}$ (leading to $P(x_{t-1} \mid \hat{x}_{0\langle t\rangle}, x_t) = P(x_{t-1} \mid \hat{x}_{0\langle t\rangle})$), this can subsequently become a quite strong assumption about the accuracy of the samples from the denoiser $\hat{x}_{0\langle t\rangle} \sim P(\hat{x}_{0\langle t\rangle} \mid x_t)$. We relax this assumption in our design. Now we get the transition probability by marginalizing over $\hat{x}_{0\langle t\rangle}$,

$$\begin{aligned} &P(x_{t-1} \mid x_t) \\ &= \sum_{\hat{x}_{0\langle t\rangle}} P(x_{t-1}, \hat{x}_{0\langle t\rangle} \mid x_t) \\ &= \sum_{\hat{x}_{0\langle t\rangle}} P(\hat{x}_{0\langle t\rangle} \mid x_t) \cdot P(x_{t-1} \mid \hat{x}_{0\langle t\rangle}, x_t) \\ &= \mathbb{E}_{\hat{x}_{0\langle t\rangle} \sim P(\hat{x}_{0\langle t\rangle}|x_t)} P(x_{t-1} \mid \hat{x}_{0\langle t\rangle}, x_t). \end{aligned} \tag{3}$$

2.2. Uncertainty-Aware Guidance

In this section, we introduce uncertainty estimation on sequence data as a way to guide discrete diffusion. While various uncertainty estimation approaches could be integrated into our framework (Liu et al., 2020; Gawlikowski et al., 2023; Kristiadi et al., 2021; Hie et al., 2020), we follow the supervised strategy of Liu et al. (2024), motivated by its success in natural language generation. Exploring alternative techniques—including non-learnable and unsupervised methods—is left for future work.

Prior Uncertainty. We start by defining a random variable $u_t \in \{0,1\}^N$, where $u_t^i = 1$ indicates the i-th residue is noisy, and $u_t^i = 0$ indicates it matches the native sequence (Figure 1c). Considering $u_t \perp\!\!\!\perp \{\hat{x}_{0\langle t\rangle}, x_{t-1}\} \mid x_t$, we can rewrite the joint probability in Equation 2 as,

$$\begin{aligned} P(x_{t-1}, \hat{x}_{0\langle t\rangle}, u_t \mid x_t) &= P(\hat{x}_{0\langle t\rangle} \mid x_t) \\ &\cdot P(u_t \mid x_t) \cdot P(x_{t-1} \mid \hat{x}_{0\langle t\rangle}, u_t, x_t). \end{aligned} \tag{4}$$

Considering u_t factorizes over residues,

$$\begin{aligned} P(x_{t-1}, \hat{x}_{0\langle t\rangle}, u_t \mid x_t) &= P(\hat{x}_{0\langle t\rangle} \mid x_t) \\ &\cdot \prod_{i\in[1,N]} P(u_t^i \mid x_t) \cdot P(x_{t-1}^i \mid \hat{x}_{0\langle t\rangle}, u_t^i, x_t). \end{aligned} \tag{5}$$

Since u_t^i follows a Bernoulli distribution, we have $\mathbb{E}[u_t^i|x_t] = P(u_t^i = 1 \mid x_t)$, which we refer to as the *prior uncertainty estimate* and model it as a learnable function.

Posterior Uncertainty. We assume imperfect denoisers, sampling from which can often lead to erroneous discrete jumps. Before taking any denoising step we want to estimate how much point-wise uncertainty would change if we updated the i-th residue with $\hat{x}_{0\langle t\rangle}^i$. To model this, we introduce another latent variable $v_t \in \{0,1\}^N$ representing the estimated per-residue correctness of $\hat{x}_{0\langle t\rangle}$, with $v_t \perp\!\!\!\perp \{x_t, x_{t-1}, u_t\} \mid \hat{x}_{0\langle t\rangle}$ (Figure 1d). The joint probability in Equation 5 then becomes,

$$P(x_{t-1}, \hat{x}_{0\langle t\rangle}, u_t, v_t \mid x_t) = P(\hat{x}_{0\langle t\rangle} \mid x_t) \cdot \prod_{i\in[1,N]} P(u_t^i \mid x_t) \cdot P(v_t^i \mid \hat{x}_{0\langle t\rangle}) \cdot P(x_{t-1}^i \mid u_t^i, v_t^i, \hat{x}_{0\langle t\rangle}, x_t). \tag{6}$$

Here the expectation $\mathbb{E}[v_t^i|\hat{x}_{0\langle t\rangle}] = P(v_t^i = 1 \mid \hat{x}_{0\langle t\rangle})$ is our point-wise *posterior uncertainty* estimate.

Together, u_t^i and v_t^i form a *prior-posterior uncertainty signal* over the i-th residue— $\mathbb{E}[u_t^i \mid x_t]$ estimates existing uncertainty in x_t^i, while $\mathbb{E}[v_t^i \mid \hat{x}_{0\langle t\rangle}]$ estimates uncertainty after updating with $\hat{x}_{0\langle t\rangle}^i$. We can now marginalize over both variables to obtain the transition probability,

$$P(x_{t-1} \mid x_t) = \prod_{i\in[1,N]} P(x_{t-1}^i \mid x_t) = \mathbb{E}_{\hat{x}_{0\langle t\rangle}\sim P(\hat{x}_{0\langle t\rangle}|x_t)}\Big[\prod_{i\in[1,N]} \sum_{u_t^i, v_t^i\in\{0,1\}} P(u_t^i \mid x_t) \cdot P(v_t^i \mid \hat{x}_{0\langle t\rangle}) \cdot P(x_{t-1}^i \mid u_t^i, v_t^i, \hat{x}_{0\langle t\rangle}, x_t)\Big]. \tag{7}$$

2.3. Conditioning on 3D Structure

Since we aim to generate a protein sequence $x_{t-1} \in \mathcal{A}$ given its 3D conformation $\psi \in \mathbb{R}^{L\times N\times 3}$, we additionally condition the transition probability $P(x_{t-1} \mid x_t)$ on ψ. This modifies the Equation 7 as,

$$P(x_{t-1} \mid x_t, \psi) = \prod_{i\in[1,N]} P(x_{t-1}^i \mid x_t, \psi) = \mathbb{E}_{\hat{x}_{0\langle t\rangle},\psi\sim P(\hat{x}_{0\langle t\rangle}|x_t,\psi)}\Big[\prod_{i\in[1,N]} \sum_{u_t^i, v_t^i\in\{0,1\}} P(u_t^i \mid x_t, \psi) \cdot P(v_t^i \mid \hat{x}_{0\langle t\rangle}, \psi) \cdot P(x_{t-1}^i \mid u_t^i, v_t^i, \hat{x}_{0\langle t\rangle}, x_t)\Big]. \tag{8}$$

Now our prior and posterior uncertainty estimates become $\mathbb{E}[u_t^i \mid x_t, \psi] = P(u_t^i = 1 \mid x_t, \psi)$ and $\mathbb{E}[v_t^i \mid \hat{x}_{0\langle t\rangle}, \psi] = P(v_t^i = 1 \mid \hat{x}_{0\langle t\rangle}, \psi)$, respectively.

2.4. Parameterization and Optimization

In this section, we describe how we parameterize the transition probability in Equation 8. We discuss our full framework as comprising five modules: (1) a structure encoder $\mathcal{E}_{\theta_1}^{st}(\cdot)$, (2) a sequence encoder $\mathcal{E}_{\theta_2}^{seq}(\cdot)$, (3) a sequence decoder $\mathcal{D}_{\theta_3}^{seq}(\cdot)$, (4) an uncertainty estimator $\mathbb{U}_{\theta_1,\theta_2,\phi}(\cdot)$, and (5) a refinement module $\mathcal{R}(\cdot)$. All modules except $\mathcal{R}(\cdot)$ are learnable and parameterized by θ_1, θ_2, θ_3, and ϕ. We also use a single uncertainty estimator to model both prior and posterior uncertainty estimates.

Denoiser Parameterization. For the first three modules, we adopt AIDO.ProteinIF (Sun et al., 2024), a state-of-the-art discrete diffusion-based inverse folding method. AIDO.ProteinIF uses ProteinMPNN-CMLM (Zheng et al., 2023b) as $\mathcal{E}_{\theta_1}^{st}(\cdot)$, which also produces the initial sequence estimate x_T which is then iteratively refined, similar to other leading methods (Zheng et al., 2023b; Wang et al., 2024).

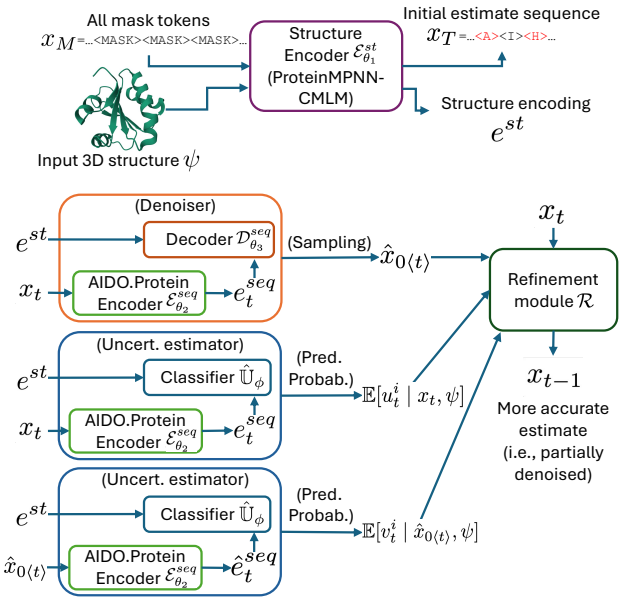

Figure 2. Overall architecture of our proposed method. Here, $(x_T, e^{st}) = \mathcal{E}_{\theta_1}^{st}(x_M, \psi)$, $e_t^{seq} = \mathcal{E}_{\theta_2}^{seq}(x_t)$, and $\hat{e}_t^{seq} = \mathcal{E}_{\theta_2}^{seq}(\hat{x}_{0\langle t\rangle})$. See Section 2.4 for details.

$\mathcal{E}_{\theta_2}^{seq}(\cdot)$ is the encoder of AIDO.Protein–a 16B parameter protein language model, while $\mathcal{D}_{\theta_3}^{seq}(\cdot)$ is a lightweight transformer (Vaswani, 2017) that fuses structure and sequence representations. At each time-step t, we obtain a crude estimate $\hat{x}_{0\langle t\rangle}$ from x_t as,

$$\hat{x}_{0\langle t\rangle} = \mathcal{D}_{\theta_3}^{seq}(\mathcal{E}_{\theta_2}^{seq}(x_t), \mathcal{E}_{\theta_1}^{st}(x_M, \psi)), \tag{9}$$

where $x_M \in \{\texttt{<MASK>}\}^L$ denotes a fully masked sequence with all residue labels unknown. We use x_M alongside the structure ψ as input to ProteinMPNN-CMLM, ensuring the structure encoding depends solely on ψ. Unlike x_t, where each x_t^i is a noisy but valid amino acid, x_M contains no residue information. Equation 9, representing the denoiser probability $P_{\theta_1,\theta_2,\theta_3}(\hat{x}_{0\langle t\rangle} \mid x_t, \psi)$, can be formulated as both deterministic or stochastic via sampling temperature (Sun et al., 2024; Wang et al., 2024). Unlike AIDO.ProteinIF, which uses time-dependent structure encoding $\mathcal{E}_{\theta_1}^{st}(x_t, \psi)$, we fix it as $\mathcal{E}_{\theta_1}^{st}(x_M, \psi)$, enabling caching and reducing computation. Empirically, this simplification does not affect performance.

Uncertainty Estimator Parameterization. To leverage the learned priors from the structure and sequence encoders, we parameterize the uncertainty estimator $\mathbb{U}_{\theta_1,\theta_2,\phi}(\cdot)$ with θ_1, θ_2, and ϕ. This allows direct prediction of the prior and posterior uncertainty for residue i as,

$$\mathbb{E}[u_t^i \mid x_t, \psi] = \mathbb{U}_{\theta_1,\theta_2,\phi}(x_t, x_T, \psi)^i, \quad \mathbb{E}[v_t^i \mid \hat{x}_{0\langle t\rangle}, \psi] = \mathbb{U}_{\theta_1,\theta_2,\phi}(\hat{x}_{0\langle t\rangle}, x_T, \psi)^i, \tag{10}$$

which can further be expanded as,

$$\mathbb{E}[u_t^i \mid x_t, \psi] = \hat{\mathbb{U}}_\phi(\mathcal{E}_{\theta_2}^{seq}(x_t), \mathcal{E}_{\theta_1}^{st}(x_M, \psi))^i \quad \mathbb{E}[v_t^i \mid \hat{x}_{0\langle t\rangle}, \psi] = \hat{\mathbb{U}}_\phi(\mathcal{E}_{\theta_2}^{seq}(\hat{x}_{0\langle t\rangle}), \mathcal{E}_{\theta_1}^{st}(x_M, \psi))^i, \tag{11}$$

where $\hat{\mathbb{U}}_\phi(\cdot) \in [0,1]$ is a soft binary classifier, parameterized by ϕ, that uses structure and sequence encodings to predict point-wise uncertainty probabilities.

Refinement Module. We design the refinement module $\mathcal{R}(\cdot)$ as a simple, non-learnable function based on the estimated point-wise uncertainties,

$$x_{t-1}^i = \mathcal{R}(x_t, \hat{x}_{0\langle t\rangle}, \psi)^i = \begin{cases} \hat{x}_{0\langle t\rangle}^i & \text{if } \mathbb{E}[u_t^i \mid x_t, \psi] > \mathbb{E}[v_t^i \mid \hat{x}_{0\langle t\rangle}, \psi] \\ x_t^i & \text{otherwise .} \end{cases} \quad (12)$$

Equation 12 updates residue i with $\hat{x}_{0\langle t\rangle}^i$ only if it reduces the estimated uncertainty. This design eliminates the need for a common hyperparameter—the number of tokens denoised per step—and makes our method less sensitive to the number of denoising steps, as uncertainty estimates serve as an implicit stopping criterion. As a result, our framework partially reduces the cost of hyperparameter tuning. In future work, we plan to explore learnable alternatives for $\mathcal{R}(\cdot)$.

Table 1. Quantitative evaluation on CATH 4.2 dataset in three different settings. Best and second best scores shown in bold and italic fonts, respectively. Here "PMPNN"= ProteinMPNN.

Models	Short chains		Single chains		All	
	PPL ↓	AAR (%) ↑	PPL ↓	AAR (%) ↑	PPL ↓	AAR (%) ↑
StructTrans	8.39	28.14	8.83	28.46	6.63	35.82
GVP	7.23	30.60	7.84	28.95	5.36	39.47
PMPNN	6.21	36.35	6.68	34.43	4.61	45.96
PMPNN-CMLM	7.16	35.42	7.25	35.71	5.03	48.62
PiFold	6.04	*39.84*	6.31	38.53	4.55	51.66
LM-Design	7.01	35.19	6.58	40.00	4.41	54.41
DPLM	-	-	-	-	-	54.54
AIDO.ProteinIF	**4.29**	38.46	**3.18**	*58.87*	**3.20**	*58.60*
Ours	*4.86*	**40.00**	*3.19*	**61.16**	*3.24*	**60.93**

Optimization. We study both individually and jointly optimized models for the denoiser (parameterized by $\theta_1, \theta_2, \theta_3$) and the uncertainty estimator (parameterized by θ_1, θ_2, ϕ). We first perform individual optimization by using the publicly available AIDO.ProteinIF model (Sun et al., 2024)[1] and training a separate uncertainty estimator with binary classification, updating only ϕ while freezing θ_1 and θ_2. We then jointly fine-tune all parameters with a multi-objective setup combining discrete diffusion (Austin et al., 2021) and classification losses. The best performance is achieved by sampling with individually trained models and refining with jointly trained ones—consistent with prior work showing the benefit of starting from reasonable initial estimates (Sun et al., 2024; Zheng et al., 2023b; Wang et al., 2024).

[1]https://huggingface.co/genbio-ai/AIDO.ProteinIF-16B

Table 2. Quantitative evaluation on TS50 and TS500 benchmark datasets. Best and second best scores shown in bold and italic fonts, respectively. Here "PMPNN"= ProteinMPNN.

Models	TS50		TS500	
	PPL ↓	AAR % ↑	PPL ↓	AAR % ↑
GVP	4.71	44.14	4.20	49.14
PMPNN	3.93	54.43	3.53	58.08
PMPNN-CMLM	3.60	54.84	3.46	57.44
PiFold	3.86	58.72	3.44	60.42
LM-Design	3.82	56.92	**2.13**	64.50
AIDO.ProteinIF	*2.93*	*66.19*	2.68	*69.66*
Ours	**2.86**	**68.85**	*2.59*	**71.18**

3. Results and Discussion

We evaluate our method on standard benchmarks—CATH-4.2 (Orengo et al., 1997), TS50 (Li et al., 2014), and TS500 (Li et al., 2014)—and compare it with several state-of-the-art baselines (Appendix C.3). Results in Tables 1 and 2 report performance using perplexity (PPL) and sequence recovery (or amino acid recovery, AAR) (Zheng et al., 2023b; Wang et al., 2024); metric and dataset details are provided in Appendix C.1 and C.2.

On the CATH-4.2 dataset, our method performs consistently well across all three standard evaluation settings (Appendix C.2), achieving the highest AAR in each. It improves over AIDO.ProteinIF by 1.54% on short sequences, 2.29% on single chains, and 2.33% overall. Our method also achieved highly competitive scores in perplexity, ranking second across all CATH-4.2 settings. On the full test set, it achieves a PPL of 3.28, closely behind AIDO.ProteinIF. Investigating the divergence between PPL and AAR remains future work.

To assess generalizability, we evaluate our method on TS50 and TS500—two distinct test-only datasets (Li et al., 2014; Zheng et al., 2023b). Our model achieves state-of-the-art AAR (68.85% on TS50, 71.18% on TS500), demonstrating strong generalization across diverse proteins. It also attains the best PPL on TS50 (2.88) and the second-best on TS500 (2.62), outperforming AIDO.ProteinIF (2.68).

4. Conclusion

We present an uncertainty-aware discrete denoising diffusion model for structure-conditioned protein sequence generation. By employing a prior-posterior uncertainty signaling mechanism, our approach enables adaptive and interpretable denoising trajectories that account for residue- and timestep-specific ambiguity. Through modular integration with pretrained large language model and structure encoder, and joint multi-objective optimization, our method achieves significant improvements over existing baselines. These results highlight the potential of incorporating uncertainty-awareness into discrete generative frameworks for advancing sequence design across biomolecular domains.

References

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Chen, S. F., Beeferman, D., and Rosenfeld, R. Evaluation metrics for language models. 1998.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378 (6615):49–56, 2022.

Ellington, C. N., Sun, N., Ho, N., Tao, T., Mahbub, S., Li, D., Zhuang, Y., Wang, H., Song, L., and Xing, E. P. Accurate and general dna representations emerge from genome foundation models at scale. *bioRxiv*, pp. 2024–12, 2024.

Gao, Z., Tan, C., Chacón, P., and Li, S. Z. Pifold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022.

Gao, Z., Tan, C., Zhang, Y., Chen, X., Wu, L., and Li, S. Z. Proteininvbench: Benchmarking protein inverse folding on diverse tasks, models, and metrics. *Advances in Neural Information Processing Systems*, 36:68207–68220, 2023.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023.

Ghazvininejad, M., Levy, O., Liu, Y., and Zettlemoyer, L. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019.

Hie, B., Bryson, B. D., and Berger, B. Leveraging uncertainty in machine learning accelerates biological discovery and design. *Cell systems*, 11(5):461–477, 2020.

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

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. Learning inverse folding from millions of predicted structures. *ICML*, 2022. doi: 10.1101/2022.04.10.487779. URL https://www.biorxiv.org/content/early/2022/04/10/2022.04.10.487779.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

John, J., Richard, E., Alexander, P., Tim, G., Michael, F., Olaf, R., Kathryn, T., Russ, B., Augustin, , Anna, P., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 2021.

Karimi, M., Zhu, S., Cao, Y., and Shen, Y. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks. *Journal of chemical information and modeling*, 60(12):5667–5681, 2020.

Kristiadi, A., Hein, M., and Hennig, P. Learnable uncertainty under laplace approximations. In *Uncertainty in Artificial Intelligence*, pp. 344–353. PMLR, 2021.

Li, Z., Yang, Y., Faraggi, E., Zhan, J., and Zhou, Y. Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles. *Proteins: Structure, Function, and Bioinformatics*, 82(10):2565–2573, 2014.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33:7498–7512, 2020.

Liu, S., Nam, J., Campbell, A., Stärk, H., Xu, Y., Jaakkola, T., and Gómez-Bombarelli, R. Think while you generate: Discrete diffusion with planned denoising. *arXiv preprint arXiv:2410.06264*, 2024.

Liu, Y. and Kuhlman, B. Rosettadesign server for protein design. *Nucleic acids research*, 34(suppl_2):W235–W238, 2006.

Mahbub, S. and Bayzid, M. S. Egret: edge aggregated graph attention networks and transfer learning improve protein–protein interaction site prediction. *Briefings in Bioinformatics*, 23(2):bbab578, 2022.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, 2021. doi: 10.1101/2021.07.09.450648. URL https://www.biorxiv.org/content/10.1101/2021.07.09.450648v1.

Nadav, B., Grant, G., Charlotte, H., W., Chun, Jimmie, Y., and Vasilis, N. Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, 2023.

Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

Sevgen, E., Moller, J., Lange, A., Parker, J., Quigley, S., Mayer, J., Srivastava, P., Gayatri, S., Hosfield, D., Korshunova, M., et al. Prot-vae: protein transformer variational autoencoder for functional protein design. *bioRxiv*, pp. 2023–01, 2023.

Sumi, S., Hamada, M., and Saito, H. Deep generative design of rna family sequences. *Nature Methods*, 21(3):435–443, 2024.

Sun, N., Zou, S., Tao, T., Mahbub, S., Li, D., Zhuang, Y., Wang, H., Cheng, X., Song, L., and Xing, E. P. Mixture of experts enable efficient and effective protein understanding and design. *bioRxiv*, pp. 2024–11, 2024.

Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Wang, X., Zheng, Z., Ye, F., Xue, D., Huang, S., and Gu, Q. Diffusion language models are versatile protein learners. *arXiv preprint arXiv:2402.18567*, 2024.

Zheng, L., Yuan, J., Yu, L., and Kong, L. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023a.

Zheng, Z., Deng, Y., Xue, D., Zhou, Y., Ye, F., and Gu, Q. Structure-informed language models are protein designers. In *International conference on machine learning*, pp. 42317–42338. PMLR, 2023b.

Zou, S., Tao, T., Mahbub, S., Ellington, C. N., Algayres, R., Li, D., Zhuang, Y., Wang, H., Song, L., and Xing, E. P. A large-scale foundation model for rna function and structure prediction. *bioRxiv*, pp. 2024–11, 2024.

A. Related Work

Early breakthroughs in this space established autoregressive models as a powerful design paradigm. ProteinMPNN (Dauparas et al., 2022) exemplified this direction, achieving state-of-the-art recovery rates across a range of structural motifs and design contexts, including oligomers and binders. Its ability to generalize across diverse topologies positioned it as a widely adopted baseline. To address the computational inefficiency inherent in autoregressive sampling, PiFold (Gao et al., 2022) proposed a hybrid framework that incorporates expressive backbone encodings with an accelerated decoding scheme, delivering order-of-magnitude speedups while preserving accuracy.

Parallel to these efforts, large-scale structure-supervised training became a promising strategy. Leveraging AlphaFold2-predicted structures (John et al., 2021), Hsu et al. (2022) trained transformer architectures to directly map backbone geometries to plausible sequences, allowing the model to internalize structural priors from millions of inferred conformations. This formulation decoupled the reliance on experimental structures and enabled broader generalization.

Pretrained language models have also been adapted for inverse folding by conditioning on structure-derived features. ESM-1b and ESM-1v (Nadav et al., 2023; Meier et al., 2021) served as foundational models, later extended in Zheng et al. (2023b) for structure-guided generation. These approaches benefit from linguistic pretraining on massive protein corpora, providing rich residue-level priors that can be fine-tuned for geometry-aware decoding.

More recently, generative models based on diffusion dynamics have introduced a new formulation. Diffusion Probabilistic Language Models (DPLMs) (Wang et al., 2024) model inverse folding as a discrete denoising process, gradually refining corrupted sequences toward structure-compatible outputs. This technique introduces temporal uncertainty modeling, which facilitates a smoother posterior landscape and often yields improved diversity and stability in generation.

At the high end of model capacity, AIDO.Protein (Sun et al., 2024) utilizes a 16-billion parameter mixture-of-experts framework pretrained across multiple sequence and structure tasks. Its architecture allows conditional computation and task-specific adaptation, leading to superior performance in benchmark evaluations and improved coverage of structurally diverse regions. Given the rapidly evolving landscape, benchmarking remains a critical bottleneck. ProteinInvBench (Gao et al., 2023) was introduced to standardize evaluation across structure-conditioned generation methods. It incorporates a wide spectrum of tasks—ranging from native sequence recovery to functional design—and includes unified metrics and competitive baselines, supporting reproducibility and comparability.

B. Preliminaries

B.1. Problem Definition

The task of protein inverse folding involves determining a plausible amino acid sequence that would adopt a given three-dimensional (3D) structure. Let the protein conformation be denoted by $\psi = \{\psi^1, \psi^2, \ldots, \psi^L\}$, where each $\psi^i \in \mathbb{R}^{N\times 3}$ represents the spatial positions of N representative atoms of the i-th residue in 3D space, and L denotes the length of the protein. The objective is to predict a corresponding primary sequence $x = [x^1, x^2, \ldots, x^L]$, where each $x^i \in \mathcal{A}$ is an amino acid drawn from the standard set $\mathcal{A}$ of canonical residues, where $\mid \mathcal{A} \mid = 20$.

The inverse folding process can be formulated as learning a function

$$\theta : \mathbb{R}^{L\times N\times 3} \to \mathcal{A}^L,$$

which takes the structural input and outputs a position-wise distribution over amino acids. Contemporary solutions rely heavily on neural architectures, particularly graph neural networks (GNNs) and 3D-aware models like convolutional networks adapted to irregular geometries, due to their capacity to model the intricate spatial and sequential dependencies inherent in protein structures (Gao et al., 2022; Dauparas et al., 2022; Jing et al., 2020; Hsu et al., 2022).

In many approaches, the protein structure is abstracted as a graph $\mathcal{G} = (\mathcal{N}, E)$, where each node $n_i \in \mathcal{N}$ corresponds to a residue and is annotated with geometric information (e.g., backbone coordinates), and each edge $e_{ij} \in E$ encodes relational features such as pairwise distance, angle, or biochemical interactions between residues i and j (Dauparas et al., 2022; Jing et al., 2020; Mahbub & Bayzid, 2022). This graph-based formalism allows the model to integrate local and non-local structural cues during sequence prediction.

Once trained, such models can infer a compatible sequence for a given structure either by sequentially choosing amino acids in an autoregressive fashion or by generating all residues simultaneously through a non-autoregressive mechanism, e.g.,

using variational autoencoders. Sampling-based techniques, including Monte Carlo simulations, and iterative refinement methods including denoising diffusion, are often used to explore diverse sequence candidates that conform to the same fold (Dauparas et al., 2022; Wang et al., 2024; Liu & Kuhlman, 2006).

C. Experimental Setup

C.1. Evaluation Metrics

We evaluate our model's performance in the protein inverse folding task using two principal metrics: Perplexity (PPL) and Amino Acid Recovery (AAR).

Perplexity (PPL). PPL provides a measure of the model's predictive certainty over amino acid choices and is commonly used in sequence modeling tasks (Chen et al., 1998). In the context of protein design, lower perplexity values suggest that the model assigns high probabilities to native-like sequences, indicating that the learned distribution aligns well with that of natural proteins (Zheng et al., 2023b).

For *autoregressive* models, PPL is generally calculated as,

$$\text{PPL}_{\text{AR}} = \exp\left(-\frac{1}{L}\sum_{i=1}^{L}\log P(x^i \mid x^{<i}, \psi)\right), \tag{13}$$

where $x^{<i}$ denotes the sequence prefix up to position $i-1$, and $\psi \in \mathbb{R}^{L\times N\times 3}$ represents the 3D structural context of the protein, where L is the number of residues in the protein and N is the number of representative atoms per-residue.

Since our model leverages a *non-autoregressive*, *iterative refinement* approach, we instead use a modified formulation where the predictions are conditioned on a noisy version of the native sequence,

$$\text{PPL}_{\text{NAR}} = \exp\left(-\frac{1}{L}\sum_{i=1}^{L}\log P(x_{t-1}^i \mid x_t, \psi)\right), \tag{14}$$

Here, x_t denotes a perturbed (noisy) variant of the true sequence at denoising time-step t, and $P(x_{t-1}^i \mid x_t, \psi)$ is the model's estimated probability of observing residue x_{t-1}^i at position i and time-step $t-1$ given x_t and ϕ.

Amino Acid Recovery (AAR). AAR, also known as sequence recovery rate, is widely recognized as the standard metric for structure-conditioned protein design (Sun et al., 2024). It measures the typical percentage of residues in a predicted sequence that match their counterparts in the native sequence for a given protein structure. For a protein of length L, AAR is defined as:

$$\text{AAR} = \frac{1}{L}\sum_{i=1}^{L}\mathbf{1}(x_t^i = x_0^i)\times 100\%, \tag{15}$$

where x_0^i is the native amino acids at position i and x_t^i is its estimate at denoising time-step t, and $\mathbf{1}(x_t^i = x_0^i)$ is an indicator function equal to 1 if the two residues match, and 0 otherwise. AAR is reported as the median over all the sequences in a test set (Zheng et al., 2023b; Wang et al., 2024; Dauparas et al., 2022).

C.2. Benchmark Datasets

We conduct our experiments using three established benchmarks in the protein inverse folding literature: CATH-4.2 (Orengo et al., 1997), TS50 (Li et al., 2014), and TS500 (Li et al., 2014). Comprehensive statistics for these datasets are presented in Table 3. Among them, CATH-4.2 is a commonly adopted dataset that serves as the primary source for training, validation, and testing in numerous prior works (Gao et al., 2022; Dauparas et al., 2022; Wang et al., 2024; Sun et al., 2024). All protein sequences in this benchmark are capped at 500 amino acids in length. Within the CATH-4.2 test set, earlier studies have delineated three experimental subsets: sequences under 100 residues (termed "short sequences"), sequences that belong to

a single protein chain (having only one entry in CATH 4.2, comprising roughly 92.86% of the test set), and the full test set itself. The short-sequence subset includes proteins with fewer than 100 amino acids, accounting for about 16.5% of all test samples. To evaluate generalization beyond the training domain, we additionally assess performance on TS50 and TS500. TS50 contains only 50 proteins, with the longest sequence spanning 173 residues. In contrast, TS500 introduces significant sequence length diversity, ranging from just 43 to as long as 1,636 residues. Following prior protocols (Zheng et al., 2023b; Gao et al., 2022), we use TS50 and TS500 exclusively for evaluation, relying on models trained solely on the training portion of CATH-4.2.

Table 3. Overview of dataset composition and sequence length distribution across CATH-4.2, TS50, and TS500. Columns: "Sample count" = number of sequences, "Residue count" = number of total amino acids, "Avg." = average sequence length, "Med." = median sequence length, "S.D." = standard deviation of sequence lengths.

Dataset	Split	Sample count	Residue count	Avg.	Med.	S.D.
CATH-4.2	Train	18,024	3,941,775	218.7	204.0	109.93
	Validation	608	105,926	174.2	146.0	92.44
	Test	1,120	181,693	162.2	138.0	82.22
	All	19,752	4,229,394	214.1	196.0	109.06
TS50	Test	50	6,861	137.2	145.0	25.96
TS500	Test	500	130,960	261.9	225.0	167.30

C.3. Baselines for Comparison

To benchmark our method, we compare against eight state-of-the-art approaches that exemplify diverse modeling paradigms for structure-conditioned protein sequence generation: AIDO.Protein (Sun et al., 2024), ProteinMPNN (Dauparas et al., 2022), its non-autoregressive counterpart ProteinMPNN-CMLM (Zheng et al., 2023b), LM-Design (Zheng et al., 2023b), DPLM (Wang et al., 2024), PiFold (Gao et al., 2022), GVP (Jing et al., 2020), StructTrans (Ingraham et al., 2019).

AIDO.Protein (Sun et al., 2024) leverages large-scale pretraining—spanning 16 billion parameters—and adapts to the inverse folding task through a discrete diffusion modeling objective that is explicitly conditioned on structural input. ProteinMPNN (Dauparas et al., 2022), an autoregressive model, generates amino acid sequences sequentially based on structural input, whereas ProteinMPNN-CMLM (Zheng et al., 2023b) replaces the autoregressive decoding with a masked prediction strategy via the conditional masked language modeling (CMLM) objective (Ghazvininejad et al., 2019), yielding improved sequence recovery. Built similarly on the CMLM framework, LM-Design (Zheng et al., 2023b) augments non-autoregressive decoding with protein language model pretraining to further boost performance on inverse folding. DPLM (Wang et al., 2024) takes a different direction by introducing discrete diffusion objectives, enabling more flexible sequence modeling through iterative refinement. PiFold (Gao et al., 2022) combines expressive structural features with efficient autoregressive decoding, offering both competitive accuracy and inference speed. GVP (Jing et al., 2020) extends classical neural architectures by incorporating geometric vector perceptrons, enabling operations directly on 3D Euclidean features. StructTrans (Ingraham et al., 2019) adopts a conditional generation framework over protein graphs to synthesize sequences compatible with given backbone structures.