Sampling from Energy distributions with Target Concrete Score Identity

Kholkin Sergei
Applied AI Institute, Moscow, Russia
kholkinsd@gmail.com

Francisco Vargas
Xaira Therapeutics
vargfran@gmail.com

Alexander Korotin
Applied AI Institute, Moscow, Russia
iamalexkorotin@gmail.com

Abstract

We introduce the *Target Concrete Score Identity Sampler* (TCSIS), a method for sampling from unnormalized densities on discrete state spaces by learning the reverse dynamics of a Continuous-Time Markov Chain (CTMC). Our approach builds on a forward in time CTMC with a uniform noising kernel and relies on the proposed *Target Concrete Score Identity*, which relates the concrete score, the ratio of marginal probabilities of two states, to a ratio of expectations of Boltzmann factors under the forward uniform diffusion kernel. This formulation enables Monte Carlo estimation of the concrete score without requiring samples from the target distribution or computation of the partition function. We approximate the concrete score with a neural network and propose two algorithms: *Self-Normalized TCSIS* and *Unbiased TCSIS*. Finally, we demonstrate the effectiveness of TCSIS on problems from statistical physics.

1 Introduction

Let $p(x)$ be a probability distribution on finite space S:

$$p(x) = \frac{\overline{p}(x)}{Z}, \quad Z = \sum_{x \in S} \overline{p}(x), \tag{1}$$

where $\overline{p}(x) : S \to \mathbb{R}_+$ is the unnormalized density, also known as Boltzmann distribution, and the normalization constant Z is generally intractable. In this work we are interested in sampling from $p(x)$ given only $\overline{p}(x)$. This task is central in Bayesian inference [16], statistical physics [18], and computational biology [11], yet remains challenging in large state spaces.

Sampling from unnormalized densities is well studied in continuous spaces, where Markov Chain Monte Carlo (MCMC) serves as the standard baseline [17, 15]. However, MCMC methods often suffer from poor mixing, slow convergence, and local trapping [17]. These challenges have motivated the development of neural samplers [24, 25, 8], parameterized by neural networks.

In discrete state spaces, MCMC methods are also a strong baseline [12, 7], but neural approaches are only beginning to emerge. Recent work on discrete diffusion models and learned CTMCs [4, 13, 21] has enabled neural samplers for combinatorial optimization [22] and statistical physics [10, 19]. For example, SDDS [22] trains a diffusion via policy gradients and self-normalized importance sampling, while LEAPS [10] and DNFS [19] use temperature annealing between a Boltzmann distribution and noise, combined with coordinate descent to learn a CTMC sampler.

The goal of this work is to leverage recent advances in discrete diffusion models and learnable CTMCs, specifically probability factorization and concrete scores, to design a learnable sampler

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference: Sampling Meets Learning.

from unnormalized densities. We state a *Target Concrete Score Identity* that allows one to estimate the concrete score at given time-moment t for uniform discrete diffusion forward process [3, 23] given only the energy function. Further we propose *Target Concrete Score Identity Sampler* (**TCSIS**) framework and two practical algorithms: 1) **Self normalized**, to learn the concrete score directly 2) **Unbiased** to learn the unnormalized marginal densities and construct concrete score out of them during the inference stage. Finally we do evaluate our method on the problems from statistical physics.

2 Continuous-Time Markov Chains

We begin by the introduction of Continuous-Time Markov Chains (CTMCs) [4, 13]. We consider finite state spaces S of the form $S = \mathcal{X}^d$, where $\mathcal{X} \in \{1, .., V\}$ is a set of tokens. robability distributions over the discrete space S are represented by probability mass vectors $p \in \mathbb{R}^{|S|}$, where each entry $p(x)$ denotes the probability of state x, and the entries sum up to one. A CTMC defines a family of time-indexed distributions p_t evolving according to linear ordinary differential equation:

$$\frac{\partial p_t}{\partial t} = Q_t p_t, \quad p_{t=0}(x) = p_0(x), \tag{2}$$

where $Q_t(y, x) : S \times S \to \mathbb{R}$ is a transition rate matrix which captures the rate of change of transition probabilities and have non-negative non-diagonal entries and columns which sum to zero. Since the ODE (2) describes the infinitesimal dynamics of a CTMC, the process can be simulated by discretizing time into small steps. Over an infinitesimal interval, the chain either stays in its current state or transitions to another, with rates governed by the entries of the transition rate matrix Q_t.

Finally, this process has a well-known time-reversal [4, 13], characterized by another matrix $\overline{Q}_t$:

$$\frac{\partial p_{T-t}}{\partial t} = \overline{Q}_{T-t} p_{T-t}, \quad \overline{Q}_t(y, x) = \begin{cases} \frac{p_t(y)}{p_t(x)} Q_t(x, y), y \neq x \\ -\sum_{y \neq x} Q_t(x, y), y = x \end{cases} \tag{3}$$

Here, ratio $\frac{p_t(y)}{p_t(x)} = s_t(x)_y$ is known as the **concrete score** [14, 13]. In literature, it is typically represented in vector form, $s_t(x) = \left[\frac{p_t(y)}{p_t(x)}\right]_{y \in \mathcal{X}^d}$, so that $s_t(x) : \mathcal{X}^d \times \mathbb{R} \to \mathbb{R}^{|\mathcal{X}^d|}$. One can see that obtaining the concrete score is **sufficient** for defining the reverse dynamics.

We call a special case of a CTMC in which the forward process, defined by Q_t, gradually corrupts the data towards a simple distribution a *discrete diffusion model*. The forward dynamics is known, while the backward process is learned, typically from data. To remain scalable, such models parameterize transition rate matrices or concrete scores in a factorized manner [4, 14], i.e., model transitions only between sequences $y, x \in \mathcal{X}^d$ restricted to their **1-Hamming neighborhoods**. Then, the concrete score notation changes $s_t(x) : \mathcal{X}^d \times \mathbb{R} \to \mathbb{R}^{d \times V}$. For additional information see Appendix D.2. For simplicity we use general notation, but in practice we adopt this factorization as well.

We focus on **uniform discrete diffusion models** [23, 13, 3], where the forward dynamics is defined via uniform noising kernel and the terminal distribution is uniform, see Appendix D.1.

3 Method

In the Section 3.1 we provide an expression for the concrete score of discrete diffusion with uniform noising process. In Section 3.2 we propose general framework for constructing a sampler from unnormalized density $\overline{p}(x) = p(x)Z$ and a way to learn related concrete scores without access to samples from $p(x)$. In Sections 3.3 and 3.4 we propose two implementations of described in Section 3.2 framework with different loss functions and different parametrizations of a neural network.

3.1 Target Concrete Score Identity

The central component of our work is the notion of the **concrete score** [14]. Prior works [14, 13] propose methodologies for learning the concrete score from data samples.

Proposition 1 (Target Concrete Score Identity)**.** *Consider a probability distribution $p(x)$ with finite support $S = \mathcal{X}^d$, where $\mathcal{X} = \{1, ..., V\}$ and its unnormalized density $\overline{p}(x) = p(x)Z$, where Z is unknown. Consider the continuous in time dynamics determined by uniform forward noising kernel (12). Then the ratio of its marginal distributions at time t, i.e, concrete score $s_t(x)$, has the form:*

$$s_t(x)_y = \frac{p_t(y)}{p_t(x)} = \frac{\mathbb{E}_{p_{t|0}(x_0|y)}[\overline{p}(x_0)]}{\mathbb{E}_{p_{t|0}(x_0|x)}[\overline{p}(x_0)]}.$$

For proof see Appendix B. This proposition can be viewed as discrete state space counterpart of Proposition 2.1 in [5]. Further, Proposition 1 enables Monte Carlo estimation of concrete score $s_t(x)_y$, given x and y. Relatedly, [27] introduces a similar to Proposition 1 identity for the concrete score of the posterior $p_{0|t}$, which they employ for training of Masked Diffusion Models [21].

3.2 Target Concrete Score Identity Sampler

Next, we use the Target Concrete Score Identity to learn a reverse discrete diffusion for sampling from an unnormalized density $\overline{p}(x) = p(x)Z$. Following [14], we approximate the concrete score with a neural network $s_t(x) \sim s_t^\theta(x)$ and train it by minimizing a loss. Since Proposition 1 gives a direct expression for the concrete score, it suffices to minimize the discrepancy between $s_t(x_t)$ and $s_t^\theta(x_t)$ over all x_t. We therefore propose the following objective:

$$L_\theta = \mathbb{E}_{q(x_t),t}[D(s_t(x_t), s_t^\theta(x_t))], \tag{4}$$

where $q(x_t)$ is *arbitrary distribution with full support*, $t \sim U[0, T]$ and D is divergence applicable for concrete scores. The neural network parametrization of the concrete score is given by $s_t^\theta(x_t) : \mathcal{X}^d \times \mathbb{R} \to \mathbb{R}^{d\times V}$. We call this framework *Target Concrete Score Identity Sampler* (**TCSIS**) and the learning procedure is described in Algorithm 1. We note that our training procedure in general *is simulation free*. After the concrete score $s_{\theta,t}$ is learned the sampling procedure follows regular discrete diffusion models practice and is given in Section 4 [13].

As one can notice the distribution $q(x_t)$ *can be varied as long as it has full support*. Following Proposition 1 [2] we suppose that more informative $q(x_t)$ that closely follows ground truth $p_t(x_t)$ is beneficial. We parametrize $q(x_t) = \mathbb{E}[q(x_0)p_{t|0}(x_t|x_0)]$ and propose several options for $q(x_0)$: 1) uniform noise 2) data generated by the learned model 3) result of MCMC sampling.

3.3 Self normalized TCSIS

Here we propose a practical algorithm for learning the concrete score. One can directly utilize the Proposition 1 and estimate both numerator and denominator by Monte Carlo.

$$s_t(x)_y \approx \hat{s}_t(x)_y = \frac{\sum_{i=0}^N \overline{p}(x_0^{i,y})}{\sum_{i=0}^N \overline{p}(x_0^{i,x})}, \quad x_0^{i,y} \sim p_{t|0}(x_0|y), x_0^{i,x} \sim p_{t|0}(x_0|x) \tag{5}$$

Then one can derive learning objective

$$\hat{L}_\theta^{\text{SNIS}} = \mathbb{E}_{q(x_t),t}[D^{\text{SE}}(\hat{s}_t(x_t), s_t^\theta(x_t))] = \mathbb{E}_{q(x_t),t}[\sum_{y\neq x_t}(s_t^\theta(x_t)_y - \hat{s}_t(x_t)_y \log s_t^\theta(x_t)_y)] + C, \tag{6}$$

where we have used Score Entropy, i.e. D^{SE}, [13] in (4) as D, but generally any D can be used. This formulation enables the parametrization of the concrete score $s_t^\theta(x_t)_y$ in log-space, i.e., $\log s_t(x_t)_y \approx \log s_t^\theta(x_t)_y$, which improves the stability of the method. In addition, similar to Eq 9 in [2] one can formulate Eq 5 as a self normalized importance sampling, see Appendix B. We call sampler parametrized and learned in such way *Self Normalized TCSIS*.

3.4 Unbiased TCSIS

On the other hand it is possible to parametrize by neural network not the concrete score, but the unnormalized marginal density and construct concrete score out of it. Than the following proposition holds.

Proposition 2 (Marginal densities parametrization)**.** *If $p_t^\theta(y)$ minimizes:*

$$L_\theta^{p_t} = \mathbb{E}_{q(x_t),t}[D^{SE}(\overline{p}_t(x_t), p_t^\theta(x_t))] = \mathbb{E}_{q(x_t),t}[p_t^\theta(x_t) - \overline{p}_t(x_t)\log p_t^\theta(x_t)] + C, \tag{7}$$

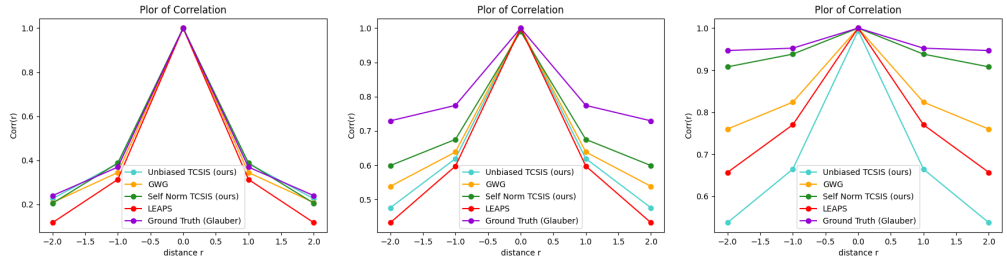
where $\overline{p}_t(x) = \mathbb{E}_{p_{t|0}(x_0|x)}[\overline{p}(x)]$. Then the concrete score estimator constructed in the following way:

$$s_t^\theta(x)_y = \frac{p_t^\theta(y)}{p_t^\theta(x)}, \tag{8}$$

minimizes (4). Furthermore, the estimation of objective $\hat{L}_\theta^{p_t} \approx L_\theta^{p_t}$, when $\overline{p}_t(x)$ is calculated via Monte Carlo, is unbiased.

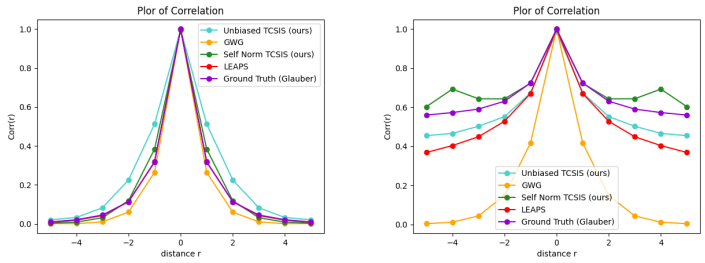
$$\hat{L}_\theta^{p_t} = \mathbb{E}_{q(x_t),t}[p_t^\theta(x_t) - \hat{\overline{p}}_t(x_t)\log p_t^\theta(x_t)] + C, \quad \hat{\overline{p}}_t(x) = \sum_i^N \overline{p}(x_0^i), \quad x_0^i \sim p_{t|0}(x_0|x) \tag{9}$$

We call sampler parametrized and learned in such way *Unbiased TCSIS*. The most important thing about Proposition 2 is unbiasedness of the learning objective, which is valuable. We note that the choice of using here the Score Entropy as D in Eq 4 is crucial as it allows for unbiased estimation with



(a) β_{high} (b) β_{critical} (c) β_{low}

Figure 1: Ising $L = 4$ lattice model correlation plots. The closer to ground truth the better.



(a) β_{high} (b) β_{critical}

Figure 2: Ising $L = 10$ lattice model correlation plots. The closer to ground truth the better.

parametrization of $p_t^\theta(x)$ in the log space. The neural network parametrization $\log p_t^\theta : \mathcal{X}^d \times \mathbb{R} \to \mathbb{R}$ allows one to assemble the concrete score in log space during the inference time, i.e., $\log s_t^\theta(x)_y = \log p_t^\theta(y) - \log p_t^\theta(x)$, however under the factorization assumption it requires $\mathcal{O}(dV)$ neural network evaluations.

4 Experiments

We evaluate both our methods: Self Normalized TCSIS and Unbiased TCSIS on problem set from statistical physics, i.e., the lattice Ising model. The Ising model on $L \times L$ lattice (with no magnetization) is defined by its energy distribution: $p(x) \propto \exp(\beta \sum_{\langle i,j \rangle} x_i x_j)$, where $x \in \{x_i\}_{i=1}^d$, $x_i \in \{0, 1\}$, β is the inverse temperature (interaction strength) and $\sum_{<i,j>}$ denotes the summation across nearest neighboring spins x_i, x_j on the lattice. Neighboring spins are uncorrelated at high temperature but reach a critical correlation when the temperature drops below a certain threshold.

We test our model on Ising lattices with inverse temperatures $\beta_{\text{high}} = 0.28$, $\beta_{\text{critical}} = 0.4407$, $\beta_{\text{low}} = 0.6$, since Ising model behaviour is known to be distinct at these temperatures by the phase transition theory [1]. In particular, we run on $L = 4$ lattice with all of the temperatures and on $L = 10$ lattice with $\beta_{\text{high}}, \beta_{\text{critical}}$. To validate our model we compute two point correlations, see Appendix E.2 and compare with another neural sampler LEAPS [10] and MCMC method GWG [7], all with 24 inference steps on $L = 4$ lattice and 64 inference steps on $L = 10$ lattice. The LEAPS sampler is run without importance sampling correction. As a ground truth we run Glauber dynamics for 10000 steps. For description of metrics, all the methods hyperparameters, experimental details and additional results, see Appendix E.

We find that Self-Normalized TCSIS consistently performs well across all temperature regimes and experimental settings, outperforming LEAPS. The Unbiased TCSIS achieves performance comparable to LEAPS overall. While the MCMC baseline performs strongly on small $L = 4$ lattices, its accuracy deteriorates as dimensionality increases and temperature decreases, which is a known challenge for MCMC methods [6]. In contrast, neural samplers demonstrate better scalability and maintain competitive performance in these regimes.

5 Discussion

We introduce the Target Concrete Score Identity and the corresponding family of Target Concrete Score Identity Samplers with simulation free training. Empirical results demonstrate that our approach is competitive across the considered tasks. Moreover, the framework of employing the concrete score for sampling establishes a promising direction for advancing the development of more efficient methods beyond our current approach.

Future improvements include approximating the concrete score via Taylor expansion [19, 7], exploring alternative architectures for Unbiased TCSIS to reduce the $\mathcal{O}(dV)$ inference cost, and developing a learning-free variant where the concrete score is estimated online using Proposition 1.

The limitations are discussed in Appendix A.

References

[1] Statistics of the two-dimensional ferromagnet. part i. *Physical Review*, 60(3):252, 1941.

[2] Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, et al. Iterated denoising energy matching for sampling from boltzmann densities. In *International Conference on Machine Learning*, pages 760–786. PMLR, 2024.

[3] 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.

[4] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

[5] Valentin De Bortoli, Michael Hutchinson, Peter Wirnsberger, and Arnaud Doucet. Target score matching. *arXiv preprint arXiv:2402.08667*, 2024.

[6] Katayoon Goshvadi, Haoran Sun, Xingchao Liu, Azade Nova, Ruqi Zhang, Will Grathwohl, Dale Schuurmans, and Hanjun Dai. Discs: A benchmark for discrete sampling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 79035–79066. Curran Associates, Inc., 2023.

[7] 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*, pages 3831–3841. PMLR, 2021.

[8] Aaron J Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Daniel S Levine, Brandon M Wood, Bin Hu, Brandon Amos, Brian Karrer, et al. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. In *Forty-second International Conference on Machine Learning*.

[9] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

[10] Peter Holderrieth, Michael Samuel Albergo, and Tommi Jaakkola. Leaps: A discrete neural sampler via locally equivariant networks. In *Forty-second International Conference on Machine Learning*.

[11] Nicolas Lartillot and Hervé Philippe. A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Molecular biology and evolution*, 21(6):1095–1109, 2004.

[12] Muheng Li and Ruqi Zhang. Reheated gradient-based discrete sampling for combinatorial optimization. *Transactions on Machine Learning Research*.

[13] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pages 32819–32848, 2024.

[14] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35:34532–34545, 2022.

[15] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[16] Iain Murray, Zoubin Ghahramani, and David JC MacKay. Mcmc for doubly-intractable distributions. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 359–366, 2006.

[17] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[18] MEJ Newman and GT Barkema. Monte carlo methods in statistical physics. *Monte Carlo methods in statistical physics/MEJ Newman and GT Barkema. Oxford: Clarendon Press*, 1999.

[19] Zijing Ou, Ruixiang Zhang, and Yingzhen Li. Discrete neural flow samplers with locally equivariant transformer. *arXiv preprint arXiv:2505.17741*, 2025.

[20] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[21] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[22] Sebastian Sanokowski, Wilhelm Franz Berghammer, Haoyu Peter Wang, Martin Ennemoser, Sepp Hochreiter, and Sebastian Lehner. Scalable discrete diffusion samplers: Combinatorial optimization and statistical physics. In *The Thirteenth International Conference on Learning Representations*.

[23] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P de Almeida, Alexander M Rush, Thomas PIERROT, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*.

[24] Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*.

[25] Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets variational inference: Controlled monte carlo diffusions. In *The Twelfth International Conference on Learning Representations*.

[26] Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*.

[27] Ruixiang ZHANG, Shuangfei Zhai, Yizhe Zhang, James Thornton, Zijing Ou, Joshua M Susskind, and Navdeep Jaitly. Target concrete score matching: A holistic framework for discrete diffusion. In *Forty-second International Conference on Machine Learning*.

A Limitations

We find both TCSIS methods struggle with growth of problem dimensionality and at lower temperatures. Self Normalized TCSIS requires a lot of energy evaluations for training, i.e., $\mathcal{O}(dVN)$ calls of energy function, which makes it inefficient for environments with costly calls of energy function, e.g., neural network. However worth noting that Taylor approximation and Top-k sampling can help to alleviate the problem [27, 7].

Unbiased TCSIS experiences costly inference stage, because one has to assemble the concrete score from approximated marginal densities, which requires $\mathcal{O}(dV)$ calls of $p_t^\theta(x)$ neural network. The Unbiased TCSIS loss $\hat{L}^{p_t}$ (9) has the $p_t^\theta(x_t)$ and $\hat{\overline{p}}_t(x_t)$ terms that do experience exponential asymptotic and can induce overflow in the values of energy function.

B Proofs

B.1 Proof of Target Concrete Score Identity

Proof.

$$p_t(x_t) = \mathbb{E}_{p(x_0)}[p_{t|0}(x_t|x_0)] = \sum_{x_0} p_{t|0}(x_t|x_0)p(x_0) = \big[\text{Eq 12}\big] = \sum_{x_0}\prod_{i=0}^{d} p_{t|0}(x_t^i|x_0^i)p(x_0)$$

Then let us analyze $p_{t|0}(x_t^i|x_0^i)$:

$$p_{t|0}(x_t^i|x_0^i) = \exp(\int_0^t \sigma_s ds Q^{\text{uniform}}(x_t^i, x_0^i)),$$

Next, recap that Q^{uniform} has the form:

$$Q^{\text{uniform}} = \begin{bmatrix} 1-N & 1 & \cdots & 1 \\ 1 & 1-N & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1-N \end{bmatrix}$$

and is *symmetric* by construction. Note that an exponent of symmetric matrix is as well symmetric. Then it follows:

$$p_{t|0}(x_t^i|x_0^i) = \exp(\int_0^t \sigma_s ds Q^{\text{uniform}}(x_t^i, x_0^i)) = \big[\text{symmetric}\big] = \exp(\int_0^t \sigma_s ds Q^{\text{uniform}}(x_t^i, x_0^i)) = p_{t|0}(x_0^i|x_t^i).$$

Getting back to marginal density:

$$p_t(x_t) = \sum_{x_0}\prod_{i=0}^{d} p_{t|0}(x_t^i|x_0^i)p(x_0) = \sum_{x_0}\prod_{i=0}^{d} p_{t|0}(x_0^i|x_t^i)p(x_0) =$$
$$= \sum_{x_0} p_{t|0}(x_0|x_t)p(x_0) = \mathbb{E}_{p_{t|0}(x_0|x_t)}[p(x_0)]$$

Then one can construct a concrete score with unnormarlized density $p(x_0) = \frac{\overline{p}(x_0)}{Z}$:

$$s_t(x)_y = \frac{p_t(y)}{p_t(x)} = \frac{\mathbb{E}_{p_{t|0}(x_0|y)}[p(x_0)]}{\mathbb{E}_{p_{t|0}(x_0|x)}[p(x_0)]} = \big[p(x_0) = \frac{\overline{p}(x_0)}{Z}\big] = \frac{\mathbb{E}_{p_{t|0}(x_0|y)}[\frac{p(x_0)}{Z}]}{\mathbb{E}_{p_{t|0}(x_0|x)}[\frac{p(x_0)}{Z}]} = \frac{\mathbb{E}_{p_{t|0}(x_0|y)}[\overline{p}(x_0)]}{\mathbb{E}_{p_{t|0}(x_0|x)}[\overline{p}(x_0)]}$$

□

Algorithm 1: Training of TCSIS.

Input : Energy function $E(x)$, Neural network parameters θ initialization, T number of grad steps, "Proposal" distribution $q(x_t)$, Batch Size M, Number of MC samples L, TCSIS Type

Output : Neural network parameters θ

for $i \leftarrow 1$ **to** T **do**

  Sample batch $\{x_t^m\}_{m=1}^M \sim q(x_t)$;

  Get all the 1-Hamming distance sequences $\{y_t^{m,k,n}\}_{n=1,k=1}^{M,V,N}$ for $\{x_t^m\}_{m=1}^M$;

  Sample $\{\hat{x}_t^{m,k,n,l}\}_{m=1,k=1,n=1,l=1}^{M,V,N,L} \sim p_{t|0}(x_0|\{y_t^{m,k,n}\})$;

  $T^E \leftarrow E(\{\hat{x}_t^{m,k,n,l}\}_{m=1,k=1,n=1,l=1}^{M,V,N,L})$;

  $T^{\text{noise}} \leftarrow \log p_{t|0}(\{\hat{x}_t^{m,k,n,l}\}_{m=1,k=1,n=1,l=1}^{M,V,N,L}|\{y_t^{m,k,n}\})$;

  Compute marginal density estimation $\hat{\overline{p}}_t(x_t) = T^E + T^{\text{noise}}$;

  if *TCSIS type = Self-Normalized* **then**

    Compute concrete score estimation $\hat{s}_t(x)_y$ using Eq 5 and $\hat{\overline{p}}_t(x_t)$;

    Compute loss L_θ using Eq 6 and $\hat{s}_t(x)_y$;

  else

    Compute loss L_θ using Eq 9 and $\hat{\overline{p}}_t(x_t)$;;

  Optimize $\theta \leftarrow \theta - \nabla_\theta L_\theta$;

B.2 Self normalized TCSIS

One can show that objective 6 can be interpreted as self normalized importance weighted estimate,

$$s_t(x)_y = \frac{\mathbb{E}_{p_{t|0}(x_0|y)}[\overline{p}(x_0)]}{\mathbb{E}_{p_{t|0}(x_0|x)}[\overline{p}(x_0)]} = \frac{\sum_{x_0 \in S} p_{t|0}(x_0|y)\overline{p}(x_0)}{\sum_{x_0 \in S} p_{t|0}(x_0|x)\overline{p}(x_0)} = \tag{10}$$

$$\frac{\sum_{x \in S} \frac{p_{t|0}(x_0|y)}{p_{t|0}(x_0|x)} p_{t|0}(x_0|x)\overline{p}(x_0)}{\sum_{x_0 \in S} p_{t|0}(x_0|x)\overline{p}(x_0)} = \frac{\mathbb{E}_{p_{t|0}(x_0|x)}\left[\frac{p_{t|0}(x_0|y)}{p_{t|0}(x_0|x)}\overline{p}(x_0)\right]}{\mathbb{E}_{p_{t|0}(x_0|x)}[\overline{p}(x_0)]}, \tag{11}$$

which can be interpreted as expectation of $\frac{p_{t|0}(x_0|y)}{p_{t|0}(x_0|x)}$ under $p_{t|0}(x_0|x)$ weighted by $w_i \approx \frac{\overline{p}(x_0^i)}{\sum_i \overline{p}(x_0^i)}$.

C Algorithm

Here we present an algorithm for training both the Self Normalized TCSIS and Unbiased TCSIS, see Algorithm 1.

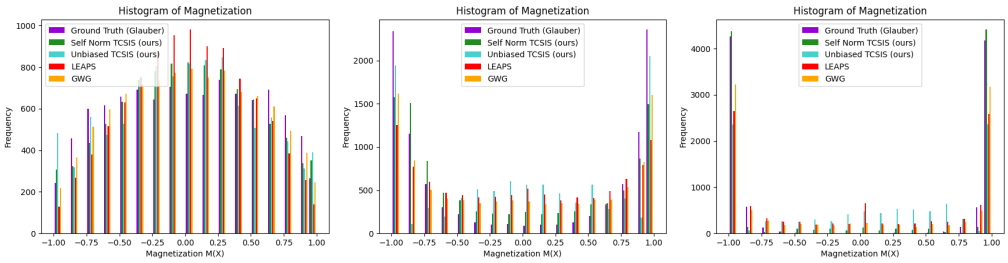
D Additional related works

D.1 Uniform discrete diffusion models

In uniform diffusion models the forward noising kernel is defined by factorized rate matrix $Q_t(x^i, y^i) = \sigma_t Q^{\text{uniform}}(x^i, y^i)$, where:

$$Q^{\text{uniform}} = \begin{bmatrix} 1-N & 1 & \cdots & 1 \\ 1 & 1-N & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1-N \end{bmatrix}$$

The marginal densities ODE, i.e., Eq 2, can be integrated analytically and transition probability $p_{t|0}(x_t|x_0)$ would take the form:



(a) High temperature setting (b) Critical temperature setting (c) Low temperature setting

Figure 3: Ising $L = 4$ lattice model with different inverse temperatures magnetization histograms.

$$p_{t|0}(x_t|x_0) = \prod_{i=0}^{N} p_{t|0}(x_t^i|x_0^i) = \prod_{i=0}^{N} \exp(\overline{\sigma}_t Q^{\text{uniform}}(x_t^i, x_0^i)), \quad Q^{\text{uniform}}(x, y) = \begin{cases} 1, y \neq x \\ 1 - N, y = x \end{cases} \tag{12}$$

where x_t^i is i-th element of x. With sufficiently large time T forward process converges to **uniform noise**.

D.2 Factorization and concrete score for 1-Hamming distance neighborhood.

In practice, concrete score is learned only for sequences within 1-Hamming neighborhood Section 3.3 [13], i.e., between sequences that do differ at most by one token. Modeling concrete score between general sequences $\mathcal{X} = \{1, ..., N\}^d$ in computationally expensive as number of states grow exponentially. In addition, modeling of concrete scores withing 1-Hamming neighborhood, i.e., factorized by sequence length, is sufficient for consistent modeling of CTMC processes [4].

Modeling of concrete scores only for sequences withing 1-Hamming neighborhood is equivalent to modeling factorized rate matrices $Q(x, y)$ or factorized probability distributions $p(x) = \prod_{i=0}^{d} p(x^i)$ [13, 4]. In that case factorized concrete score $s_t(x) : \mathcal{X}^d \times \mathbb{R} \to \mathbb{R}^{N \times d}$ would be represented by a *matrix*, rather then the full concrete score $s_t(x) : \mathcal{X}^d \times \mathbb{R} \to \mathbb{R}^{|\mathcal{X}^d|}$ that can be represented by the *vector of exponential size* $|\mathcal{X}^d|$.

E Experimental details

E.1 Magnetization

Here we present the magnetization histograms for our experiments. The magnetization is the sum of all coordinate values on a lattice computed as follows: $M(x) = \sum_i x_i$. The histograms of magnetizations for Ising models on $L = 4$ and $L = 10$ lattices are presented in Figures 4 and 4 correspondingly. Magnetization histograms for each method were calculated using 10000 samples.

E.2 Two point Correlation

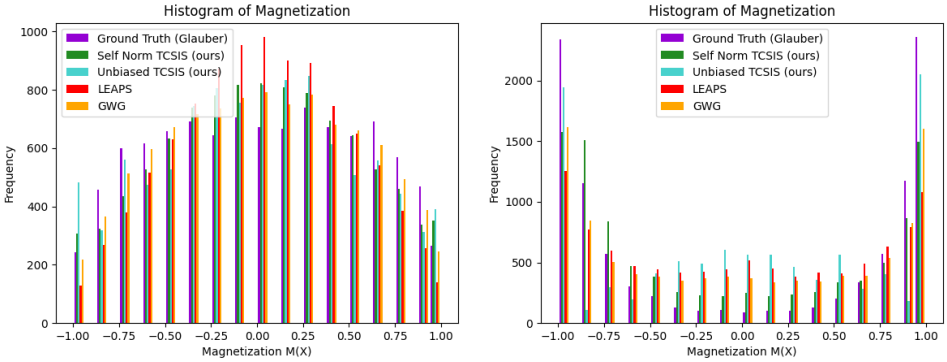
The two point correlation on lattice is defined as follows:

$$G(r) = \mathbb{E}[x_i x_{i+r}] - \mathbb{E}[x_i]\mathbb{E}[x_{i+r}],$$

where x_i and x_{i+r} are separated by distance of r on the lattice. The latter is a measure of the dependency between spin values per distance r lattice separation.

E.3 TCSIS hyperparameters

The main part of the hyperpameters is described in the Table 1. Separately, as a neural network we use diffusion transformer (DiT) [20] with 2-dimensional rotary embeddings [9] and for Unbiased



(a) High temperature setting

(b) Critical temperature setting

Figure 4: Ising $L = 10$ lattice model with different inverse temperatures magnetization histograms.

TCSIS parametrization we add a regression head on top. In all the cases neural network has fewer then 1M parameters. In "proposal" $q(x_t)$ column: Noise - random uniform noise, Model and Noise - a mixture of samples from model and random noise (with weight of 0.9 for model samples), GWG for samples from GWG [7] MCMC method with 1000 steps. For Model and GWG samples replay buffer was used. Precond means preconditioning that was used for the Unbiased TCSIS. Following [26] we model marginal densities as $\log p_t^\theta(x) = NN_\theta(x, t) + NN_\theta(t) * E(x)$, where $E(x)$ is the energy function of corresponding Ising model in our case. Learning of Unbiased TCSIS in $L = 10$ Ising model setting was carried out in *torch.double* because the loss values could induce overflow. We use a 0.999 EMA in all the TCSIS setups. Training of TCSIS took less then 6 hours on NVIDIA A100 GPU and inference took less than 5 minutes on NVIDIA A100 GPU.

Method	Lattice dim, L	MC N	"Proposal" $q(x_t)$	LR	Grad steps	Batch size	Precond
Self Norm TCSIS	4	500	Noise	$1e-5$	50k	64	✗
Unbiased TCSIS	4	100	GWG	$1e-4$	50k	64	✓
Self Norm TCSIS	10	2000	Model and Noise	$3e-6$	150k	64	✗
Unbiased TCSIS	10	500	GWG	$1e-4$	100k	64	✓

Table 1: Comparison of methods with their configurations.

E.4 LEAPS

LEAPS [10] official github implementation was taken:

`https://github.com/malbergo/leaps`

Locally-equivariant convolutional (LEC) network was taken as a neural network with kernel sizes $[3, 3, 3, 3]$ for $L = 4$ and $[3, 5, 7, 9]$ for $L = 10$, 20 channels were taken for each setup. MLP with $[200, 200, 200, 200]$ channels was taken as a free energy network. 20000 geometric annealing gradient steps were taken as a warmup. All the other hyperparameters were taken as default from the proposed by authors configuration. LEAPS was ran without correction during sampling. LEAPS training took less than 24 hours on NVIDIA A100 GPU and inference took less than 5 minutes on NVIDIA A100 GPU.

E.5 GWG

The implementation was taken from the DMALA [12] repository:

`https://github.com/ruqizhang/discrete-langevin`

We used a Taylor approximation for the difference function, disallowed multiple jumps per step, and ran the $L = 4$ setup with 24 transition steps and $L = 10$ setup with 64 transition steps, a randomized proposal, temperature set to 2, and step size of 0.05. Inference of the MCMC took less than 1 minute on NVIDIA A100 GPU.

NeurIPS Paper Checklist

[Yes] , [No] , or [NA]

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the Limitations section in the Appendix A.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide necessary proofs in the Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide all the necessary technical details in the Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We do not provide code at the moment, however we do provide all the algorithm descriptions in the main text and provide all the necessary hyperparameters in the Appendix E. Do do not use any data in our experiments, only energy functions.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all the necessary technical details in the Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following the research field practices we do not run experiments with several random seed and do not provide error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide all the necessary technical details in the Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There are no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Not applicable

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.