Gradient-guided discrete walk-jump sampling for biological sequence generation

Zarif Ikram[1,2] Dianbo Liu[2*] M Saifur Rahman[1*]

[1]Department of CSE, BUET, Bangladesh

[2]National University of Singapore, Singapore

[*]Equal supervision.

Outline

Motivation

Preliminaries

Gradient-guided discrete Walk-Jump Sampling (gg-dWJS)

Results

Accepted at Transactions on Machine Learning Research

Motivation

Why generate antibodies?

- Vital for rapid discovery and delivery of antibody-based drugs.
- Important considerations include efficacy, safety, and manufacturability.
- Challenges include enormous state space of protein sequence and high-entropy variable regions.
- Experimental validation is time-consuming and expensive.

Why structure based generation?

- Simpler in continuous settings, enabling direct optimization in structural space

Why structure based generation?

- Simpler in continuous settings, enabling direct optimization in structural space
- Necessitates inverse-folding of structures.

Why structure based generation?

- Simpler in continuous settings, enabling direct optimization in structural space
- Necessitates inverse-folding of structures.
- Optimized structure does not guarantee a realizable sequence.

Why structure based generation?

- Simpler in continuous settings, enabling direct optimization in structural space
- Necessitates inverse-folding of structures.
- Optimized structure does not guarantee a realizable sequence.
- Even if the sequence exists, inverse-folding could be difficult

Sequence based generation

- Two general directions: diffusion and autoregressive models

Sequence based generation

- Two general directions: diffusion and autoregressive models
- Diffision models
  - Generate from noise
  - Challenge: intricate noise scheduling

Sequence based generation

- Two general directions: diffusion and autoregressive models
- Diffision models
  - Generate from noise
  - Challenge: intricate noise scheduling
- Autoregressive models
  - Next-token prediction
  - Challenge: error accumulation

Preliminaries

Score based models

- Learns the "direction" that removes noise

$$\nabla \log p(y)$$

Score based models

- Learns the "direction" that removes noise
- Add noise to the original data and learn the perturbation

$$\nabla \log p(y)$$

Score based models

- Learns the "direction" that removes noise
- Add noise to the original data and learn the perturbation
- Works well for continuous data

$$\nabla \log p(y)$$

Neural Empirical Bayes (NEB)

- Formalism to recover X for a smoothed random variable $Y = X + \mathcal{N}(0, \sigma^2 I_d)$

$$\hat{x} = y + \sigma^2 \nabla \log p(y)$$

Neural Empirical Bayes (NEB)

- Formalism to recover X for a smoothed random variable $Y = X + \mathcal{N}(0, \sigma^2 I_d)$
- $\nabla \log p(y)$ is the score model!

$$\hat{x} = y + \sigma^2 \nabla \log p(y)$$

discrete Walk-Jump Sampling (dWJS)

- Smooth the discrete data to continuous data using noise

discrete Walk-Jump Sampling (dWJS)

- Smooth the discrete data to continuous data using noise
- Perform the MCMC walk on continuous noisy data manifold using the denoising score model

discrete Walk-Jump Sampling (dWJS)

- Smooth the discrete data to continuous data using noise
- Perform the MCMC walk on continuous noisy data manifold using the denoising score model
- Jump back to the discrete data using NEB

Gradient-guided discrete Walk-Jump Sampling (gg-dWJS)

We need targeted generation

- Generated sequences need to be antibody-like

We need targeted generation

- Generated sequences need to be antibody-like
- ... but also optimized for a specific attribute!

Gradient guidance

- We can learn attributes of a noised sequence using discriminator model $f_\theta(Y)$
- Attributes can be antigen affinity, aromaticity, instability, etc.

Gradient guidance

- We can learn attributes of a noised sequence using discriminator model $f_\theta(Y)$
- Attributes can be antigen affinity, aromaticity, instability, etc.
- Use $\nabla f_\theta(Y)$ to maximize the attribute

Gradient guidance

- We can learn attributes of a noised sequence using discriminator model $f_\theta(Y)$
- Attributes can be antigen affinity, aromaticity, instability, etc.
- Use $\nabla f_\theta(Y)$ to maximize the attribute Gradient guidance!

Steps

1. Learn a noisy discriminator model
2. Learn a noisy score model
3. Gradient guided walk
4. Gradient guided jump

Step 1: Learn a noisy discriminator model

- Add noise to the discrete sequences while keeping their ground truth
- Learn a predictor!

$$f_\theta(Y)$$

Step 2: Learn a noisy score model

- Add noise to the discrete sequences
- Train the score based model

$$g_\phi(y)$$

Step 3: Gradient guided walk

- Use MCMC to denoise
- We walk in the smoothed manifold, so score-based model will work!

Step 3: Gradient guided walk

- Use MCMC to denoise
- We walk in the smoothed manifold, so score-based model will work!
- Combine discriminator gradient for guided walk towards optimized attribute region

Step 4: Gradient guided jump

- Use NEB to jump to discrete region

Step 4: Gradient guided jump

- Use NEB to jump to discrete region
- Conditional NEB: use discriminator gradient for guided jump

In summary...

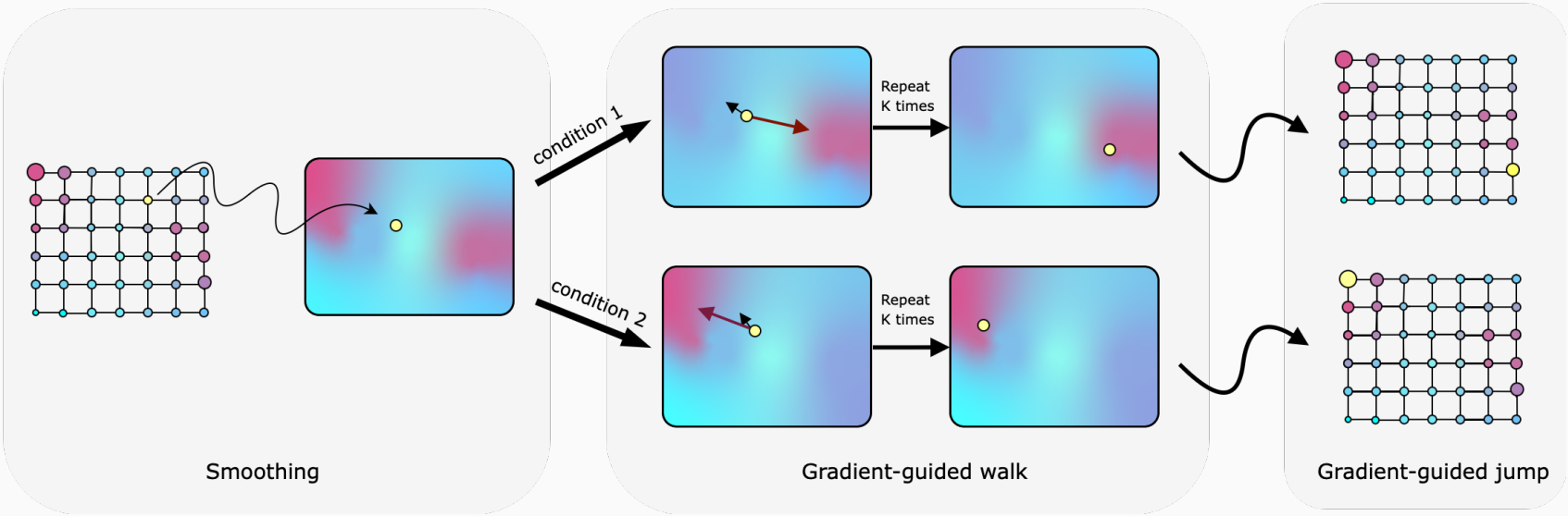

Figure 1: Gradient-guided discrete walk-jump sampling process. We begin the sampling process by smoothing some discrete seed. Next, we conduct the gradient-guided walk process by combining denoising gradient with the discriminator gradient. Finally, we perform gradient-guided jump to return to the discrete data manifold. Here, purple tint represents higher data density and larger circle represents higher data distribution.

Multi-objective optimization (MOO)

- Finding $x^* \in \mathcal{X}$ such that d different rewards are maximized simultaneously

$$R(x^*) = \max_{x \in \mathcal{X}} R(x) =$$
$$[\max_{x \in \mathcal{X}} R_1(x), \max_{x \in \mathcal{X}} R_2(x), \ldots, \max_{x \in \mathcal{X}} R_d(x)]$$

Multi-objective optimization (MOO)

- We can do weighted sum of different objectives to turn them into a single objective
- We can learn the smoothed predictor $F_\theta(x, w)$

$$F_\theta(x, w) = \sum w_i F_{\theta,i}(x)$$

Multi-objective optimization (MOO)

- We can do weighted sum of different objectives to turn them into a single objective
- We can learn the smoothed predictor $F_\theta(x, w)$
- ... and use its gradient to guide the Langevin walk in the noisy manifold

$$F_\theta(x, w) = \sum w_i F_{\theta,i}(x)$$

Multi-objective optimization with gg-dWJS

- Perform the single-objective decomposition of the MOO problem using preference conditioning

$$
\begin{aligned}
g &= g_\phi(y) + \nabla_y f_{1,\theta}(y, w_1) + \nabla_y f_{2,\theta}(y, w_2) + \cdots + \nabla_y f_{d,\theta}(y, w_d) \\
&= g_\phi(y) + \nabla_y w_1 R_1(y) + \nabla_y w_2 R_2(y) + \cdots + \nabla_y w_d R_d(y) \\
&= g_\phi(y) + \nabla_y (w_1 R_1(y) + w_2 R_2(y) + \cdots + w_d R_d(y)) \\
&= g_\phi(y) + \nabla_y f_\theta(y, w)
\end{aligned}
$$

Multi-objective optimization with gg-dWJS

- Perform the single-objective decomposition of the MOO problem using preference conditioning
- For multiple preference weights w, sample and create Pareto front

$$
\begin{aligned}
g &= g_\phi(y) + \nabla_y f_{1,\theta}(y, w_1) + \nabla_y f_{2,\theta}(y, w_2) + \cdots + \nabla_y f_{d,\theta}(y, w_d) \\
&= g_\phi(y) + \nabla_y w_1 R_1(y) + \nabla_y w_2 R_2(y) + \cdots + \nabla_y w_d R_d(y) \\
&= g_\phi(y) + \nabla_y (w_1 R_1(y) + w_2 R_2(y) + \cdots + w_d R_d(y)) \\
&= g_\phi(y) + \nabla_y f_\theta(y, w)
\end{aligned}
$$

Results

Discretized MNIST generation

- Can we generate binarized (discrete) image conditionally?

Discretized MNIST generation

- Can we generate binarized (discrete) image conditionally?
- Yes!

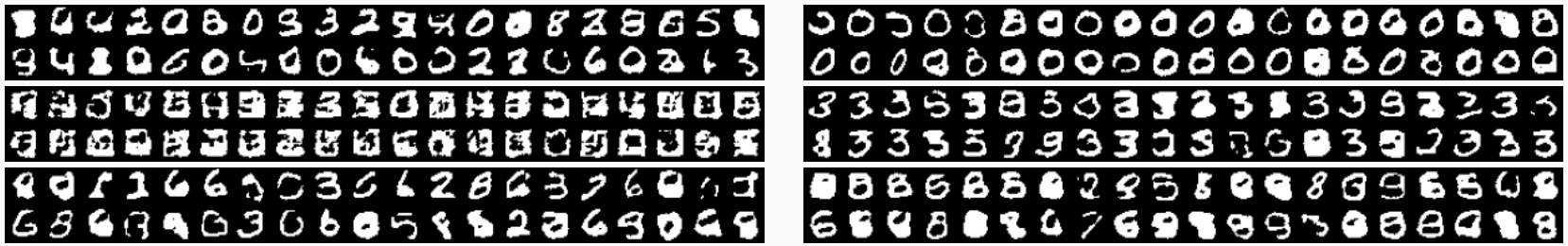

Figure 2: Comparison of binarized MNIST samples generated by different dWJS methods. *Left:* from top to bottom: dwjs, gg-dWJS w/o denoising gradient, gg-dWJS. *Right:* from top to bottom: gg-dWJS generated samples with label 0, 3, and 8.

Antibody sequence optimization

Table 1: Experiment results for antibody sequence generation for single-task optimization task. The results show that gg-dWJS-generated sequences are better optimized and of higher quality.

Method	dcs ↑	Instability index ↓	% Beta sheets ↑
dWJS	0.49 ± 0.30	34.14 ± 6.38	0.393 ± 0.02
gg-dWJS w/ Beta sheet discriminator	0.51 ± 0.28	35.92 ± 6.25	**0.408± 0.02**
gg-dWJS w/ Instability discriminator	**0.56 ± 0.27**	**31.32 ± 5.21**	0.402 ± 0.023
VDM	0.34 ± 0.31	39.50 ± 6.79	0.37 ± 0.02
IgLM	0.19 ± 0.29	38.84 ± 6.18	0.37 ± 0.02
GPT-4o	0.31 ± 0.30	42.63 ± 2.47	0.37 ± 0.01
GFlowNets	0.0 ± 0.0	39.04 ± 1.04	N/A

Antimicrobial peptide (AMP) generation

Table 2: Results on the AMP Task.

	Performance	**Diversity**	**Novelty**
gg-dWJS	$\mathbf{0.98 \pm 0.015}$	$\mathbf{25.78 \pm 1.22}$	15.021 ± 1.02
GFlowNet-AL	0.932 ± 0.002	22.34 ± 1.24	$\mathbf{28.44 \pm 1.32}$
DynaPPO	0.938 ± 0.009	12.12 ± 1.71	9.31 ± 0.69
COMs	0.761 ± 0.009	19.38 ± 0.14	26.47 ± 1.3
GFlowNet	0.868 ± 0.015	11.32 ± 0.67	15.72 ± 0.44

Antimicrobial peptide (AMP) generation

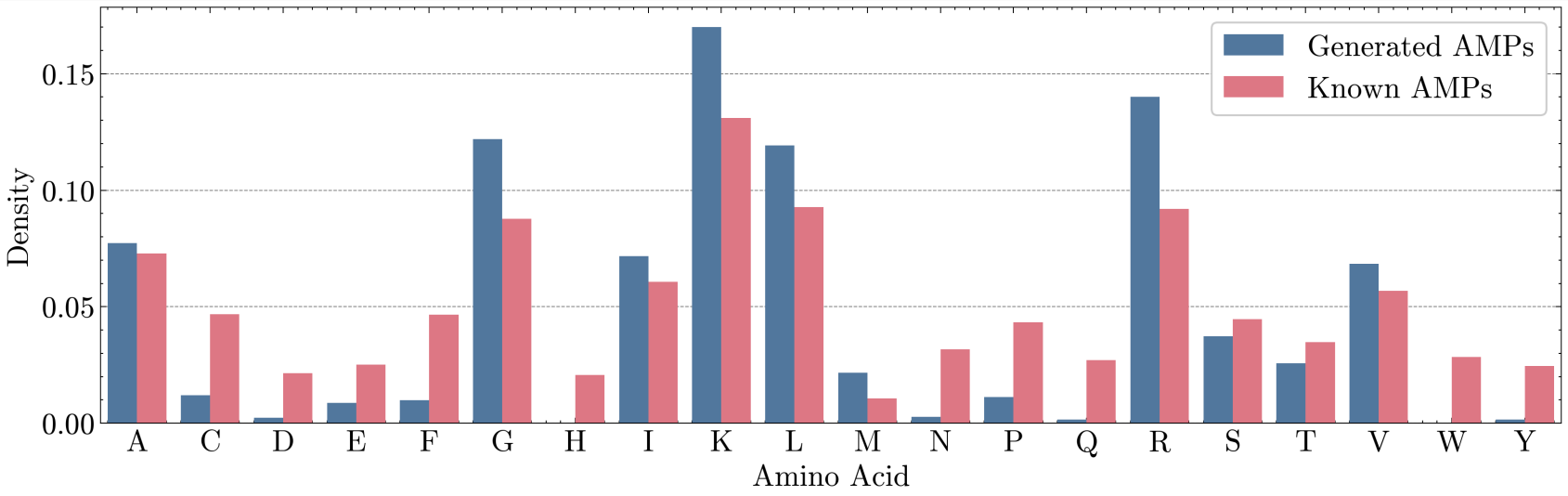

Figure 3: Distribution of amino acids found in the generated AMPs by gg-dWJS matches that of known AMPs while maintaining focusing on amino acid "K", which is dominant in peptides with anti-microbial activity.

Antibody multiobjective optimization

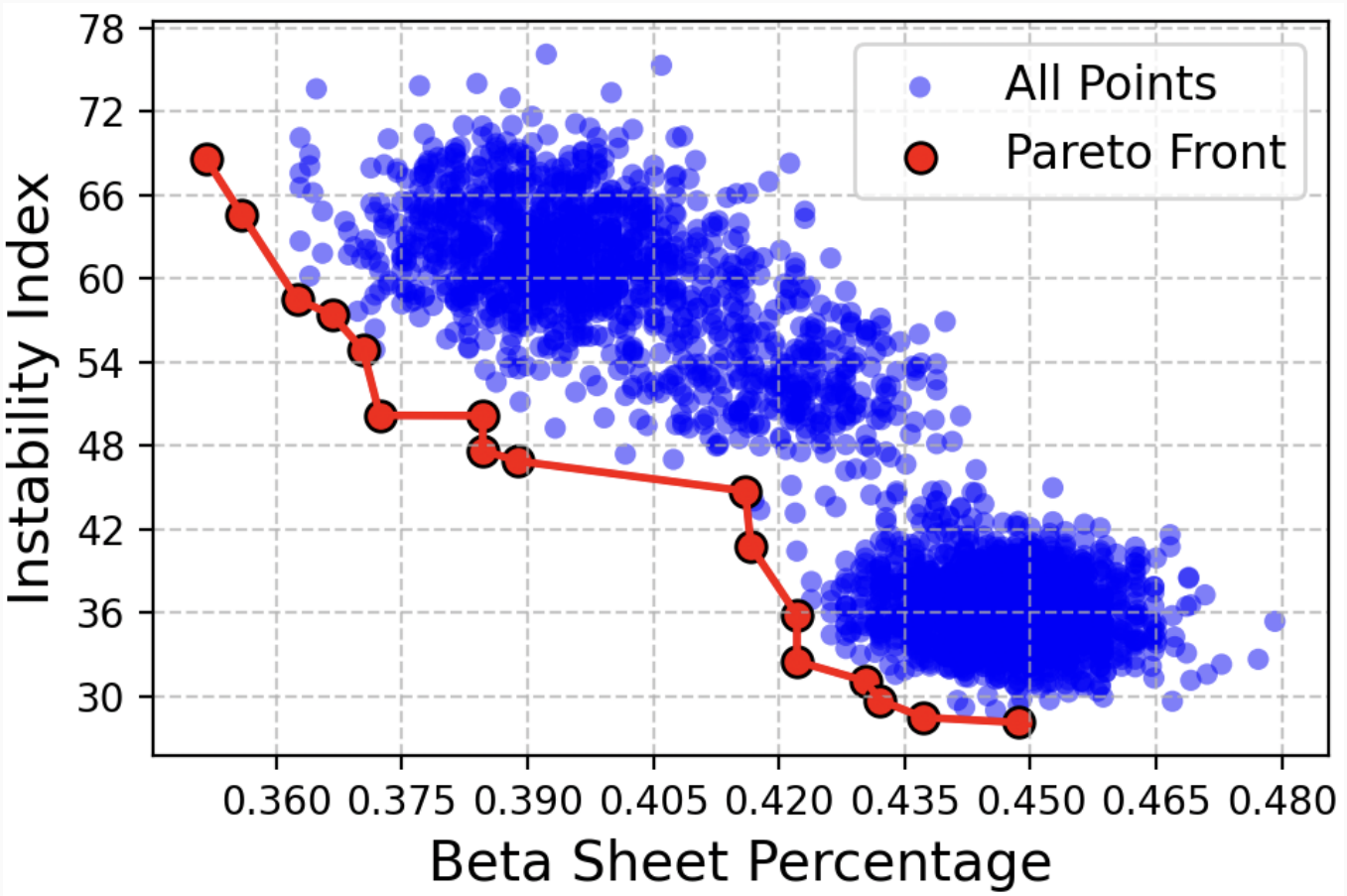

Figure 4: Pareto front of the samples generated using three preference weights with gg-dWJS.

Project URL

https://zarifikram.github.io/gg-dWJS/

Thank You!