To Optimize, Not to Invent: RNAGenScape for mRNA Sequence Generation and Optimization Without *de novo* Design

Danqi Liao [*1] **Chen Liu** [*1] **Xingzhi Sun** [1] **Dié Tang** [2] **Haochen Wang** [2]
Scott Youlten [2] **Antonio J. Giraldez** [2] **Smita Krishnaswamy** [1 2]

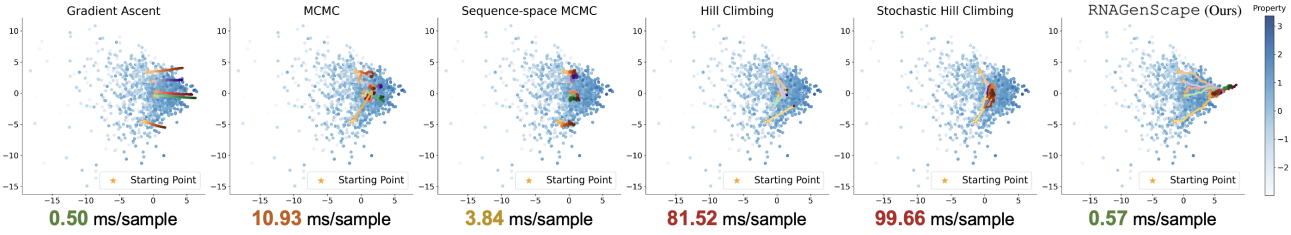

Figure 1. Comparison of latent space trajectories over 100 optimization steps for RNAGenScape and existing methods, visualized using the top two principal components. Each trajectory is shown as a line fading from bright to dark in a consistent color. Although most methods progress in generally correct directions, our proposed RNAGenScape follows more visually reasonable paths and achieves more optimized results. Besides, RNAGenScape is among the most efficient methods during inference.

Abstract

Designing mRNA sequences with optimized biological properties remains a fundamental challenge in synthetic biology and therapeutic development. Deep generative models have enabled data driven sequence generation, but most are designed for *de novo* generation, meaning generating entirely from scratch. However, refine existing sequences, interpolate between sequences, or producing interpretable optimization steps remain important tasks in mRNA design. In this work, we introduce RNAGenScape, a framework for mRNA design that combines Langevin-dynamics with a learned manifold projector. Operating entirely in the latent space of a pretrained encoder, RNAGenScape updates latent representations using property guided gradients and then projects each noisy step back onto the learned manifold to ensure biological plausibility. This approach enables property-guided optimization, smooth interpolation between arbitrary mRNA sequences, and tracking of interpretable latent trajectories, all without requiring explicit the density estimation or the score learning typically utilized in score-matching diffusion models. We demonstrate results on zebrafish mRNA datasets. We show that RNAGenScape can continuously steer sequences toward target properties while remaining close to natural sequences, and can generate intermediate variants along each trajectory. Our results establish a scalable and generalizable paradigm for controllable mRNA design and latent space exploration in biological sequence modeling.

1. Introduction

Designing biological sequences using machine learning has emerged as a critical objective in computational biology. Recent efforts have focused on *de novo* design (Prykhodko et al., 2019; Méndez-Lucio et al., 2020; Meyers et al., 2021; Munson et al., 2024; Watson et al., 2023), that is, creating protein and mRNA sequences from scratch. These methods are typically evaluated on the novelty,

[*]Equal contribution [1]Department of Computer Science, Yale University, CT, USA [2]Department of Genetics, Yale University, CT, USA. Correspondence to: Smita Krishnaswamy <smita.krishnaswamy@yale.edu>, Antonio Giraldez <antonio.giraldez@yale.edu>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

diversity, and biological plausibility of the generated sequences. Empowered by recent advances in deep generative modeling (Goodfellow et al., 2020; Ho et al., 2020; Lipman et al., 2022), recent works have demonstrated strong generative performance and have opened promising new directions in synthetic biology (Méndez-Lucio et al., 2020; Dauparas et al., 2022; Repecka et al., 2021; Madani et al., 2023; Watson et al., 2023).

However, *de novo* design typically operates without reference to natural biological sequences. The generated outputs are often detached from biological context and offer limited insight into how specific changes in sequence affect function, or vice versa. In many cases, the designed sequences are difficult to interpret or validate experimentally, and do not reflect the constraints or patterns observed in the real data. As a result, these models can achieve high scores on synthetic benchmarks while failing to advance biological understanding or utility.

In this work, we propose `RNAGenScape`, a novel approach to refining existing sequences on the learned latent manifold. Rather than generating sequences from scratch, we begin with real mRNA untranslated region (UTR) sequences and optimize them to improve a desired property, such as translation efficiency. We achieve this by designing a Langevin dynamics that can steer existing points toward more optimized regions along the latent manifold. This allows us to optimize from real sequences while preserving interpretability. By tracing the entire sequence optimization process, we can examine intermediate variants, measure how specific edits influence the target properties, gain a more detailed view of the sequence landscape, and discover new biological insights.

In summary, our main contributions are as follows.

1. We propose a Langevin-dynamics framework that enables interpolation and continuous property-guided optimization of mRNA sequences starting from real data points, rather than generating from scratch. This framework shifts the focus from generation to refinement and offers a path toward biologically grounded sequence modeling.
2. We introduce a learned manifold-projection mechanism using a denoising autoencoder to constrain the sampling process and ensure that updates remain close to the biological data manifold.
3. We demonstrate that this combination yields interpretable trajectories in the latent space for both property optimization and target-directed interpolation, enabling analysis of how sequence edits affect properties at each step.
4. We provide empirical evidence that our method improves target properties (e.g., translation efficiency) while maintaining manifold fidelity, outperforming various optimization and generation methods.

2. Preliminaries

2.1. Manifold hypothesis and manifold learning

The manifold hypothesis (Cayton et al., 2008; Narayanan & Mitter, 2010; Fefferman et al., 2016) posits that high-dimensional data commonly encountered in machine learning tasks lie near a low-dimensional manifold embedded in the ambient space. Under this assumption, each observation $x_i \in \mathbb{R}^n$ arises from a smooth nonlinear map $\mathbf{f} : \mathcal{M}^d \to \mathbb{R}^n$ applied to a latent variable $z_i \in \mathcal{M}^d$, where $\mathcal{M}^d$ is a d dimensional manifold with $d \ll n$. Manifold learning methods aim to recover this latent structure by constructing representations that preserve the intrinsic geometry of the data (Van Dijk et al., 2018; Moon et al., 2019; Burkhardt et al., 2021; Liu et al., 2024; Liao et al., 2024; Liu et al., 2025a;b; Sun et al., 2025).

A point is considered "on-manifold", if it lies within the range of the generative map $\mathbf{f}$, reflecting the learned structure of the data (Rifai et al., 2011). In contrast, off-manifold points deviate from this structure and may correspond to invalid samples or adversarial perturbations (Zhang et al., 2022; Li et al., 2023). Projecting off-manifold points back onto the manifold is critical in tasks that require robustness or structure-aware optimization, where staying close to the data manifold is desirable (He et al., 2023b).

2.2. Langevin-dynamics and beyond

Diffusion Models (Ho et al., 2020) are generative frameworks that learn a data distribution $p(x)$ by reversing a fixed Markov diffusion process of length T. Starting from Gaussian noise, they are trained to iteratively denoise samples through a sequence of learned denoising functions over T steps. The training objective (Eqn (1)) is a reweighted form of the variational lower bound, closely related to denoising score matching (Song et al., 2021).

$$\mathcal{L}_{\text{DM}} := \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[||\epsilon - \epsilon_\theta(x_t, t)||_2^2\right] \tag{1}$$

Latent Diffusion Models (Rombach et al., 2022) present an extension of the concept. Instead of performing the reverse diffusion process in the data space, they operate in a latent space after embedding the data with an encoder $\mathcal{E}$, where $z = \mathcal{E}(x)$. The modified objective is shown in Eqn (2).

$$\mathcal{L}_{\text{LDM}} := \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,1),t}\left[||\epsilon - \epsilon_\theta(z_t, t)||_2^2\right] \tag{2}$$

Langevin Dynamics (Song & Ermon, 2019) has been employed in generative models to sample from high-dimensional data distributions using only an estimate of

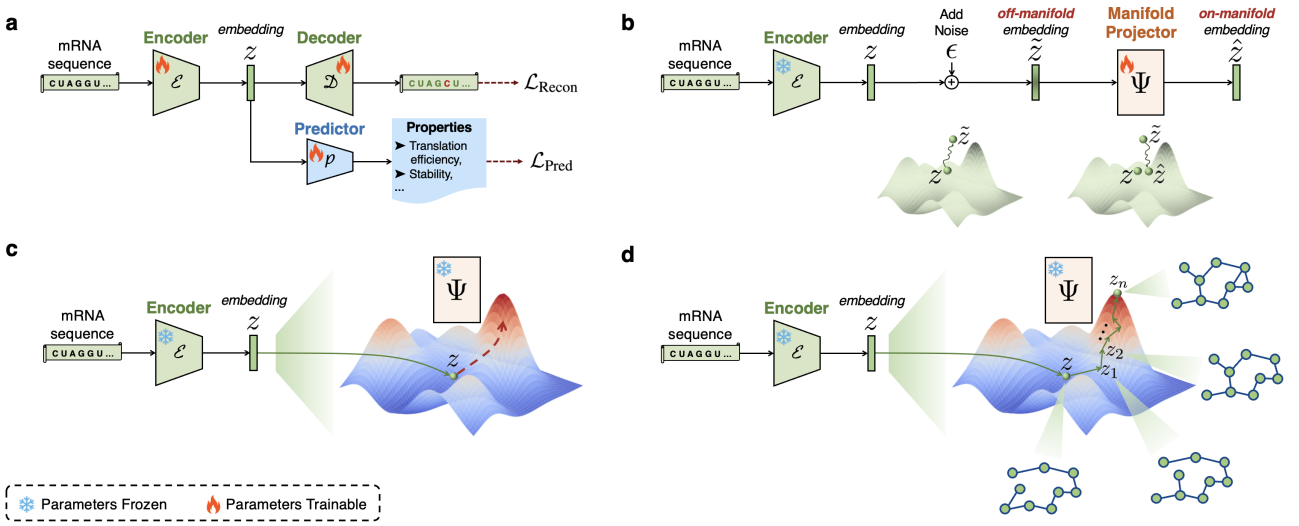

Figure 2. Schematic of RNAGenScape. **(a)** We first train an organized latent space for mRNA sequences by jointly optimizing reconstruction and property prediction objectives. **(b)** We then train a manifold projector to project perturbed samples back to the embedding manifold, while the encoder's weights are frozen. **(c)** We can use the encoder and the manifold projector to optimize the properties of given input mRNA sequences. **(d)** Notably, the intermediate products during the optimization process can also be generated.

the score function $\nabla_x \log p(x)$. In particular, it first trains a neural network $\boldsymbol{s_\theta}$ to approximate the score function of data perturbed by Gaussian noise. Sampling is then performed via annealed Langevin dynamics, given by Eqn (3).

$$\tilde{x}_t = \tilde{x}_{t-1} + \frac{\eta_i}{2}\boldsymbol{s_\theta}(\tilde{x}_{t-1}, \sigma_i) + \sqrt{\eta_i}\boldsymbol{z_t} \tag{3}$$

Here, $\boldsymbol{s_\theta}(\tilde{x}_{t-1}, \sigma_i)$ is the learned score function at noise level σ_i, and η_i is the step size at that level. By gradually annealing from high to low noise, this procedure enables generation of high-quality samples without an explicit likelihood or energy model.

Neural Stochastic Differential Equations (Kidger et al., 2021), abbreviated as neural SDEs, are differential equations simultaneously modeling two terms: a drift term $f(\cdot)$ depicting the true time-varying dynamics of the variable, and a diffusion term $g(\cdot)$ representing stochasticity using the Brownian motion W_t (Eqn (4)). From a high level, Langevin dynamics is a special case of neural SDEs after discretization.

$$\mathrm{d}X_t = f(t, X_t)\mathrm{d}t + g(t, X_t) \circ \mathrm{d}W_t \tag{4}$$

3. Methods

In this section, we will describe RNAGenScape in detail. The key components of our framework are:

1. An autoencoder whose latent space is organized by the target property (Section 3.1 and Figure 2a),
2. A manifold projector that brings perturbed embeddings to the data manifold (Section 3.2 and Figure 2b), and
3. Property-guided on-manifold Langevin dynamics in the latent space (Section 3.3 and Figure 2c-d).

Once trained, these components allows RNAGenScape to (1) optimize the target property of a given sequence (Section 3.4 and Figure 3) and (2) interpolate between existing sequences (Section 3.5 and Figure 4).

3.1. Learning a latent space organized by property

We begin by training an **organized autoencoder (OAE)**, where the latent space is implicitly structured via supervision from a property prediction task (Figure 2a). Similar to a vanilla autoencoder (Hinton & Salakhutdinov, 2006), the encoder $\mathcal{E}$ maps the input mRNA sequence x to a latent representation z, which is decoded by $\mathcal{D}$ back to the sequence space. In addition to this standard architecture, a predictor $\mathcal{P}$ infers properties from the embedding z (Eqn (5)).

The latent space $\mathcal{Z}$ is thus shaped by jointly optimizing the reconstruction loss (Eqn (6)) and the prediction loss (Eqn (7)), encouraging it to capture sequence-relevant information while being organized by the target properties.

$$z = \mathcal{E}(x), \quad \hat{x} = \mathcal{D}(z), \quad \hat{y} = \mathcal{P}(z) \tag{5}$$

$$\mathcal{L}_{\text{Recon}} = \frac{1}{N}\sum_i^N ||\hat{x}_i, x_i||_2^2 \tag{6}$$

$$\mathcal{L}_{\text{Pred}} = \frac{1}{N}\sum_i^N ||\hat{y}_i, y_i||_2^2 \tag{7}$$

3.2. Training a manifold projector

To ensure that the generated trajectories remain close to the latent data manifold, we introduce a manifold projector Ψ, implemented as a denoising autoencoder (DAE) (Vincent et al., 2008). As shown in Figure 2b, given a slightly perturbed input $\tilde{z}$ derived from a clean data point z on the manifold, Ψ projects $\tilde{z}$ back onto or near the manifold.

We train Ψ by corrupting the input samples and minimizing the reconstruction error between the projected noisy inputs and their clean counterparts (Eqn (8)).

$$\mathcal{L}_\Psi = \frac{1}{N}\sum_{i}^{N} ||\Psi(C(\tilde{z}|z)) - z||_2^2, \tag{8}$$

Here, $C(\tilde{z}|z)$ is the conditional distribution of the corrupted data. We also incorporate **multi-step denoising** to help improve the performance of the manifold projector Ψ. The training procedure of Ψ is described in Algorithm 1.

Algorithm 1 Training Denoiser Ψ for Manifold Projection

Input: Dataset $\mathcal{Z} = \{z_i\}_{i=1}^N$, denoiser Ψ, noise levels $\{\sigma_1, \ldots, \sigma_K\}$, denoising steps K, learning rate η
for each z_i in minibatch $\{z_i\}_{i=1}^B \subset \mathcal{Z}$ **do**
  Initialize $\tilde{z}^{(0)} \leftarrow z_i$
  for $k = 1$ to K **do**
    $\tilde{z}^{(k)} \sim C(\tilde{\mathcal{Z}}|\tilde{\mathcal{Z}}^{(k-1)}, \sigma_k)$
    $\mathcal{L}^{(k)} = \|\Psi(\tilde{z}^{(k)}) - \tilde{z}^{(k-1)}\|_2^2$
  end for
  $\mathcal{L}_i = \sum_{k=1}^K \mathcal{L}^{(k)}$
  $\Psi \leftarrow \Psi - \eta\nabla_\Psi \left(\frac{1}{B}\sum_{i=1}^B \mathcal{L}_i\right)$
end for

3.3. Property-guided on-manifold Langevin dynamics

Next, we introduce a novel property-guided on-manifold Langevin-dynamics framework that iteratively adjusts latent embeddings to optimize a target property, while ensuring the resulting trajectories remain close to the data manifold.

Given a pretrained encoder $\mathcal{E}$, a property predictor $\mathcal{P}$, and a manifold projector Ψ, our Langevin-dynamics framework optimizes sequences for a target property. Starting from the latent embedding $z = \mathcal{E}(x)$ of a sequence x, we iteratively update it using a gradient-based drift term $f(z)$, inject Gaussian noise ϵ, and apply a manifold projection $\Psi(\cdot)$ to ensure biological plausibility and interpretability.

We define the update rule as follows.

$$\begin{aligned} \mathrm{d}z_t &= \eta\nabla_z f(z_t) + \sqrt{2\eta\tau}\cdot\epsilon_t, \\ \epsilon_t &\sim \mathcal{N}(0, I) \end{aligned} \tag{9}$$

$$z_{t+1} = \Psi(z_t + \mathrm{d}z_t) \tag{10}$$

The temperature τ can be tuned to control the scale of the random noise during each update. A smaller τ makes a more focused update, while a larger one makes more diverse samples.

Here, the drift function $f(z)$ guides the movement along the property gradient given by the pretrained property predictor $\mathcal{P}$ while being regularized by a sparsity term $f_{\text{sparsity}}(z)$ that encourages exploration of sparse regions in the latent space.

$$f(z) = \mathcal{P}(z) + \lambda_{\text{sparsity}} f_{\text{sparsity}}(z) \tag{11}$$

To construct the sparsity estimator f_{sparsity}, we first fit a kernel $\mathcal{K}$ to the input batch data to encode pairwise affinities, and use the resulting affinity matrix to approximate the local sparsity of the input region. Specifically, we define f_{sparsity} as the negative row sum of the affinity matrix:

$$f_{\text{sparsity}}(z) = -\|\mathcal{K}(z, \cdot)\|_1 \tag{12}$$

We adopt an anisotropic kernel (Coifman & Lafon, 2006) on the batch latent embeddings z:

$$\begin{aligned} \mathcal{K}(z_1, z_2) &= \frac{\mathcal{G}(z_1, z_2)}{\|\mathcal{G}(z_1, \cdot)\|_1^\alpha \|\mathcal{G}(z_2, \cdot)\|_1^\alpha}, \text{ where} \\ \mathcal{G}(z_1, z_2) &= e^{-\frac{\|z_1 - z_2\|^2}{\sigma}} \end{aligned} \tag{13}$$

Here, $0 \le \alpha \le 1$ controls the separation of geometry from data density. With $\alpha = 0$ producing the classic Gaussian kernel, and $\alpha = 1$ completely removing density and providing a geometric equivalent to uniform sampling of the underlying manifold.

The manifold projector Ψ is applied after each update to ensure that each step remains near the biologically valid latent manifold, enabling interpretable and controllable generation trajectories.

3.4. Optimizing the property of a sequence

With the pretrained components $\mathcal{E}$, $\mathcal{P}$ and Ψ, we can optimize the target property of any given sequence using the Langevin dynamics described in Section 3.3. Notably, optimization entails both maximization and minimization: users can choose to increase or decrease the target property, depending on the application.

3.5. Interpolating between sequences

In addition to optimizing a single sequence, we can interpolate between two existing sequences by guiding the latent embedding of one sequence toward that of another. Specifically, given a source sequence x_{source} and a target sequence x_{target}, we first obtain their latent embeddings via the encoder: $z_{\text{source}} = \mathcal{E}(x_{\text{source}})$ and $z_{\text{target}} = \mathcal{E}(x_{\text{target}})$.

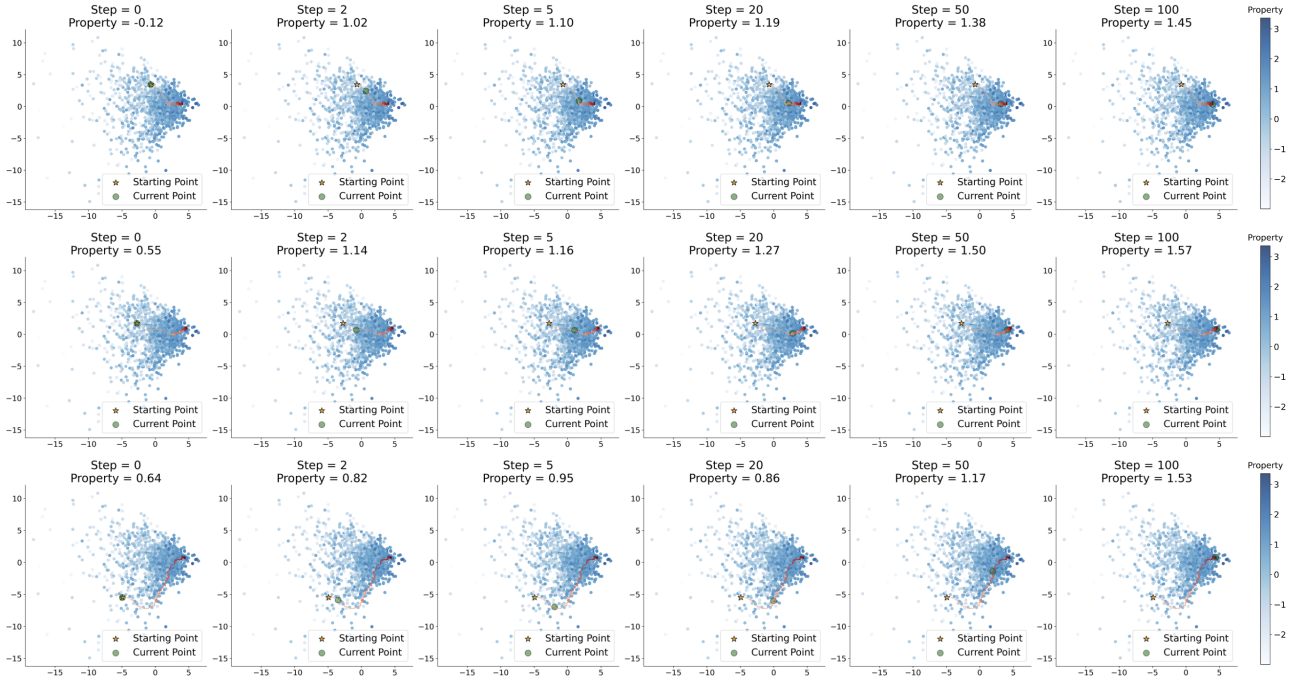

Figure 3. Latent space trajectories of RNAGenScape over 100 optimization steps, visualized using the top two principal components. The trajectories follow smooth, reasonable and coherent paths with steady improvement in the target property.

We then perform property-guided on-manifold Langevin dynamics starting from $z = z_{\text{source}}$, with an added force term that drives the embedding toward z_{target}:

$$F(z, z_{\text{target}}) = -\frac{z - z_{\text{target}}}{\|z - z_{\text{target}}\|_2} \tag{14}$$

In this case, we slightly modify the update rule of the Langevin dynamics (Eqn (9)) as follows:

$$\mathrm{d}z_t = \eta \left(\nabla_z f(z_t) + \lambda_{\text{interp}} F(z_t)\right) + \sqrt{2\eta\tau} \cdot \epsilon_t \tag{15}$$

By setting the interpolation weight $\lambda_{\text{interp}} > 0$ in Eqn (15), we add a directional bias that drives the latent trajectory toward the target point. All other components of the Langevin framework remain the same, including the use of Gaussian noise for exploration and the manifold projection Ψ to maintain plausibility.

4. Empirical Results

In this section, we demonstrate the effectiveness of RNAGenScape on two key tasks: (1) RNA sequence optimization and (2) RNA sequence interpolation.

The first task is broadly relevant to applications in therapeutics and synthetic biology. For example, enhancing the translation efficiency and stability of an mRNA vaccine can increase its protein yield and persistence, thereby boosting therapeutic efficacy while reducing the required dose.

The second task facilitates the exploration of intermediate variants. This can provide insights into the functional and structural landscape of regulatory elements within the RNAs of interest.

Our results demonstrate that:

1. RNAGenScape enables smooth latent space trajectories that stay on the data manifold (Section 4.2),
2. it achieves stronger property improvements than both *de novo* generative models and optimization baselines (Section 4.3),
3. it does so with substantially higher efficiency and directionality (Section 4.4), and
4. it allows interpolation between arbitrary sequences (Section 4.5).

4.1. Experimental Settings

Datasets In this study, we focus on the 5' untranslated region (UTR) of mRNAs, a non-coding segment located upstream of the coding sequence. 5' UTR plays a crucial role in regulating translation initiation and protein expression levels, without modifying the encoded protein sequence. These properties make a 5' UTR a biologically meaningful target for sequence optimization in synthetic biology and therapeutic applications.

We trained and evaluated RNAGenScape on five diverse

Table 1. Quantitative comparison of *de novo* sequence generation and property optimization methods. Our proposed RNAGenScape outperforms others in property optimization while also being inference-efficient. Top performers among property optimization methods are bolded. For *de novo* generative models, the optimization columns are grayed out, as they cannot explicitly steer properties; reported values instead reflect samples from their learned distributions. Note that in our context lower $\mathcal{W}_2$ distances do not necessarily indicate better performance, as property-optimized distributions are not expected to replicate the data distribution.

Methods ↓ Metrics →	Inference Speed	Distribution Alignment	Property Optimization (+)		Property Optimization (−)	
	ms/sample ↓	$\mathcal{W}_2$ distance	median Δproperty ↑	% mRNAs improved ↑	median Δproperty ↓	% mRNAs improved ↑
***de novo* generative models**						
VAE (Kingma et al., 2013)	0.06	0.60	0.13	65.7	0.13	34.3
WGAN-GP (Gulrajani et al., 2017)	0.07	0.69	0.72	53.0	0.72	47.0
DDPM (Ho et al., 2020)	0.91	0.62	-1.06	39.2	-1.06	60.8
LDM (Rombach et al., 2022)	0.74	0.62	-0.78	46.2	-0.78	53.8
FM (Lipman et al., 2022)	5.82	0.62	-1.09	34.7	-1.09	65.3
Property optimization methods						
OAE + Gradient Ascent	**0.50**	0.28	0.30	97.8	-0.06	73.5
OAE + MCMC	10.93	0.47	0.25	83.5	-0.58	95.1
OAE + Sequence-space MCMC	3.84	0.44	0.29	90.4	0.13	28.0
OAE + Hill Climbing	81.52	0.56	0.09	62.1	-0.28	85.0
OAE + Stochastic Hill Climbing	99.66	0.60	0.02	53.6	-0.46	86.2
OAE + RNAGenScape **(Ours)**	**0.57**	0.67	**0.79**	**98.9**	**-1.17**	**99.5**

5' UTR datasets of zebrafish, experimentally collected using Nascent Peptide Translating Ribosome Affinity Purification (NaP-TRAP (Strayer et al., 2023)), a massively parallel reporter assay for quantifying translation control. These datasets span multiple developmental stages and experimental conditions, including 2 hours post-fertilization (hpf) and 6 hpf with both polyadenylated and SV40 late polyadenylation signal contexts, as well as a 12 hpf dataset using HEK293T cells expressing zebrafish 5' UTRs. Each dataset has around 11,000 5' UTR sequences of length 124.

We used translation efficiency as the optimization target. Translation efficiency reflects how many copies of proteins each mRNA produces, and maximizing this property can enhance protein production for therapeutic, synthetic biology, or developmental applications.

Baselines We compared our method with a range of popular *de novo* generative modeling approaches, including variational autoencoder (VAE) (Kingma et al., 2013), Wasserstein generative adversarial network with gradient penalty regularization (WGAN-GP) (Gulrajani et al., 2017), denoising diffusion probabilistic model (DDPM) (Ho et al., 2020), latent diffusion model (LDM) (Rombach et al., 2022), and flow matching (FM) (Rombach et al., 2022). To benchmark against optimization-based methods, we also experiments with gradient ascent (Zinkevich, 2003), Markov chain Monte Carlo (MCMC) (Brooks, 1998; Andrieu et al., 2003), and hill climbing (Selman & Gomes, 2006). All optimization baselines were GPU-compatible adaptions from the implementation in (Castro et al., 2022).

Hardware All experiments were carried out on the five aforementioned datasets with three random seeds, and we reported the averaged results. The evaluations were performed on a single NVIDIA A100 GPU. With that said, RNAGenScape requires minimal GPU memory and can be run efficiently on more modest hardware.

Evaluation Since the optimization process could and should result in mRNA sequences not covered by the dataset, to quantify their properties, we trained a separate property prediction model $\mathcal{P}_{\text{true}}(x)$ to serve as a proxy of the ground truth. Note that $\mathcal{P}_{\text{true}}(x)$ is not accessible by RNAGenScape or competing methods during optimization, and is only used for evaluation.

4.2. RNAGenScape produces structured, data-aligned trajectories

RNAGenScape operates within a learned latent space that reflects the manifold of real biological sequences. As shown in Figure 1, the optimization trajectories of different methods tend to point in generally correct directions, but often veer off-manifold, stagnate in local optimal regions, or oscillate erratically. In contrast, RNAGenScape consistently traces smooth and data-consistent paths that preserve proximity to the natural sequence manifold.

To further illustrate this behavior, we visualize individual optimization runs in Figure 3. Each trajectory exhibits (roughly) monotonic increases in the target property while remaining near regions populated by real sequences. These trajectories are direct consequences of the manifold-constrained dynamics, which guided each step toward high-property regions while staying on the manifold.

Importantly, all intermediate steps during optimization can be decoded into mRNA sequences, allowing researchers to examine how sequences evolve step by step as specific properties are optimized. A deeper analysis of these

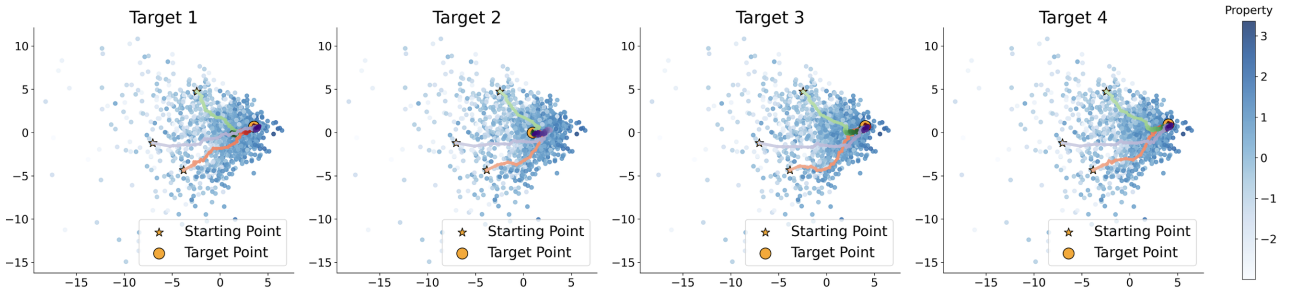

Figure 4. Latent space interpolation trajectories of RNAGenScape over 100 optimization steps, visualized using the top two principal components. Each trajectory is shown as a line fading from bright to dark in a consistent color. By incorporating the directional force toward the target, RNAGenScape produces smooth and coherent paths between arbitrary input–target pairs on the manifold.

trajectories is left for future work.

4.3. Superior property optimization in both directions

We quantitatively compare RNAGenScape against a range of *de novo* generative models and optimization baselines (Table 1). Although *de novo* approaches are effective in modeling the data distribution, they offer limited explicit control over the target properties. As a result, their performance in property optimization is limited.

In contrast, RNAGenScape consistently achieves the strongest performance among property optimization methods. It achieves the highest median property change and the highest success rate in both positive and negative directions. In particular, its median property improvement is approximately twice that of the runner-up, and its success rate is the highest among all methods compared.

For sanity check, we also computed the 2-Wasserstein ($\mathcal{W}_2$) distance between the generated sequence distribution and the test set sequence distribution. We note that lower $\mathcal{W}_2$ distances to the test distribution should not be interpreted as better in this context, since property optimization naturally shifts the output distribution away from the original data distribution.

4.4. Efficiency and scalability

In addition to its strong property control, RNAGenScape is also highly efficient at inference time. As reported in Table 1, it achieves an inference speed of 0.57 ms/sample, nearly matching the fastest method (gradient ascent at 0.50 ms/sample) and substantially faster than other methods such as hill climbing (81.52 ms/sample) and MCMC (10.93 ms/sample). This efficiency makes RNAGenScape well suited for large-scale or iterative design workflows where fast feedback is essential.

4.5. Interpolating between arbitrary sequences

Besides being able to optimize mRNA sequences for target properties, RNAGenScape enables interpolation between arbitrary sequences by leveraging the directional drift term (Eqn (11)).

We qualitatively illustrate the resulting interpolation trajectories in Figure 4. Guided by a directional force toward a specified target, RNAGenScape generates smooth and coherent trajectories on the learned manifold while preserving biological plausibility and continuity. These trajectories connect arbitrary input–target sequence pairs in a structured manner, reflecting semantically meaningful transitions that can be decoded back for further biological interpretation and investigation. Again, a deeper analysis of these trajectories is left for future work.

5. Conclusion

We introduced RNAGenScape, a Langevin-dynamics framework that refines real mRNA sequences in a learned latent space rather than generating from scratch. By combining an organized autoencoder with a denoising-based manifold projector, RNAGenScape steers existing sequences along smooth, manifold-aligned trajectories that both improve target properties and preserve biological plausibility. Empirically, RNAGenScape outperforms a suite of *de novo* generative models and optimization methods in property control, while matching or exceeding their inference efficiency. With this work, we also hope to shift the paradigm of biological sequence design from unconstrained, *de novo* generation to guided refinement of real data points.

6. Limitations and Future Work

One limitation of our approach is its dependence on the fidelity of the organized latent space: if the organized autoencoder fails to capture critical sequence

constraints, manifold projections may permit small but functionally invalid drifts. Additionally, our current formulation optimizes a single scalar property; extending RNAGenScape to multi-objective settings would broaden its applicability. Finally, while we have demonstrated compelling in silico gains, integrating real-world experimental feedback remains an important avenue to validate and refine the learned manifold.

In future work, we will analyze the intermediate outputs from both property optimization and targeted interpolation to uncover new biological insights. We will also study the possibility to perform sequence-structure joint modeling and optimization. Beyond mRNA, we plan to extend RNAGenScape to other modalities such as protein sequences and regulatory elements, and integrate active learning frameworks that guide wet lab experimentation. By grounding sequence optimization in the manifold of real data, we aim to provide a versatile platform for interpretable and high-throughput design in synthetic biology.

7. Related Works

Machine learning is becoming increasingly popular for optimizing biological sequences such as DNA, RNA, and proteins. This section reviews recent advances in sequence modeling and optimization, with an emphasis on mRNAs.

Sequence-to-function modeling A central goal in biological sequence modeling is predicting quantitative properties (e.g., expression level, stability) directly from the sequence (Oliver, 1996). Recent deep learning models trained on high-throughput experimental data have demonstrated strong performance in this setting, particularly for regulatory regions such as 5'UTRs and promoters (Sample et al., 2019; Vaishnav et al., 2022). Models such as ConvNets (Chen et al., 2024) and Transformers (He et al., 2023a) have been used to capture complex dependencies in mRNA space, and form the basis for downstream prediction of properties.

Generative models for design Generative models enable sampling of novel sequences enriched for desired traits. Variational autoencoders (VAEs) (Kingma et al., 2013) have been applied to proteins to learn smooth latent spaces that are amenable to gradient-based optimization (Sinai et al., 2017; Castillo-Hair et al., 2024). ProteinMPNN (Dauparas et al., 2022), although described as a message-passing neural network by the authors, shares core design principles with autoencoders. Generative adversarial networks (Goodfellow et al., 2020) such as Méndez-Lucio et al. (Méndez-Lucio et al., 2020) or ProteinGAN (Repecka et al., 2021) and autoregressive language models such as ProGen (Madani et al., 2023) have also been used to generate diverse protein sequences. More recently, diffusion models (Ho et al., 2020) have shown promise in discrete domains. For example, RFdiffusion (Watson et al., 2023) generates proteins unconditionally or conditioned on structural constraints. These methods can be readily adapted to mRNA design.

Optimization of biological sequences Sequence optimization can be framed as a black-box search or a differentiable surrogate-guided process. Several approaches relax discrete inputs for gradient-based updates, such as using straight-through estimators (Linder et al., 2019). ReLSO learns a continuous latent space and performs gradient ascent (Castro et al., 2022). Others apply reinforcement learning (Eastman et al., 2018) or Monte Carlo algorithm (Wirecki et al., 2023) for sequence optimization. Methods such as Fast SeqProp (Linder & Seelig, 2021) and LaMBO (Stanton et al., 2022) have demonstrated success in optimizing sequences under multi-objective constraints.

Integration of structural context While the present work strictly focuses on the mRNA sequence, many successful models incorporate inductive biases from the structures. ProteinMPNN (Dauparas et al., 2022) and diffusion-based inverse folding (Yi et al., 2023) condition sequence generation on 3D structures. ImmunoStruct (Givechian et al., 2025) jointly models protein sequence, structure, and biochemical properties to predict immunogenicity. CellSpliceNet (Afrasiyabi et al., 2025) integrates long-range sequence, local regions of interest, secondary structure, and gene expression to predict alternative slicing. EternaFold (Wayment-Steele et al., 2022) incorporate predicted secondary structures to improve fitness prediction. Although in our work we did not incorporate mRNA structures, extending RNAGenScape to sequence-structure joint modeling and optimization could be a promising direction.

8. Acknowledgements

S.K. is funded by the NIH (NIGMSR01GM135929, R01GM130847), NSF CAREER award IIS-2047856, NSF IIS-2403317, NSF DMS-2327211 and NSF CISE-2403317. S.K is also funded by the Sloan Fellowship FG-2021-15883, the Novo Nordisk grant GR112933. A.J.G. is funded by the NIH (R01HD100035, R35GM122580).

References

Afrasiyabi, A., Kovalic, J., Liu, C., Castro, E., Weinreb, A., Varol, E., Miller, D., Hammarlund, M., and Krishnaswamy, S. Cellsplicenet: Interpretable multimodal modeling of alternative splicing across neurons in c. elegans. *bioRxiv*, pp. 2025–06, 2025.

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. An introduction to mcmc for machine learning. *Machine learning*, 50:5–43, 2003.

Brooks, S. Markov chain monte carlo method and its application. *Journal of the royal statistical society: series D (the Statistician)*, 47(1):69–100, 1998.

Burkhardt, D. B., Stanley III, J. S., Tong, A., Perdigoto, A. L., Gigante, S. A., Herold, K. C., Wolf, G., Giraldez, A. J., van Dijk, D., and Krishnaswamy, S. Quantifying the effect of experimental perturbations at single-cell resolution. *Nature biotechnology*, 39(5):619–629, 2021.

Castillo-Hair, S., Fedak, S., Wang, B., Linder, J., Havens, K., Certo, M., and Seelig, G. Optimizing 5'utrs for mrna-delivered gene editing using deep learning. *Nature Communications*, 15(1):5284, 2024.

Castro, E., Godavarthi, A., Rubinfien, J., Givechian, K., Bhaskar, D., and Krishnaswamy, S. Transformer-based protein generation with regularized latent space optimization. *Nature Machine Intelligence*, 4(10):840–851, 2022.

Cayton, L. et al. *Algorithms for manifold learning*. eScholarship, University of California, 2008.

Chen, Y., Du, Z., Ren, X., Pan, C., Zhu, Y., Li, Z., Meng, T., and Yao, X. mrna-cla: An interpretable deep learning approach for predicting mrna subcellular localization. *Methods*, 227:17–26, 2024.

Coifman, R. R. and Lafon, S. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

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

Eastman, P., Shi, J., Ramsundar, B., and Pande, V. S. Solving the rna design problem with reinforcement learning. *PLoS computational biology*, 14(6):e1006176, 2018.

Fefferman, C., Mitter, S., and Narayanan, H. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Givechian, K. B., Rocha, J. F., Yang, E., Liu, C., Greene, K., Ying, R., Caron, E., Iwasaki, A., and Krishnaswamy, S. Immunostruct: a multimodal neural network framework for immunogenicity prediction from peptide-mhc sequence, structure, and biochemical properties. *bioRxiv*, pp. 2024–11, 2025.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

He, S., Gao, B., Sabnis, R., and Sun, Q. Rnadegformer: accurate prediction of mrna degradation at nucleotide resolution with deep learning. *Briefings in Bioinformatics*, 24(1):bbac581, 2023a.

He, Y., Murata, N., Lai, C.-H., Takida, Y., Uesaka, T., Kim, D., Liao, W.-H., Mitsufuji, Y., Kolter, J. Z., Salakhutdinov, R., et al. Manifold preserving guided diffusion. *arXiv preprint arXiv:2311.16424*, 2023b.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

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

Kidger, P., Foster, J., Li, X., and Lyons, T. J. Neural sdes as infinite-dimensional gans. In *International conference on machine learning*, pp. 5453–5463. PMLR, 2021.

Kingma, D. P., Welling, M., et al. Auto-encoding variational bayes, 2013.

Li, Q., Hu, Y., Liu, Y., Zhang, D., Jin, X., and Chen, Y. Discrete point-wise attack is not enough: Generalized manifold adversarial attack for face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20575–20584, 2023.

Liao, D., Liu, C., Christensen, B. W., Tong, A., Huguet, G., Wolf, G., Nickel, M., Adelstein, I., and Krishnaswamy, S. Assessing neural network representations during training using noise-resilient diffusion spectral entropy. In *2024 58th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6. IEEE, 2024.

Linder, J. and Seelig, G. Fast activation maximization for molecular sequence design. *BMC bioinformatics*, 22: 1–20, 2021.

Linder, J., Bogard, N., Rosenberg, A. B., and Seelig, G. Deep exploration networks for rapid engineering of functional dna sequences. *BioRxiv*, pp. 864363, 2019.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Liu, C., Amodio, M., Shen, L. L., Gao, F., Avesta, A., Aneja, S., Wang, J. C., Del Priore, L. V., and Krishnaswamy, S. CUTS: A Deep Learning and Topological Framework for Multigranular Unsupervised Medical Image Segmentation . In *proceedings of Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume LNCS 15008. Springer Nature Switzerland, October 2024.

Liu, C., Liao, D., Parada-Mayorga, A., Ribeiro, A., DiStasio, M., and Krishnaswamy, S. Diffkillr: Killing and recreating diffeomorphisms for cell annotation in dense microscopy images. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025a.

Liu, C., Xu, K., Shen, L. L., Huguet, G., Wang, Z., Tong, A., Bzdok, D., Stewart, J., Wang, J. C., Del Priore, L. V., and Krishnaswamy, S. Imageflownet: Forecasting multiscale trajectories of disease progression with irregularly-sampled longitudinal medical images. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025b.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8):1099–1106, 2023.

Méndez-Lucio, O., Baillif, B., Clevert, D.-A., Rouquié, D., and Wichard, J. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nature communications*, 11(1):10, 2020.

Meyers, J., Fabian, B., and Brown, N. De novo molecular design and generative models. *Drug discovery today*, 26 (11):2707–2715, 2021.

Moon, K. R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D. B., Chen, W. S., Yim, K., Elzen, A. v. d., Hirn, M. J., Coifman, R. R., et al. Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, 37(12):1482–1492, 2019.

Munson, B. P., Chen, M., Bogosian, A., Kreisberg, J. F., Licon, K., Abagyan, R., Kuenzi, B. M., and Ideker, T. De novo generation of multi-target compounds using deep generative chemistry. *Nature Communications*, 15(1): 3636, 2024.

Narayanan, H. and Mitter, S. Sample complexity of testing the manifold hypothesis. *Advances in neural information processing systems*, 23, 2010.

Oliver, S. G. From dna sequence to biological function. *Nature*, 379(6566):597–600, 1996.

Prykhodko, O., Johansson, S. V., Kotsias, P.-C., Arús-Pous, J., Bjerrum, E. J., Engkvist, O., and Chen, H. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of cheminformatics*, 11(1):74, 2019.

Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Rokaitis, I., Zrimec, J., Poviloniene, S., Laurynenas, A., Viknander, S., Abuajwa, W., et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333, 2021.

Rifai, S., Dauphin, Y. N., Vincent, P., Bengio, Y., and Muller, X. The manifold tangent classifier. *Advances in neural information processing systems*, 24, 2011.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Sample, P. J., Wang, B., Reid, D. W., Presnyak, V., McFadyen, I. J., Morris, D. R., and Seelig, G. Human 5' utr design and variant effect prediction from a massively parallel translation assay. *Nature biotechnology*, 37(7): 803–809, 2019.

Selman, B. and Gomes, C. P. Hill-climbing search. *Encyclopedia of cognitive science*, 81(333-335):10, 2006.

Sinai, S., Kelsic, E., Church, G. M., and Nowak, M. A. Variational auto-encoding of protein sequences. *arXiv preprint arXiv:1712.03346*, 2017.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Stanton, S., Maddox, W., Gruver, N., Maffettone, P., Delaney, E., Greenside, P., and Wilson, A. G. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *International conference on machine learning*, pp. 20459–20478. PMLR, 2022.

Strayer, E. C., Krishna, S., Lee, H., Vejnar, C., Beaudoin, J.-D., and Giraldez, A. J. Nap-trap, a novel massively parallel reporter assay to quantify translation control. *bioRxiv*, pp. 2023–11, 2023.

Sun, X., Liao, D., MacDonald, K., Zhang, Y., Liu, C., Huguet, G., Wolf, G., Adelstein, I., Rudner, T. G., and Krishnaswamy, S. Geometry-aware generative autoencoders for warped riemannian metric learning and generative modeling on data manifolds. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2025.

Vaishnav, E. D., de Boer, C. G., Molinet, J., Yassour, M., Fan, L., Adiconis, X., Thompson, D. A., Levin, J. Z., Cubillos, F. A., and Regev, A. The evolution, evolvability and engineering of gene regulatory dna. *Nature*, 603 (7901):455–463, 2022.

Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., Burdziak, C., Moon, K. R., Chaffer, C. L., Pattabiraman, D., et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3): 716–729, 2018.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620 (7976):1089–1100, 2023.

Wayment-Steele, H. K., Kladwang, W., Strom, A. I., Lee, J., Treuille, A., Becka, A., Participants, E., and Das, R. Rna secondary structure packages evaluated and improved by high-throughput experiments. *Nature methods*, 19(10): 1234–1242, 2022.

Wirecki, T., Lach, G., Jaryani, F., Badepally, N. G., Moafinejad, S. N., Klaudel, G., and Bujnicki, J. M. Desirna: structure-based design of rna sequences with a monte carlo approach. *bioRxiv*, pp. 2023–06, 2023.

Yi, K., Zhou, B., Shen, Y., Liò, P., and Wang, Y. Graph denoising diffusion for inverse protein folding. *Advances in Neural Information Processing Systems*, 36:10238–10257, 2023.

Zhang, W., Zhang, Y., Hu, X., Goswami, M., Chen, C., and Metaxas, D. N. A manifold view of adversarial risk. In *International Conference on Artificial Intelligence and Statistics*, pp. 11598–11614. PMLR, 2022.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.