ANTIBODY SEQUENCE OPTIMIZATION WITH GRADIENT-GUIDED DISCRETE WALK-JUMP SAMPLING

Zarif Ikram
Department of CSE, BUET
Dhaka, Bangladesh
zarifikram003@gmail.com

Dianbo Liu*
National University of Singapore
Singapore
dianbo@nus.edu.sg

M Saifur Rahman*
Department of CSE, BUET
Dhaka, Bangladesh
mrahman@cse.buet.ac.bd

ABSTRACT

For enhancing the rapid discovery and delivery of antibody-based drugs, antibody attribute optimization is essential for the real-world application of therapeutic antibody sequence design. Using a generative machine learning model for antibody design is a promising direction for such tasks. However, existing methods struggle in balancing error accumulation, scalability, and targeted attribute optimization. In this work, we propose gradient-guided discrete walk-jump sampling (gg-dWJS), a novel discrete sequence generation method for antibody attribute optimization. Leveraging gradient guidance in the noisy manifold, we sample from the smoothed data manifold by applying discretized Markov chain Monte Carlo (MCMC) using a denoising model with the gradient-guidance from a discriminative model. This is followed by jumping to the discrete data manifold using a conditional one-step denoising. Through evaluation on both discrete image and antibody sequence generation tasks, we show that our method generates high-quality samples that are well-optimized for specific tasks.

1 INTRODUCTION

Therapeutic antibody optimization is vital for the rapid discovery and delivery of antibody-based drugs in the context of their efficacy, safety, and manufacturability. These optimized drugs find diverse practical applications, from molecular imaging of cancer (Wu, 2014) to immunotherapy of cancer and viral diseases (Hudson & Souriau, 2003; Lu et al., 2020). Antibody optimization is challenging: the protein sequence has an enormous state space with high-entropy variable regions. Besides, experimental validation is both time-consuming and expensive. Generally, the antibody engineering process begins with *in vitro* experiments to determine aspects of interest such as humanization, specificity (Makowski et al., 2022), efficacy, and pharmacokinetic properties, followed by mutation to the antibody samples for improved properties.

Ab initio antibody sequence generation is a fascinating direction for generating novel antibody sequences given prior samples. Using a generative model, these methods attempt to generate antibody sequences that are *antibody-like* and similar to prior data. We can roughly divide these models into two groups: autoregressive models (Wang et al., 2022; Jain et al., 2022) and denoising models Luo et al. (2022); Gruver et al. (2023). Albeit compelling, both come with their drawbacks. On one hand, autoregressive models suffer from error accumulation and even scalability in the case of reward distribution fitting. (e.g., Jain et al. (2022)). On the other hand, denoising models require intricate noise scheduling, making the real discovery task difficult.

To combat the inefficiency of the autoregressive and denoising models, discrete walk-jump sampling (dWJS) (Frey et al., 2024) recently proposed sampling antibody sequences by walking on noisy manifolds, followed by denoising jumping to the discrete data manifold. While dWJS benefits from sampling from noisy manifolds for discrete sequence generation, its only objective is to sample from

*These authors co-supervised the work.

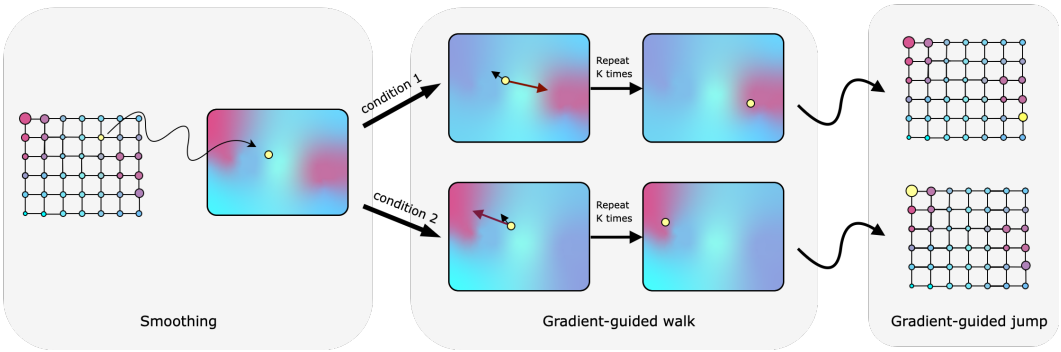

Figure 1: Gradient-guided discrete walk-jump sampling process. We begin the sampling process by smoothing some discrete seed. Next, we conduct the gradient-guided walk process by combining denoising gradient with the discriminator gradient. Finally, we perform gradient-guided jump to return to the discrete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

the data distribution, leading to sequences that are not necessarily optimized for a chosen attribute. However, in a real-world antibody optimization setting, we are interested in sequences that are not only antibody-like but also attribute-optimized.

A simple but effective way of optimizing properties in a generative model is to train a discriminative model and use its gradient to guide the sampling process towards a high-fitness region. While this approach is simpler in continuous settings, enabling direct optimization in structural space, it still requires consideration of the amino acid sequence for actual synthesis, necessitating inverse-folding (Dauparas et al., 2022) of structures. However, the optimized structure does not guarantee a realizable sequence, and even if the sequence exists, there is no assurance of inverse-folding it, motivating the need for direct optimization in sequence space. In a discrete setting such as antibody sequence generation, gradient guidance is difficult. The sampling step in dWJS, thankfully, takes place in the smoothed data manifold, which lets these discriminative models provide gradient guidance.

In this work, we present gradient-guided discrete walk-jump sampling (gg-dWJS), a novel approach for *ab initio* sequence generation for antibody attribute optimization, building on (Frey et al., 2024). To improve the likelihood of a sequence attribute, gg-dWJS learns a denoising and discriminative model on the noisy data manifold and then **walks** on the noisy data manifold using discretized Langevin MCMC with **gradient guidance**. Finally, using the same model, our method performs one-step conditional denoising to **jump** back to the true data manifold. We evaluate our method on image and antibody sequence generation tasks, showing that our method generates high-quality samples with optimized attributes.

2 Preliminary

2.1 Neural Empirical Bayes (NEB)

By combining kernel density estimation and empirical Bayes, NEB (Saremi & Hyvärinen, 2019) provides a denoising method to recover X for a smoothed random variable $Y = X + \mathcal{N}(0, \sigma^2 I_d)$ such that $X \rightharpoonup Y$ using the gradient log density of Y. Formally, to retrieve $Y \rightharpoonup X$, given Y=y, we can get the least square estimator of $X = \hat{x}$ according to Robbins (1992); Miyasawa et al. (1961)–

$$\hat{x} = y + \sigma^2 \nabla \log p(y) \tag{1}$$

Where $\nabla \log p(y)$ is known as the *score function* (Hyvärinen & Dayan, 2005).

2.2 Discrete walk-jump sampling (dWJS)

Building on the NEB, dWJS learns the score function on the noisy data manifold parameterized as $g_\phi : \mathbb{R}^d \to \mathbb{R}^d$. Hence, equation 1 now looks like the following:

$$\hat{x} = y + \sigma^2 g_\phi(y) \tag{2}$$

We can learn the parameter ϕ by performing stochastic gradient descent on the following loss function.

$$\mathcal{L}(\phi) = \mathbb{E}_{x\sim p(x), y\sim p(y|x)}||x - \hat{x_\phi}(y)||^2 \tag{3}$$

To sample discrete under this formulation, we perform Langevin MCMC using g_ϕ on the noisy data manifold, returning to the true data manifold using equation 2.

3 GRADIENT-GUIDED DISCRETE WALK-JUMP SAMPLING

gg-dWJS augments dWJS by learning a discriminator model $P(C|Y)$ **(Step A)** on the noisy data Y given the true data X and its label C. Next, we train the score model g_ϕ on the noisy data Y **(Step B)**. Later, we utilize the discriminator's gradient to guide the Langevin MCMC walk **(Step C)**. Finally, we jump from the noisy data manifold to the true discrete data manifold using the gradient guidance **(Step D)**.

3.1 STEP A: LEARNING A NOISY DISCRIMINATOR MODEL

Given the true data X and its label C, we smooth X by adding gaussian noise, obtaining $Y = X + \mathcal{N}(0, \sigma^2 I_d)$. Next, we learn the discriminator model $\log P(C|Y)$ by parameterizing it with $f_\theta(Y)$. Note that C can be both categorical and continuous. In our work, we parameterize $f_\phi(y)$ to return logits over the labels given y. So, $f_\theta(y, c)$ refers to logit for a given label c.

3.2 STEP B: LEARNING A NOISY SCORE MODEL

Similar to step B, we learn the score model on the noisy data Y, parameterizing it with g_ϕ. We learn the model by minimizing the loss $\mathcal{L}$ from equation 3.

3.3 STEP C: GRADIENT GUIDED WALK

We start by sampling random discrete data $y_0 \sim \mathcal{U}_d$. Next, we perform discretized Langevin MCMC (Sachs et al., 2017) on y with gradient guidance from $f_\theta(y)$ for K steps. Formally, we use

$$g_\phi(y_k) + \nabla_Y f_\theta(y_k, c)$$

to perform MCMC given y_k and label c at the k_{th} step. Intuitively, one can see this as performing a gradient accent on the label c's probability with respect to the input y_k to find the local maxima while walking towards the smoothed data density with $g_\phi(y_k)$. We summarize the process in algorithm 1.

3.4 STEP D: GRADIENT GUIDED JUMP

After the K steps of walking, we jump from the noisy data manifold to the clear discrete data using the gradient guidance and score model given smoothed data y_k and label c. From NEB, given a smoothed input y, we can jump back to $\hat{x}$ using the equation 1. Without loss of generality, we can extend that, for a conditional variable c and $y \sim p(y|c)$,

$$\begin{aligned}
\hat{x} &= y + \sigma^2 \nabla_y \log p(y|c) \\
&= y + \sigma^2 \nabla_y \log p(c|y) + \sigma^2 \nabla_y \log p(y) - \sigma^2 \nabla_y \log p(c) \\
&= y + \sigma^2 \nabla_y \log p(c|y) + \sigma^2 \nabla_y \log p(y) \\
&\approx y + \sigma^2 \nabla_y f_\theta(y, c) + \sigma^2 g_\phi(y)
\end{aligned}$$

Therefore, according to the above demonstration, we perform a one-step denoising to return to the true discrete data manifold using f_θ and g_ϕ from steps A & B.

4 EXPERIMENT ON DISCRETE IMAGE GENERATION ($|\mathcal{X}| \approx 10^{236}$)

To validate our method, we compare it against dWJS for the binarized static MNIST image generation task (Salakhutdinov & Murray, 2008; Larochelle & Bengio, 2008). The high-dimensionality of this task ($28 \times 28 \times 2$) makes it an attractive one to validate our approach. For this task, two questions interest us.

Algorithm 1 gradient-guided discrete walk-jump sampling

Input:
c: targetted label
δ: step size
u: inverse mass
γ: friction
K: number of steps taken
$g_\phi \approx \nabla \log p(y)$: learnt score function
$f_\theta \approx \log p(c|y)$: learnt discriminator model
$y_0 \sim \mathcal{N}(0, \sigma^2 I_d) + \mathcal{U}_d(0, 1)$
$v_0 \leftarrow 0$
for $k = 0, \ldots, K-1$; // gradient-guided walk
do
  $y_{k+1} \leftarrow y_k + \frac{\sigma}{2} v_k$
  $s_{k+1} \leftarrow g_\phi(y_{k+1})$; // score of y_{k+1}
  $d_{k+1} \leftarrow \nabla f_\theta(y_{k+1}, c)$; // discriminator gradient guidance
  $g_{k+1} \leftarrow s_{k+1} + d_{k+1}$
  $v_{k+1} \leftarrow v_k + \frac{u\delta}{2} g_{k+1}$
  $\epsilon \sim \mathcal{N}(0, I_d)$
  $v_{k+1} \leftarrow e^{-\gamma\delta} v_{k+1} + \frac{u\delta}{2} g_{k+1} + \sqrt{u(1 - e^{-2\gamma\delta})}\epsilon$
  $y_{k+1} \leftarrow y_{k+1} + \frac{\delta}{2} v_{k+1}$
$\hat{x}_K \leftarrow y_K + \sigma^2 g_\phi(y_K) + \sigma^2 \nabla f_\theta(y_K, c)$; // gradient-guided jump

- *Does gradient guidance enable more realistic generation than denoising walk?*
- *Can gg-dWJS generate conditionally, i.e., can we generate binarized MNIST images with label 0?*

In figure 2, we show the results of our experiment. On the left, we compare the samples generated by dWJS, gg-dWJS without denoising walk, and gg-dWJS. The samples show that gg-dWJS produces the most realistic and diverse samples, which is further affirmed by the lowest Fréchet inception distance (FID) (Heusel et al., 2017) in table 1. Besides, we experiment with performing the walk step using only gradient-guidance, but we find that it cannot generate high-quality samples. On the right, we show the samples of labels 0, 3, and 8 produced by gg-dWJS. They showcase the method's ability to conditionally generate samples. We report the details related to this experiment in appendix B.

Table 1: Experiment results for binarized static MNIST image generation. Here, we calculate FID on the test data. Note that by binarizing MNIST images, we lose important pixel information, contributing to the high FID.

Method	FID ↓
dWJS	54.62
dWJS w/o denoising walk w/ gradient guidance	89.42
gg-dWJS	**51.88**

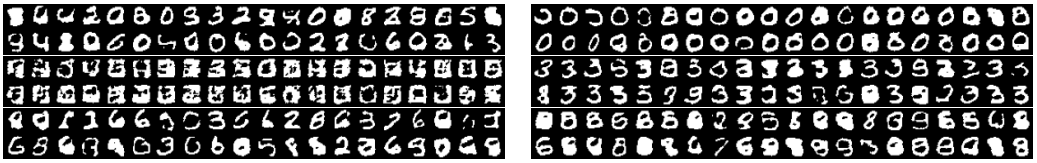

Figure 2: Comparison of binarized MNIST samples generated by different dWJS methods. *Left:* from top to bottom: dWJS, gg-dWJS w/o denoising gradient, gg-dWJS. *Right:* from top to bottom: gg-dWJS generated samples with label 0, 3, and 8.

5 EXPERIMENT ON ANTIBODY SEQUENCE OPTIMIZATION ($|\mathcal{X}| \approx 10^{393}$)

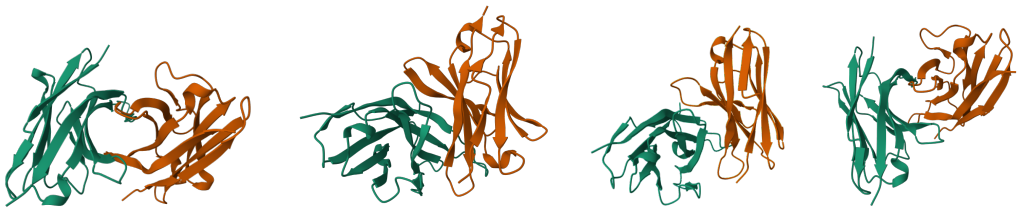

Figure 3: Visualization of generated heavy and light chain sequences using gg-dWJS optimized for instability index. We use IgFold (Ruffolo et al., 2023) for sequence folding and Mol* viewer (Sehnal et al., 2021) for visualization.

Now, we turn our attention to antibody sequence generation. Following Frey et al. (2024); Gruver et al. (2023), we represent antibody sequences as $x = (x_1, \ldots, x_d)$ where $x_i \in 1, \ldots, 21$ refers to the 20 amino acid (AA) type, augmented by the gap token `'-'`. We use the $\sim$1.3M sequences provided by Frey et al. (2024). These sequences are from observed antibody space (OAS) database, aligned according to the AHo number scheme (Honegger & PluÈckthun, 2001) using the ANARCI package (Dunbar & Deane, 2016). The gapped heavy and light chain lengths are respectively 149 and 148, making the total dimension $(149 + 148) \times 21$. For the optimization task, we experiment on two simple single-objective tasks: the percentage of beta sheets and the protein instability index (Guruprasad et al., 1990). For each task, we train a smoothed predictor on the antibody sequences and use its gradient guidance to optimize the single objective.

Table 2: 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	0.170 ± 0.02
gg-dWJS w/ Beta sheet discriminator	0.55 ± 0.28	35.93 ± 6.26	$\mathbf{0.173 \pm 0.02}$
gg-dWJS w/ Instability discriminator	$\mathbf{0.56 \pm 0.27}$	$\mathbf{31.32 \pm 5.21}$	0.170 ± 0.018

We report the results of our experiments in table 2. Specifically, we report the distributional conformity score (DCS), percentage of beta sheets, and instability index for samples generated by dWJS and gg-dWJS with the two discriminators.[1] The result shows that not only does gg-dWJS generate antibody sequences that are better optimized for their respective objectives, but it also produces sequences of higher quality overall. Details of this experiment are in appendix C

6 CONCLUSION

To speed up the discovery and delivery of antibody-based drugs, it is important to optimize the therapeutic antibody sequence, considering factors such as efficacy, safety, and manufacturability. While there are established language models for protein sequence generation using language modeling, using them for antibody sequences is difficult because of high-quality data scarcity and high-entropy variable regions in the sequence. Thus, denoising models can be a promising alternative for modeling such discrete modality through learning the denoising gradient in continuous space. One such method is dWJS, which uses smoothed data to learn the denoising gradient and NEB to jump back to the discrete data manifold. One drawback of this method is that it cannot produce samples that optimize specific antibody attributes, which is important for real-world antibody discovery. Our proposed method gg-dWJS learns a discriminative model on the smoothed data and uses the gradient information to augment its denoising walk towards the local maxima given some attributes. Finally, using a conditional jump using the same model, it returns to the clean data manifold. Thus, our method requires no additional score model training for different optimization tasks.

[1]Because of the unavailability of the open-source implementation of DCS by Frey et al. (2024), we implement the metric following the provided algorithm, which produces the same result for dWJS (0.49) reported by the authors.

REFERENCES

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. 10

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021. 10

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023. 10

Miguel A Carreira-Perpinan and Geoffrey Hinton. On contrastive divergence learning. In *International workshop on artificial intelligence and statistics*, pp. 33–40. PMLR, 2005. 10

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3itjR9QxFw. 10

Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009. 11

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022. 2

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 12

James Dunbar and Charlotte M Deane. Anarci: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 2016. 5

Noelia Ferruz and Birte Höcker. Controllable protein design with language models. *Nature Machine Intelligence*, 4(6):521–532, 2022. 11

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022. 11

Nathan C Frey, Daniel Berenberg, Karina Zadorozhny, Joseph Kleinhenz, Julien Lafrance-Vanasse, Isidro Hotzel, Yan Wu, Stephen Ra, Richard Bonneau, Kyunghyun Cho, et al. Protein discovery with discrete walk-jump sampling. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zMPHKOmQNb. 1, 2, 5, 10, 11, 12

Alan E Gelfand. Gibbs sampling. *Journal of the American statistical Association*, 95(452):1300–1304, 2000. 10

Edward I George and Robert E McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993. 10

Vladimir Gligorijević, Daniel Berenberg, Stephen Ra, Andrew Watkins, Simon Kelow, Kyunghyun Cho, and Richard Bonneau. Function-guided protein design by deep manifold sampling. *bioRxiv*, pp. 2021–12, 2021. 11

Will Grathwohl, Kevin Swersky, Milad Hashemi, David Duvenaud, and Chris Maddison. Oops i took a gradient: Scalable sampling for discrete distributions. In *International Conference on Machine Learning*, pp. 3831–3841. PMLR, 2021. 10

Nate Gruver, Samuel Don Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=MfiK69Ga6p. 1, 5, 11

Kunchur Guruprasad, BV Bhasker Reddy, and Madhusudan W Pandit. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Engineering, Design and Selection*, 4(2):155–161, 1990. 5, 12

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 4

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 10

Annemarie Honegger and Andreas PluÈckthun. Yet another numbering scheme for immunoglobulin variable domains: an automatic modeling and analysis tool. *Journal of molecular biology*, 309 (3):657–670, 2001. 5

Peter J Hudson and Christelle Souriau. Engineered antibodies. *Nature medicine*, 9(1):129–134, 2003. 1

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 2

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure FP Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, et al. Biological sequence design with gflownets. In *International Conference on Machine Learning*, pp. 9786–9801. PMLR, 2022. 1

Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi S Jaakkola. Iterative refinement graph neural network for antibody sequence-structure co-design. In *International Conference on Learning Representations*, 2021. 12

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016. 11

Bahjat Kawar, Roy Ganz, and Michael Elad. Enhancing diffusion-based image synthesis with robust classifier guidance. *Transactions on Machine Learning Research*, 2022. 12

Jack Kyte and Russell F Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology*, 157(1):105–132, 1982. 12

Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pp. 536–543, 2008. 3

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 10

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. 11

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`. 11

Ruei-Min Lu, Yu-Chyi Hwang, I-Ju Liu, Chi-Chiu Lee, Han-Zen Tsai, Hsin-Jung Li, and Han-Chung Wu. Development of therapeutic antibodies for the treatment of diseases. *Journal of biomedical science*, 27(1):1–30, 2020. 1

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. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022. 1, 11

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020. 11

Emily K Makowski, Patrick C Kinnunen, Jie Huang, Lina Wu, Matthew D Smith, Tiexin Wang, Alec A Desai, Craig N Streu, Yulei Zhang, Jennifer M Zupancic, et al. Co-optimization of therapeutic antibody affinity and specificity using machine learning models that generalize to novel mutational space. *Nature communications*, 13(1):3788, 2022. 1

Koichi Miyasawa et al. An empirical bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist*, 38(181-188):1–2, 1961. 2

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. 11

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 11

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011. 12

Zhangzhi Peng, Chenchen Han, Xiaohan Wang, Dapeng Li, and Fajiie Yuan. Generative diffusion models for antibody design, docking, and optimization. *bioRxiv*, pp. 2023–09, 2023. 11

Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and basic theory*, pp. 388–394. Springer, 1992. 2

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015. 11

Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature communications*, 14(1):2389, 2023. 5

Matthias Sachs, Benedict Leimkuhler, and Vincent Danos. Langevin dynamics with variable coefficients and nonconservative forces: from stationary states to numerical methods. *Entropy*, 19(12): 647, 2017. 3

Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pp. 872–879. ACM, 2008. 3

Saeed Saremi and Aapo Hyvärinen. Neural empirical bayes. *Journal of Machine Learning Research*, 20(181):1–23, 2019. 2

David Sehnal, Sebastian Bittrich, Mandar Deshpande, Radka Svobodová, Karel Berka, Václav Bazgier, Sameer Velankar, Stephen K Burley, Jaroslav Koča, and Alexander S Rose. Mol* viewer: modern web app for 3d visualization and analysis of large biomolecular structures. *Nucleic acids research*, 49(W1):W431–W437, 2021. 5

Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. Generative language modeling for antibody design. *bioRxiv*, pp. 2021–12, 2021. 11

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 10

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=PxTIG12RRHS`. 10

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=BYWWwSY2G5s. 10

Natasa Tagasovska, Nathan C. Frey, Andreas Loukas, Isidro Hotzel, Julien Lafrance-Vanasse, Ryan Lewis Kelly, Yan Wu, Arvind Rajpal, Richard Bonneau, Kyunghyun Cho, Stephen Ra, and Vladimir Gligorijevic. A pareto-optimal compositional energy-based model for sampling and optimization of protein sequences. In *NeurIPS 2022 AI for Science: Progress and Promises*, 2022. URL https://openreview.net/forum?id=U2rNXaTTXPQ. 11

Danqing Wang, YE Fei, and Hao Zhou. On pre-training language model for antibody. In *The Eleventh International Conference on Learning Representations*, 2022. 1

Anna M Wu. Engineered antibodies for molecular imaging of cancer. *Methods*, 65(1):139–147, 2014. 1

Jin Xu, Zishan Li, Bowen Du, Miaomiao Zhang, and Jing Liu. Reluplex made more practical: Leaky relu. In *2020 IEEE Symposium on Computers and communications (ISCC)*, pp. 1–7. IEEE, 2020. 11

Dinghuai Zhang, Nikolay Malkin, Zhen Liu, Alexandra Volokhova, Aaron Courville, and Yoshua Bengio. Generative flow networks for discrete probabilistic modeling. In *International Conference on Machine Learning*, pp. 26412–26428. PMLR, 2022. 10

Supplementary material for

Antibody sequence optimization with gradient-guided discrete walk-jump sampling

Zarif Ikram, Dianbo Liu, M Saifur Rahman

ACRONYMS

AA amino acid. 5

DAG directed acyclic graph. 10

DCS distributional conformity score. 5, 12

dWJS discrete walk-jump sampling. 1–5

EBM energy based model. 10

FID Fréchet inception distance. 4

GFlowNets Generative Flow Networks. 10

gg-dWJS gradient-guided discrete walk-jump sampling. 1–5

GRAVY grand average of hydropathicity. 12

KDE kernel density estimation. 12

MCMC Markov chain Monte Carlo. 1–3

NEB Neural empirical Bayes. 2, 3, 5, 10

OAS observed antibody space. 5

A RELATED WORK

A.1 DISCRETE GENERATIVE MODELS

A large class of discrete generative models consist of autoregressive models (i.e., language models). Among many others, Austin et al. (2021) learns the posterior alphabet distribution over the prior generated data by learning the bidirectional context in a language model. Recent works such as Generative Flow Networks (GFlowNets) (Bengio et al., 2023; 2021) model the flow as a DAG, and learn to sample directly proportional to a given reward function. Zhang et al. (2022) uses this idea to train and sample from an energy based model (EBM) (LeCun et al., 2006) using contrastive divergence (Carreira-Perpinan & Hinton, 2005). Besides, Grathwohl et al. (2021) improves the sampling process by leveraging local gradients using a Gibbs sampler (George & McCulloch, 1993; Gelfand, 2000).

Another way to generate data is through denoising models (Ho et al., 2020; Song & Ermon, 2020; Song et al., 2021). These models learn the gradient log density of the data and generate continuous data from the perturbed data distribution. Thus, they are faster and more efficient than autoregressive models. Of course, one cannot simply apply the denoising gradient to the discrete data distribution because the gradients are not defined there. Many promising works attempt to remedy this by proposing different techniques. Chen et al. (2023) simply transforms the discrete data into analogue bits, learns the denoising gradient, and applies thresholding to return the categorical data. Sun et al. (2023) applies denoising via a continuous-time Markov chain to the categorical data using a stochastic jump process. Frey et al. (2024) leverages the gradient information in the smoothed data manifold and jumps to the true data manifold using NEB.

A.2 PROTEIN SEQUENCE OPTIMIZATION

With the advancement of language models, many recent works (Madani et al., 2020; Nijkamp et al., 2023; Ferruz et al., 2022) propose language models pre-trained on protein data to generate protein sequences, often for tasks such as structure prediction (Lin et al., 2023). These protein language models have also been adopted for antibody generation tasks, both unguided (Shuai et al., 2021) and guided (Gligorijević et al., 2021; Ferruz & Höcker, 2022; Tagasovska et al., 2022). A key problem with language modeling of antibodies is that they struggle to capture the data distribution of the antibody sequences given a limited amount of high-quality data and high-entropy variable regions of the antibody sequences. As such, recent works such as Luo et al. (2022); Gruver et al. (2023); Peng et al. (2023) leverage the continuous space of antibody structures to generate antibody structures using diffusion. In this work, we focus on the problem of generating targeted antibody sequences.

B ADDITIONAL DETAILS ON THE STATIC MNIST EXPERIMENT

We train the noisy score and discriminator model on the binarized static MNIST dataset.[2] For score model architecture, we use a U-Net architecture (Ronneberger et al., 2015) that takes the smoothed one-hot representation of the discrete images and returns the score of the same shape. Finally, we train a CNN architecture [3] on smoothed data and their corresponding labels for the discriminator model.

Figure 4: Randomly chosen 400 static MNIST test-set samples.

C ADDITIONAL DETAILS ON THE ANTIBODY SEQUENCE OPTIMIZATION EXPERIMENT

C.1 SCORE MODEL

We follow Frey et al. (2024) for the score model implementation available at `https://github.com/Genentech/walk-jump`. We use a 35-layer Bytenet (Kalchbrenner et al., 2016) architecture with a hidden layer of 128. To train the model from scratch, we utilize a batch size of 64 and the AdamW optimizer (Loshchilov & Hutter, 2019) in PyTorch (Paszke et al., 2019) with early stopping. The training parameters include a learning rate of 10^{-4} and a weight decay of 0.01. We conducted the training using four NVidia A100 GPUs. The training process takes approximately 10 hours. For training the score model, we use $\sigma = 1$.

C.2 DISCRIMINATOR MODELS

We train the discriminator model on two sequence-related properties: the percentage of beta sheets and the protein instability index. We label the training sequences using BioPython (Cock et al., 2009). For the architecture of the smoothed predictor, we adopt a 3-layer 1D-CNN followed by a 3-layer MLP integrated into the existing ByteNet architecture, incorporating leakyReLU (Xu et al., 2020) activations between the layers. Finally, we follow the same training parameters as the score model training.

[2] Collected from Kaggle

[3] Based on the PyTorch example, adapted for discrete data.

C.3 EVALUATION METRICS

The fraction of beta-pleated sheets provides insights into the structural characteristics of antibody sequences, influencing their stability and functionality. The protein stability index is a test proposed by Guruprasad et al. (1990). A protein sequence has a short half-life if its stability index is above 40. In the context of therapeutic antibody generation, stability is, thus, a very important metric, as it is linked with manufacturability. The recently proposed DCS measures sample quality using a sample-to-distribution metric, as opposed to previous sample recovery metrics (Jin et al., 2021). We implement DCS following Frey et al. (2024), using two sequence based properties: molecular weight and grand average of hydropathicity (GRAVY) (Kyte & Doolittle, 1982). We used the kernel density estimation (KDE) implementation by Scikit-learn (Pedregosa et al., 2011) to approximate the joint distribution using a Gaussian kernel with a bandwidth of 0.15.

D DISCUSSION

Why does discriminator gradient guidance produces higher quality samples? We speculate that the guidance assists the denoising walk by reducing the discrete space. Besides, by working alongside the denoising score, it reinforces the step taken with respect to a specific label, thereby improving the fidelity of the data generated. There are many works such as Dhariwal & Nichol (2021); Kawar et al. (2022) that report the same phenomenon. We leave the experimental validation of this underlying effect for a future work.

E FUTURE WORK

An exciting direction of our current work is to generate antibody sequences targeting specific antigens by estimating their binding affinities. Besides, in a real-world setting, we are more interested in multi-objective optimization than the single-objective optimization presented in this work. Thus, using our method in a multi-objective optimization setting by combining different discriminator can be an interesting direction. Besides, we did not explore hyperparameter tuning such as σ and gradient guidance strength in this work. We leave these for the future iteration of this work.