Score-Based Generative Models for Molecule Generation

Dwaraknath Gnaneshwar
Julia Computing, Inc
dwaraknath.gnaneshwar@juliacomputing.com

Bharath Ramsundar
Deep Forest Sciences Inc.
bharath@deepforestsci.com

Dhairya Gandhi
Julia Computing, Inc
dhairya@juliacomputing.com

Rachel Kurchin
Department of Mechanical Engineering
Carnegie Mellon University
rkurchin@cmu.edu

Venkatasubramanian Viswanathan
Department of Mechanical Engineering
Carnegie Mellon University
venkvis@cmu.edu

Abstract

Recent advances in generative models have made exploring design spaces easier for de novo molecule generation. However, popular generative models like GANs and normalizing flows face challenges such as training instabilities due to adversarial training and architectural constraints, respectively. Score-based generative models sidestep these challenges by modelling the gradient of the log probability density using a score function approximation, as opposed to modelling the density function directly, and sampling from it using annealed Langevin Dynamics. We believe that score-based generative models could open up new opportunities in molecule generation due to their architectural flexibility, such as replacing the score function with an SE(3) equivariant model. In this work, we lay the foundations by testing the efficacy of score-based models for molecule generation. We train a Transformer-based score function on Self-Referencing Embedded Strings (SELFIES) representations of 1.5 million samples from the ZINC dataset and use the Moses benchmarking framework to evaluate the generated samples on a suite of metrics.

1 Introduction

Recent advances in generative models have enabled the design of novel molecules with targeted properties. Prior work has explored Generative Adversarial Networks (GANs) [1], variational autoencoders (VAEs) [2, 3] and normalizing flows (NFs) [4–6] for molecule generation. There are a few significant challenges with each of these approaches. Likelihood-based models language models and NFs place restrictions on the model architecture to ensure a tractable normalizing constant for log-likelihood computation. Such architectural constraints limit model expressivity and increase the complexity of both the training procedure and the model design. Moreover, altering these models to include recent advances requires significant efforts in redesigning to accommodate the constraints. GANs side-step these training challenges by modelling the probability distribution implicitly. However, their training procedure is prone to instabilities [7, 8].

ELLIS Machine Learning for Molecule Discovery Workshop 2021. https://moleculediscovery.github.io/workshop2021/

Recently, score-based generative models [9] have garnered increasing interest due to their success in generating data of various modalities such as images [10, 11], audio [12, 13], and graphs [14]. Score-based models model the gradient of the log probability density function, a quantity known as *score* [15], using a model called the *score function*. They do not require having a tractable normalizing constant as they don't use the density function directly in their formulation. An extensive review of score-based models can be found in 2, but the key attribute for our purposes is the flexibility of architecture of the score function.

1.1 Why Score-Based Models?

Having no restrictions on the score function opens up interesting new avenues in molecule generation that are more challenging with other generative models. For example, consider equivariance under symmetry groups, a desirable property to increase data efficiency and reduce model size without sacrificing expressivity. Reference [4] explores a NF model that is equivariant under Euclidean symmetries to generate 3D molecules. They parameterize a continuous-time flow, where the first derivative is modelled by an Equivariant Graph Neural Network (EGNN) [4]. Implementing novel equivariant architectures or other improvements to generative models like GANs or NFs might require significant efforts due to normalization constraints. However, given the fact that score-based models do not place any such restrictions on the architecture of the score function, leveraging new developments is a straightforward matter of drop-in replacement.

We believe that we can leverage this flexibility to enable novel and high-performance generative models for molecules. However, score-based generative models are not easy to train for molecule generation and it is not yet clear what best practices are to train state-of-the-art score-based models for molecules. In this work, we lay the foundations to establish such best practices, and hope it can serve to ignite a broader discussion within the community. As an initial effort to this end, we present results from training a Transformer-based score function on the ZINC dataset [16].

2 Score-Based Generative Models

Consider a set of samples $\{\mathbf{x}_i\}_{i=1}^N$ from a data-generating distribution $p_d(\mathbf{x})$. Score-Based generative models model the *score* of the probability density $p_d(\mathbf{x})$, defined as $\nabla_\mathbf{x} \log p_d(\mathbf{x})$, the gradient of the log probability density at the input data point. Score is a vector field which points in the direction of greatest growth in log density. The procedure is to generate a starting point and move it towards high density regions along the direction of this vector field. We train a model (typically a neural network) to serve as this *score function* $\mathbf{s}_\theta : \mathbb{R}^D \to \mathbb{R}^D$, to learn the vector field from a data point using denoising score matching [17] such that $\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_\mathbf{x} \log p_d(\mathbf{x})$.

Score matching We train the score function on i.i.d. samples from the data distribution with the goal of directly estimating the score without estimating the density function $p_d(\mathbf{x})$. However, naïve implementations of such score matching can lead to biased results because the loss function is weighted by the $p_d(\mathbf{x})$, resulting in inaccurate estimates of the score in low-density regions. This challenge is addressed in [9] by training the score functions on perturbed data points (that is, Gaussian noise is added to data and the model is trained to denoise the sample). They observe that perturbing data with random Gaussian noise makes the data distribution more amenable to score-based generative modelling. The overall training procedure then becomes 1) perturbing the data using various levels of noise; and 2) simultaneously estimating scores corresponding to all noise levels by training a single conditional score network. Reference [10] proposes using a geometric series of increasing standard deviations $\sigma_1 < \sigma_2 .. < \sigma_L$ to obtain the noise-perturbed distribution $p_{\sigma_i}(\mathbf{x})$. The training objective then becomes

$$\sum_{i=1}^{L} \lambda(i) \mathbb{E}_{p_{\sigma_i}(\mathbf{x})} \left[||\mathbf{s}_\theta(\mathbf{x}, i) - \nabla_\mathbf{x} \log p_{\sigma_i}(\mathbf{x})||_2^2 \right], \tag{1}$$

where $\lambda(i) \in \mathbb{R}_{>0}$ is a positive weighting function. Reference [10] also provides practical recommendations to choose the various hyperparameters such as σ_L, σ_1, L, etc.

Langevin Dynamics We now elaborate the iterative procedure used to draw samples from the trained score-based models. Given a step size $\alpha > 0$ and a total number of iterations T, an initial sample is drawn from a prior distribution $p_\pi(\mathbf{x})$. We iteratively move the sample along the vector

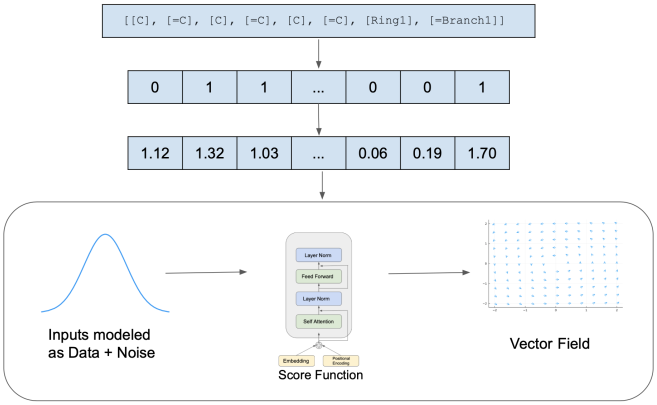

Figure 1: The training flow for our score-based architecture. Compounds are input as SELFIES strings, which are converted into a one-hot encoding. Small amounts of Gaussian noise are added to this encoding, and is then passed into a transformer score function which outputs the vector field.

field provided by the score function for T steps (we are following standard terminology in this field, but note that the update equation doesn't directly match the standard physics definition of Langevin dynamics):

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \alpha \nabla_{\mathbf{x}} \log p(\mathbf{x}_{t-1}) + \sqrt{2\alpha}\mathbf{z}_t, \quad 1 \le t \le T, \tag{2}$$

where $\mathbf{z}_i \sim \mathcal{N}(0, I)$. When $\alpha \to 0$ and $T \to \infty$, the $\mathbf{x}_T$ obtained converge to a sample from $p_d(\mathbf{x})$. Langevin Dynamics does not access $p_d(\mathbf{x})$ directly, instead we use $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ through the score function $\mathbf{s}_\theta(\mathbf{x})$. Reference [9] incorporates the multiple noise levels solution to address the challenge of inaccurate score estimates by running Langevin Dynamics sequentially on the noise levels. This method is called annealed Langevin Dynamics.

3 Experiments

We test the capacity of score-based generative models to generate molecules from their text representations in a canonical setting. We train our models on 1.6M drug-like molecules from the ZINC dataset [16] for 150k steps with a batch size of 256. We use a Transformer model as our score function. The molecules are represented using the SELF-Referencing Embedded Strings (SELFIEs) representation [18]. We use the MOSES framework [19] to evaluate our generated samples. All metrics are calculated on a set of 30k newly generated molecules from our model, following the recommendation from the MOSES framework. We trained all our models on a single Nvidia Tesla V100 GPU. Training the model took approximately 12 hours while time taken for sampling was largely dependent on the number of time steps for each noise scale (T). With $T = 10$, generating 30k samples took approximately 4 hours.

Hyperparameters The score functions are modelled after the encoder layers of Transformer [20] with six layers. The hidden dimensions of the layers are 512 with four attention heads. We use the Adam optimizer [21] with learning rate 10^{-4}. Following the recommendations made in [10], we set σ_1 to 10, which is the maximum Euclidean distance between all pairs of training data points. (We encode the SELFIES string for a sample as a $V \times L$ matrix. Here V is the vocabulary size for tokens, and L is the maximum string length in the dataset. Each column is a one-hot encoded representation of the token at that position. We flatten this binary matrix into a vector and compute the L^2 distance). We chose $\sigma_1 = 0.01$ and a total of 350 levels between σ_L, σ_1.

4 Discussion

Results from running MOSES benchmarks are shown in Table 1. High novelty and validity percentages indicate that the model is able to generate a novel set of structurally valid compounds.

Metric	HMM	VAE	JTN-VAE	This Work
Valid (↑)	0.07	0.97	1.0	**1.0** ± 0.0
Unique@1k (↑)	0.62	1.0	1.0	0.88 ± 0.06
Unique@10k (↑)	0.56	0.99	0.99	0.82 ± 0.05
Filters (↑)	0.90	0.99	0.97	0.37 ± 0.11
Novelty (↑)	0.99	0.69	0.91	**1.0** ± 0.0
IntDiv$_1$	0.84	0.85	0.85	**0.90** ± 0.04
IntDiv$_2$	0.81	0.85	0.84	**0.88** ± 0.04
FCD/Test(↓)	24.46	0.09	0.39	39.84 ± 2.63
FCD/TestSF(↓)	25.43	0.56	0.93	40.92 ± 2.77

Table 1: Experiment results on the 1.6M sample version of Zinc dataset with a Tranformer model as the score function. All results are reported over three independent model initializations. Generated samples are novel and diverse, but don't closely match the training distribution. ↑ and ↓ mean higher and lower are better.

‘

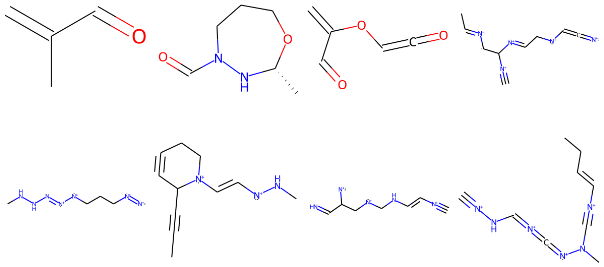

Figure 2: Randomly sampled molecules from a set of 30k molecules generated from score-based model.

Uniqueness scores are lower than VAE models but are still competitive. Uniqueness could possibly be improved with finer tuning of the noise scales. High **IntDiv**$_1$, **IntDiv**$_2$ scores, which assess the chemical diversity within the generated set, suggest that the model is not suffering from mode collapse, as mode collapse often leads to producing limited variety of samples. Low **Filters** scores and **FCD** scores indicate that while the generated molecules are chemically valid, they are far from the training distribution. On the one hand, generating diverse chemical matter distinct from the training distribution can prove valuable, but it is possible the model lacks the representation power to fully model the training distribution.

5 Acknowledgements

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001211. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

6 Conclusion and Future Work

In this work, we explore the use of score-based generative models for the task of molecule generation. We train a Transformer-based score-based generative model on the 1.6 million sample version of the ZINC dataset and evaluate the generated molecules using the MOSES framework. The results indicate that score-based models are capable of generating diverse and unique molecules but the generated points, however, are far from the training distribution.

Future work will improve the score function models to include structural information about the molecule and train score-based models to generate molecule graphs directly instead of SELFIES. We plan to incorporate recent advances in equivariant operations to test the efficacy of equivariant score-based generative models for molecule graph generation. We hypothesize that richer choices of models may enable score-based methods to fully learn rich training distributions. Alternatively, it is possible that training a large enough transformer model may suffice to learn these distributions. We plan to explore both hypotheses.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[2] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[3] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," in *International conference on machine learning*, PMLR, 2018, pp. 2323–2332.

[4] V. G. Satorras, E. Hoogeboom, F. B. Fuchs, I. Posner, and M. Welling, "E (n) equivariant normalizing flows for molecule generation in 3d," *arXiv preprint arXiv:2105.09016*, 2021.

[5] N. C. Frey and B. Ramsundar, "Flow-based models for active molecular graph generation,"

[6] K. Madhawa, K. Ishiguro, K. Nakago, and M. Abe, "Graphnvp: An invertible flow model for generating molecular graphs," *arXiv preprint arXiv:1905.11600*, 2019.

[7] R. Durall, A. Chatzimichailidis, P. Labus, and J. Keuper, "Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues," *arXiv preprint arXiv:2012.09673*, 2020.

[8] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in gans," in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–10.

[9] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *arXiv preprint arXiv:1907.05600*, 2019.

[10] ——, "Improved techniques for training score-based generative models," *arXiv preprint arXiv:2006.09011*, 2020.

[11] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *arXiv preprint arXiv:2105.05233*, 2021.

[12] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020.

[13] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[14] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover, and S. Ermon, "Permutation invariant graph generation via score-based generative modeling," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 4474–4484.

[15] Q. Liu, J. Lee, and M. Jordan, "A kernelized stein discrepancy for goodness-of-fit tests," in *International conference on machine learning*, PMLR, 2016, pp. 276–284.

[16] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, and R. G. Coleman, "Zinc: A free tool to discover chemistry for biology," *Journal of chemical information and modeling*, vol. 52, no. 7, pp. 1757–1768, 2012.

[17] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching.," *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.

[18] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "Self-referencing embedded strings (selfies): A 100% robust molecular string representation," *Machine Learning: Science and Technology*, vol. 1, no. 4, p. 045 024, 2020.

[19] D. Polykovskiy, O. Tatanov, A. Zhebrak, S. Belyaev, B. Sanchez-Lengeling, R. Kurbanov, and A. Zhavoronkov, "Molecular sets (moses): A benchmarking platform for molecular generation models. arxiv: 181112823," 2018.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.