Latent Diffusion Models for Controllable RNA Sequence Generation

Kaixuan Huang[1*] **Yukang Yang**[1*] **Kaidi Fu**[2♮] **Yanyi Chu**[3] **Le Cong**[3] **Mengdi Wang**[1†]
[1]Princeton University [2]Tsinghua University [3]Stanford University

Abstract

This work presents `RNAdiffusion`, a latent diffusion model for generating and optimizing discrete RNA sequences of variable lengths. RNA is a key intermediary between DNA and protein, exhibiting high sequence diversity and complex three-dimensional structures to support a wide range of functions. We utilize pretrained BERT-type models to encode raw RNA sequences into token-level, biologically meaningful representations. A Query Transformer is employed to compress such representations into a set of fixed-length latent vectors, with an autoregressive decoder trained to reconstruct RNA sequences from these latent variables. We then develop a continuous diffusion model within this latent space. To enable optimization, we integrate the gradients of reward models—surrogates for RNA functional properties—into the backward diffusion process, thereby generating RNAs with high reward scores. Empirical results confirm that `RNAdiffusion` generates non-coding RNAs that align with natural distributions across various biological metrics. Further, we fine-tune the diffusion model on mRNA 5' untranslated regions (5'-UTRs) and optimize sequences for high translation efficiencies. Our guided diffusion model effectively generates diverse 5'-UTRs with high Mean Ribosome Loading (MRL) and Translation Efficiency (TE), outperforming baselines in balancing rewards and structural stability *trade-off*. Our findings hold potential for advancing RNA sequence-function research and therapeutic RNA design.

1 Introduction

Diffusion models demonstrate exceptional performances in modelling continuous data, with applications in images synthesis [98, 100, 16], point clouds generation [92], video synthesis [58], reinforcement learning [3, 62, 79], time series [112] and molecule structure generation [122]. An important advantage of diffusion models is that their generation process can be "controlled" to achieve specific objectives via incorporating additional *guidance* signal. The guidance can steer the backward process toward generating samples with desired properties, without additional training [34, 17, 31].

While diffusion models have been widely applied in continuous domains, they have also proven useful in generating discrete sequences, inheriting the benefits of controllability through guidance. This has been shown both in the textual domain [14, 77] and in the biology domain [37, 122, 88, 45]. In the domain of text generation, state-of-the-art performances are achieved by scaling up autoregressive models [2, 10, 113, 115, 116]. However, for generating biological sequences such as proteins, DNA, and RNA, language models have to be adapted to handle their specific sequence characteristics [99, 65, 93, 19, 63, 22].

*Equal contribution. ♮ Intern at Stanford University.
†Corresponding author: mengdiw@princeton.edu

AI for New Drug Modalities at NeurIPS 2024.

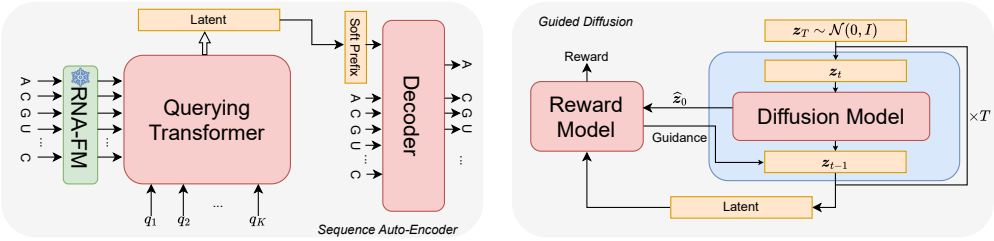


Figure 1: `RNAdiffusion` : Latent diffusion model for RNA sequences. Three parts of `RNAdiffusion` : (1) RNA **sequence auto-encoder**, consisting of a pretrained RNA-FM model, a Querying Transformer, and a decoder, for translating between the sequence space and the latent space; (2) **Guided diffusion** model with a pre-trained score network, for generating latent RNA embeddings under external guidance; (3) **Latent reward model**, trained on the latent space to predict functional properties of RNA, for computing guidance of diffusion.

In this work, we focus on developing generative models for RNA, with applications to both non-coding RNAs (ncRNAs) and mRNA untranslated regions (UTRs).

For ncRNAs, they are crucial regulators of gene expression and cellular processes, influencing pathways involved in development, disease, and immune responses. In parallel, the UTRs of mRNAs play a central role in regulating gene expression [91, 25, 13] and mRNA stability [12]. The ability to generate optimized UTR sequences offers significant potential for controlling gene expression, advancing mRNA vaccine designs, and enabling targeted therapies [101, 24, 18]. Therefore, our goal is to develop a robust and versatile language model capable of generating ncRNAs, and optimizing UTRs for enhanced functionality.

For ncRNAs and UTRs, individual nucleotides (A, U, G, C) often do not independently exhibit functions. Instead, the nucleotides may form clusters of higher-order structures, such as motifs, contributing to the functional properties of the RNA. This introduces distinct challenges in sequence modeling. Specifically, the denoising network has to account for the high-level picture of the entire sequence and the low-level syntax within a local cluster. Besides, these RNAs have dramatically different lengths, adding difficulty to efficient training and generalization. Although Transformer denoisers can handle variable-length inputs in theory, existing studies mostly pad or truncate input to a fixed length, or build an additional length prediction module to ensure robust practical performances [103, 64, 126].

To overcome these challenges, we develop diffusion models on a latent space for RNA sequences. Inspired by recent advances [100, 87], our approach utilizes a sequence auto-encoder to map raw input sequences into a latent embedding space. We then train a score-based continuous diffusion model to learn this latent distribution. Please see Figure 1 for a detailed schematic. This latent-space approach is effective for RNA sequences: *First*, it delegates the generation of the intricate raw RNA sequences to the sequence decoder, allowing the diffusion model to concentrate on capturing the coarse-grained distribution in the meaningful latent space. *Second*, the latent space is of fixed size, simplifying the design and training of the diffusion model by eliminating the need for additional modules or special adaptations for varying sequence lengths.

For the sequence autoencoder, we use a pretrained language model RNA-FM [26] to map raw sequences to meaningful, token-level representations. A trainable Query Transformer (Q-Former) [76] summarizes these representations into fixed-length latent vectors. Next, we train a score-based denoising diffusion model to capture the distribution of RNAs in latent space. We then assess the generated ncRNAs using biological metrics such as Levenshtein distance, 4-mer distance, G/C content, minimum free energy, and secondary structure. Results confirm that the generated ncRNAs align with natural ncRNA distributions across these indicators. Thus, our latent diffusion model can produce novel RNA sequences with high similarity to native ncRNAs.

Since latent variables from Q-Formers provide biologically meaningful summaries, we train reward networks directly in the latent space to predict functional properties of UTRs, key elements in mRNAs. The rewards, Mean Ribosome Loading (MRL) [101] and Translation Efficiency (TE) [24], measure mRNA-to-protein production levels and efficiencies, respectively. Using the trained reward models, we compute gradients and guide the diffusion model. Results show that `RNAdiffusion` generates novel UTRs with substantially higher MRL and TE values by incorporating guidance into

the backward generation process. This guidance is a plug-and-play method that requires no extra training. Overall, the generated UTRs from guided diffusion achieve a 166.7% improvement of TE and a 52.6% improvement of MRL compared to unguided generation.

We summarize the contributions of `RNAdiffusion` :

- Our sequence autoencoder, a novel adaptation of Q-Former, summarizes varying-length outputs from pretrained encoders into fixed-length embeddings, accurately reconstructing input sequences.
- Our latent diffusion model generates novel RNA sequences, and empirical studies confirm that these sequences align with natural distributions across various biological metrics.
- `RNAdiffusion` enables **controlled generation** via guidance, optimizing 5'-UTRs for higher protein production (166.7% improved TE and 52.6% improved MRL compared to unguided generation), while achieving a better reward–structural stability *trade-off* than baselines.

2 Methodology

`RNAdiffusion` has three parts: (1) a sequence auto-encoder and decoder that map between the space of variable-length discrete sequence and a continuous latent space, (2) a continuous diffusion model trained to model the distributions in the latent space, with optionally added guidance, and (3) a reward model trained on top of the latent space for computing guidance. See Figure 1 for an overview of our methodology.

2.1 Sequence Auto-Encoder

We aim for a sequence auto-encoder that removes the information redundancy in the raw sequence of tokens and compress them into a fixed-length set of continuous embedding vectors. We propose to use a sequence auto-encoder that consists of a pretrained encoder, a Querying Transformer (Q-Former) [76], and a decoder (Figure 1 left panel).

Encoder. We adopt the pretrained RNA-FM [26], a BERT-type encoder-only model, as the first component of the sequence auto-encoder [33]. BERT-type encoders are pretrained via masked language modeling task to predict randomly masked tokens. We denote the encoder by Enc. It maps a sequence $\boldsymbol{x} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^L)$ to a sequence of contextualized token embeddings of the same length L. We fix the encoder during the training of the sequence auto-encoder.

Querying-Transformer. Next we use a Q-Former, denoted by $\mathrm{QFormer}_\phi(\cdot)$, to summarize the embeddings of the pre-trained encoder $\mathrm{Enc}(x)$ into K latent vectors $\boldsymbol{z} = (\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^K)$, where K is a *fixed* number independent of L. We aim to achieve three goals: (1) To retains enough information in latent vectors, making it easy to train a decoder for raw sequence reconstruction; (2) To use latent vectors later for predicting RNA functional properties, via training additional prediction models on top of these embedding; (3) To reduce the dimensionality of raw sequences, so that diffusion models can focus on learning intrinsic structure of latent RNA vectors.

The Querying Transformer (Q-Former) [76] takes K trainable query token embeddings $(\boldsymbol{q}^1, \ldots, \boldsymbol{q}^K)$ as input. The query tokens go through the transformer with cross-attentions to the embedding sequence given by the encoder and progressively summarize the original token embeddings of varying lengths into a fixed-size sequence of embedding vectors. This is a flexible approach compared to using only the `<cls>` token embedding [75]. As when we increase K, we retain more information in the latent vectors and allow accurate reconstruction, which helps to mitigate the KL-vanishing problem [21]. Our ablation study in Section 3.3 validates the necessity of the Q-former.

Decoder. We use a standard causal Transformer [120] as the decoder, denoted by Dec_ψ. We build a simple linear projection layer to convert the K latent vectors into a soft prompt embeddings of equal size for the decoder to condition on. Then the decoder autoregressively reconstructs the original input sequence given the latent vectors.

Traning Objective. With the pre-trained encoder fixed, we train the Q-Former and the decoder from scratch in an end-to-end manner. The training objective is to reconstruct the original sequence from the embeddings given by the fixed pretrained encoder model. Our loss function is given by

$$\mathcal{L}_{\text{reconstruct}}(\psi, \phi; \boldsymbol{x}) = -\log p_{\mathrm{Dec}_\psi}(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^L \mid \mathrm{QFormer}_\phi(\mathrm{Enc}(\boldsymbol{x}))), \tag{1}$$

where ϕ, ψ denote the weights of the QFormer (including the query tokens $\boldsymbol{q}$) and Decoder, respectively. As people often use a very small (e.g. $1e-6$) KL-regularization term in latent diffusion models [100, 131], we choose to not include any KL-regularization, simplifying the design and helping to avoid KL-vanishing [21]. Empirically, the training works well without the KL term.

2.2 Latent Diffusion Model

We build a classical continuous diffusion model [57] to model the distribution of the latent variables $\boldsymbol{z}$, which generates samples from the target distribution by a series of noise removal processes. Specifically, we adopt a denoising network to predict the added noise $\boldsymbol{\epsilon}$ from the noisy input $(\sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon})$, where t is a time step of the forward process, $\bar{\alpha}_t$ is the corresponding signal strength level, and $\boldsymbol{\epsilon}$ is the added Gaussian noise.

For the denoising score network $\mathrm{Denoiser}_\theta$, we use transformer denoising networks following [77]. We use the ϵ-prediction scheme in Denoising Diffusion Probabilistic Models (DDPM) [57] and adopt the simplified training objective given by

$$\mathcal{L}(\theta) = \mathbb{E}_{\boldsymbol{z}_0 = \mathrm{QFormer}_\phi(\mathrm{Enc}(\boldsymbol{x}))}\mathbb{E}_{t,\boldsymbol{\epsilon}}\|\boldsymbol{\epsilon} - \mathrm{Denoiser}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2, \tag{2}$$

In our framework, the latent embedding have fixed sizes, making it easy to train diffusion models and treating data as if they are continuous like images. Unlike other discrete diffusion models, our framework does not require truncating/padding sequences to a fixed length compared with [77, 14], and it does not require training an additional length predictor as done by [103, 64, 126]. Our auto-regressive decoder has been trained to generate sequences with variable lengths; see Figure 2.

2.3 Guided Generation in the Latent Space

To generate RNA sequences of desired properties, we need a reward model for predicting the property of new sequences. Suppose we have a labeled dataset $\{(\boldsymbol{x}, r)\}$, where r is the measured reward of interest, e.g., the translation efficiency. We train a reward model R_ξ on top of the latent space to predict r given the input sequence $\boldsymbol{x}$. The training objective is

$$\mathcal{L}(\xi) = \mathbb{E}_{(\boldsymbol{x},r)}\, \ell(R_\xi(\boldsymbol{z}), r), \tag{3}$$

where ℓ is a loss function, $\boldsymbol{z} = \mathrm{QFormer}_\phi(\mathrm{Enc}(\boldsymbol{x}))$ is the embedding obtained from Q-Former, and ξ are weights of the reward model.

Finally, we use the reward model to compute a gradient-based guidance (e.g., via universal guidance [17]) and add it to the backward generation process. This guidance steers the random paths of the backward generation process of the diffusion model toward a higher reward region in a plug-and-play manner. Specifically, at the step t of the backward diffusion process, given the current noisy sample $\boldsymbol{z}_t$, we first estimate the clean sample

$$\widehat{\boldsymbol{z}}_0 = \widehat{\boldsymbol{z}}_0(\boldsymbol{z}_t) = \frac{\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\mathrm{Denoiser}_\theta(\boldsymbol{z}_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{4}$$

And we use $R_\xi(\widehat{\boldsymbol{z}}_0)$ as a surrogate reward for the noisy sample $\boldsymbol{z}_t$. We plug in the gradient $\nabla_{\boldsymbol{z}_t}\ell(r^*, R_\xi(\widehat{\boldsymbol{z}}_0))$ into the current predicted noise as follows, with r^* being the targeted reward and λ being the guidance strength.

$$\text{predicted noise} = \mathrm{Denoiser}_\theta(\boldsymbol{z}_t, t) + \sqrt{1-\bar{\alpha}_t} \cdot \lambda \cdot \nabla_{\boldsymbol{z}_t}\ell(r^*, R_\xi(\widehat{\boldsymbol{z}}_0)). \tag{5}$$

We note that controlling the generation for decoder-only models such as GPTs in the raw sequence space is generally hard without fine-tuning the model (see [70] and the references therein).

3 RNA Sequence AutoEncoder

3.1 Experiment Setup

Models. We adopt the pretrained RNA-FM [26] as the encoder. RNA-FM is an encoder-only model trained with masked language modelling on non-coding RNAs (ncRNAs). The architecture of our

Q-Former is modified from the ESM-2 model structure [82]. For the decoder model, we adopt the same model architecture of ProGen2-small [93] with our customized tokenizers for RNAs.

Dataset. We pretrain our RNA sequence autoencoder with the ncRNA subset of Ensembl database [89] downloaded from RNAcentral [1], which contains collected ncRNA sequences from a variety of vertebrate genomes and model organisms. We select the sequences shorter than 768 bp to avoid exceeding RNA-FM's maximum length limit and exclude those with rare tokens more than {A, C, G, U}, curating a dataset of 1.1M samples for training and testing.

Hyperparameter Search. There are two hyperparameters that influence the dimension of the latent space: the number of query tokens K and the hidden dimension D of each latent vector. We tune K among $\{16, 32\}$ and D among $\{40, 80, 160, 320\}$.

Evaluation Metrics. For each input sequence $\boldsymbol{x}$ of length L, we calculate two metrics: (1)length-normalized reconstruction loss (i.e., negative log-likelihood, **NLL**), i.e., $-\frac{1}{L}\log p_{\mathrm{Dec}_\psi}(\boldsymbol{x} \mid \mathrm{QFormer}_\phi(\mathrm{Enc}(\boldsymbol{x})))$; (2)length-normalized edit distance (**NED**) between the reconstructed sequence $\widehat{\boldsymbol{x}} \sim p_{\mathrm{Dec}_\psi}(\cdot \mid \mathrm{QFormer}_\phi(\mathrm{Enc}(\boldsymbol{x})))$ and the original input sequence $\boldsymbol{x}$, i.e., $\frac{1}{L}\mathrm{EditDistance}(\widehat{\boldsymbol{x}}, \boldsymbol{x})$. Due to the space limit, additional experimental setup can be found in Appendix D.1.

3.2 Results

We perform a sweep over hyperparameters and obtain a set of RNA sequence autoencoders with high reconstruction capabilities. For each of the two metrics, we report the empirical average on a held-out test set in Table 1. We see that increasing either the number of query tokens K or the hidden dimension D reduces the reconstruction errors and the error rates. Increasing the dimension of the latent space retains more information in the latent embeddings, and hence it is easier to reconstruct the input sequence.

Table 1: The performance of the RNA sequence autoencoders and the latent diffusion models. For diffusion models, the upper bound of NLL, i.e., ELBO, is calculated using Equation (12) of [57].

K	D	Sequence AutoEncoder NLL↓	Sequence AutoEncoder NED ↓	Latent Diffusion NLL (bits/dim)↓
16	40	0.2631	19.36% ± 21.76 %	≤ 2.93
16	80	0.0186	6.00% ± 13.40%	≤ 1.64
16	160	0.0156	2.19% ± 5.50 %	≤ 1.78
16	320	0.0023	0.12% ± 0.66 %	≤ 1.61
32	40	0.0995	13.50% ± 19.06 %	≤ 2.35
32	80	0.0125	4.40% ± 10.73 %	≤ 1.29
32	160	0.0003	0.02% ± 0.40 %	≤ **1.01**
32	320	**0.0001**	**0.003% ± 0.08 %**	≤ 1.68

3.3 Ablation Studies.

We perform ablation studies to validate the necessities of the components of our sequence autoencoder. When we replace the Q-former with a naive average pooling of RNA-FM output token embeddings $\mathrm{Enc}(x)$, the reconstruction performance of the Sequence AutoEncoder is significantly dampened, resulting in a 34.8% normalized edit distance (see Table 2).

We also experiment with replacing the RNA-FM with naive one-hot token embeddings and the results indicate that both the reconstruction capability of the sequence autoencoder and the reward modeling ability of the latent vectors are negatively affected, which validate the necessity of using the pre-trained RNA-FM as the encoder. Full results are deferred to Appendix D.4.

4 Latent Diffusion Model for ncRNAs

4.1 Experimental Setup

Models and Datasets. We build a 24-layer Transformer model [120] with hidden dimension 2048 as the denoising network for our latent diffusion model. As the latent vectors are continuous, we

Table 2: **Ablations on Q-former**. We remove the Q-former by averaging all output token embeddings (dim: 640) of RNA-FM, i.e., latent space dim 1×640, and train the decoder for one epoch.

Settings ($K \times D$)	Sequence AutoEncoder	
	NLL ↓	NED ↓
w/o Q-former (1×640)	1.0061	34.78 % ± 17.81 %
w/ Q-former (16×160)	**0.0156**	**2.19% ± 5.50 %**

do not need any additional embedding layer or rounding operation. We use the same Ensembl ncRNA dataset and apply the same preprocessing as our RNA sequence Auto-encoder. We trained the diffusion model for 10 epochs with the Sequence Autoencoder frozen. Full training details are given in Appendix D.2.

Evaluation Metrics. To evaluate the performance of our latent diffusion model, we calculate the weighted loss $\mathcal{L}_{1:T-1}$ on the test dataset according to Eqn. (12) of [57], which is an upper bound of the negative log-likelihood (NLL). We also visualize RNAs in the latent space by using a **t-SNE map** [119], and compare with reference natural RNAs and randomly generated sequences, which match the length distribution of natural data. Besides, we feed the generated latent vectors through the decoder to get the RNA sequences and we adopt the following biological metrics following [95, 18].

- **Minimum Levenshtein Distance**. The Levenshtein distance [73] is defined as the minimum number of edits (insertions, deletions, or substitutions) required to transform one sequence into another. For each generated RNA sequence, we calculate the smallest Levenshtein distance to all sequences in a reference dataset of natural RNAs, and normalized the distance by the length of the generated sequence. [3] By looping over all the generated RNA sequences, we obtain a distribution of minimum *normalized* Levenshtein distances.
- **Minimum 4-mer Distance**. We calculate the frequencies of 4-mers of each sequence [121]. The 4-mer distance of two sequences is defined as the Euclidean distance between the two *frequency* vectors. Similarly, for each generated RNA sequence, we calculate the minimum 4-mer distance against a reference natural RNA dataset.[3] We obtain the distribution of minimum 4-mer distances for the generated RNA sequences.
- **G/C content** [69]. G/C content, defined as the percentage of guanine (G) and cytosine (C), is an essential indicator of molecular stability. A higher G/C content generally implies higher stability due to the stronger hydrogen bonds between G and C compared to A and T.
- **Minimum Free Energy (MFE)** [118]. It represents the minimum energy required for the molecule to maintain its conformation. The MFE is crucial measure of an RNA segment's structural stability, reflecting how tightly the RNA molecule can fold based on its nucleotide composition and sequence arrangement. A lower MFE suggests a more stable and tightly folded RNA structure. We calculate the MFE of generated RNA sequences using the ViennaRNA Package [85].
- **Minimum Levenshtein Distance of RNA Secondary Structure**. RNA secondary structure, which involves local interactions like hairpins and stem-loops, plays a crucial role in the RNA's biological functions [55]. Dot-bracket notation represents these structures by using dots for unpaired nucleotides and matching parentheses for base pairs, simplifying the visualization and analysis of RNA sequences [85]. We calculate the secondary structure of RNA sequences using ViennaRNA Package [85], and then calculate the distribution of the minimum *normalized* Levenshtein distances of the secondary structures of the generated RNAs to those of natural referential RNAs.

4.2 Results: Samples from `RNAdiffusion` Resemble Natural ncRNAs

We report the test NLL of the latent generated distribution in Table 1. We see that increasing the query token length K and the embedding dimension D generally decreases the test NLL per dimension. For subsequent experiments, we choose to use the trained RNA auto-encoders that have average normalized edit distance $\leq 1\%$. They correspond to hyperparameters $(K, D) = (16, 320)$, $(32, 160)$, and $(32, 320)$. We further select the parameter **(32, 160)** for generation, which achieves the smallest *total* NLL, i.e, NLL per dimension *multiplied by* the dimension of latent space ($K \times D$).

[3] We exclude the identical sequence when calculating the minimum Levenshtein distance and minimum 4-mer distance.

We compare the generated samples (with $K = 32$, $D = 160$) with natural ncRNAs. We plot the length distribution of the sequences in Figure 2 and demonstrate that the generated sequences vary in length, with a distribution closely aligned with that of natural ncRNAs.

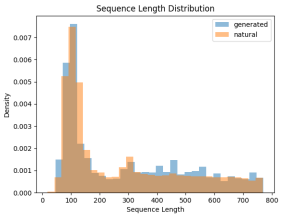


Figure 2: **Sequence length comparison** between the natural **ncRNA** test set and generated sequences (sample size: 20000).

Additionally, we create a curated random sequence set by replacing each token in the reference sequences with nucleotide randomly sampled at a probability of 0.5 for G/C, approximating the average G/C content ratio of natural ncRNAs, while maintaining the original sequence length. Next, we plot the distributions of the aforementioned metrics of our generated sequences and compare them to a held-out natural ncRNA reference set and the set of random sequences in Figure 3, which demonstrates that our generated sequences exhibit greater similarity to the natural sequences compared to the random sequences in terms of sequence-level metrics (Levenshtein and 4-mer distances, G/C content ratios), structural-level metrics (Minimum Free Energies and secondary structure distances), and latent space similarities (t-SNE maps).

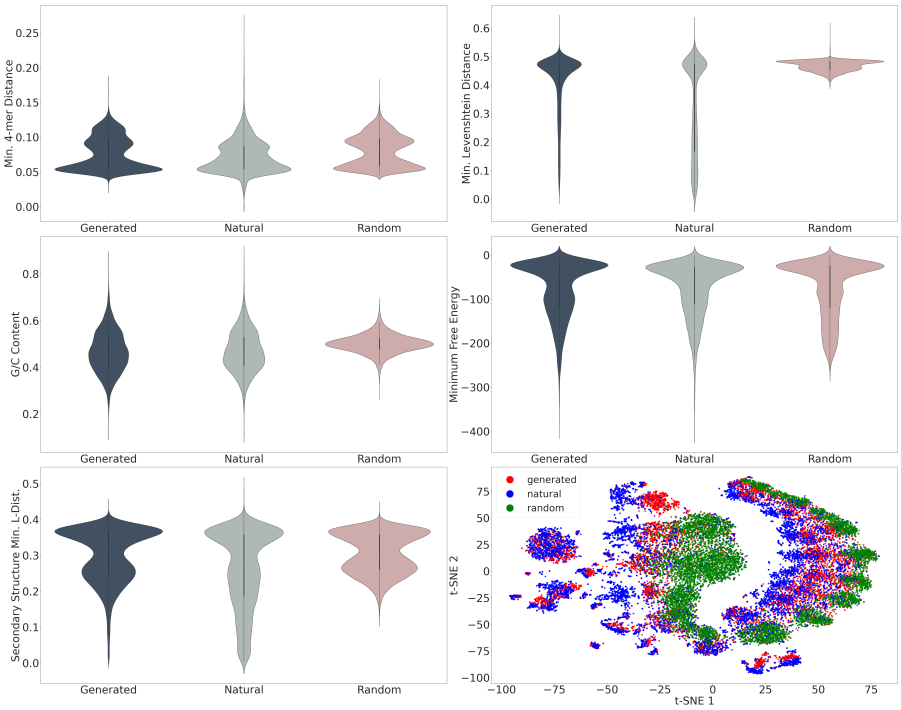


Figure 3: **Generated samples from** `RNAdiffusion` **compared to natural ncRNAs and random sequences.** Each includes 9000 sequences. (a) minimum 4-mer distances. (b) minimum sequence Levenshtein distances. (c) G/C content ratios. (d) minimum free energy. (e) minimum Levenshtein distances of RNA secondary structure. (f) t-SNE visualizations of RNAs in latent embedding space.

5 Optimizing 5'-UTR Sequences with Guided Diffusion

The 5' untranslated region (5'-UTR), located at the beginning of mRNA before the coding sequence, plays a crucial role in regulating protein synthesis by impacting mRNA stability, localization, and translation. In this section, we focus on generating 5'-UTR sequences with high protein production levels through guided diffusion.

We consider two functional properties of 5'-UTRs that correlate with protein production levels [101, 24, 26, 132, 67]: (1) **Mean Ribosome Loading (MRL)** measures the number of ribosomes actively translating an mRNA, reflecting its translation efficiency [101]. In molecular biology, techniques like polysome profiling (Ribo-seq) are employed to determine MRL. (2) **Translation efficiency (TE)**

represents the rate at which mRNA is translated into protein [24]. It is determined by dividing the Ribo-seq RPKM [60], which shows ribosomal footprints, by the RNA-seq RPKM that measures the mRNA's abundance in the cell. This calculation helps gauge how efficiently an mRNA is being translated, independent of its expression level.

5.1 Fine-tuning `RNAdiffusion` for 5'-UTRs

Although both UTRs and ncRNAs share similarities in their non-coding properties, UTRs contain important functional elements that are irrelevant to ncRNAs. Thus, our pretrained `RNAdiffusion` could not be directly applied to generate UTR sequences due to the distributional shift between UTRs and ncRNAs. To adapt `RNAdiffusion` to generate 5'-UTR sequences, we *fine-tune* the score function of `RNAdiffusion` using the five-species 5'-UTR dataset from UTR-LM [30], where we elected sequences with lengths less than 768 and get 205K samples. We finetune the diffusion model for 1 epoch on top of the frozen sequence auto-encoder with $(K, D) = (32, 160)$.

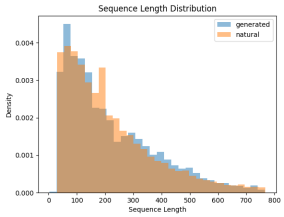


Figure 4: **Sequence length histogram** of the natural **UTR** test set and generated sequences (sample size: 20000).

Upon testing the autoencoder on UTR test set, it shows considerable low reconstruction errors (NLL$= 0.0001$, NED$=0.01\% \pm 0.31\%$), confirming that the autoencoder pretrained with ncRNAs could be transferred onto UTRs. To compare the generated sequences of the fine-tuned diffusion model with natural 5'-UTRs, we present the sequence length histograms (Figure 4) and compute the metrics proposed in Section 4.1 (Figure 6 in Appendix D.5). The results demonstrate that they align closely with the natural distribution.

5.2 Latent Reward Modeling for MRL and TE

Datasets. For the *TE prediction* task, we use three endogenous human 5'-UTR datasets analyzed by Cao et al. [24], which contain 28,246 sequences and their measured TE. We reserve 10% of the dataset for testing, 10% for validation, and use the remaining 80% for training. For the *MRL prediction* task, following the length-based held-out testing approach recommended by Sample et al. [101], we train our model on a collection of 83,919 random 5'-UTR sequences ranging from 25 to 100 bp with MRL measurement. We reserve 10% of the data for validation. Subsequently, we test the model on 7,600 human 5'-UTR sequences.

Model Architecture. The latent reward networks take as input a list of K latent vectors obtained by Q-former. We construct a 6-layer 1D convolutional neural network with residual blocks [53] and apply it on both the MRL and TE prediction tasks. Additional details are deferred to Appendix D.3.

Results. For each task, we train two reward models on top of the latent spaces from the pretrained RNA Sequence Autoencorder ($(K, D) = (32, 160)$), each initialized under distinct random seeds. One model is utilized for guided generation in Section 5.3 while the other is employed for validation assessment. We calculate the Spearman correlation coefficients between the predicted rewards and the actual rewards (shown in Table 3). Notably, the Spearman R values on the held-out test sets are 0.56 for the TE task and 0.69 for the MRL task, achieving slightly inferior yet comparable performance compared with baselines [24, 101]) and the state-of-the-art predictor built upon UTR-LM ([30]).

Table 3: **Spearman R-values of MRL and TE reward modeling.** The embeddings after Q-Former remain useful for downstream tasks, allowing us to build effective reward models based on them.

	TE			MRL		
	Ours	UTR-LM [30]	[24]	Ours	UTR-LM [30]	[101]
Test Spearman R	0.56	**0.64**	0.60	0.69	**0.85**	0.83

5.3 Reward-Guided Diffusion

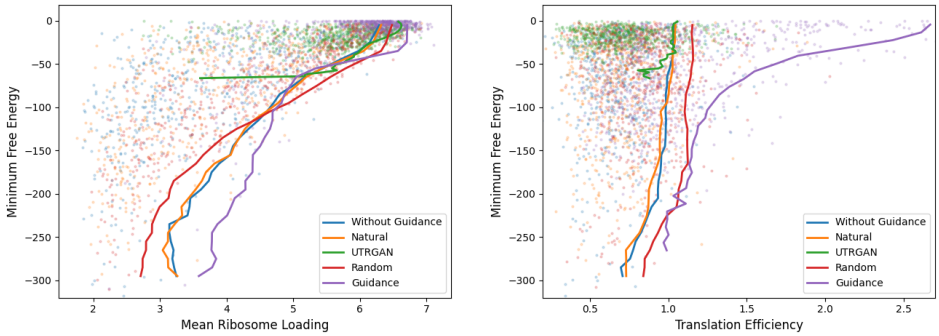
We compute gradient-based forward universal guidance [17] in the latent space with the trained reward models to shift the latent backward sample paths towards generating high MRLs/TEs sequences as described in Section 2.3, where we choose the loss function ℓ to be L_2 loss and vary the target reward

r^* and the guidance strength λ. For guidance, we take the latent reward models trained with one random seed for TE and MRL optimization. For cross-validation, we employ the reward models trained with another random seed as the *validation* model to evaluate the rewards of the generated sequence. The best performing guidance parameters are able to improve the mean TE by up to 166.7% and mean MRL by up to 52.6% compared to the unguided baseline. We defer our systematic analysis of the effects of different guidance parameters to Appendix D.6.

Baselines. We compare the performance of our guidance pipeline with the following two baselines:

- **Random**. We adopt a best-of-N sequence generation method similar to Sample et al. [101], which uses a genetic algorithm to generate random sequences and merely retain ones with high rewards. In our approach, we generate random sequences as in Section 4.2 and select the sequence with the highest reward according to our guidance reward model.
- **UTRGAN**, a 5'-UTR sequence generative model targeting translation rate optimization. We directly generate 6000 reward-optimized sequences using the official codes release by UTRGAN [18].

Results. For each generated sequence, we evaluate the reward using the *validation* reward models and also calculate the Minimum Free Energy (MFE) as the proxy for structural stability. We plot the validation reward (MRL/TE) v.s. MFE of samples generated by different methods in the Pareto fronts (Figure 5). Our gradient-guided latent diffusion model achieves higher rewards with the same MFE level compared to the two baselines. This demonstrates that our method has a better reward–structural stability *trade-off*.



(a) MRL (b) TE

Figure 5: **Pareto front curves** between Minimum Free Energy (MFE) and Mean Ribosome Loading (MRL)/Translation Efficiency (TE). The curves are generated by selecting the top 10% quantile MRL/TE within sliding windows around each MFE value. Each shaded dot denotes a sequence. All the sequences are evaluated with the same *validation* reward models.

6 Related Work

People have been exploring building D3PM-type [14] discrete diffusion models and Diffusion-LM type [77] for proteins sequences and 3D structures [72, 123, 40, 6, 80, 117, 122, 127, 61, 45, 4, 88] and DNAs [15, 78, 37]. Besides, latent diffusion models for text sequences have been studied in [124, 131, 87]. For a detailed discussion on continuous/discrete diffusion models and their applications in biology, see the full related work section in Appendix C.

7 Conclusion

We present `RNAdiffusion` , a diffusion-based generative model that captures the RNA sequence distribution within a latent space constructed by a Sequence AutoEncoder, which utilizes a Querying Transformer to manage the variable-length nature of RNA sequences. By incorporating gradient guidance from latent reward models, our diffusion model can generate novel 5'-UTRs with up to 166.7% higher TE and 52.6% higher MRL, while achieving a superior *trade-off* between rewards and structural stability. These results provide a biologically relevant demonstration of RNA sequence generation, an underexplored yet exciting area of research.

References

[1] Rnacentral 2021: secondary structure integration, improved sequence search and new member databases. *Nucleic acids research*, 49(D1):D212–D220, 2021.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sP1fo2K9DFG.

[4] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pages 2023–09, 2023.

[5] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=EbMuimAbPbs.

[6] Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

[7] Namrata Anand and Possu Huang. Generative modeling for protein structures. *Advances in neural information processing systems*, 31, 2018.

[8] Namrata Anand, Raphael Eguchi, and Po-Ssu Huang. Fully differentiable full-atom protein backbone generation. 2019.

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

[10] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[11] Ivan Anishchenko, Samuel J Pellock, Tamuka M Chidyausiku, Theresa A Ramelot, Sergey Ovchinnikov, Jingzhou Hao, Khushboo Bafna, Christoffer Norn, Alex Kang, Asim K Bera, et al. De novo protein design by deep network hallucination. *Nature*, 600(7889):547–552, 2021.

[12] Patricia R Araujo, Kihoon Yoon, Daijin Ko, Andrew D Smith, Mei Qiao, Uthra Suresh, Suzanne C Burns, Luiz OF Penalva, et al. Before it gets started: regulating translation at the 5' utr. *International Journal of Genomics*, 2012, 2012.

[13] John F Atkins, Raymond F Gesteland, and Thomas Cech. Rna worlds: from life's origins to diversity in gene regulation. 2011.

[14] 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.

[15] Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. *arXiv preprint arXiv:2305.10699*, 2023.

[16] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[17] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023.

[18] Sina Barazandeh, Furkan Ozden, Ahmet Hincer, Urartu Ozgur Safak Seker, and A Ercument Cicek. Utrgan: Learning to generate 5'utr sequences for optimized translation efficiency and gene expression. *bioRxiv*, pages 2023–01, 2023.

[19] Gonzalo Benegas, Sanjit Singh Batra, and Yun S Song. Dna language models are powerful predictors of genome-wide variant effects. *Proceedings of the National Academy of Sciences*, 120(44):e2311219120, 2023.

[20] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

[21] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002. URL https://aclanthology.org/K16-1002.

[22] 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.

[23] 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.

[24] Jicong Cao, Eva Maria Novoa, Zhizhuo Zhang, William CW Chen, Dianbo Liu, Gigi CG Choi, Alan SL Wong, Claudia Wehrspaun, Manolis Kellis, and Timothy K Lu. High-throughput 5' utr engineering for enhanced protein production in non-viral gene therapies. *Nature communications*, 12(1):4138, 2021.

[25] Mark G Caprara and Timothy W Nilsen. Rna: versatility in form and function. *Nature structural biology*, 7(10):831–833, 2000.

[26] Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Irwin King, et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *arXiv preprint arXiv:2204.00300*, 2022.

[27] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. *arXiv preprint arXiv:2302.07194*, 2023.

[28] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

[29] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[30] Yanyi Chu, Dan Yu, Yupeng Li, Kaixuan Huang, Yue Shen, Le Cong, Jason Zhang, and Mengdi Wang. A 5' utr language model for decoding untranslated regions of mrna and function predictions. *Nature Machine Intelligence*, pages 1–12, 2024.

[31] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

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

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[34] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[35] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

[36] Raphael R Eguchi, Christian A Choe, and Po-Ssu Huang. Ig-vae: Generative modeling of protein structure by direct 3d coordinate generation. *PLoS computational biology*, 18(6): e1010271, 2022.

[37] Lucas Ferreira DaSilva, Simon Senan, Zain Munir Patel, Aniketh Janardhan Reddy, Sameer Gabbita, Zach Nussbaum, Cesar Miguel Valdez Cordova, Aaron Wenteler, Noah Weber, Tin M Tunjic, et al. Dna-diffusion: Leveraging generative models for controlling chromatin accessibility and gene expression via synthetic regulatory elements. *bioRxiv*, pages 2024–02, 2024.

[38] Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

[39] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

[40] Zhangyang Gao, Cheng Tan, and Stan Z Li. Diffsds: A language diffusion model for protein backbone inpainting under geometric conditions and constraints. *arXiv preprint arXiv:2301.09642*, 2023.

[41] Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Difformer: Empowering diffusion model on embedding space for text generation. *arXiv preprint arXiv:2212.09412*, 2022.

[42] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

[43] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *arXiv preprint arXiv:2206.09012*, 2022.

[44] Joe G Greener, Lewis Moffat, and David T Jones. Design of metalloproteins and novel protein folds using variational autoencoders. *Scientific reports*, 8(1):16189, 2018.

[45] Nate Gruver, Samuel Stanton, Nathan C Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. *arXiv preprint arXiv:2305.20009*, 2023.

[46] Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[47] Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. *arXiv preprint arXiv:2404.14743*, 2024.

[48] Zhiye Guo, Jian Liu, Yanli Wang, Mengrui Chen, Duolin Wang, Dong Xu, and Jianlin Cheng. Diffusion models in bioinformatics and computational biology. *Nature reviews bioengineering*, 2(2):136–154, 2024.

[49] Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *arXiv preprint arXiv:2210.17432*, 2022.

[50] Xiaochuang Han, Sachin Kumar, Yulia Tsvetkov, and Marjan Ghazvininejad. Ssd-2: Scaling and inference-time fusion of diffusion language models. *arXiv preprint arXiv:2305.14771*, 2023.

[51] Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, pages 1188–1205, 1986.

[52] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[54] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. *arXiv preprint arXiv:2211.15029*, 2022.

[55] Paul G Higgs. Rna secondary structure: physical and computational aspects. *Quarterly reviews of biophysics*, 33(3):199–253, 2000.

[56] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[57] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[58] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[59] Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.

[60] Nicholas T Ingolia, Sina Ghaemmaghami, John RS Newman, and Jonathan S Weissman. Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. *science*, 324(5924):218–223, 2009.

[61] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

[62] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

[63] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

[64] Daniel D Johnson, Jacob Austin, Rianne van den Berg, and Daniel Tarlow. Beyond in-place corruption: Insertion and deletion in denoising probabilistic models. *arXiv preprint arXiv:2107.07675*, 2021.

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

[66] Mostafa Karimi, Shaowen Zhu, Yue Cao, and Yang Shen. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks. *Journal of chemical information and modeling*, 60(12):5667–5681, 2020.

[67] Alexander Karollus, Žiga Avsec, and Julien Gagneur. Predicting mean ribosome load for 5'utr of any length using deep learning. *PLoS computational biology*, 17(5):e1008982, 2021.

[68] Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2678–2687. PMLR, 2018.

[69] Özlen Konu and Ming D Li. Correlations between mrna expression levels and gc contents of coding and untranslated regions of genes in rodents. *Journal of Molecular Evolution*, 54: 35–41, 2002.

[70] Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pages 17506–17533. PMLR, 2023.

[71] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence of score-based generative modeling for general data distributions. In *International Conference on Algorithmic Learning Theory*, pages 946–985. PMLR, 2023.

[72] Jin Sub Lee, Jisun Kim, and Philip M Kim. Score-based generative modeling for de novo protein design. *Nature Computational Science*, 3(5):382–392, 2023.

[73] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[74] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

[75] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.378. URL https://aclanthology.org/2020.emnlp-main.378.

[76] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[77] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022.

[78] Zehui Li, Yuhao Ni, William AV Beardall, Guoxuan Xia, Akashaditya Das, Guy-Bart Stan, and Yiren Zhao. Discdiff: Latent diffusion model for dna sequence generation. *arXiv preprint arXiv:2402.06079*, 2024.

[79] Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. *arXiv preprint arXiv:2302.01877*, 2023.

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

[81] Zeming Lin, Tom Sercu, Yann LeCun, and Alexander Rives. Deep generative models create new and diverse protein structures. In *Machine Learning for Structural Biology Workshop, NeurIPS*, 2021.

[82] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, 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.

[83] Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning*, pages 21051–21064. PMLR, 2023.

[84] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[85] Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. Viennarna package 2.0. *Algorithms for molecular biology*, 6:1–14, 2011.

[86] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

[87] Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Weinberger. Latent diffusion for language generation. *arXiv preprint arXiv:2212.09462*, 2022.

[88] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *bioRxiv*, 2022. doi: 10.1101/2022.07.10.499510. URL https://www.biorxiv.org/content/early/2022/10/16/2022.07.10.499510.

[89] Fergal J Martin, M Ridwan Amode, Alisha Aneja, Olanrewaju Austine-Orimoloye, Andrey G Azov, If Barnes, Arne Becker, Ruth Bennett, Andrew Berry, Jyothish Bhai, Simarpreet Kaur Bhurji, Alexandra Bignell, Sanjay Boddu, Paulo R Branco Lins, Lucy Brooks, Shashank Budhanuru Ramaraju, Mehrnaz Charkhchi, Alexander Cockburn, Luca Da Rin Fiorretto, Claire Davidson, Kamalkumar Dodiya, Sarah Donaldson, Bilal El Houdaigui, Tamara El Naboulsi, Reham Fatima, Carlos Garcia Giron, Thiago Genez, Gurpreet S Ghattaoraya, Jose Gonzalez Martinez, Cristi Guijarro, Matthew Hardy, Zoe Hollis, Thibaut Hourlier, Toby Hunt, Mike Kay, Vinay Kaykala, Tuan Le, Diana Lemos, Diego Marques-Coelho, José Carlos Marugán, Gabriela Alejandra Merino, Louisse Paola Mirabueno, Aleena Mushtaq, Syed Nakib Hossain, Denye N Ogeh, Manoj Pandian Sakthivel, Anne Parker, Malcolm Perry, Ivana Piližota, Irina Prosovetskaia, José G Pérez-Silva, Ahamed Imran Abdul Salam, Nuno Saraiva-Agostinho, Helen Schuilenburg, Dan Sheppard, Swati Sinha, Botond Sipos, William Stark, Emily Steed, Ranjit Sukumaran, Dulika Sumathipala, Marie-Marthe Suner, Likhitha Surapaneni, Kyösti Sutinen, Michal Szpak, Francesca Floriana Tricomi, David Urbina-Gómez, Andres Veidenberg, Thomas A Walsh, Brandon Walts, Elizabeth Wass, Natalie Willhoft, Jamie Allen, Jorge Alvarez-Jarreta, Marc Chakiachvili, Bethany Flint, Stefano Giorgetti, Leanne Haggerty, Garth R Ilsley, Jane E Loveland, Benjamin Moore, Jonathan M Mudge, John Tate, David Thybert, Stephen J Trevanion, Andrea Winterbottom, Adam Frankish, Sarah E Hunt, Magali Ruffier, Fiona Cunningham, Sarah Dyer, Robert D Finn, Kevin L Howe, Peter W Harrison, Andrew D Yates, and Paul Flicek. Ensembl 2023. *Nucleic Acids Research*, 51(D1):D933–D941, 11 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac958. URL https://doi.org/10.1093/nar/gkac958.

[90] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *arXiv preprint arXiv:2211.00802*, 2022.

[91] Zhichao Miao and Eric Westhof. Rna structure: advances and assessment of 3d structure prediction. *Annual review of biophysics*, 46:483–503, 2017.

[92] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[93] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

[94] Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023. URL https://openreview.net/forum?id=6961CeTSFA.

[95] Furkan Ozden, Sina Barazandeh, Dogus Akboga, Urartu Ozgur Safak Seker, and A Ercument Cicek. Rnagen: A generative adversarial network-based model to generate synthetic rna sequences to target proteins. *bioRxiv*, pages 2023–07, 2023.

[96] Seongmin Park and Jihwa Lee. Finetuning pretrained transformers into variational autoencoders. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 29–35, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.insights-1.5. URL https://aclanthology.org/2021.insights-1.5.

[97] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[98] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[99] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[100] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[101] Paul J Sample, Ban Wang, David W Reid, Vlad Presnyak, Iain J McFadyen, David R Morris, and Georg Seelig. Human 5' utr design and variant effect prediction from a massively parallel translation assay. *Nature biotechnology*, 37(7):803–809, 2019.

[102] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[103] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. *arXiv preprint arXiv:2112.06749*, 2021.

[104] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[105] Emre Sevgen, Joshua Moller, Adrian Lange, John Parker, Sean Quigley, Jeff Mayer, Poonam Srivastava, Sitaram Gayatri, David Hosfield, Maria Korshunova, Micha Livne, Michelle Gill, Rama Ranganathan, Anthony B. Costa, and Andrew L. Ferguson. Prot-vae: Protein transformer variational autoencoder for functional protein design. *bioRxiv*, 2023. doi: 10.1101/2023.01.23.525232. URL https://www.biorxiv.org/content/early/2023/01/24/2023.01.23.525232.

[106] Kim T Simons, Rich Bonneau, Ingo Ruczinski, and David Baker. Ab initio protein structure prediction of casp iii targets using rosetta. *Proteins: Structure, Function, and Bioinformatics*, 37(S3):171–176, 1999.

[107] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[108] 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.

[109] Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*, 2022.

[110] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[111] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

[112] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

[113] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[114] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[115] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[116] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[117] Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

[118] Edoardo Trotta. On the normalization of the minimum free energy of rnas by sequence length. *PloS one*, 9(11):e113380, 2014.

[119] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[121] Le Van Vinh, Tran Van Lang, Le Thanh Binh, and Tran Van Hoai. A two-phase binning algorithm using l-mer frequency on groups of non-overlapping reads. *Algorithms for Molecular Biology*, 10:1–12, 2015.

[122] Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv*, 2022. doi: 10.1101/2022.12.09.519842. URL https://www.biorxiv.org/content/early/2022/12/10/2022.12.09.519842.

[123] 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.

[124] Xingqian Xu, Zhangyang Wang, Eric Zhang, Kai Wang, and Humphrey Shi. Versatile diffusion: Text, images and variations all in one diffusion model. *arXiv preprint arXiv:2211.08332*, 2022.

[125] Tianyu Yang, Thy Thy Tran, and Iryna Gurevych. Dior-cvae: Diffusion priors in variational dialog generation. *arXiv preprint arXiv:2305.15025*, 2023.

[126] 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*, 2023.

[127] Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

[128] Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Minshuo Chen, and Mengdi Wang. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement. *Advances in Neural Information Processing Systems*, 36, 2024.

[129] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

[130] Yinan Zhang, Eric Tzeng, Yilun Du, and Dmitry Kislyuk. Large-scale reinforcement learning for diffusion models. *arXiv preprint arXiv:2401.12244*, 2024.

[131] Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Josh Susskind, and Navdeep Jaitly. Planner: Generating diversified paragraph via latent language diffusion model. *arXiv preprint arXiv:2306.02531*, 2023.

[132] Weizhong Zheng, John HC Fong, Yuk Kei Wan, Athena HY Chu, Yuanhua Huang, Alan SL Wong, and Joshua WK Ho. Translation rate prediction and regulatory motif discovery with multi-task learning. In *International Conference on Research in Computational Molecular Biology*, pages 139–154. Springer, 2023.

A Limitations

The performance of our latent diffusion models depends on the total dimension of the latent space $K \times D$, where K is the number of query embeddings and D is the embedding dimension for each query. Increasing K and D makes the reconstruction task and the reward prediction tasks easier, but it allows more redundant information in the latent space, which may hinder the training of the diffusion model. Therefore, we require a hyperparameter search to find the best-performing K and D. To perform reward-guided diffusion, we require a trained differentiable reward network, whose performance crucially depends on the number and the quality of the labeled data. To extend to non-differentiable rewards, additional fine-tuning with reinforcement learning methods [20] is still required. Besides, we did not synthesize the generated 5'-UTRs in wet lab experiments to test the performance, which we leave as future work.

B Broader Impacts

Our work aims to generate 5'-UTRs with optimized protein production levels, which could significantly enhance the understanding and engineering of gene expression. This can lead to breakthroughs in developing new therapies for diseases such as cancer and genetic disorders. It can also improve synthetic biology systems that produce valuable compounds, such as pharmaceuticals and bio-fuels, more efficiently and sustainably. Nevertheless, the capability to alter gene expression and produce biological molecules could be misused to create harmful biological agents or cause unintended harm to ecosystems.

C Additional Related Work

C.1 Continuous Diffusion Models

A majority of works on diffusion models focus on modelling continuous data [57, 107, 98, 100, 16, 92, 58, 3, 62, 79, 112, 122]. During training, continuous Gaussian noises are sequentially injected to the sample. The core of diffusion models is a denoising network that takes in a Gaussian-noise-perturbed sample and predicts the clean version of the sample. Backed by the theory of forward/backward SDEs [108, 94, 71, 28, 27, 9, 51], diffusion models are guaranteed to recover the true data distribution.

The behavior of diffusion models can be controlled through *guidance*, where an additional guidance term is added into each step of the backward diffusion process to drive the sample toward the regions with desired properties [34, 17, 31, 43, 47, 128]. Classifier-based guidance [34] and its extentions utilize a separate classifier or a reward model to calculate the guidance term, while classifier-free guidance [56] calculates the guidance term based on the conditional diffsuion model itself. More complated controls can be done via finetuning the diffusion models with additional networks [129].

C.2 Diffusion Models for discrete variables

The essential components of diffusion models are the forward process, which defines how a clean sample is sequentially corrupted into a pure random noise, and the corresponding backward process, which involves training a denoising network to learn the score function to revert the forward process and sequentially recover the input.

Diffusion on Discrete Domains. To extend diffusion models to discrete domains, the forward/backward proocesses and the score matching objectives need to be redefined [14, 64, 103, 59]. Notably, BERT-type encoder-only models can be naturally modified into diffusion models, where masking & demasking operations serve as the noising/denoising counterparts [14, 54, 126]. Researchers have also explored score matching for discrete variables [90, 111, 23, 86].

Diffusion on Continuous Embedding Spaces. One straightforward idea to circumvent the discreteness is to learn diffusion models on the continuous embedding spaces. The pioneering work Diffusion-LM [77] proposes to jointly learn a token embedding together with the continuous diffusion model and develops the re-parametrization method and the clamping trick to reduce rounding error induced when projecting the embeddings back to discrete tokens. Afterwards, various work have been done to extend the Diffusion-LM framework to conditional generation and improve the token embeddings and training [42, 83, 41, 29, 49, 50, 15, 109, 35, 46].

Model Structures. As BERT-type models [33, 84] have demonstrated success in reconstructing the masked tokens from the remaining tokens, people adapt Transformers as the denoising networks for sequences, either operating directly on the discrete space with adding masks as one type of sample corruption, or on the continuous embedding space with the standard Gaussian noise corrupting.

C.3 Latent Diffusion Model for Sequences

Latent diffusion models for sequences are essentially sequence variational auto-encoders with diffusion latent priors, which contain a chain of conditional Guassian distributions, rather than a simple Gaussian distribution. The difference between latent diffusion models and the Diffusion-LM scheme [77] (Section C.2) lies in that latent diffusion models incorporate complex sequence (variational) auto-encoder models while Diffusion-LM only utilizes a token embedding layer.

Sequence Variational Auto-encoders. The design of sequence VAEs can be traced back to the era of RNNs [21, 68, 52]. Afterwards, OPTIMUS [75] is a sequence VAE that connects pretrained BERT and GPT-2, where the authors only use the `<cls>` token embedding of BERT to extract the latent vector. Similarly, Park and Lee [96] finetune a pretrained T5 model ino a sequence VAE and they use mean/max pooling over all the embeddings to obtain the latent.

A notorious challenge for sequence VAEs (with Gaussian priors) is the KL vanishing problem [21], where the encoder produces nearly identical latents for all the inputs (a.k.a. posterior-collaspe) and the decoder completely ignores the latents. The intuitive reason behind the KL vanishing problem is that the KL-regularization is too strong and the encoder fails to extract useful information for the decoder to reconstruct the sequence. Several tricks are proposed to address the issues, such as anealling the KL-regularization coefficient, memory dropout, etc [75, 125].

Latent Diffusion Model for Sequences. People have been building latent diffusion model for sequences, such as Versatile Diffusion [124], LD4LG [87], Planner [131]. With diffusion priors, the KL vanishing problem can be reduced by using a tiny or zero KL-regularization, as the diffusion models are strong enough for modeling complex multimodal distributions [100]. Versatile Diffusion [124] trains a textual diffusion model on top of the latent space of OPTIMUS, with a focus on the multimodality setting, e.g. image captions.

Addressing the similarities and differences of Planner [131].

Planner [131] proposes to use the first K token embeddings from the BERT model as the latent variable to increase the information within the latent embedding, but the BERT model needs to be fine-tuned. In comparsion, we keep the pretrained encoder model fixed to avoid the deterioration of the encoder, e.g., the encoder may be finetuned to forget the semantics, as the reconstruction task only requires the sequence auto-encoder to reproduce the raw input sequence. We train a Q-Former [76] to better summarize the entire embedding sequence through cross-attention. Our experiment on MRL/TE predictions also empirically verifies that the latent vectors contain biologically meaningful information.

Addressing the similarities and differences of LD4LG [87].

In methodology, the most similar work to us is LD4LG [87]. Specifically, LD4LG uses a pretrained encoder-decoder model such as BART [74] or T5 [97]. Similarly the encoder is fixed and a Perceiver Resampler [5] is trained to summarize the embeddings. Perceiver Resampler and Q-Former are essentially the same in this setting.

However, LD4LG incorporates an additional transformer Reconstruction Network, which transforms the latent embeddings for the fixed pretrained decoder to cross-attend to. Instead, we perform a simple linear transform to the latent vectors and use them as the soft prefix for the decoder. We initialize the decoder from stratch and train the decoder together with the Q-Former, so the decoder focuses only on translating the latent variables back to discrete sequences. Our design choice reduces the computation for the pipeline and also does not require a paired pretrained encoder-decoder model to work. Additionally, LD4LG solely focuses on NLP tasks such as summarization and translation. Our work focuses on RNA sequence modeling. We further show that our latent embeddings can be used to predict reward of interests (MRL and TE), and study using the trained reward model as guidance to optimize the RNA sequence.

C.4 Generaitve Models for Biological Sequences

Generative models have achieved considerable success in designing biological sequences [48], with applications ranging across protein and DNA. Among these, diffusion models have emerged as a particularly powerful tool for protein design, enabling the generation of large and diverse protein structures through guided iterative processes. This overcomes the limitations of traditional deep generative models like VAEs and GANs [7, 81, 36, 11, 44, 8, 66, 105], which often generate only small proteins or protein domains. For instance, ProteinSGM [72] employs diffusion models to generate protein structures from residue distances and angles, enabling both unconditional and targeted generation with post-processing by Rosetta [106]. Foldingdiff [123] uses a transformer model and DDPM to unconditionally model protein backbones using angles, while DiffSDS [40] enhances backbone structure quality by using an encoder-decoder language model to generate a 1D directional representation of invariant atom features for the diffusion process. The SE(3)-equivariant [114, 102] DDPMs [6] generate fully atomistic protein structures and sequences by equipping with invariant point attention [65] structural modules. Genie [80] improves noise prediction with geometric asymmetry in residue coordinates. SMCDiff [117] utilizes a GNN [104] within a diffusion model to address motif-scaffolding in proteins, utilizing conditional sampling formulated as a sequential Monte Carlo simulation solved by particle filtering. RFdiffusion [122] combines DDPM with RoseTTAFold for generating unconditional large protein generation and conditional scaffolding using a self-conditioning strategy. FrameDiff [127] efficiently generates protein backbones using SE(3)-diffusion with denoising score matching. Chroma [61] uses a GNN-based diffusion model to create large, diverse proteins and protein complexes conditioned on specific properties.

Guided diffusion models significantly enhance protein design by integrating generative and discriminative approaches to optimize sequence fitness. Diffusion Optimized Sampling (NOS) [45] boosts diffusion models with gradient-guided denoising and incorporates LaMBO, a Bayesian optimization process that accommodates multiple objectives and edit-based constraints. The EvoDiff framework [4] utilizes evolution-guided generation to create proteins that are structurally plausible and meet evolutionary and functional requirements, pushing protein engineering towards a programmable, sequence-first paradigm. Additionally, DiffAb [88] applies an antigen-specific guided diffusion method to iteratively refine antibody CDR sequences and structures based on the 3D structure of targeted antigens, facilitating the precise and efficient design of functionally optimized antibodies.

In the realm of DNA, the application of diffusion models is emerging but less extensive than in protein design. The Dirichlet Diffusion Score Model [15] specifically generates human promoter DNA sequences, exploring structural plausibility and diversity. Alongside, latent diffusion models such as DiscDiff [78] and DNA-Diffusion [37] have tailored the diffusion approach for discrete data, generating DNA sequences and regulatory elements that closely mimic natural genomic variability. These models transform discrete data into a continuous latent space, a necessary adaptation for applying diffusion principles that require differentiable probability density functions.

Despite the successes with proteins and DNA, the application of diffusion models to RNA, particularly 5'-UTR sequences, has been limited. UTRGAN [18] uses GANs to generate novel 5'-UTR sequences, establishing itself as the state-of-the-art generative model for this purpose. In this study, we focus on generating ncRNAs with latent diffusion models and adopt guided diffusion for high MRL/TE 5'-UTR generation as a downstream task.

D Additional Experimental Results

D.1 Additional Experimental Setup of RNA AutoEncoder

Details of the Models of Sequence AutoEncoder. We adopt the pretrained RNA-FM [26] as the encoder. RNA-FM is an encoder-only model trained with masked language modelling on non-coding RNAs (ncRNAs), which can extract biologically meaningful embeddings from ncRNAs and can be adapted for various downstream tasks, such as UTR function prediction, RNA-protein interaction prediction, secondary structure prediction, and etc. For the Q-Former, we modify the ESM-2 model architecture [82] by adding K trainable query token embeddings and adding cross-attention layers in each transformer block that attends to the embeddings from the RNA-FM. For the decoder model, we adopt the same model architecture of ProGen2-small [93] with our customized tokenizers for RNAs.

Notes on ESM-2 and ProGen2. ESM-2 [82] is an encoder-only model trained on protein sequences. The model architecture of ESM-2 improves upon BERT by replacing the absolute positional embeddings with Rotary Position Embedding (RoPE) [110]. The model architecture of ESM-2 has been adapted to model other biological sequences as in RNA-FM [26], UTR-LM [30], etc.

ProGen2 models [93] are originally designed for protein sequence modeling. The architecture differs from standard GPT-like models in (1) Rotary Position Embedding and (2) parallel feed-forward and attention computation.

Preprocessing. We adopt the following data preprocessing method following Chen et al. [26]. First, we replace base T with base U. Next, we eliminate identical sequences by applying the CD-HIT-EST clustering algorithm with a cut-off at 100% [38]. Afterwards, we remove RNA sequences with lengths > 768 and RNA sequences containing rare symbols other than A,U,G,C. The resulting dataset contains 1.1M RNA sequences. We adopt a simple tokenization method, where each base (A,U,G,C) is viewed as one token.

Training details. We split the dataset into the train split and the test split using a 9:1 ratio. We tune the number of query tokens K among $\{16, 32\}$ and the hidden dimension D of each latent vector among $\{40, 80, 160, 320\}$. We freeze the encoder and pretrain the Q-Former and the decoder from scratch using Adam optimizer with initial learning rate $1e{-}4$ and the cosine learning rate schedule. For $(K, D) = (16, 160), (16, 320), (32, 320)$, we pre-train the auto-encoder for 1 epoch; while for $(K, D) = (32, 160)$, the pre-training processes last 2 epochs and the remaining settings take 3 epochs to converge.

Experiments Compute Resources We train the auto-encoder on one NVIDIA H100-80G GPU. The estimated time for one epoch of training is around 8.5 hours.

D.2 Additional Experimental Setup of Latent Diffusion Models

Models. We adopt the following modifications to our Transformer denoising networks: (1) we add a skip-connection directly between the final output and the input, and (2) we use linear projection layers before and after the transformer to accommodate for different latent dimensions.

Training Details. We train a diffusion model on top of the latent space of each of the auto-encoders we obtain in Section 3. We use ϵ-prediction scheme so the denoising network predicts the added noise of the noisy sample. We adopt the linear noise schedule as DDPM [57] and set the number of diffusion steps $T = 1000$. For each auto-encoder setting, the diffusion model is trained for 3 epochs by using AdamW optimizer with initial learning rate $1e{-}4$ and the cosine learning rate schedule.

Generation Details. We use the DDIM sampling method [107] with $\eta = 0$ (deterministic) and 50 denoising steps to get samples from our latent diffusion models. Then we transform the latent vectors into soft prompts for the decoder and auto-regressively generate a sample sequence nucleus-by-nucleus with top-$p = 0.95$ and temperature 1. We remove any invalid sample sequence containing special tokens in the middle (e.g., <bos>).

Experiments Compute Resources We train the diffusion model on one NVIDIA H100-80G GPU. The estimated time for one epoch of pre-training on 1.1M ncRNA dataset is 3.5 hours; while for one-epoch fine-tuning on 205K 5'UTR dataset, the training process approximately costs 40 minutes.

D.3 Additional Experimental Setup of Latent Reward Models

Additional Details on Datasets. For the *TE prediction* task, the three endogenous human 5' UTR datasets we used are derived from different cell lines or tissue types: human embryonic kidney 293T (HEK), human prostate cancer cell line PC3 (PC3), and human muscle tissue (Muscle). These datasets contained 14,410, 12,579, and 1,257 sequences, respectively. In training the reward model, we combined these three datasets into a larger one.

Model Architecture. We build Convolutional ResNets on top of the latent space to predict the rewards. Each layer in this network, referred to as a ResBlock, consists of two 1D convolutional layers activated by ReLU functions and followed by Batch Normalization. Convolution operations are applied along the query dimension, with the number of input channels matching the embedding dimension of the output from the querying transformer, and the channel count doubles between ResBlocks. We employ the smallest kernel size of 3 to accommodate our constrained query length,

along with padding of 1 to preserve dimensional integrity. The output from the ResNet is averaged across the query dimension before being processed through a Multi-Layer Perceptron (MLP) network to produce the final scalar output. To enhance model robustness, we incorporate a dropout layer after each Resblock to prevent overfitting — a strategy we found more beneficial than downsizing the model, which could compromise its learning capacity.

Training Details. We trained the reward model for 50 epochs on the MRL prediction dataset and for 100 epochs on the TE datasets. We utilized the AdamW optimizer, setting the initial learning rate 1e-4 with a warm-up phase of 3,000 steps, followed by a cosine learning rate schedule. We implemented a dropout rate of 0.2 and an Adam weight decay of 0.02 to combat overfitting. We applied data augmentation to further ensure robust learning by introducing random Gaussian noise to the input latent vectors. We adopted the same noise scheduler as used in diffusion models. Instead of sampling noise from all 1,000 steps, we restricted sampling to only the first 10 steps.

Experiments Compute Resources We train the reward models on one NVIDIA A100-80G GPU. The estimated time for training one reward model is 5 hours for the MRL task (50 epochs of 83k samples) and 2 hours for the TE task (100 epochs of 28k sequences).

D.4 Ablations Studies on RNA-FM

We validate the necessity of the pre-trained RNA-FM in this section. When we replace the pre-trained encoder RNA-FM with naive one-hot embeddings, both the reconstruction capability of the sequence autoencoder and the reward model performance are affected, where we observe significant increases in the reconstruction error and declines in the R-values of the reward model (Table 4).

Table 4: **Ablations on RNA-FM**. We replace RNA-FM with naive one-hot token embedding and test the performances on sequence autoencoder and MRL reward model training under different Q-former settings (K, D). NLL: reconstruction error, NED: normalized edit distance.

(a) (K, D) = (32, 160).

	Sequence AutoEncoder		MRL Reward Model	
	NLL ↓	NED ↓	Test Spearman R ↑	Test Pearson R ↑
w/o RNA-FM	0.0018	0.10 % ± 4.91 %	0.5983	0.6969
w/ RNA-FM	**0.0003**	**0.02% ± 0.40 %**	**0.6901**	**0.7655**

(b) (K, D) = (16, 160).

	Sequence AutoEncoder		MRL Reward Model	
	NLL ↓	NED ↓	Test Spearman R ↑	Test Pearson R ↑
w/o RNA-FM	0.1755	30.30 % ± 36.13 %	0.2526	0.3377
w/ RNA-FM	**0.0156**	**2.19% ± 5.50 %**	**0.3699**	**0.4599**

D.5 Evaluation of 5'-UTR `RNAdiffusion` Fine-tuning

We compare the samples generated by fine-tuned latent diffusion models with natural 5'UTR reference dataset and random samples following the same evaluation procedure in Section 4.1. The results are shown in Figure 6.

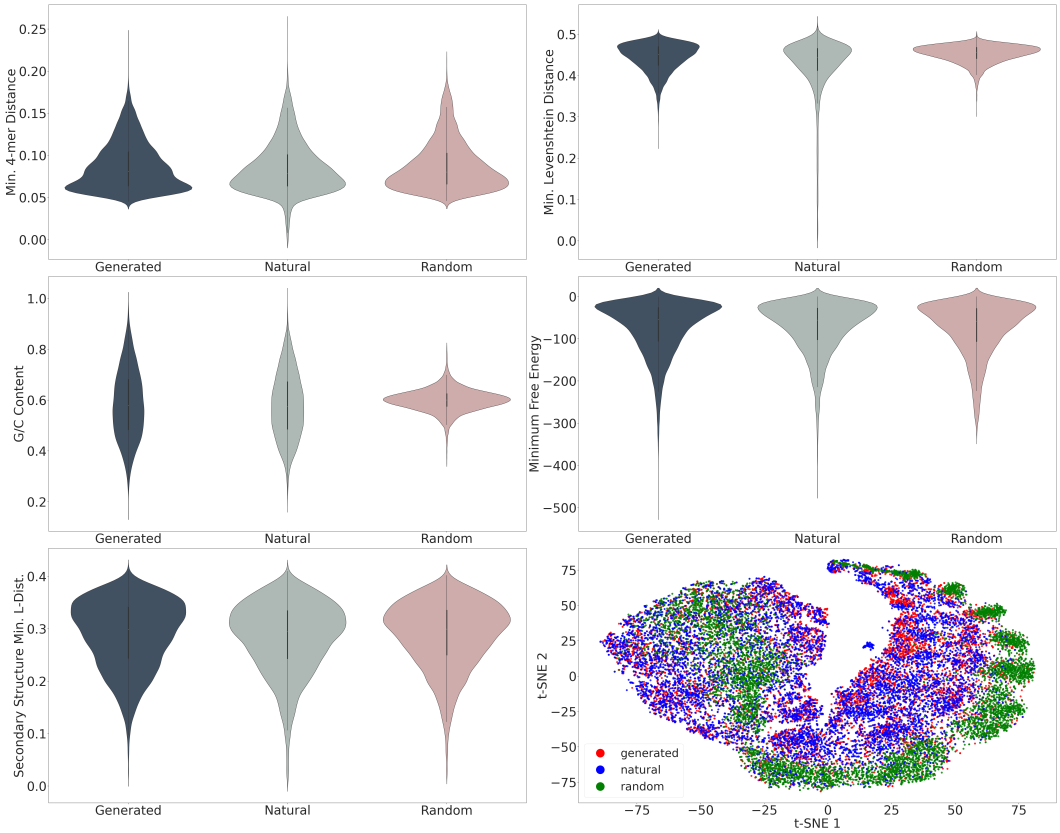
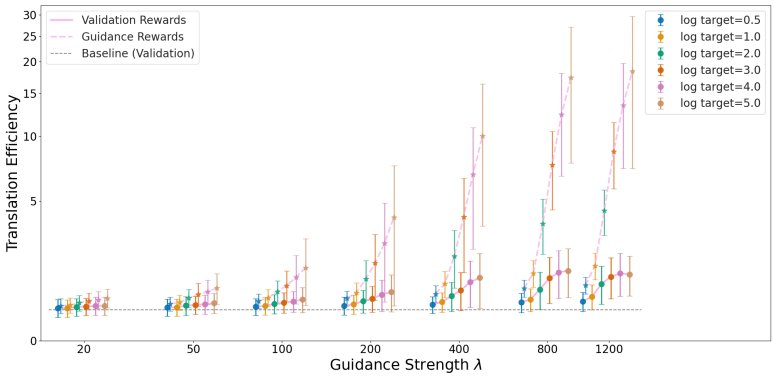


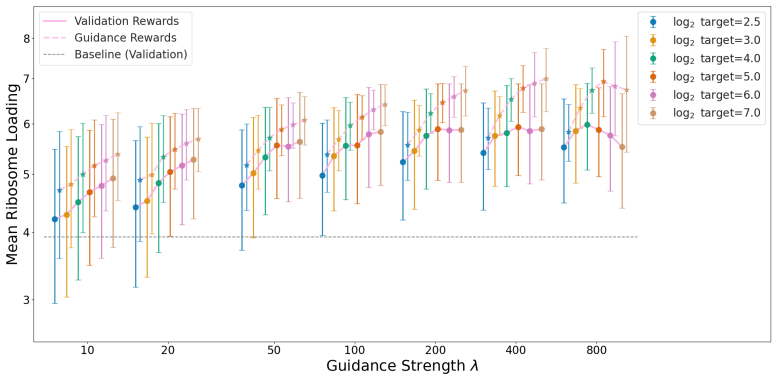
Figure 6: Comparing the sequences generated by the *fine-tuned* latent diffusion model with natural 5'UTR sequences (test set) and random sequences in terms of biological metrics. (a) minimum 4-mer distances. (b) minimum sequence Levenshtein distances. (c) G/C content ratios. (d) minimum free energy. (e) minimum second structure Levenshtein distances. (f) t-SNE of latent space. The random sequences follow the same variable length distribution as natural UTR sequences, and each nucleotide is sampled at a G/C content ratio of 0.6, resembling the average level of natural UTR.

D.6 Ablations on Guidance Parameters for Guided 5'UTR Generation

Guidance Parameter Search. We tune the guidance for generation by varying the values of guidance strength λ and target reward r^* as of Equation 5, and plot the rewards of guided generation under different guidance parameters λ and r^* in Figure 7. Among all the guidance settings, the best *validation* TE (1.84 ± 0.82) is achieved with the moderate guidance strength as 800 and the log-target value as 5 , showing **166.7%** improvement compared to the *validation* TE of unguided generation (0.69 ± 0.22). Under the MRL-guidance setting of $\lambda = 800$ and $\log r^* = 4$, mean *validation* MRL of the generated sequences increases from 3.92 ± 1.31 to 5.98 ± 0.90 with maximum relative increase of **52.6 %**.



(a) TE Optimization



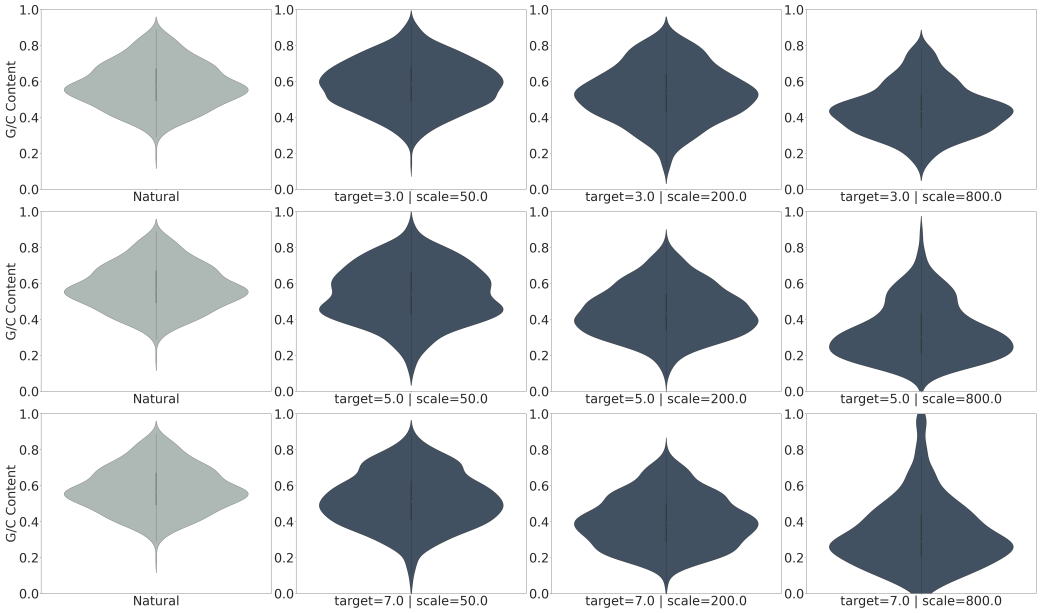
(b) MRL Optimization

Figure 7: **Performances of reward-guided diffusion on TE and MRL tasks.** Baseline results refer to the evaluation of unguided generation with validation reward models. Guidance Reward Evaluation indicates the assessment with the same reward models which are used for guided generation. All evaluated sequences are over 30-bp.

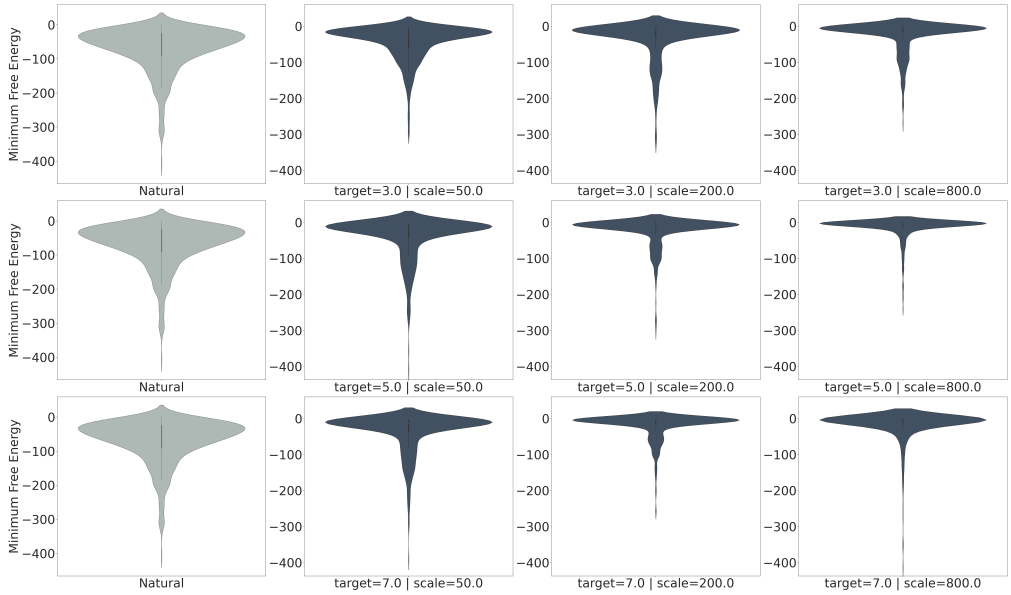
Biological Metrics. We compare the generated distributions under different guidance strengths in terms of biological metrics (see Figure 8). With the λ and r^* increasing, the disparities in G/C content and Minimum Free Energy (MFE) between the generated sequences and natural sequences become larger. Specifically, the distribution of G/C content is shifting downwards while the mean MFE is increasing, indicating that the stability of generated sequences has been affected.

Reward Hacking in Reward-guided Diffusions. As shown in Figure 7, at a mild guidance strength, increasing the target TE or MRL values generally brings the reward improvements of generated sequences, demonstrating the efficacy of our reward-guided generation approach. For higher guidance strength, the improvements can be *artificially* amplified when evaluated with the same reward model used for guidance. Continuing to increase λ and r^* will eventually hurt the cross-validation performance, since the generation process is over-adapted towards the imperfect guidance reward

model, which is a general phenomenon known as reward hacking [39, 32, 130]. Additionally, increasing λ or r^* may cause the distribution shift of biological properties of sequences such as G/C content and MFE (see Figure 8), affecting the stability of UTR sequences. Thus, adopting moderate λ's and r^*'s would be important for reward-guided generation in terms of the *trade-off* between achieving higher reward values and maintaining generation qualities.



(a) G/C Content



(b) Minimum Free Energy

Figure 8: Biological performance of the sequences generated with the MRL-guided diffusion model. The leftmost column shows the result of natural 5' UTR sequences. In the 2nd to 4th column, the guidance strength increases from 50 to 200 and 800. From top to bottom, each row corresponds to log_2-MRL targets of 3, 5 and 7.