TOWARDS A GENERATIVE PROTEIN EVOLUTION MACHINE WITH DPLM-EVO

Anonymous authors
Paper under double-blind review

ABSTRACT

Proteins are shaped by evolution under biophysical and functional constraints. Protein language models can learn rich evolutionary constraints, and discrete diffusion-based PLMs (DPLMs) are promising for both understanding and generation. However, existing DPLMs rely on masking-based diffusion, which is a loose proxy for evolution, and difficult to model the edit operations that drive sequence change in nature: substitutions and insertions/deletions (indels). In this paper, we present DPLM-EVO, an evolutionay discrete diffusion protein language model that explicitly predicts substitution, insertion, and deletion actions during denoising. To make indel-aware generation tractable, we introduce a latent alignment formulation that supports variable-length sequences. To make substitution corruption informative, we propose a contextual evolutionary noising kernel that generates biologically plausible mutations. On ProteinGym, DPLM-EVO achieves state-of-the-art mutation effect prediction in the single-sequence setting, and it enables variable-length generation and post-editing via explicit edit trajectories.

1 INTRODUCTION

Today's rapidly growing sequence databases archive the results of protein evolution over millions of years, capturing both conserved patterns and extensive natural variation across families. For protein engineering, the practical goal is often not only to generate "protein-like" sequences, but also to leverage this evolutionary information to (i) predict the functional impact of mutations and (ii) propose variants that preserve the structure while improving or reprogramming function.

Protein language models (PLMs) trained on large protein sequence corpus have become a dominant paradigm for learning such evolutionary regularities (Lin et al., 2022; Hayes et al., 2024; Nijkamp et al., 2022; Wang et al., 2024b;a). By modeling the statistics of natural sequence variation, PLMs enable diverse applications including sequence-only, zero-shot mutation effect prediction (Meier et al., 2021) and protein structure prediction (Lin et al., 2022). In many real design workflows, however, the problem is inherently *edit-based*: engineers start from a natural scaffold and iteratively introduce substitutions and indels to modify loops, linkers, termini, or binding interfaces while preserving the overall fold and key functional sites.

Diffusion-based protein language models (Sohl-Dickstein et al., 2015; Austin et al., 2021; Hoogeboom et al., 2021; Zheng et al., 2023a; Sahoo et al., 2024; Shi et al., 2024; Nie et al., 2025) provide a powerful bidirectional modeling paradigm, but most existing methods use *masked diffusion*: the forward process hides tokens and generation recovers them. Masking is a loose proxy for evolution, which changes sequences through *substitutions and indels* and often requires variable-length edits for remodeling loops, linkers, and motifs. This mismatch makes it difficult to express realistic evolutionary trajectories and to post-edit existing proteins when length changes are needed.

We propose DPLM-EVO, an evolutionary discrete diffusion framework that predicts substitution, insertion, and deletion actions during denoising (Fig. 1). Our key design is to introduce a latent alignment space for diffusion, coupled to the observed sequence space, which makes indel-aware modeling tractable. We further introduce a contextual evolutionary noising kernel for substitution corruption to provide informative, biologically plausible mutation noise.

In short, our contributions are: (i) introducing an evolutionary discrete diffusion with explicit substitution, insertion, and deletion predictions; (ii) decoupling a latent alignment formulation for length-adaptive modeling; and (iii) biologically informed noising for substitution. Empirically, DPLM-EVO achieves state-of-the-art single-sequence mutation effect prediction on ProteinGym and supports variable-length generation and post-editing via explicit edit trajectories.

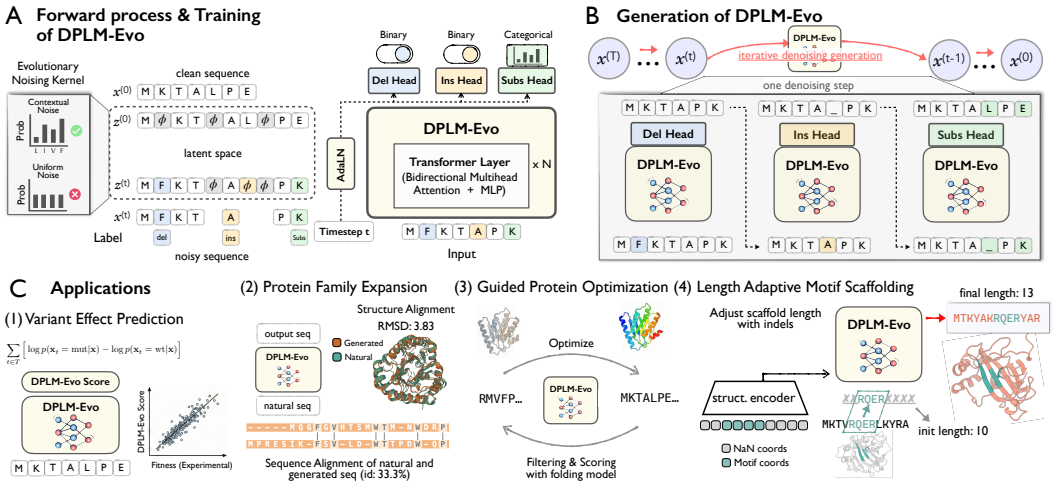


Figure 1: (A) DPLM-EVO models evolution with explicit substitution, insertion, and deletion predictions via a latent alignment space that supports variable length. (B) Sampling iteratively applies these edits to generate sequences. (C) Applications include mutation effect prediction, simulated evolution, optimization, and conditional motif scaffolding.

2 PRELIMINARIES

We briefly review diffusion protein language models (DPLMs) under the masked diffusion framework, which underpins our evolutionary extension.

DPLM with masked diffusion. The *masked* discrete diffusion framework (Sahoo et al., 2024; Liu et al., 2025; Nie et al., 2025) is characterized by a forward and backward Markov process. Let $\texttt{Cat}(\mathbf{x}; \mathbf{p})$ be a categorical distribution on protein sequence $\mathbf{x}$ parameterized by a probability vector $\mathbf{p}$ over the vocabulary $\mathcal{A}$ of K amino acids. The forward process of masked diffusion is governed by

$$q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = \texttt{Cat}\big(\mathbf{x}^{(t)}; \bar{\alpha}_t \delta_{\mathbf{x}^{(t-1)}} + (1-\bar{\alpha}_t)\pi_{\text{mask}}\big),$$

which gradually perturb the data $\mathbf{x}_0 \sim q(\mathbf{x})$ into an absorbing state $\mathbf{x}^{(T)} \sim \pi_{\text{mask}}$. The learned *backward* process $p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$ reversely denoises $\mathbf{x}^{(T)}$ towards the data distribution $\mathbf{x}$.

The learning objective can be simplified into weighted cross-entropies, resembling masked language modeling at arbitrary noise levels:

$$\begin{aligned}\mathcal{L}_t &= \mathbb{E}_{q(\mathbf{x})}\text{KL}\big[q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x})\|p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})\big] \\ &= \mathbb{E}_{q(\mathbf{x})}\Big[\lambda_t \textstyle\sum_{1\le i\le L} \mathbb{I}_{x_t^{(i)} \neq x_0^{(i)}} \cdot \log p_\theta(x_0^{(i)}|\mathbf{x}^{(t)})\Big],\end{aligned}$$

where λ_t is a weighting coefficient induced by the noising schedule. For inference, DPLM samples from an all-mask initialization and iteratively denoises in a *mask-predict* manner.

Motivation. The masked diffusion formulation of makes it difficult to (i) represent the elementary evolutionary edits that biologists and engineers apply in practice—*substitutions, insertions, and deletions*—and (ii) support flexible, variable-length trajectories during generation and post-editing. To address this mismatch, we extend the DPLM framework to explicitly model evolutionary edit operations in next section.

3 METHODOLOGY

Standard discrete diffusion is typically fixed-length and therefore cannot explicitly represent insertions and deletions. We extend masked discrete diffusion to an evolutionary framework by coupling a variable-length *observed sequence space* to a *latent alignment space* (Havasi et al., 2025; Graves et al., 2006). The alignment is served as a latent variable for the forward noising process between observed clean and noisy sequence, enabling variable length modeling. The model predicts a substitution distribution together with insertion and deletion probabilities, thereby supporting explicit edit trajectories and variable-length sampling.

3.1 AN EVOLUTIONARY DISCRETE DIFFUSION FRAMEWORK

Accomodating length-adaptive *indel* modeling with latent alignment. Let $\mathcal{V}$ represent the amino acid vocabulary. To handle variable-length sequences, we draw inspiration from latent-alignment methods (Graves et al., 2006; Havasi et al., 2025) and distinguish between two spaces:

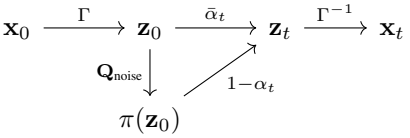
- **Observed Space $\mathcal{X}$:** Sequences $\mathbf{x} \in \mathcal{V}^L$ with $\mathcal{V} = \mathcal{A} \cup \{\mathbf{m}\}$, where $\mathbf{m}$ is a mask token.
- **Latent Alignment Space $\mathcal{Z}$:** Sequences of length $2L$ over $\mathcal{V}^+ = \mathcal{V} \cup \{\phi\}$, where ϕ is a gap token.

We define a deterministic *collapse function* $\Gamma^{-1}(\mathbf{z}) : \mathcal{Z} \to \mathcal{X}$ that maps a latent alignment $\mathbf{z}$ to an observed sequence $\mathbf{x}$ by removing all ϕ tokens, *i.e.*, $\Gamma^{-1}(\mathbf{z}) = [\mathbf{z}^{(j)} \mid \mathbf{z}^{(j)} \neq \phi]_{j=1}^{N}$. Conversely, $\Gamma(\mathbf{x})$ denotes the set of all latent alignments $\mathbf{z}$ that collapse to $\mathbf{x}$, obtained by inserting exactly L gap tokens ϕ into $\mathbf{x}$ at arbitrary positions, for example $[A, B, C] \mapsto [A, \phi, \phi, B, \phi, C]$. This expanded latent canvas, which is strictly longer than observed $\mathbf{x}$, enables length-changing generation through explicit gap token transitions.

Given an observed protein $\mathbf{x}_0$, we treat its alignment $\mathbf{z}_0$ as a latent variable, the training objective is to maximize the evidence lower bound (ELBO) of the log-likelihood:

$$\log p_\theta(\mathbf{x}_0) = \log \sum_{\mathbf{z}_0 \in \Gamma(\mathbf{x}_0)} p_\theta(\mathbf{z}_0) \geq \mathbb{E}_{\mathbf{z}_0 \in \Gamma(\mathbf{x}_0)} \Big[\mathbb{E}_{\mathbf{z}_t \sim q_t(\mathbf{z}_t|\mathbf{z}_0)} \big[\log p_\theta(\mathbf{z}_0|\mathbf{z}_t) \big] \Big]. \tag{1}$$

Forward noising process for sequence with holistic edit operations. Unlike predominant masked diffusion models that use absorbing-state (mask) noise, we introduce a new noising prior $\pi(\mathbf{z}_0)$ that respects all possible sequence edit operations.



Specifically, the forward transition $q(\mathbf{z}_t|\mathbf{z}_0)$ is defined as an interpolant of the original data and the noise distribution: $q_t(\mathbf{z}_t|\mathbf{z}_0) = \bar{\alpha}_t \delta_{\mathbf{z}_0} + (1 - \bar{\alpha}_t)\pi(\mathbf{z}_0)$, where $\bar{\alpha}_t$ is the noise schedule. The noise distribution $\pi(\mathbf{z}_0)$ depends on the initial state $\mathbf{z}_0$ via a transition matrix $\mathbf{Q}_{\text{noise}} \in \mathbb{R}^{(K+2)\times(K+2)}$, defining transitions between K amino acids, $\mathbf{m}$ and ϕ:

$$\pi(\mathbf{z}_0) = \texttt{Cat}(\cdot \mid \mathbf{Q}_{\text{noise}} \mathbf{z}_0).$$

To explicitly control the ratio of substitutions, insertions, and deletions, we parameterize $\mathbf{Q}_{\text{noise}}$ with a deletion ratio ω_{del} and an insertion ratio ω_{ins}:

$$\mathbf{Q}_{\text{noise}} = \begin{array}{c} \begin{matrix} z \in \mathcal{A} & \quad z = \mathbf{m} & \quad z = \phi \end{matrix} \\ \left(\begin{array}{c|c|c} (1-\omega_{\text{del}})(1-\rho_{\text{mask}})\mathbf{U}_K & \mathbf{0}_K & \omega_{\text{ins}} \frac{1}{K} \mathbf{1}_K \\ (1-\omega_{\text{del}})\rho_{\text{mask}} \mathbf{1}_K^\top & 1 & 0 \\ \omega_{\text{del}} \mathbf{1}_K^\top & 0 & 1-\omega_{\text{ins}} \end{array} \right) \end{array},$$

where $\mathbf{1}_K$ denotes a K-dime column vector of all ones, and $\mathbf{U}_K = \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top$ denotes the uniform transition matrix of size $K \times K$ over amino acids. Intuitively, this implies:

- If $z_0^{(j)} \in \mathcal{A}$ (amino acid): It is either substituted (with probability $1 - \omega_{\text{del}}$) by another amino acid or mask token (up to ρ_{mask}, or deleted (becomes ϕ, with probability ω_{del}).
- If $z_0^{(j)} = \phi$ (gap): It either becomes an inserted amino acid, with probability ω_{ins} or remains a gap (with probability $1 - \omega_{\text{ins}}$).

Simulating biological sequence mutations as evolutionary noising via contextualized on-policy substitution. Uniform substitution noise ignores biophysical constraints and is often uninformative for learning evolutionary structure. We therefore use a *contextual evolutionary noising kernel* that draws substitution noise from the model's on-policy predictions after a warmup stage, yielding more plausible and informative corruptions. We provide the details in §A.1.

Connections to existing discrete diffusion. By setting $(\omega_{\text{del}}, \omega_{\text{ins}}, \rho_{\text{mask}})$ in $\mathbf{Q}_{\text{noise}}$, our model recovers masked diffusion and uniform diffusion as special cases, and enables initializing from pretrained masked diffusion models. We provide a discussion in Appendix A.2.

3.2 Training

Overall Objective. We optimize a weighted sum of substitution, deletion, and insertion losses over observed tokens (collapsed from the latent alignment), with weights λ_{t-1} and γ controlling the denoising schedule and edit balance:

$$\mathcal{L}_t = \mathbb{E}_{\mathbf{x}_0, \mathbf{z}_0, \mathbf{z}_t} \Bigg[\sum_{k=1}^{|\Gamma^{-1}(\mathbf{z}_t)|} \lambda_{t-1} \big(\gamma_{\text{sub}} \mathcal{L}_{\text{sub}}^{(k)} + \gamma_{\text{del}} \mathcal{L}_{\text{del}}^{(k)} + \gamma_{\text{ins}} \mathcal{L}_{\text{ins}}^{(k)} \big) \Bigg].$$

The introduced decomposed losses enables precise control over the model's propensity for different evolutionary operations. We use three heads on the observed sequence $\mathbf{x}_t = \Gamma^{-1}(\mathbf{z}_t)$: a substitution head p_θ^{sub} (multiclass over $\mathcal{V}$), and deletion/insertion heads $p_\theta^{\text{del}}, p_\theta^{\text{ins}}$ (binary). We provide the formulation of full decomposed objectives and the practical implementation in Appendix A.3.

3.3 Generation of DPLM-Evo

DPLM-Evo follows standard iterative denoising for discrete diffusion, while enabling variable-length trajectories through explicit insertion and deletion actions in the observed sequence space. At each step, we apply deletion/insertion decisions from the indel heads, then fill substituted and newly inserted residues using the substitution head. Appendix B provides the full sampling procedure.

4 Experiments

In this section, we evaluate DPLM-Evo across various understanding and generative tasks. We assess variant effect prediction to validate the model's understanding of protein evolution, and evaluate generation capabilities through unconditional generation and motif scaffolding. Full experimental setups, additional analyses, and case studies are provided in Appendix C.

4.1 Variant Effect Prediction

Setup. Modeling the effect of sequence variation on function is fundamental for understanding and designing proteins.

DPLM-Evo predicts variant effects using protein sequence only, without supervision from experimental data. Unlike the common masked-residue scoring pipeline, DPLM-Evo is a substitution-based model that natively scores variants without masking by evaluating the substitution distribution at the mutated site(s). Please refer to Appendix C.1 for more details.

Results. On the ProteinGym DMS substitution zero-shot benchmark (Notin et al., 2023), DPLM-Evo *achieves the highest correlation score among all the single sequence foundation models for variant effect prediction* (Fig.2A). We attribute this strong correlation to the model's evolutionary pretraining, which fundamentally enables it to learn mutation preferences from natural proteins, effectively capturing the constraints imposed by natural selection.

Moreover, *explicitly aligning with evolutionary kernel further unlocks the potential of* DPLM-Evo *in mutation effect prediction.* This makes the scores correlate more closely with natural mutations (Appendix C.1). Illustrated in Fig.2B, this alignment yields further enhancements, outperforming SaProt (Su et al., 2023) that takes additional structure and TranceptEVE L with supplementary MSA.

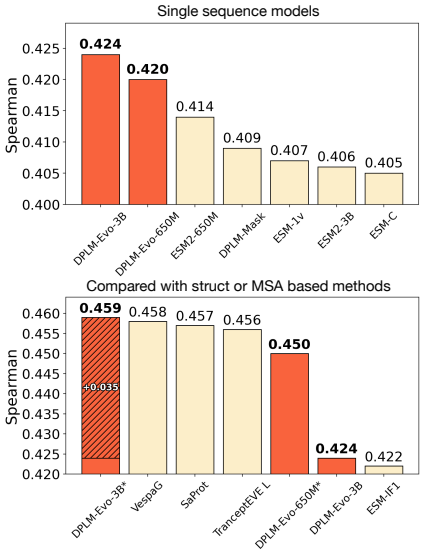


Figure 2: *Results of ProteinGym zero-shot DMS substitution benchmark.* * represents the model which explicitly aligns the output distribution with MSA-based mutation effect prediction model.

4.2 Unconditional Protein Sequence Generation

DPLM-Evo performs unconditional generation by applying substitution, deletion, and insertion, enabling variable-length generation that more closely mirrors natural evolutionary trajectories. Full experimental settings can refer to Appendix C.2.

Fig.3 demonstrates the evaluation results of unconditional generation in various perspectives. (1) *Overall performance:* Fig. 3A shows DPLM-Evo achieves consistent high foldability across length as measured by ESMFold pLDDT. Fig.3B-C demonstrates the output lengths remain concentrated near their initial values without excessive expansion or collapse. This indicates that insertion and deletion prediction are invoked conservatively, resulting in a refinement process that prioritizes substitutions over drastic length changes. (2) *Diversity and Reduced Mode Collapse:* Compared with DPLM based on masked diffusion, Fig.3D-F shows DPLM-Evo achieves comparable foldability while possessing greater generation diversity and higher sequence entropy, indicating fewer repetition patterns and alleviating the mode collapse issue. (3) *Effect of Evolutionary Kernel:* Training with the *contextual evolutionary noising kernel* substantially outperforms uniform noising, as shown in Fig.3D. This indicates that biologically grounded corruptions encourages DPLM-Evo to learn more

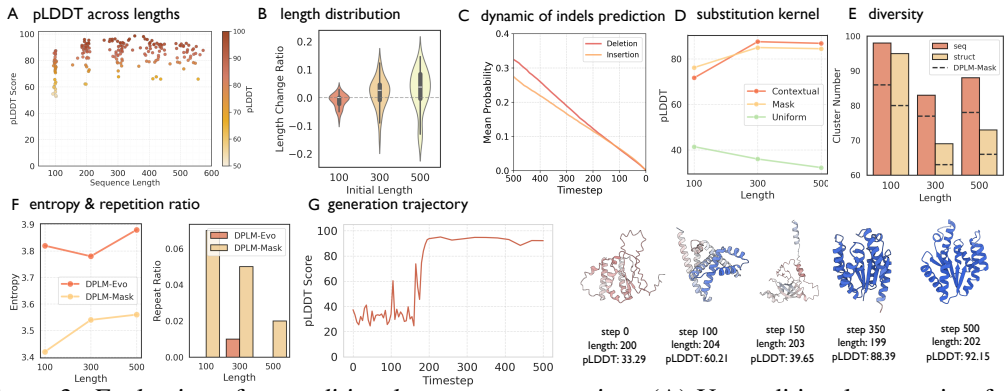


Figure 3: *Evaluation of unconditional sequence generation.* (A) Unconditional generation from length 100 to 500 evaluated by pLDDT. (B) Length distribution of DPLM-EVO generations from fixed initial lengths. (C) Insertion/deletion head predicted probability under different timesteps. (D) Abaltion on different substitution kernels. (E) Sequence and structure diversity of DPLM-EVO compared with DPLM-Mask in different lengths. (F) Entropy and repetition comparison between DPLM-EVO and DPLM-Mask. (G) Demonstration of the generation trajectory.

evolutionarily plausible substitution predictions, yielding higher-quality samples at generation time. Further details and analyses about unconditional generation can be found in Appendix C.2.

4.3 LENGTH-ADAPTIVE SCAFFOLDING OF FUNCTIONAL MOTIFS

Setup. Motif scaffolding aims to generate a protein scaffold for a given functional motif. We evaluate DPLM-EVO in *zero-shot* and *continued fine-tuning* settings. For finetuning, DPLM-EVO incorporates structural constraints for motif structure features, as illustrated in Fig.1C(4). Further details on settings and evaluation metrics are provided in the Appendix C.3 for more details.

Results. In the zero-shot setting, DPLM-EVO achieves higher overall success rate than EvoDiff and DPLM-Mask (Fig.4); we attribute this to the capability for dynamic scaffolding length adjustment and evolutionarily plausible mutations provided by the substitution head. In contrast, fixed-length sequence models require manually scaffold length enumeration and cannot revise length once an unsuitable initialization is chosen. Continued finetuning brings further improvements, highlighting the importance of multimodal conditioning (Appendix C.3).

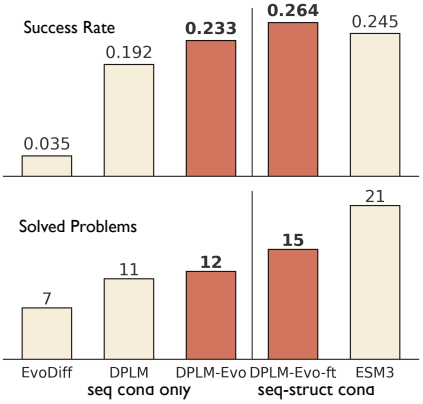


Figure 4: *Evaluation of motif scaffolding task.*

4.4 CASE STUDIES

In-silico sequence family expansion. As shown in Fig. 5 and Appendix C.4, DPLM-EVO is able to generate diverse yet structurally similar relatives of a given protein via unconstrained post-editing. This suggest that DPLM-EVO performs unconstrained sequence optimization that preserves a shared structural scaffold while producing diverse sequences, with sequence identity mostly below 50%.

Directed evolution of GFP. Fig. 6 and Appendix C.5 demonstrates that DPLM-EVO improves the structural stability score (pTM) over iterations while keeping the chromophore site RMSD below a strict threshold, indicating it can leverage priors from evolution-scale sequences to optimize sequences under hard structural constraints.

5 CONCLUSION

In this work, we present DPLM-EVO, a unified framework for evolutionary discrete diffusion that explicitly models substitutions, insertions, and deletions. We decouple the latent space from observed sequences, enabling efficient indel-aware diffusion. We further enhance the substitution learning through a contextual evolutionary noising kernel. Extensive empirical evaluations demonstrate that DPLM-EVO not only achieves state-of-the-art performance in mutation effect prediction on ProteinGym but also opens new avenues for variable-length protein generation and optimization, bridging the gap between deep generative modeling and evolutionary biology.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Ethan Baron, Alan N Amin, Ruben Weitzman, Debora Marks, and Andrew Gordon Wilson. A diffusion model to shrink proteins while maintaining their function. *arXiv preprint arXiv:2511.07390*, 2025.

Armin Behjati, Fatemeh Zare-Mirakabad, Seyed Shahriar Arab, and Abbas Nowzari-Dalini. Protein sequence profile prediction using protalbert transformer. *Computational Biology and Chemistry*, 99:107717, 2022.

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Sebastian Deorowicz, Agnieszka Debudaj-Grabysz, and Adam Gudyś. Famsa: Fast and accurate multiple sequence alignment of huge protein families. *Scientific reports*, 6(1):33964, 2016.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

ESM Team. Esm cambrian: Revealing the mysteries of proteins with unsupervised learning, 2024. URL https://evolutionaryscale.ai/blog/esm-cambrian.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376, 2006.

Jiatao Gu and Xiang Kong. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 120–133, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.11. URL https://aclanthology.org/2021.findings-acl.11.

Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, volume 32, pp. 11179–11189, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/675f9820626f5bc0afb47b57890b466e-Abstract.html.

Yong Guo, Yin Zheng, Mingkui Tan, Qi Chen, Jian Chen, Peilin Zhao, and Junzhou Huang. Nat: Neural architecture transformer for accurate and compact architectures. *Advances in Neural Information Processing Systems*, 32, 2019.

Marton Havasi, Brian Karrer, Itai Gat, and Ricky TQ Chen. Edit flows: Flow matching with edit operations. *arXiv preprint arXiv:2506.09018*, 2025.

Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *bioRxiv*, pp. 2024–07, 2024.

Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

Cheng-Yen Hsieh, Xinyou Wang, Daiheng Zhang, Dongyu Xue, Fei Ye, Shujian Huang, Zaixiang Zheng, and Quanquan Gu. Elucidating the design space of multimodal protein language models. In *Forty-second International Conference on Machine Learning*, 2025.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8946–8970. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hsu22a.html.

Elodie Laine, Yasaman Karami, and Alessandra Carbone. Gemme: a simple and fast global epistatic model predicting mutational effects. *Molecular biology and evolution*, 36(11):2604–2619, 2019.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022.

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.

Yangzhou Liu, Yue Cao, Hao Li, Gen Luo, Zhe Chen, Weiyun Wang, Xiaobo Liang, Biqing Qi, Lijun Wu, Changyao Tian, et al. Sequential diffusion language models. *arXiv preprint arXiv:2509.24007*, 2025.

Céline Marquet, Julius Schlensok, Marina Abakarova, Burkhard Rost, and Elodie Laine. Expert-guided protein language models enable accurate and blazingly fast fitness prediction. *Bioinformatics*, 40(11):btae621, 11 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae621. URL https://doi.org/10.1093/bioinformatics/btae621.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, pp. 29287–29303, 2021.

Ananthan Nambiar, Maeve Heflin, Simon Liu, Sergei Maslov, Mark Hopkins, and Anna Ritz. Transforming the language of life: transformer neural networks for protein prediction tasks. In *Proceedings of the 11th ACM international conference on bioinformatics, computational biology and health informatics*, pp. 1–8, 2020.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *The Thirteenth International Conference on Learning Representations*, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Erik Nijkamp, Jeffrey Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *arXiv preprint arXiv:2206.13517*, 2022.

Pascal Notin, Aaron W Kollasch, Daniel Ritter, Lood Van Niekerk, Steffan Paul, Han Spinner, Nathan J Rollins, Ada Shaw, Ruben Weitzman, Jonathan Frazer, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*, 2024.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

Machel Reid, Vincent Josua Hellendoorn, and Graham Neubig. Diffuser: Diffusion via edit-based reconstruction. In *International Conference on Learning Representations*, 2022.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Forty-second International Conference on Machine Learning*.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander M Rush, Thomas PIERROT, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR, PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020b.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pp. 2023–10, 2023.

Robert Verkuil, Ori Kabeli, Yilun Du, Basile IM Wicky, Lukas F Milles, Justas Dauparas, David Baker, Sergey Ovchinnikov, Tom Sercu, and Alexander Rives. Language models generalize beyond natural proteins. *bioRxiv*, pp. 2022–12, 2022.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Dplm-2: A multimodal diffusion protein language model. *arXiv preprint arXiv:2410.13782*, 2024a.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *arXiv preprint arXiv:2402.18567*, 2024b.

Zirui Wu, Lin Zheng, Zhihui Xie, Jiacheng Ye, Jiahui Gao, Yansong Feng, Zhenguo Li, Victoria W., Guorui Zhou, and Lingpeng Kong. Dreamon: Diffusion language models for code infilling beyond fixed-size canvas, 2025. URL https://hkunlp.github.io/blog/2025/dreamon.

Yijia Xiao, Jiezhong Qiu, Ziang Li, Chang-Yu Hsieh, and Jie Tang. Modeling protein using large-scale pretrain language model. *arXiv preprint arXiv:2108.07435*, 2021.

Kevin K Yang, Alex X Lu, and Nicolo Fusi. Convolutions are competitive with transformers for protein sequence pretraining. *bioRxiv*, pp. 2022–05, 2022.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Quanquan Gu. Diffusion language models can perform many tasks with scaling and instruction-finetuning. *arXiv preprint arXiv:2308.12219*, 2023a.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. Dinoiser: Diffused conditional sequence learning by manipulating noises. *arXiv preprint arXiv:2302.10025*, 2023b.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023a.

Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei YE, and Quanquan Gu. Structure-informed language models are protein designers. In *International Conference on Machine Learning*, 2023b.

A Methodology Details.

A.1 Substitution Learning with Contextual Evolutionary Noise

In this section, we discuss the noise kernel for the substitution head. Standard multinomial diffusion models typically employ un-informative uniform noise for corruption, which ignores the biophysical constraints of the protein fitness landscape and leads to inefficient training. To address this, we propose a *contextual evolutionary noising kernel* that instantiates the substitution noise sampled from a contextualized distribution. The contribution is obtained using the model's own on-policy denoising predictions, where the noise position is first noised with $\mathbf{Q}_{\text{noise}}$, then predicted. After a warmup stage trained with the original non-informative $\mathbf{Q}_{\text{noise}}$, this model's on-policy substitution prediction provides biologically plausible and evolutionarily reasonable mutation as noise, which is more informative than random noise. Meanwhile, it encourages the model to capture evolutionary and homologous dependencies between amino acids and sequences. We will introduce this in detail in the following part.

A.1.1 Motivation: from uniform noise to biological manifolds

Standard discrete diffusion models typically employ an uniform noise kernel (Hoogeboom et al., 2021). In the context of protein engineering, this implies that a mutation from a hydrophobic residue (e.g., Leucine) to a charged residue (e.g., Arginine) is as probable as a mutation to another hydrophobic residue (e.g., Isoleucine). This assumption fundamentally contradicts the biophysical constraints of the protein fitness landscape. Training with uniform noise suffers from significant inefficiency: the model spends a large portion of training correcting biologically obvious errors (e.g., restoring a hydrophobic core disrupted by charged noise) rather than learning evolutionary dependencies. To address this, we propose a **contextual evolutionary noising kernel**. Instead of corrupting data towards random chaos, we utilize the model's own prediction capability to generate noise that lies on the protein manifold. This not only provides more informative noisy tokens, which is helpful for denoising, but also encourages the model to capture the dependencies between the wild-type sequence and plausible homologous variants generated by model prediction.

A.1.2 Formalization: construct the confidence-aware kernel with mask prediction

We leverage mask token $\mathbf{m}$ (distinct from the gap token ϕ used for deletions) to represent unknown semantic identity in the protein sequence. Let p_θ denote the neural network parameterized by θ. At time step t, we construct a contextual distribution for an amino acid at index j in latent alignment sequence $\mathbf{z}$ (where $\mathbf{z}_0^{(j)} \in \mathcal{V}$) by masking the input at that position:

$$p_{\text{ctx}}(\cdot|\mathbf{z}_0^{(j)}) = \text{Softmax}\left(p_\theta(\Gamma^{-1}(\mathbf{z}_0^{\setminus j} \cup \{\mathbf{m}\}))^{(j)}\right) \tag{2}$$

where $\mathbf{z}_0^{\setminus j} \cup \{\mathbf{m}\}$ represents the latent sequence with the j-th token replaced by $\mathbf{m}$. Specifically, contextual noise is sampled from p_{ctx}, then the model is trained to denoise based on contextual noise.

A.1.3 IMPLEMENTATION: ON-POLICY CONFIDENCE-AWARE KERNEL FOR MORE EFFECTIVE TRAINING AND GENERATION

We leverage p_θ^{subs} of DPLM-EVO for the mask prediction to construct p_{ctx}. However, during training, the model is trained to denoise based on the contextual noise, which consists of the standard amino acids. As training processes, DPLM-EVO will lost its the mask prediction capabilities. To prevent catastrophic forgetting of the masking prediction, we construct a mixing noising kernel for the contextual noise and the mask state. Therefore, DPLM-EVO can learn the mask prediction explicitly during training. Specifically, we employ a *confidence-aware* gating mechanism to construct the noising kernel:

$$\pi(v \mid z_0^{(j)}) = \mathbb{I}(c_j > \tau) \cdot p_{\text{ctx}}(v|z_0^{(j)}) + \mathbb{I}(c_j \leq \tau) \cdot \delta_{\mathbf{m}}(v) \tag{3}$$

Here, $c_j = \max_{v \in \mathcal{V}} p_{\text{ctx}}(v|z_0^{(j)})$ is the confidence score of the model's prediction, and τ is a dynamic threshold. If $c_j \leq \tau$, this indicates the low confidence prediction that the model is uncertain, which represents insufficiently valuable or evolutionarily relevant information. Therefore, we fallback to the mask token $\mathbf{m}$ for these positions. This reinforces the fundamental masked prediction objective, ensuring the model remains robust to missing information and avoid forgetting about mask prediction.

Crucially, this process is **on-policy**: the noise is generated by the model state θ at the current training step. To enhance the quality of contextual noise at the early training stage and prevent the training instability, we initialize the model parameters from a pre-trained MLM-based pLM or an absorbing discrete diffusion-based pLM.

A.1.4 BIOLOGICAL INTERPRETATION: TRAVERSING THE FITNESS LANDSCAPE

By replacing the static uniform kernel with our dynamic contextual kernel, we reframe the diffusion training process as a traversal on the fitness landscape:

(1) Denoising as Error Correction (Lethal Mutations): When the contextual noise p_{ctx} generates a residue that violates structural constraints (e.g., steric clashes), the training objective is to enable the model to identify and correct these erroneous mutations that are evolutionarily unacceptable.

(2) Denoising as Homology Modeling (Neutral Mutations): The contextual noise sampled from p_{ctx} may also be biologically valid substitutions. In this regime, the ground truth $\mathbf{x}_0$ represents a specific instance (wild type), while the noise $\mathbf{x}_t$ represents a high-fitness neighbor (homologue). The denoising loss encourages the model to learn the *evolutionary dependencies* between the original functional sites $\mathbf{x}_0$ and variable sites $\mathbf{x}_t$.

A.1.5 THE LEARNABLE PRIOR

We discuss the sampling prior of the contextual noising kernel when $t = T$. In standard multinomial diffusion, the sampling prior is a uniform distribution $\mathcal{U}(\mathcal{V})$, which is a poor approximation of natural proteins. In our framework, the effective prior at T steps becomes the model's prediction given a fully masked sequence:

$$p(z_T) \approx p_\theta(\cdot \mid \mathbf{m}^L) \tag{4}$$

This **learnable prior** captures the natural background frequencies of amino acids and global sequence statistics (e.g., length distributions and domain compositions) inherent in the pre-trained weights. Consequently, the reverse generation process initializes from a biologically informed distribution rather than random chaos, improving sampling efficiency and stability.

A.1.6 OTHER ALTERNATIVE NOISING KERNEL: BLOSUM-INFORMED SUBSTITUTION

While the contextual evolutionary noising kernel provides dynamic, instance-specific noise, it incurs additional computational overhead due to the required forward pass of the model. For scenarios requiring high computational efficiency, we propose a static but biologically grounded alternative based on the BLOSUM substitution matrices.

Standard discrete diffusion typically uses a uniform transition matrix $\mathbf{U}_K$, which implies that all amino acid substitutions are equally probable. In contrast, the BLOSUM62 matrix encodes empirical substitution frequencies observed in homologous protein alignments. Let $\mathbf{B} \in \mathbb{R}^{K \times K}$ be the BLOSUM62 scoring matrix, where $\mathbf{B}_{ij}$ represents the log-odds score of substituting amino acid i with j.

We construct the static substitution noise matrix $\mathbf{M}_{\text{BLOSUM}}$ by applying a row-wise Softmax operation over the scaled scores:

$$[\mathbf{M}_{\text{BLOSUM}}]_{ij} = \frac{\exp(\mathbf{B}_{ij}/\tau)}{\sum_{k=1}^{K} \exp(\mathbf{B}_{ik}/\tau)} \tag{5}$$

where $\tau > 0$ is a temperature hyperparameter that controls the entropy of the noise distribution:

- As $\tau \to \infty$, the distribution approaches the uniform distribution $\mathbf{U}_K$.
- As $\tau \to 0$, the distribution collapses to the identity matrix (no mutation).
- At moderate τ, the distribution favors physico-chemically conservative mutations (e.g., $L \leftrightarrow I$) over radical changes (e.g., $L \leftrightarrow K$).

This static kernel can be directly plugged into our proposed evolutionary discrete diffusion framework by replacing the uniform component in the noise transition matrix. Although less expressive than the contextual noising kernel, it still has large potential to outperform uniform noise by respecting fundamental biochemical properties.

A.2 Connections to existing discrete diffusion

Our model can recover existing discrete diffusion models by manipulating coeffiients in $\mathbf{Q}_{\textbf{noise}}$. For example, when $\omega_{\text{del}} = 0$ and $\omega_{\text{ins}} = 0$, we disable indels and our model becomes a fixed-length sequence diffusion model. In such circumstances,

- if $\rho_{\text{mask}} = 1$, $\mathbf{Q}_{\textbf{noise}}$ will always transits any token into $\mathbf{m}$ and our model reduces to classicial masked diffusion (Sahoo et al., 2024; Shi et al., 2024).
- if $\rho_{\text{mask}} = 0$, $\mathbf{Q}_{\textbf{noise}}$ will transits each token into another random token, and our model reduces to classicial uniform diffusion (Austin et al., 2021; Schiff et al.).
- if $\rho_{\text{mask}} \in (0, 1)$, $\mathbf{Q}_{\textbf{noise}}$ will transits each token into either a random token or $\mathbf{m}$, and our model reduces to a generalized diffusion with mixed masked-unifrom noising paths (Austin et al., 2021; von Rütte et al., 2025).

With these connections, we can also initialize our model from pretrained masked diffusion-based models, efficiently reprogramming classical discrete diffusion to enable the full spectrum of sequence edit operations. We note that there are also recent work on variable-length diffusion/flow models (Havasi et al., 2025; Wu et al., 2025) for text generation, and Baron et al. (2025) learning to shrink protein sequences. We extend our discussion to other related work in §D.

A.3 Decomposed objectives and practical implementations

The deomposed training objectives. To make the training tractable, we should solve a critical issue: the diffusion is defined on the latent sequence $\mathbf{z}_t$, but in practice the neural network f_θ operates on the original sequence $\mathbf{x}_t = \Gamma^{-1}(\mathbf{z}_t)$, which is collapsed by $\mathbf{z}_t$. To bridge this gap, we define the *Index Mapping Function* $\mathcal{I} : \{1, \dots, L_t\} \to \{1, \dots, N\}$ such that $\mathcal{I}(k)$ is the index of the k-th non-gap token in the latent sequence $\mathbf{z}_t$. Then, we decompose the loss defined on the latent sequence $\mathbf{z}$ into three mutually exclusive components defined in the observed space $\mathbf{x}$, i.e., substitution loss, deletion loss and insertion loss, based on the token category between $\mathbf{z}_t$ (current noisy state) and $\mathbf{z}_0$ (ground truth).

To more clearly decouple the prediction of the three operations, we leverage separate and independent heads, p_θ^{sub}, p_θ^{del} and p_θ^{ins} for the substitution, deletion and insertion prediction for each token in the original sequence $\mathbf{x}_t$. We define the loss for the k-th token of the input sequence $\mathbf{x}_t$:

(1). Substitution Loss. It is active only when the input and target token are both valid amino acids:

$$\mathcal{L}_{\text{sub}}^{(k)} = \mathbb{I}_{(\mathbf{z}_0^{(\mathcal{I}(k))} \in \mathcal{V})} \cdot \mathbb{I}_{(\mathbf{z}_t^{(\mathcal{I}(k))} \in \mathcal{V})} \cdot \mathbb{I}_{(\mathbf{z}_0^{(\mathcal{I}(k))} \neq \mathbf{z}_t^{(\mathcal{I}(k))})} \cdot \text{CE}\left(\mathbf{z}_0^{(\mathcal{I}(k))}, p_\theta^{\text{sub}}(\cdot|\mathbf{x}_t)\right).$$

(2). Deletion Loss. It encourages the model to predict ϕ if the current token is noise when its target is a gapin $\mathbf{z}_0$:

$$\bar{\mathcal{L}}_{\text{del}}^{(k)} = \mathbb{I}_{(\mathbf{z}_0^{(\mathcal{I}(k))} = \phi)} \cdot \mathbb{I}_{(\mathbf{z}_t^{(\mathcal{I}(k))} \in \mathcal{V})} \cdot \text{CE}\Big(\mathbf{z}_0^{(\mathcal{I}(k))}, p_\theta^{\text{del}}(\cdot|\mathbf{x}_t)\Big).$$

(3). Insertion Loss. Let $v_{\text{next}}^{(k)}$ be the first non-gap token in $\mathbf{z}_0$ between indices $\mathcal{I}(k)$ and $\mathcal{I}(k+1)$. If no such token exists, i.e., there is no insertion needed between $\mathbf{x}_t^k$ and $\mathbf{x}_t^{k+1}$, the $v_{\text{next}}^{(k)}$ is $\emptyset$. The loss is calculated on the positions that need insertion for reconstruction:

$$\bar{\mathcal{L}}_{\text{ins}}^{(k)} = \mathbb{I}_{(v_{\text{next}}^{(k)} \neq \emptyset)} \cdot \text{CE}\left(v_{\text{next}}^{(k)}, p_\theta^{\text{ins}}(\cdot|\mathbf{x}_t)\right).$$

Practical considerations for $\mathcal{L}_{del}$ and $\mathcal{L}_{ins}$. In our prelimiary experiments, we find that training with the original $\bar{\mathcal{L}}_{\text{del}}$ imposes a significant risk of mode collapse, while $\bar{\mathcal{L}}_{\text{ins}}$ leads to unstable training.

We find that only training $\mathcal{L}_{\text{del}}$ to predict the ϕ token poses a significant risk of mode collapse. Without exposure to tokens that should *not* be deleted, the head may converge to a degenerate solution that always predicts the gap token ϕ for any input, resulting in excessive deletion during generation.

Therefore, we introduce negative samples that represents tokens that should be preserved for deletion training. Given that there are only two distinct prediction targets: either a gap token or retaining the original token, deletion is essentially a *binary classification* task. Therefore, we parameterize the deletion head as a binary prediction head, and define the binary deletion target $y_k^{\text{del}} = \mathbb{I}(\mathbf{z}_0^{(\mathcal{I}(k))} = \phi)$. The $\mathcal{L}_{\text{del}}$ can be formalized as the Binary Cross-Entropy (BCE) computed over all the positions:

$$\mathcal{L}_{\text{del}} = \sum_{k}^{|\Gamma^{(-1)}(\mathbf{z}_t)|} \text{BCE}(y_k^{\text{del}}, p_\theta^{\text{del}}(\cdot|\mathbf{x}_t)) \tag{6}$$

where $\text{BCE}(y, p) = -[y \log p + (1-y)\log(1-p)]$.

Meanwhile, we find that parameterize the insertion head as a binary classifier leads to more stable training. For each token position k in the observed sequence $\mathbf{x}_t$, the head predicts whether an additional token should be inserted immediately to the right of $\mathbf{x}_t^{(k)}$. We define the binary insertion target as $y_k^{\text{ins}} = \mathbb{I}(v_{\text{next}}^{(k)} \neq \emptyset)$, and train the head with a BCE objective:

$$\mathcal{L}_{\text{ins}} = \sum_{k}^{|\Gamma^{(-1)}(\mathbf{z}_t)|} \text{BCE}\left(y_k^{\text{ins}}, p_\theta^{\text{ins}}(\cdot|\mathbf{x}_t)\right). \tag{7}$$

When an insertion is triggered, we first insert a special mask token $\mathbf{m}$ as a noisy placeholder at that location, and then reuse the substitution head for filling the masked position.

B Sampling Details

B.1 Overview

The generation process of DPLM-Evo follows the standard iterative denoising paradigm of discrete diffusion models. Starting from a completely noisy sequence $\mathbf{x}_T$, which is initialized with mask tokens with length L, and sampled from the learned prior $p_\theta(\cdot|\mathbf{m}^L)$, the model iteratively refines the sequence by sampling from the reverse distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Formally, the reverse transition is derived by marginalizing over the predicted ground truth $\hat{\mathbf{x}}_0$:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \sum_{\hat{\mathbf{x}}_0} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_0) p_\theta(\hat{\mathbf{x}}_0|\mathbf{x}_t) \tag{8}$$

In practice, since summing over the entire sequence space is intractable, we approximate this process that we first sample a $\hat{\mathbf{x}}_0$ via $p_\theta(\cdot|\mathbf{x}_t)$, and then samples the previous state $\mathbf{x}_{t-1}$ using the tractable posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_0)$. According to Zheng et al. (2023a), this can be implemented with a predict-then-renoise strategy: first update a set of indices in $\mathbf{x}_t$ with $\hat{\mathbf{x}}_0$, then renoise the indices with low confidence from the noise distribution.

B.2 Evolutionary sampling with deletion, insertion and substitution

We maintain a noisy state $\mathcal{N}_t$ during sampling, which tracks the indices of tokens that are considered "noisy" at the current step t and will be updated for the next step $t-1$. The denoising process at each iteration step t proceeds through four steps.

Step 1: Deletion prediction. We first modify the sequence $\mathbf{x}_t$ by removing noise. The deletion head is applied to the indices in the noisy set $j \in \mathcal{N}_t$. If the model predicts deletion (i.e., $p_\theta^{\text{del}}(\mathbf{x}_t^j) > \tau_{\text{del}}$), the token is removed from the sequence. The noisy set $\mathcal{N}_t$ is updated to reflect the shifted indices of the remaining tokens.

Step 2: Insertion prediction. The insertion head scans the current noisy set. If an insertion is predicted at index j (i.e., $p_\theta^{\text{ins}}(\mathbf{x}_t^j) > \tau_{\text{ins}}$), a mask token $\mathbf{m}$ is inserted into the sequence at position $j+1$. Crucially, since these new tokens lack semantic content, their indices are immediately added to the noisy set $\mathcal{N}_t$.

Step 3: Substitution prediction, along with the insertion content. The substitution head makes prediction for all tokens, yielding a $\hat{\mathbf{x}}_0$ sampled from $p_\theta^{\text{subs}}(\cdot|\mathbf{x}_t)$ and the corresponding confidence score. We update the indices in the noisy set $\mathcal{N}_t$ of $\mathbf{x}_t$ with $\hat{\mathbf{x}}_0$, including the substitution prediction and inserted content prediction. Then, we update the noisy set for the next step, i.e., $\mathcal{N}_{t-1}$, by selecting

Algorithm 1 Evolutionary sampling with DPLM-Evo

Require: Trained prediction heads p_θ^{del}, p_θ^{ins} and p_θ^{subs}, Prior length L_{init}, Steps T

Initialize: $x_T \sim p_\theta(\cdot \mid \mu^{L_{\text{init}}})$, $\mathcal{N}_T \leftarrow \{1, \ldots, L_{\text{init}}\}$
for $t = T$ **down to** 1 **do**
  // Step 1: Deletion
  Predict deletion: $\mathcal{D} \leftarrow \{j \in \mathcal{N} \mid p_\theta^{\text{del}}(x_t^j) > \tau_{\text{del}}\}$
  $x_t \leftarrow \text{Delete}(x_t, \mathcal{D})$, Update indices in $\mathcal{N}_t$
  // Step 2: Insertion
  Predict insertion: $\mathcal{I} \leftarrow \{j \in \mathcal{N} \mid p_\theta^{\text{ins}}(x_t^j) > \tau_{ins}\}$
  $x_t \leftarrow \text{Insert}(x_t, \mathcal{I}, \text{token} = \mathbf{m})$
  $\mathcal{N}_t \leftarrow \mathcal{N}_t \cup \text{Indices}(\text{InsertedToken})$
  // Step 3: Substitution
  Sample $\hat{x}_0 \sim p_\theta^{\text{subs}}(\cdot|x_t)$
  for j in $\mathcal{N}_t$ **do**
    $x_t^{(j)} \leftarrow x_0^{(j)}$
  end for
  $k_t \leftarrow \text{Schedule}(t)$
  $\mathcal{N}_{t-1} \leftarrow \text{TopK\_Lowest\_Confidence}(c, k_t)$
  // Step 4: Renoise
  Sample $x_{t-1} \sim \pi_{\text{noise}}(\cdot|x_t)$ for indices in $\mathcal{N}_{t-1}$.
end for
Return: x_0

the $k_t\%$ tokens with the *lowest* confidence scores, where $k_t\%$ follows a linear decay schedule from 100% to 0%.

Step 4: Renoising. Finally, we perform renoising for the indices in $\mathcal{N}_{t-1}$. This also prevents the model from collapsing into local optima. For every index $j \in \mathcal{N}_{t-1}$, we sample $\mathbf{x}_{t-1}^{(j)}$ from the noise distribution. This noise distribution can be instantiated as the contextual evolutionary noising kernel (using the model's own predictions) or the BLOSUM-based kernel, ensuring that the noise state remains biologically plausible, which is consistent with the training.

The full procedure is summarized in Algorithm 1.

C Full Experimental Results

In this section, we evaluate DPLM-Evo across various understanding and generative tasks. First, we assess variant effect prediction to validate the model's understanding of protein evolution. Subsequently, we examine the model's generation capabilities, including unconditional generation (covering both substitution-only and full edit operations) and conditional motif scaffolding scenario. Finally, we demonstrate the potential application of DPLM-Evo in protein sequence optimization.

C.1 Variant Effect Prediction

Setup. Modeling the effect of sequence variation on function is fundamental for understanding and designing proteins. DPLM-Evo predicts variant effects using protein sequence only, without supervision from experimental data. **Unlike** the common masked-residue scoring pipeline used by masked language models (i.e., masking the residue(s) of interest and reading out the logits at the masked positions), DPLM-Evo is a substitution-based model that natively scores variants **without masking.** Instead, we directly input the wild-type sequence and evaluate the model's substitution distribution at the mutated site(s), which better matches the model design and avoids introducing an artificial mask token.

For a variant with mutation set T, similar to ESM-1v (Meier et al., 2021), we use a log-odds mutation score that compares the mutant residue to the wild-type residue:

$$\sum_{t \in T} \Big[\log p(\mathbf{x}_t = \text{mut} \mid \mathbf{x}) - \log p(\mathbf{x}_t = \text{wt} \mid \mathbf{x}) \Big],$$

where $\mathbf{x}$ denotes the wild-type sequence and $p(\mathbf{x}_t = \cdot \mid \mathbf{x})$ is the substitution probability at position t predicted by DPLM-Evo conditioned on the unmodified wild-type context. In contrast, masked-

residue approaches typically score mutations via

$$\sum_{t \in T} \Big[\log p(\mathbf{x}_t = \text{mut} \mid \mathbf{x}_{\setminus T}) - \log p(\mathbf{x}_t = \text{wt} \mid \mathbf{x}_{\setminus T}) \Big],$$

where $\mathbf{x}_{\setminus T}$ indicates that the mutated positions are masked/removed from the input; DPLM-Evo does not require this step.

Results. We evaluate the performance on the ProteinGym DMS substitution zero-shot benchmark (Notin et al., 2023) by calculating the correlation between DPLM-EVO's score and experimental fitness score across all 217 DMS assays. As shown in Fig.2 upper figure, ***(1)* DPLM-EVO *achieves the highest correlation score among all the single sequence foundation models for variant effect prediction in ProteinGym.*** DPLM-EVO outperforms the ESM model series, including ESM-2 (Lin et al., 2022), ESM-C (ESM Team, 2024), ESM-1v (Meier et al., 2021), and DPLM (Wang et al., 2024b). According to Fig.2(b), DPLM-EVO even surpasses ESM-IF1 (Hsu et al., 2022), which utilizes additional structure information, despite DPLM-EVO using a single sequence. Crucially, we observe that scaling up the model leads to further improvements, as evidenced by the 3B model outperforming the 650M model. This scalability stands in contrast to ESM-2, which exhibits a performance regression as model size increases (with ESM-2 3B underperforming ESM-2 650M by approximately 0.01 in correlation). We attribute this strong correlation to the model's evolutionary pretraining, which fundamentally enables it to learn mutation preferences from natural proteins, effectively capturing the constraints imposed by natural selection.

(2) Explicitly aligning with evolutionary kernel further unlocks the potential of* DPLM-EVO *in mutation effect prediction. We adopted the strategy proposed by VespaG Marquet et al. (2024) to explicitly align DPLM-EVO output distribution with GEMME (Laine et al., 2019), a state-of-the-art evolutionary-informed prediction model. Leveraging multiple sequence alignment information (Deorowicz et al., 2016), GEMME analyzes the evolutionary mutation sensitivity of individual sites, thereby providing a substitution distribution at each position. By aligning the substitution kernel of DPLM-EVO with that of GEMME, the scores correlate more closely with natural mutations. Illustrated in Fig.2B, this alignment yields further enhancements, outperforming SaProt (Su et al., 2023) that takes additional structure, TranceptEVE L with supplementary MSA, and the original VespaG method (based on ESM2-3B).

C.2 UNCONDITIONAL PROTEIN SEQUENCE GENERATION

Setup. We initialize DPLM-EVO from a pretrained DPLM-650M model (Wang et al., 2024b). Specifically, the backbone parameters (token embeddings and Transformer blocks) and the substitution head are initialized from DPLM, while the two binary operation heads for indel prediction (deletion and insertion) are randomly initialized. We train on the UniRef50 dataset for 100,000 steps, using 2,000 warmup steps to a peak learning rate of 10^{-4}, followed by linear decay to 0.1×10^{-4} by the end of training. The diffusion timestep is set to $T = 500$. For unconditional generation, we consider initial lengths $L_{\text{init}} \in \{100, 200, 300, 400, 500\}$.

Results. DPLM-EVO performs iterative denoising by jointly applying substitution, deletion, and insertion, enabling variable-length generation that more closely mirrors natural evolutionary trajectories. DPLM-EVO generation starts from corrupted sequences sampled from the learnable diffusion prior rather than all-mask initialization used in masked diffusion. Fig.3 demonstrates the evaluation results of unconditional generation in various perspectives: (1) ***Diversity and Foldability:*** Fig. 3A shows DPLM-EVO achieves consistent high foldability across length as measured by ESMFold pLDDT. Compared with DPLM based on masked diffusion, Fig.3D-E shows DPLM-EVO achieves comparable foldability while possessing greater generation diversity, reflected by a larger number of clusters in both sequence and structure. (2) ***Reduced Mode Collapse:*** DPLM-EVO produces higher sequence entropy than DPLM, as is shown in Fig.3F, indicating fewer repetition patterns and alleviating the mode collapse issue. (3) ***Effect of Evolutionary Kernel:*** Training with the *contextual evolutionary noising kernel* substantially outperforms uniform noising, as shown in Fig.3D. This indicates that biologically grounded corruptions encourages DPLM-EVO to learn more evolutionarily plausible substitution predictions, yielding higher-quality samples at generation time. (4) ***Length Control:*** Output lengths remain concentrated near their initial values without excessive expansion or collapse. The distribution is visualized in Fig.3B. This indicates that insertion and deletion prediction are invoked conservatively, resulting in a refinement process that prioritizes substitutions over drastic length changes.

To better understand how indel operations are scheduled over the diffusion trajectory, we probe the deletion and insertion heads across timesteps on a representative natural sequence (Fig.3C).

sample 50 candidates starting from the original sequence

Iter number	pLDDT		Clusters			
	mean/std	min/max	Seq Id 0.5	Seq Id 0.4	Seq Id 0.3	Struct TM 0.5
0 (original)	87.43	-/-	-	-	-	-
100	83.75/2.8	74.75/88.31	51	37	4	1

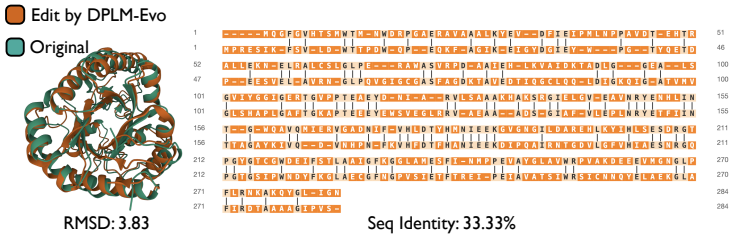


Figure 5: *Unconstrained in-silico family expansion by DPLM-Evo.* 50 candidates are sampled from the original sequence show an overall high pLDDT, structure consistency and sequence diversity. One case is visualized to show structure alignment with low RMSD below 4 and sequence alignment with low identity at 33%.

The predicted indel probabilities decrease as the timestep decreases towards clean, suggesting the model primarily uses indels for coarse adjustments during high-noise stages, and gradually shifts to fine-grained refinements later. This behavior indicates that indel operations can be manipulated through timestep control, e.g., fixing thedeletion timestep to 0 for insertion-only tasks.

C.3 Length-adaptive Scaffolding of Functional Motifs

Setup. Motif scaffolding aims to generate a protein scaffold for a given functional motif. We evaluate DPLM-Evo in *zero-shot* and *continued finetuning* settings. For finetuning, DPLM-Evo incorporates structural constraints for motif structure features, as illustrated in Fig.1C(4).

During generation, DPLM-Evo edits only the scaffold region and never modifies motif residues, allowing dynamic length adjustment to better accommodate the motif. In contrast, fixed-length sequence models require manually scaffold length enumeration and cannot revise length once an unsuitable initialization is chosen. For each motif instance, we sample 100 candidate scaffolds. Success is defined as pLDDT > 70, and motif RMSD < 1 Å.

Results. As shown in Fig.4, in the zero-shot setting, DPLM-Evo solves more motif problems than EvoDiff and DPLM-Mask, and achieves higher overall success rate (0.23). We attribute this to the capability for dynamic scaffolding length adjustment and evolutionarily plausible mutations provided by the substitution head. Continued finetuning brings further improvements, highlighting the importance of multimodal conditioning. Compared to multi-modal models like ESM-3 (Hayes et al., 2024), the finetuned DPLM-Evo achieves a higher overall success rate but resolves slightly less targets. We hypothesize this gap arises because DPLM-Evo only supports multimodal conditioning, without native end-to-end training for structural understanding. We leave multimodal evolutionary discrete diffusion modeling as an exciting direction for future work.

C.4 Case Study: In-silico Sequence Family Expansion

Setup. To assess whether DPLM-Evo can generate diverse yet structurally consistent relatives of a given protein, we perform unconstrained post-editing starting from natural sequences. Specifically, we randomly select sequences from the CAMEO dataset and let DPLM-Evo refine them without imposing explicit functional constraints. We evaluate both structural preservation relative to the starting sequence and sequence diversification.

Results. DPLM-Evo generates diverse, yet structurally similar protein sequences in the unconstrained optimization setting. **Structural preservation:** We find that DPLM-Evo preserves structural plausibility (evaluated via comparing predicted structure to wild type structure) and at the same time introduces substantial edits. While it does not necessarily increase the initial pLDDT, it effectively explores the sequence space around a given fold without catastrophic structural degradation. **Sequence diversification and family expansion:** Meanwhile, DPLM-Evo modifies a large proportion of the initial sequence(with sequence identity mostly below 50%). Fig.5 shows a case where the highly modified sequence still aligns structurally with the original. These results suggest that DPLM-Evo performs unconstrained sequence optimization that preserves a shared structural scaffold while

producing diverse sequences. This implies that DPLM-EVO captures latent regularities of natural proteins, including constraints related to fold and stability. In this way, the generated sequences can be potentially viewed as in silico expanded homologs of the starting protein, holding the potential for purely sequence based orpha protein understanding.

C.5 CASE STUDY: DIRECTED EVOLUTION OF GFP

Setup. We optimize the green fluorescent protein (GFP) via directed evolution using DPLM-EVO as illustrated in Algorithm 2.

Starting from the template, we adopt the beam search strategy to maintain a candidate set: in each iteration, 10 optimized sequences are generated for each sequence in the candidate set. In each step, we employ the Chai-1 model for filtering and structure scoring to keep only the top-scoring candidates retained for the next iteration. Following ESM3 (Hayes et al., 2024), The criteria for filter is that the template chromophore site RMSD is less than 1.5Å, while the scoring term is the pTM score produced by Chai-1.

Results. Fig.6A depicts the trajectory of optimization. We observed that as the iteration processes, the pTM value gradually increases, while the RMSD remained consistently below 1.5Å. After 20 iterations, the pTM increased from an initial 0.263 to 0.792 while a random mutation baseline converges below 0.6. Fig.6B and Fig.6C visualize the structures of the GFP before and after optimization, respectively. Residues are colored according to its pLDDT value, and indicate a significant increase in stability. These results prove that DPLM-EVO can leverage the priors learned from evolution-scale protein sequences to optimize the GFP sequence toward greater overall structural stability, while preserving the structure of the chromophore site to maintain its fluorescence effect.

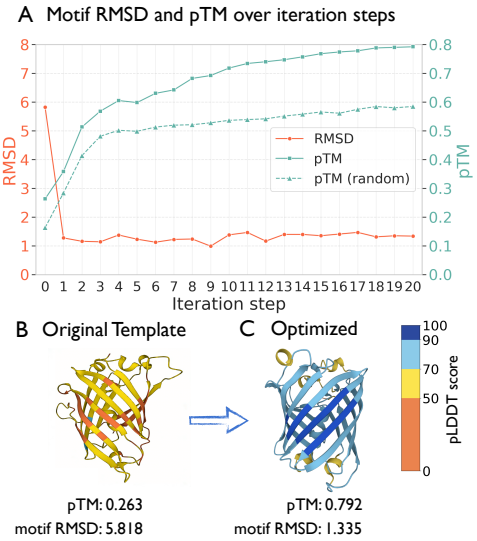


Figure 6: *A case study of sequence optimization: directed evolution for GFP. (A) The motif RMSD and pTM scores are monitored through time step.* DPLM-EVO *maintains low motif RMSD while boosting pTM to over 0.8, compared to a random mutation baseline having pTM capped at 0.6. (B-C) Visualization of the optimization starting template and the final result.*

D RELATED WORK

Discrete diffusion model are diffusion models operating in discrete state space (Sohl-Dickstein et al., 2015; Austin et al., 2021). They noises samples with discrete transistion probabilities and learn to denoise them iteratively, in comparison to their continuous counterpart using continuous distribution such as Gaussian kernels (Ho et al., 2020; Song et al., 2020a;b). Various transition kernels have been explored, typically uniform transitions (Austin et al., 2021; Sahoo et al.) and masking (He et al., 2023; Ye et al., 2023a; Zheng et al., 2023a; Sahoo et al., 2024; Shi et al., 2024; Nie et al., 2024). Among the variants, masked diffusion has attracted the most recent interest for their simplicity (Zheng et al., 2023a; Ou et al., 2024), scalability (Ye et al., 2023a; Nie et al., 2024), and empirical effectiveness, gaining success as protein language models (Zheng et al., 2023b; Wang et al., 2024b;a; Hsieh et al., 2025) and large language models (Ye et al., 2023a; Nie et al., 2025; Ye et al., 2025; Gong et al., 2025).

Flexible length generative model. Pre-deciding the length of a answer to many problems can be hard due to the uncertain computation budget and answer shapes required. While autoregressive models, which iteratively produce one token each step, naturally provide flexible length generation, they peforms inferiorly in generating data without fixed-order structure such as protein sequences (Zheng et al., 2023b). Non-autogressive geneartive models (*e.g.*, diffusion models), on the other hand, does not require presumtion on the orders but mostly require preset or predicted answer lengths (Guo et al., 2019; Gu & Kong, 2021). To address this, researchers have explored enabling non-autoregressive models to vary the output lengths by introducing indels operations in their predictions, which are also referred to as edit-based models. As early attempt, Levenshtein Transformer (Gu et al., 2019)

studies non-autoregressive machine translations and show edit-based models perform on par with autoregressive models while showing much lower sampling latency thanks to parallel generation. Later progress revisit edit-based models under the formulation of diffusion models (Reid et al., 2022). More recently, DreamOn (Wu et al., 2025) introduces a large diffusion language model with indels as speical tokens in vocabulary to vary the length during sampling, tailored for coding tasks. EditFlow (Havasi et al., 2025), most related to our work, takes a flow-based model perspective to construct indels training signals by aligninig the noisy samples, which are interpolations between clean samples and noises, with ground truth target. Although our work also extend fixed-length diffusion to supports indels, we highlight the perspective of diffusion transition kernels and their evolutionary significance.

Protein language model. Motivated by the success of large language models (LLMs), similar practice has been extended to the development of protein language models. ESM-1b Rives et al. (2019) utilizes self-supervised masked language modeling on 250 million protein sequences spanning evolutionary diversity, later leading to the development of ESM-2 Lin et al. (2023) scales further. ProtTrans Elnaggar et al. (2021), ProteinBERT Brandes et al. (2022), PRoBERTa Nambiar et al. (2020), ProtAlbert Behjati et al. (2022), TAPE Rao et al. (2019), ProteinLM Xiao et al. (2021), and CARP Yang et al. (2022) involve several other representative masked language modeling (MLM) paradigm. These sequence-based PLMs perform competitively with classic methods that rely on multiple sequence alignments, indicating that PLMs have captured some of the evolutionary information from sequences alone. In particular, these protein language models achieve powerful generalization on various downstream tasks involving the secondary and tertiary structures. Recent findings further showcase their capabilities in predicting protein functions Meier et al. (2021), structure folding Lin et al. (2023), and de novo designs Verkuil et al. (2022). Beyond representation learning, DPLM (Wang et al., 2024b) unlocks the generation capabilities of protein MLM through scalable discrete diffusion training process (Ye et al., 2023b;a; Zheng et al., 2023b), enabling generating high-quality protein sequences. Buiding upon DPLM, DPLM-2 (Wang et al., 2024a) and DPLM-2.1 (Hsieh et al., 2025) further enhance the model with multimodal understanding and generation capablities. The series lay the foundation of pretrained models and diffusion algorithms for ours.

E GFP OPTIMIZATION PIPELINE

Algorithm 2 GFP Directed Evolution by DPLM-EVO

Initialization: Starting from the GFP template sequence, add it to the candidate set $\mathcal{C}$.
Hyperparameters:
  max iteration $T = 20$
  search width $w = 100$
  beam size $b = 10$
for $i = 1, \ldots, T$ **do**
  Generate mutated sequences: For each sequence in $\mathcal{C}$, generate w mutated sequences (one position is mutated at a time).
    Obtain a total of $|\mathcal{C}| \times w$ samples.
  Filter candidates: Filter the generated samples by {Common Filters}.
  Sort candidates: Sort the filtered candidates according to {Common Score Terms}.
  Update candidate set: Select the sequences with $\mathrm{top}\ b$ scores as the next-iteration candidate set $\mathcal{C}$.
end for
Return: The final candidate set $\mathcal{C}$