GLID^{2}E: A GRADIENT-FREE LIGHTWEIGHT FINE-TUNE APPROACH FOR DISCRETE SEQUENCE DESIGN

Hanqun Cao[1]* **Haosen Shi**[1]* **Chenyu Wang**[2] **Sinno Jialin Pan**[1] **Pheng-Ann Heng**[1]†

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2]CSAIL, Massachusetts Institute of Technology
{hqcao23, hsshi23, sinnopan, pheng}@cse.cuhk.edu.hk,
wangchy@mit.edu

ABSTRACT

The design of biological sequences is essential for engineering functional biomolecules that contribute to advancements in human health and biotechnology. Recent advances in diffusion models, with their generative power and efficient conditional sampling, have made them a promising approach for sequence generation. To enhance model performance on limited data and enable multi-objective design and optimization, reinforcement learning (RL)-based fine-tuning has shown great potential. However, existing fine-tuning methods are often unstable in discrete optimization when not using gradients or become computationally inefficient when relying on gradient-based approaches, creating significant challenges for achieving both control and stability in the tuning process. To address these issues, we propose GLID^{2}E, a gradient-free RL-based tuning approach for discrete diffusion models. Our method introduces a clipped likelihood constraint to regulate the exploration space and reward shaping to better align the generative process with design objectives, ensuring a more stable and efficient tuning process. By integrating these techniques, GLID^{2}E mitigates training instabilities commonly encountered in RL and diffusion-based frameworks, enabling robust optimization even in challenging biological design tasks. In DNA sequence design system, GLID^{2}E achieves competitive performance in function-based design while ensuring lightweight and efficient tuning.

1 INTRODUCTION

Designing biological sequences with specific functional or structural properties, such as foldability and biological activity, is a critical task in computational biology (Abramson et al., 2024; Dauparas et al., 2022). While diffusion-based generative models have demonstrated strong capability in capturing the data distribution of structures and sequences, adapting these models for controllable and task-specific sequence design remains an open challenge (Ho et al., 2020; Song et al., 2020; Campbell et al., 2022; Lou et al., 2023; Xu et al., 2024). Existing approaches to fine-tuning diffusion models in discrete spaces primarily rely on the reward evaluated at the final states of the generative process as supervision signals (Rector-Brooks et al., 2024; Wang et al., 2024). DRAKES (Wang et al., 2024) applies Gumbel-Softmax to mask discrete trajectories generated by diffusion models differentiable, enabling direct back-propagation of rewards. Besides, it incorporates KL regularization to ensure that the optimized sequences achieve high reward values while remaining consistent with the pretrained model, preventing overfitting or distributional drift. However, this approach can be computationally demanding, as it requires storing intermediate states across multiple timesteps. Furthermore, it does not fully utilize information from intermediate sampling states, which could otherwise contribute to greater flexibility and controllability in fine-tuning. These issues are further exacerbated in discrete spaces, where gradient discontinuities pose additional optimization challenges.

*Equal Contribution.
†Corresponding Author.

To address these limitations, we propose GLID^{2}E, a **G**radient-free **LI**ghtweight fine-tuning framework for **D**iscrete sequence **DE**sign that reformulates the fine-tuning process as a policy optimization problem. Our method introduces value approximation to efficiently guide the generation process and incorporates intermediate scoring mechanisms to enable fine-grained control at arbitrary time steps. This eliminates the need for expensive back-propagation through the entire generative trajectory, significantly reducing computational overhead while maintaining high-quality and controllable sequence generation. Moreover, our framework improves related properties while optimizing a single objective, enhancing its effectiveness in complex tasks like biological sequence engineering. Additionally, it has the potential to be extended for multi-objective optimization, enabling the simultaneous optimization of competing properties. We validate the effectiveness of our method through extensive experiments on two distinct biological sequence systems. Results demonstrate that our framework achieves state-of-the-art controllability and generation quality while substantially reducing computational costs compared to existing approaches. Furthermore, ablation studies show the contributions of key components in our framework and the superiority in efficiency and flexibility.

2 Related Work

Diffusion models for discrete sequence modeling Diffusion models have recently been extended from continuous domains to discrete domains, such as text and biological sequence generation (Li et al., 2022; Lou et al., 2023; Wu et al., 2024; Wang et al.; Huang et al., 2024). These extensions enable diffusion models to handle discrete data by adapting the denoising and sampling processes, making them applicable to tasks like text generation and biological sequence design. However, applying diffusion models to discrete spaces introduces two key challenges. First, unlike continuous diffusion models that rely on Gaussian noise, discrete diffusion models require a discrete noising process with categorical state transitions, making it challenging to design a well-calibrated transition mechanism and ensure effective learning. Recent works address this by introducing transition matrices or Markovian corruption processes to model discrete noise (Austin et al., 2021; Hoogeboom et al., 2021) and applying auto-regressive sampling to avoid complex categorical transition design (Wu et al., 2023). Second, efficient sampling remains a major bottleneck. Continuous diffusion models benefit from well-established solvers, whereas discrete models often require iterative categorical sampling, which can be computationally expensive. To mitigate this, techniques such as accelerated sampling strategies and partial noising approaches have been proposed to reduce the number of denoising steps while maintaining generation quality (Chen et al.; Song et al., 2023).

Reinforcement-learning in generative modes Reinforcement learning (RL) has become a powerful tool for optimizing generative models, particularly when dealing with non-differentiable objectives or trade-offs across multiple constraints (Uehara et al., 2024b; Clark et al., 2023; Fan et al., 2024). In text generation, RL has been used to enhance fluency and semantic consistency (Ouyang et al., 2022; Yang et al., 2024; Guo et al., 2025). Similarly, RL has shown promise in improving functionality and stability in protein design and other biological sequences, leveraging reward-driven optimization akin to its success in text-based tasks.

The key to conditional biological sequence design is ensuring controllability and generation quality. Recent works have explored fine-tuning and training-free guidance strategies to optimize diffusion-based generation (Wu et al., 2024; Zhou et al., 2024; Lisanza et al., 2024; Gruver et al., 2024; Wang et al.; Yi et al., 2024). Fine-tuning allows generative models to specialize in specific functional objectives, while training-free guidance techniques enable generation control without requiring extensive retraining. However, maintaining stability during conditional sampling remains challenging, as models often struggle to balance trade-offs between generation quality and control (Epstein et al., 2023; Bar-Tal et al., 2023). Recent studies have combined policy gradient methods with generative models to guide sample generation for tasks such as molecular activity and protein stability optimization (Wang et al., 2024). In this context, RL-based approaches provide a promising direction for enhancing controllability in discrete diffusion models through reward-driven optimization. Nevertheless, applying these methods in high-dimensional discrete spaces can be challenging, particularly in terms of efficiency and optimization stability.

3 PRELIMINARY

Our goal is to map a known and easy-to-access prior distribution $p_{\text{data}(x)} \in \Delta(\mathcal{X})$ into a distribution $p^*(x) \in \Delta(\mathcal{X})$, where $\mathcal{X} \subseteq \{1, \ldots, N\}^n$ is a discrete domain and Δ is the simplex over the domain. For a given reward model $r(x)$, $p^*(x)$ assigns higher probabilities to samples x for which $r(x)$ has a higher value. This work considers the scenario in which a diffusion model has been pre-trained on large-scale real data and a reward model for evaluating the value we focus on for any given data. This pre-trained diffusion model can generate data that is close to real data; however, it cannot ensure that the generated data has a high value. The reward model can approximately assess the value of the given data; however, it may give some invalid data the highest reward.

For the problems we study, protein and DNA sequence generation, the continuous-time Markov chain (CTMC) is used to model a series of distributions so that a given sequence is transformed into a special mask sequence from $t = 0$ to $t = T$ under a time-dependent transitions matrix $Q(t)$. The transition matrix $Q(t)$ is typically designed manually. It is computationally straightforward and guarantees that the sequence x_T at the time T is a sequence of the special MASK tokens. For the corresponding time-reversal CTMC $\frac{dx_{T-t}}{dt} = \bar{Q}^\theta(T-t)x_{T-t}$, a deep diffusion model is trained to approximate the reverse-time transition matrix $\bar{Q}^\theta(T-t)$. Subsequently, the original data distribution can be retrieved by time-reversal CTMC from this special sequence according to the trained diffusion model. For convenience of presentation, we often regard the original pre-trained diffusion model as an unconditional model whose input is a vector of fixed length and does not consider the length of the vector explicitly.

4 METHOD

In this section, we explore two issues that may be encountered when directly applying reinforcement learning methods for fine-tuning discrete diffusion models. The detailed algorithm is shown in A.3.

4.1 CLIPPED LIKELIHOOD CONSTRAIN

Due to the imperfections of the reward model, directly using the reward model as the optimization objective in reinforcement learning can lead to issues such as unstable training, over-optimization, and policy collapse Clark et al. (2023); Uehara et al. (2024a). Recent research works Wang et al. (2024) employ regularization terms to encourage the optimized policy close to a reference policy, and KL divergence is frequently used for this. The optimization policy is obtained by solving

$$\max_\theta \mathbb{E}_{x\sim p_\theta}\left[r(x)\right] - \beta KL(p_\theta, p_{prior}),$$

where $r(x)$ is the reward model. Assuming that the model has sufficient expressive ability, the optimal solution $p_{\theta^*}(x)$ to the above optimization problem satisfies the form $p_{\theta^*}(x) \propto p_{prior}(x)\exp(\frac{1}{\beta}r(x))$, but in general this posterior is intractable and hard to approximate.

The optimal policy can be seen as a product of the prior distribution and a reward-induced distribution, modulated by the KL regularization hyperparameter. A large regularization hyperparameter may cause the final policy to generate conservative samples close to those from the prior distribution, with consistently lower rewards. This happens because the prior distribution is often trained on task-agnostic data. Conversely, a small regularization hyperparameter may cause the policy to generate unreasonable samples due to flaws in the reward function.

The analysis of optimal policy formulation reveals two observations. First, the pre-trained diffusion models possess a superior ability to evaluate the validity of generated samples, as they were trained on an extensive real-world dataset that effectively captures fundamental rationality criteria. Second, over-reliance on prior distributions may limit the exploration space of maximizing task-specific reward functions, especially since prior distributions inherently lack task-specific inductive biases. This raises a question: *Could the pre-trained diffusion model alone sufficiently enforce sample rationality constraints? Furthermore, if good rationality could be guaranteed through such constraints, would the reward function-induced distribution dominate the optimal policy?*

A natural idea is to use the magnitude of the marginal distribution in the pre-trained model to assess whether the generated sample is reasonable. However, accurately estimating the marginal distribu-

tion is unrealistic for the diffusion model. A heuristic criterion is proposed to determine whether the generated samples are reasonable relative to the prior distribution, namely, checking whether the estimated upper bound of the log-likelihood of diffusion models exceeds a constant.

To select the constant, we use the pre-trained diffusion model to generate a large number of samples, and then calculate the mean and variance of their approximate log-likelihood. We choose the value of the constant as the mean minus $k = 1$ times the standard variance, detailed in (1). This approach is inspired by the fact that a diffusion model pre-trained on a large scale can generate a rich enough set of reasonable samples with few repetitions. In other words, it generates samples that the prior distribution deems reasonable with a high probability. Therefore, determining the rationality of samples by identifying outliers beyond the standard deviation can effectively indicate the confidence level of the diffusion model.

Finally, due to the flexibility of the reinforcement learning problem setting, the heuristic can be naturally introduced by the modified reward model. We use standardization to transfer this intuition to the reward function to avoid the difference in the absolute value of the log-likelihood over different pretrained diffusion models and tasks, which is

$$\bar{r}(x) = r(x) + \beta \ \min\left(\frac{\log p_{prior}(x) - \mu}{\sigma} + k, 0\right), \tag{1}$$

where μ and σ are the mean and standard deviation of the log-likelihood and β is a hyperparameter to control the strength of the penalty. The diffusion strategy is then fine-tuned without any KL regularization. We use PPO (Schulman et al., 2017) with GAE (Schulman et al., 2015) and an entropy penalty as a standard RL optimization policy, and the clipping ratio of PPO is set to 0.05.

4.2 Reward shaping

To encourage the RL policy to explore high-value and reasonable solutions, we adopt reward shaping as an additional information source to guide the training process. Reward shaping is a well-studied technique in RL, which aims to guide an agent towards the desired behavior in a more effective manner Ng et al. (1999). One common approach is potential-based reward shaping. A potential function $\Phi(s)$ is defined over the state space S of the environment. The reward at a state s is then modified as $r(s) = \mathrm{r_o}(s) + \gamma\Phi(s') - \Phi(s)$, where $\mathrm{r_o}(s)$ is the original reward in state s, s' is the next state and γ is the discount factor. Recall our RL formulation following Wang et al. (2024), the state space constitutes the set of all sequences with a length of t. The diffusion model conducts a single denoising action at each discrete time step. The reward is 0 for all non-final state transitions, i.e. $r(s_t) = 0, \forall t \in \{0, \ldots, T-1\}$. Until all time steps are concluded, a reward generated by the reward function is received at the final step as $r(s_T) = \bar{r}(x_T)$. The discount factor is 1.0. For states earlier in time, it is more difficult to estimate the final reward due to the randomness of the policy and the complex state transitions. Adding extra rewards at each step can guide the policy in choosing better actions early on. A natural choice is to reuse the trained reward function as a hint. However, due to the internal structure and calculation mechanism of the reward function, for a partial sequence that has MASK tokens, the reward function cannot be processed directly. This is because the MASK tokens are used for training the diffusion model and are outside of the reward model training. To address this issue, we introduce a new concept - the indicator sequence. Here we define the best possible sequence without MASK token as an indicator sequence x_b. We design our potential function as

$$\Phi(s_t) = \begin{cases} r(x_t^b) & 1 \le t \le T \\ 0, & other \end{cases}, \tag{2}$$

where $x_t^b = \arg\max_{x \neq \mathrm{MASK}} \ p_\theta(x|x_{t-1}, t)$, and the potential function on the final state is 0, i.e. $r'(s_{T-1}, a, s_T) = \bar{r}(s_T) - \Phi(s_{T-1})$. This function design determine that for all $(s_0, a_0, \ldots, s_{T-1}, a_{T-1}, s_T = x)$, the sum $\sum_{i=1}^{T} r(s_{i-1})$ is equal to $\bar{r}(x)$. As the denoising process progresses, the number of MASK tokens in the sequence gradually decreases. The closer the distance between the indicator sequence and the true sequence is, the more accurate the estimation of the reward function will be. This information can be delivered to the policy via the potential function at these early states to encourage the policy to learn more effectively.

Table 1: General performance for DNA sequence design models. State-of-the-art performance is **bold**, and the second-highest performance is underlined. KL, M1, and M2 denote KL regularization, reward shaping, and likelihood penalty, respectively.

Method	Pred-Activity (median)↑	ATAC-Acc↑ (%)	3-mer Corr↑	Log-Lik (median)↑
Pretrained	0.17(0.04)	1.5(0.2)	-0.061(0.034)	-261(0.6)
CG	3.30(0.00)	0.0(0.0)	-0.065(0.001)	-266(0.6)
SMC	4.15(0.33)	39.9(8.7)	0.840(0.045)	-259(2.5)
TDS	4.64(0.21)	45.3(16.4)	0.848(0.008)	-257(1.5)
CFG	5.04(0.06)	92.1(0.9)	0.746(0.001)	-265(0.6)
DRAKES w/o KL	6.44(0.04)	82.5(2.8)	0.307(0.001)	-281(0.6)
DRAKES	5.61(0.07)	**92.5(0.6)**	**0.887(0.002)**	-264(0.6)
GLID^{2}E	**7.35(0.067)**	90.6(0.26)	0.49(0.074)	-239.889(14.2)
GLID^{2}E w/o M1	2.57(0.60)	0.63(0.3)	0.473(0.078)	**-239.12(10.07)**
GLID^{2}E w/o M2	6.62(0.42)	67.3(39.4)	0.458(0.009)	-244.65(21.5)

5 EXPERIMENT

Dataset We apply the same dataset and oracle as DRAKES (Wang et al., 2024), where the pre-training dataset consists of approximately 700k DNA sequences (200 bp each) and the oracles' dataset is split into two subsets for different reward oracle training.

Metric Four metrics—Pred-Activity, ATAC-Acc, 3-mer Corr, and Log-Lik—are applied to the model's performance in enhancer activity, chromatin accessibility, sequence similarity, transcription factor binding site correlations, and the naturalness of generated sequences.

Baseline The benchmark baselines include controlled generation methods for discrete diffusion, including conditional guidance, classifier-free guidance, and RL-based fine-tuning methods.

5.1 DNA SEQUENCE DESIGN

Benchmark analysis Our method achieves superior enhancer activity compared to other approaches. By effectively balancing exploration and exploitation, it identifies sequences with higher functional potential while maintaining biological plausibility. Reinforcement learning enables efficient optimization, allowing the model to explore diverse sequence spaces without excessive deviation from natural distributions. Despite optimizing for enhancer activity, our method ensures that generated sequences remain biologically plausible. Likelihood constraints prevent the model from overfitting to the reward function, reducing the risk of generating unrealistic sequences. This balance allows for performance improvements while preserving key sequence characteristics. Our method explores a broader sequence space compared to baselines, occasionally deviating from natural DNA motifs to discover high-activity variants. However, the likelihood constraint mitigates excessive divergence, ensuring that generated sequences remain within a biologically reasonable range.

Ablation study Removing reward shaping (M1) significantly decreases enhancer activity and sequence accessibility. Without M1, the model struggles to focus on high-value regions of the sequence space, resulting in suboptimal optimization. Reward shaping is essential for guiding the reinforcement learning process toward functionally relevant solutions. Likelihood constraints (M2) help maintain sequence plausibility. Without M2, the optimization process prioritizes reward maximization without regard for biological validity, leading to sequences that deviate from natural distributions. This constraint ensures that improvements in enhancer activity do not come at the cost of generating unrealistic sequences. M1 and M2 work together to balance optimization and plausibility. M1 directs the search toward high-reward sequences, while M2 prevents excessive deviation from natural DNA distributions. Removing either component reduces performance, underscoring their complementary roles in reinforcement learning-based sequence design.

6 CONCLUSION

We propose GLID^{2}E, a gradient-free reinforcement learning (RL)-based tuning method designed for biological sequence generation. Unlike gradient-dependent approaches, GLID^{2}E leverages clipped likelihood constraints and reward shaping techniques to effectively address the high computational cost and instability issues associated with previous methods. These innovations make it a lightweight yet robust solution for conditional discrete generation tasks. In experiments on DNA sequence activity optimization, GLID^{2}E demonstrates promising potential for multi-objective function-based design across different biological systems. This highlights its flexibility and effectiveness in tackling diverse challenges in biological sequence generation. Future work will explore broader applications of the proposed framework across more biological sequence systems. With its computational efficiency and versatility, GLID^{2}E provides a promising foundation for advancing discrete sequence design in both biological and other domain-specific contexts.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

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

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18 (10):1196–1203, 2021.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

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

Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *Advances in Neural Information Processing Systems*, 36:16222–16239, 2023.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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.

Han Huang, Ziqian Lin, Dongchen He, Liang Hong, and Yu Li. Ribodiffusion: tertiary structure-based rna inverse folding with generative diffusion models. *Bioinformatics*, 40(Supplement_1): i347–i356, 2024.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel WK Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J Hendel, Miriam K Simma, Ge Liu, Muna Yase, Hongwei Wu, et al. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature biotechnology*, pp. 1–11, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. 2023.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287, 1999.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Zachary Quinn, Chenghao Liu, Sarthak Mittal, Nouha Dziri, Michael Bronstein, Yoshua Bengio, Pranam Chatterjee, et al. Steering masked discrete diffusion models via discrete denoising posterior prediction. *arXiv preprint arXiv:2410.08134*, 2024.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32211–32252. PMLR, 2023.

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024a.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024b.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-tuning discrete diffusion models via reward optimization with applications to dna and protein design. *arXiv preprint arXiv:2410.13643*, 2024.

Xinyou Wang, Zaixiang Zheng, YE Fei, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *Forty-first International Conference on Machine Learning*.

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

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *arXiv preprint arXiv:2410.21357*, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Kai Yi, Bingxin Zhou, Yiqing Shen, Pietro Liò, and Yuguang Wang. Graph denoising diffusion for inverse protein folding. *Advances in Neural Information Processing Systems*, 36, 2024.

Bingxin Zhou, Lirong Zheng, Banghao Wu, Kai Yi, Bozitao Zhong, Yang Tan, Qian Liu, Pietro Liò, and Liang Hong. A conditional protein diffusion model generates artificial programmable endonuclease sequences with enhanced activity. *Cell Discovery*, 10(1):95, 2024.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

The hyper-parameters used for table 1 are listed in table 2. The neural networks are based on the setting from (Wang et al., 2024). The value network uses Enformer(Avsec et al., 2021) architecture and is changed from the reward oracles from (Wang et al., 2024) and initialized from the pre-trained reward oracles weights open published by Wang et al. (2024). The policy network is initialized via the pretrained masked discrete diffusion model open published by Wang et al. (2024). For each epoch, we generated 1024 samples with 128 discrete sampling steps and formed the replay buffer for mini-batch training. To stabilize the training process and final performance, we utilized a cosine learning rate scheduler and decayed the learning rate to 1e-8 at the last epoch. For evaluation purposes, we generate 640 sequences per method. This is achieved by using a batch size of 64 over 10 batches. For each random seed, this process is repeated. We report the mean and standard deviation of model performance across three random seeds.

Table 2: Hyper-parameters used in the training process. This table lists the key training settings, including learning rate, optimizer, scheduler, and other important parameters such as entropy penalty, gradient clipping, and Generalized Advantage Estimation (GAE) parameters. The highlighted row indicates the entropy penalty term, which helps regulate policy entropy during training.

Training Setting	
Learning rate	1×10^{-4}
Optimizer	Adam
Scheduler	Cosine scheduler
Number of epochs	1000
Batch size	128
Total number of steps	128
Entropy penalty	1×10^{-3}
ϵ	0.05
Gradient clip	1.0
Clipped Likelihood Constraint	
β	1.0
GAE	
λ	0.95
γ	1.0

A.2 TRAINING TIME COMPARISON

We compared the training times of DRAKES and GLID^{2}E, as shown in 3. The Gradient-Free setting enhances the algorithm's efficiency.

Method	DRAKES	GLID^{2}E (Ours)
Time	23.28 ± 0.14	13.54 ± 0.04

Table 3: Training time per epoch for different methods.

A.3 DETAILED TRAINING ALGORITHM OF GLID^{2}E

Algorithm 1 GLID^{2}E Training Algorithm based on PPO algorithm

Input: Policy network p_θ, Value network V_ϕ, Training epochs K, Advantage estimate $\hat{A}_t$, Clipping parameter ϵ
Initialization: Policy network parameters θ and value network parameters ϕ
while Termination condition is not met **do**
    Collect N trajectories $\tau_i = (s_{i,t}, a_{i,t}, \log p(s_{i,t}))_{t=0}^{T}$, where $i = 1, \ldots, N$
    Calculate indicate states $x_{i,t}^b$
    Calculate reward $r_{i,t} = \Phi(x_{i,t+1}^b) - \Phi(x_{i,t}^b)$, where $t = 0, \ldots, T-1$
    and $r_{i,T} = r(x_{i,T}) - \Phi(x_{i,T}^b) + \beta \min\left(\frac{\log p_{prior}(x_{i,T}) - \mu}{\sigma} + k, 0\right)$
    Compute advantage estimates $\hat{A}_{i,t}$ for each trajectory via GAE
    for $k = 1$ to K **do**
        for Each mini-batch B containing M samples **do**
            Compute the policy loss $\mathcal{L}_{CLIP}(\theta)$

$$\mathcal{L}_{CLIP}(\theta) = -\hat{\mathbb{E}}_t\left[\min\left(\frac{p_\theta(a_t|s_t)}{p_{\theta_{old}}(a_t|s_t)}\hat{A}_t, \text{clip}(\frac{p_\theta(a_t|s_t)}{p_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$

            Compute the value loss $\mathcal{L}_{VF}(\phi)$

$$\mathcal{L}_{VF}(\phi) = \hat{\mathbb{E}}_t\left[\left(V_\phi(s_t) - \hat{V}_t\right)^2\right]$$

where $\hat{V}_t$ is the estimated value
            Compute the entropy bonus $\mathcal{S}(\theta)$

$$\mathcal{S}(\theta) = \hat{\mathbb{E}}_t\left[\mathcal{H}\left(\pi_\theta(\cdot|s_t)\right)\right]$$

where $\mathcal{H}$ is the entropy function
            Compute the total loss $\mathcal{L}(\theta, \phi)$

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{CLIP}(\theta) + \mathcal{L}_{VF}(\phi) - c_1\mathcal{S}(\theta)$$

where c_1 is a hyperparameter
            Update policy network parameters θ and value network parameters ϕ:

$$\theta \leftarrow \theta - \alpha_\theta \nabla_\theta \mathcal{L}(\theta, \phi)$$

$$\phi \leftarrow \phi - \alpha_\phi \nabla_\phi \mathcal{L}(\theta, \phi)$$

where α_θ and α_ϕ are learning rates
        end for
    end for
end while