Dale's Law Meets Geometric Brownian Motion: Multiplicative Updates for Sampling

Anonymous Author(s)
Affiliation
Address
email

Abstract

Gradient descent has proven to be a powerful and effective technique for optimization in numerous machine learning applications. Recent advances in computational neuroscience have shown that learning in standard gradient descent optimization formulation is not consistent with learning in biological systems. This has opened up interesting avenues for building biologically inspired learning techniques. One such approach is inspired by Dale's law, which states that inhibitory and excitatory synapses do not swap roles during the course of learning. The resulting *exponential gradient descent* optimization scheme leads to log-normally distributed synaptic weights. Interestingly, the density that satisfies the Fokker-Planck equation corresponding to the stochastic differential equation (SDE) with geometric Brownian motion (GBM) is the log-normal density. Leveraging this connection, we start with the SDE governing geometric Brownian motion, and show that discretizing the corresponding reverse-time SDE yields a multiplicative update rule, which surprisingly, coincides with the sampling equivalent of the exponential gradient descent update founded on Dale's law. Proceeding further, we propose a new formalism for multiplicative denoising score-matching, which subsumes the loss function proposed by Hyvärinen for non-negative data. Indeed, log-normally distributed data is positive and the proposed score-matching formalism turns out to be a natural fit. This allows for training of score-based models for image data and results in a novel multiplicative update scheme for sample generation starting from a log-normal density. Experimental results on MNIST, Fashion MNIST, and Kuzushiji datasets demonstrate generative capability of the new scheme. To the best of our knowledge, this is the first instance of a biologically inspired generative model employing multiplicative updates, founded on geometric Brownian motion.

1 Introduction

An interesting problem in computational neuroscience is training artificial neural networks (ANNs) in a fashion that is consistent with learning and optimization seen in biological systems. Several studies [50, 36, 8, 38, 41] have confirmed that synaptic weight distributions in biological systems are log-normally distributed and that the neurons obey Dale's law [15], which states that excitatory (inhibitory) neurons stay excitatory (inhibitory) throughout the course of learning without synaptic flips. Artificial neural networks trained with gradient descent seldom obey Dale's law. Recently, Cornford et al. [12] proposed the use of exponentiated gradient descent (EGD) to train neural networks and have observed that the training is consistent with Dale's law and leads to log-normally distributed synaptic weights, in alignment with experimental findings. Exponentiated gradient descent is derived using mirror descent for a particular variant of Bregman divergence.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

In this paper, we establish a concrete link between exponentiated gradient descent optimization to sampling from stochastic differential equations (SDEs) inspired by geometric Brownian motion (GBM). Whereas most diffusion modeling and sampling schemes rely on standard Brownian motion, to the best of our knowledge, this is the first instance where GBM is used. We show that the proposed framework captures the multiplicative nature of updates seen in EGD. The ability of geometric Brownian motion to model processes with proportional changes makes it an ideal candidate for developing biologically inspired generative models. For the purpose of generation, we need the underlying score function used in the reverse-time SDE, for which we develop a novel multiplicative score-matching loss. While a large body of contemporary generative modeling literature is based on SDEs with additive Gaussian noise, our novel formalism relies on an SDE that governs the forward noising process dynamics with multiplicative log-normal noise. We develop the corresponding reverse-time SDE and show that it results in a multiplicative update rule that is structurally equivalent to the exponential gradient-descent scheme Cornford et al. [12]. The multiplicative update rule obtained as a consequence of the discretization of the SDE can be used to sample from the desired distribution whose score function is learnt using a neural network. We support the theoretical developments with experimental results on MNIST [31], Fashion MNIST [60] and Kuzushiji image datasets [10].

1.1 Related Works

Recent developments in generative modelling employing generative adversarial networks [18], diffusion models [21], score-based models [51, 52, 54], flow-based models [40] have produced stunning examples across a variety of modalities spanning images, video, audio, etc.. In the context of diffusion models, a seminal contribution has been the early work by Sohl-Dickstein et al. [47]. Inspired by non-equilibrium thermodynamics, they introduced the diffusion probabilistic model as a tractable and flexible model for sampling and inference. They demonstrated generative capability on toy datasets in two dimensions and image datasets like binarized MNIST and CIFAR-10. Ho et al. [21] demonstrated that denoising diffusion probabilistic models (DDPMs) could be used for high quality image synthesis. They vastly improved the results from Sohl-Dickstein et al. [47] and showed a performance comparable to state-of-the-art generative models [26, 27] of that time. Progress in score-matching by Song et al. [53], Song and Ermon [51, 52] demonstrated the potential of score-based generative models to be competitive with diffusion models. In the seminal work of Song et al. [55], it was shown that an SDE framework unifies both approaches. These SDEs were based on standard Brownian motion. Several alternative formulations that obviate the need for Brownian motion were also proposed. In particular, Bansal et al. [2] propose generative models that are based on more generic degradation operations and their corresponding restoration operations. They consider blurring and masking among others as degradation operators and show that such generalized degradations could also be used to formulate generative models. Rissanen et al. [43] proposed that generation could be viewed as the time-reversal of a heat equation. Additionally, they showed that their approach allows for certain image properties like shape and colour to be disentangled and they also discuss spectral properties that reveal inductive biases in generative models. Santos et al. [45] developed a discrete state-space diffusion model that relies on a pure-death random process and demonstrate competitive generative ability on binarized MNIST, CIFAR-10, and CelebA-64 datasets.

A recent preprint on image denoising by Vuong and Nguyen [57] is perhaps the closest to the multiplicative noise model considered in this paper. They consider a forward process where images are corrupted by multiplicative log-normal or gamma distributed noise. However, instead of proceeding with the multiplicative noise model, they convert it to an additive one by applying a logarithmic transformation. While the log-transformation simplifies the calculations, it reduces the problem to the additive noise setting, losing out completely on the richness of the original multiplicative noise framework. Vuong and Nguyen [57] remark explicitly that the reverse-time SDE in the multiplicative noise setting comes with a lot of complications, which are overcome by converting it to an additive noise model. They also restrict the scope of their work to denoising and do not propose a generative framework.

1.2 Organization of the paper

Section 2 gives an account of Dale's law and progress in computational neuroscience in deploying exponentiated gradient descent to enforce Dale's law – all of these form the inspiration for this work. In Section 3, we present the essential mathematics behind SDEs and generative modeling

required for understanding the contributions of this paper. Section 4 introduces Geometric Brownian Motion (GBM) and its corresponding reverse-time SDE based sampler for image generation. This necessitates a new score-matching framework for multiplicative noise which we define in Section 5. Finally, Section 6 presents experiments on MNIST, Fashion-MNIST, and Kuzushiji MNIST datasets, demonstrating the effectiveness and potential of the proposed model.

2 Dale's Law and Exponentiated Gradients

In computational neuroscience, Dale's law [15] has been empirically observed to hold in many biological systems barring certain exceptions. Dale's law states that presynaptic neurons can only *exclusively* affect their corresponding postsynaptic counterparts in an excitatory or inhibitory manner. The implication of the law is that the synapses continue to be inhibitory or excitatory during the course of learning without flipping. On the contrary, artificial neural networks have synaptic weights that can flip from excitatory to inhibitory or vice versa during training. Previous attempts [4, 58, 34] to incorporate biologically inspired learning rules to train neural networks have had limited success on standard benchmark tasks. Recently, Cornford et al. [11] demonstrated that ANNs that obey Dale's law, which they name Dale's ANNs (DANNs), can be constructed without loss in performance compared to weight updates done using standard gradient descent. They show that the ColumnEI models proposed by Song et al. [48] are suboptimal and can potentially impair the ability to learn by limiting the solution space of weights. DANNs outperform ColumnEI models on tasks across MNIST [31], Fashion-MNIST [60] and Kuzushiji MNIST datasets [10]. Cornford et al. [11] posit that the emergence and prevalence of Dale's law in biological systems is a possible evolutionary local minima and that the presence of inhibitory units in learning could help avoid catastrophic forgetting [3].

Li et al. [32] demonstrated that methods such as ColumnEI proposed by Song et al. [48] to incorporate Dale's law into the training of recurrent neural networks (RNNs) lead to suboptimal performance on sequence learning tasks, which is primarily attributed to poor spectral properties of the weight matrices, in particular, the multimodal, dispersed nature of the singular value spectrum of the weight matrix. Li et al. [32] extended the architecture developed by Cornford et al. [11] to handle sequences using RNNs and showed that these networks are on par with RNNs that are trained without incorporating Dale's law. The spectral properties of DANN RNNs are also better than the ColumnEI networks and the singular value spectrum is unimodal and clustered leading to superior performance on tasks such as the adding problem [22], sequential MNIST task [30] and language modelling using the Penn Tree Bank [37].

Cornford et al. [12] demonstrated that gradient descent is a suboptimal phenomenological fit to learning experiments in biologically relevant settings. While stochastic gradient descent for training ANNs is an exceptionally successful and robust model in general, it violates Dale's law [15] by allowing for synaptic flips. This leads to the distribution of weights not being log-normal, which contradicts experimentally observed data. Cornford et al. [12] showed that exponentiated gradient descent (EGD) introduced by Kivinen and Warmuth [29] respects Dale's law and consequently produces log-normally distributed weights. In experiments performed on the Mod-Cog framework [28] using RNNs, EGD outperforms gradient descent and is superior to GD for synaptic pruning. The learning task is formulated utilizing the mirror descent framework [39, 7] as changes to synaptic weights in a neural network such that a combination of task error and "synaptic change penalty" must be minimized. This leads to the update rule:

$$\boldsymbol{X}_{k+1} = \arg\min_{\boldsymbol{X}} \left[\bar{\ell}(\boldsymbol{X}) + \frac{1}{\eta} D_{\phi}(\boldsymbol{X}, \boldsymbol{X}_k) \right], \tag{1}$$

where $\bar{\ell}(\boldsymbol{X}) = \ell(\boldsymbol{X}_k) + \nabla \ell(\boldsymbol{X})^{\top} \mid_{\boldsymbol{X}=\boldsymbol{X}_k} (\boldsymbol{X} - \boldsymbol{X}_k)$ is the linearization of the task error $\ell(\boldsymbol{X})$ about the point $\boldsymbol{X}_k$ and $D_{\phi}(\boldsymbol{X}, \boldsymbol{X}_k)$ is the synaptic change penalty. The penalty $D_{\phi} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is chosen as the Bregman divergence corresponding to a strictly convex function $\phi : \mathbb{R}^d \to \mathbb{R}$. Depending on the choice of ϕ, we get different update rules. For instance, when $\phi(\boldsymbol{X}) = \|\boldsymbol{X}\|_2^2$, the corresponding synaptic change penalty is $D_{\phi}(\boldsymbol{X}, \boldsymbol{X}_k) = \|\boldsymbol{X} - \boldsymbol{X}_k\|_2^2$, and Eq. (1) results in the familiar gradient-descent update $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \eta \nabla \ell(\boldsymbol{X}) \mid_{\boldsymbol{X}=\boldsymbol{X}_k}$. This update rule for the weights does not guarantee that the entries of $\boldsymbol{X}_{k+1}$ and $\boldsymbol{X}_k$ have the same sign, which allows for synaptic flips during training, as also confirmed by Cornford et al. [12].

Cornford et al. [12] chose $\phi(\boldsymbol{X}) = \sum_{i=1}^{d} |X^{(i)}| \log |X^{(i)}|$, where $X^{(i)}$ denotes the ith entry of $\boldsymbol{X}$, which results in D_ϕ being the unnormalised relative entropy,

$$D_\phi(\boldsymbol{X}, \boldsymbol{X}_k) = \sum_{i=1}^{d} X^{(i)} \log \frac{X^{(i)}}{X_k^{(i)}} - X^{(i)} + X_k^{(i)}.$$

For this choice of D_ϕ, the update rule in Eq. (1) takes the form

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k \circ \exp\left(-\eta \nabla_{\boldsymbol{X}} \ell(\boldsymbol{X}) \mid_{\boldsymbol{X}=\boldsymbol{X}_k} \circ \operatorname{sign}(\boldsymbol{X}_k)\right), \tag{2}$$

where $\circ$ denotes element-wise multiplication. The update in Eq. 2 is different from standard gradient-descent update in many ways: the update is multiplicative as opposed to additive, involves exponentiation, and preserves the sign of the entries of $\boldsymbol{X}_k$ as iterations proceed. Effectively, the entries in $\boldsymbol{X}_k$ for any k have the same sign as those in $\boldsymbol{X}_0$. The update rule in Eq. (2) is referred to as *exponentiated gradient descent (EGD)* [29].

By design, EGD doesn't allow synaptic flips and automatically respects Dale's law during the course of training. The update rule also leads to the weights being distributed log-normally as demonstrated by Pogodin et al. [41]. Exponentiated gradient-descent has been shown to perform on par with gradient descent for models trained on Mod-Cog tasks, although the final weight distributions are different. The networks for both updates are initialized with log-normal weights to adhere to experimental data that shows that the synaptic strengths of neurons in the brain are log-normally distributed [14, 35]. The network trained with gradient descent had a final weight distribution that was different from log-normal whereas the network trained with exponentiated gradient was log-normally distributed. Additionally, Cornford et al. [12] have shown that learning with EGD is more robust to synaptic weight pruning and EGD outperforms gradient descent when relevant inputs are sparse and in particular, for continuous control tasks. Pogodin et al. [41] showed that the distribution of converged weights depends on the geometry induced by the choice of the update algorithm. Gradient-descent updates implicitly assume Euclidean geometry, which is inconsistent with the log-normal weight distribution that is experimentally observed and is ill-suited to data arising in neuroscience.

A quick glance at Eq. (2) prompts the question: **Does there exist a sampling equivalent for the exponentiated gradient-descent update rule?** This is inspired by the link between gradient-descent and Langevin dynamics as enunciated by Wibisono [59]. In pursuit of an answer to this question, we realised the connection between the log-normally distributed weights observed at the end of exponentiated gradient descent and the sampling equation lies in geometric Brownian motion. The equilibrium distribution of GBM is the log-normal density and its time-reversal would give us the sampling formula we seek (discussed in Sec. 4).

3 Stochastic Differential Equations and Generative Modelling

Recent generative models such as diffusion models [21, 49] and score-based models rely heavily on the SDE framework. These models have been immensely successful in generating realistic samples across different data modalities such as images [55] and audio [42]. The key idea is to construct a stochastic process such that one starts with samples from the true, unknown density and progressively transforms them to samples from a noisy, easy-to-sample-from density such as the isotropic Gaussian. The task of generation requires inverting the forward process which goes beyond mere time reversal due to the stochastic nature of the dynamics. Theoretical results [1, 9, 55] show that there exists a corresponding reverse-time SDE for the forward process. The forward process is represented as

$$\mathrm{d}\boldsymbol{X}_t = h(\boldsymbol{X}_t, t)\,\mathrm{d}t + g(\boldsymbol{X}_t, t)\,\mathrm{d}\boldsymbol{W}_t, \tag{3}$$

where $h(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is the *drift* function, $g(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is the *diffusion* function, and $\boldsymbol{W}_t$ denotes the standard Wiener process. We follow the Itô interpretation of SDEs throughout this paper. The corresponding reverse-time SDE for Eq. (3) is given by

$$\begin{aligned} \mathrm{d}\boldsymbol{X}_t \quad = \quad & \left(h(\boldsymbol{X}_t, t) - \nabla \cdot [g(\boldsymbol{X}_t, t) g(\boldsymbol{X}_t, t)^\top] - g(\boldsymbol{X}_t, t) g(\boldsymbol{X}_t, t)^\top \nabla \log f_{\boldsymbol{X}}(\boldsymbol{X}_t, t)\right) \mathrm{d}t \\ & + g(\boldsymbol{X}_t, t) \mathrm{d}\bar{\boldsymbol{W}}_t, \end{aligned} \tag{4}$$

where $\mathrm{d}\bar{\boldsymbol{W}}_t$ is the standard Brownian motion and $\nabla \cdot F(\boldsymbol{x}) := (\nabla \cdot f^1(\boldsymbol{x}),\ \nabla \cdot f^2(\boldsymbol{x}), \cdots,\ \nabla \cdot f^d(\boldsymbol{x}))^\top$ is the row-wise divergence of the matrix-valued function $F(\boldsymbol{x}) := (f^1(\boldsymbol{x}), f^2(\boldsymbol{x}), \cdots, f^d(\boldsymbol{x}))^\top \in \mathbb{R}^{d\times d}$. The issue with generating new samples from Eq. (4) is that we usually do not have access to the score function $\nabla \log f_{\boldsymbol{X}}(\boldsymbol{X}_t, t)$ and this quantity is approximated using a neural network $s_{\boldsymbol{\theta}} : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, which is trained by optimizing the denoising score-matching loss [55]

$$\mathcal{L}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}[0,1]}\left[\mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0}\\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}}\left[\lambda(t)\left\|s_{\boldsymbol{\theta}}(\boldsymbol{X}_t,t) - \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0)\right\|_2^2\right]\right], \tag{5}$$

where $\nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0)$ is determined by the forward SDE (Eq. (3) [46]) and $\lambda(t)$ is designed to stabilise training.

4 Geometric Brownian Motion

Brownian motion, originally introduced to model random particle motion [16], is widely used in physics, biology, and signal processing to describe processes with independent and identically distributed (i.i.d.) increments. The resulting distribution is Gaussian following the Central Limit Theorem. For example, the Ornstein-Uhlenbeck SDE (OU-SDE) [13] models the position Y_t of a Brownian particle as $\mathrm{d}Y_t = \mu\,\mathrm{d}t + \sigma\,\mathrm{d}W_t$, where W_t is a Wiener process, yielding $Y_t = Y_0 + \mu t + \sigma W_t$, a Gaussian process with mean μ and variance σ^2. Alternatively, when the relative increments (or ratios) follow the Brownian motion, the resulting stochastic process is called the Geometric Brownian Motion (GBM). Black and Scholes [6] pioneered the use of GBM for modeling the evolution of stock prices and financial assets in mathematical finance. Just as the normal distribution plays a crucial role in Brownian motion, the log-normal distribution plays a vital role in the analysis of GBM. Formally, a random process X_t is said to follow a Geometric Brownian Motion if it satisfies the SDE:

$$\mathrm{d}X_t = \mu X_t\,\mathrm{d}t + \sigma X_t\,\mathrm{d}W_t, \tag{6}$$

where W_t is the Wiener process, and μ and σ are known as the *percentage drift* representing a general trend and *volatility coefficients* representing the inherent stochasticity, respectively. The solution of Eq. (6) X_t evolves to follow a log-normal distribution with parameters μ and σ^2, i.e.,

$$X_t = X_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W_t\right).$$

There exist several multivariate extensions of GBM [23]. We consider the element-wise extension of Eq. (6) for image data with the forward SDE for time $t \in [0,1]$:

$$\mathrm{d}\boldsymbol{X}_t = \boldsymbol{\mu} \circ \boldsymbol{X}_t\,\mathrm{d}t + \sigma \boldsymbol{X}_t \circ \mathrm{d}\boldsymbol{W}_t, \tag{7}$$

where $\circ$ denotes element-wise multiplication, $\boldsymbol{\mu} \in \mathbb{R}^d$, $\sigma > 0$ and $\boldsymbol{W}_t$ denotes the multivariate Wiener process. This can be written equivalently, using Itô's lemma, as

$$\mathrm{d}\log \boldsymbol{X}_t = \left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\mathbf{1}\right)\mathrm{d}t + \mathrm{d}\boldsymbol{W}_t, \tag{8}$$

where $\log$ is applied element-wise. The distribution of $\boldsymbol{X}_t$, as it evolves according to Eq. (8), has i.i.d. entries that are log-normally distributed with parameters $\boldsymbol{\mu}$ and $\sigma^2\mathbb{I}$, $\mathbb{I}$ being the $d \times d$ identity matrix. Starting from a sample $\boldsymbol{X}_0$ from the unknown density $p_{\boldsymbol{X}_0}$, the solution to Eq. (8) is

$$\boldsymbol{X}_t = \boldsymbol{X}_0 \circ \exp\left(\left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\mathbf{1}\right)t + \sigma \boldsymbol{W}_t\right).$$

This closed-form expression allows us to easily generate samples from the forward process at arbitrary time instants $t \in [0,1]$. The samples at the end of the forward process are log-normally distributed. We now seek to derive the corresponding reverse-time SDE that would enable us to generate samples from the unknown density $p_{\boldsymbol{X}_0}$ starting from samples from the log-normal density. While one could use Eq. (4) to derive the corresponding reverse-time SDE, we propose a simpler approach by defining an auxiliary stochastic process $\boldsymbol{Y}_t = \log \boldsymbol{X}_t$ and leveraging *score change-of-variables formula* [44]. This allows us to rewrite Eq. (8) as

$$\mathrm{d}\boldsymbol{Y}_t = \left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\mathbf{1}\right)\mathrm{d}t + \mathrm{d}\boldsymbol{W}_t. \tag{9}$$

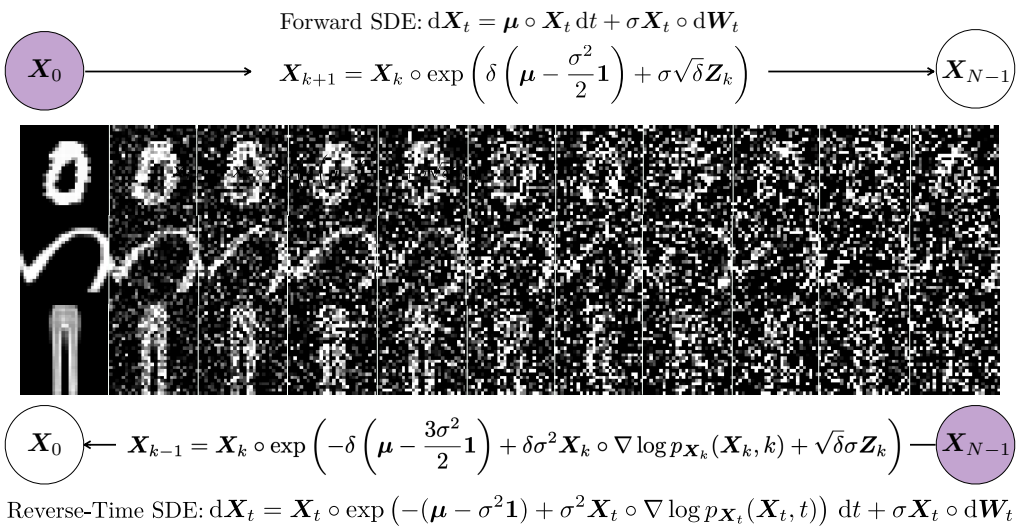

Figure 1: The forward and reverse-time SDEs for Geometric Brownian Motion (GBM). The forward SDE describes the evolution of a clean image sample to a noisy one that eventually becomes log-normally distributed, while the reverse-time SDE captures the dynamics of the process and generates new samples from the unknown density starting from log-normal noise. This is enabled by the knowledge of the unknown density manifesting through the score function.

The reverse-time SDE corresponding to the forward SDE in Eq. (9) can be obtained by invoking Eq. (4) and is given by

$$\mathrm{d}\boldsymbol{Y}_t = \left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\mathbf{1} - \sigma^2 \nabla \log p_{\boldsymbol{Y}_t}(\boldsymbol{Y}_t, t)\right)\,\mathrm{d}t + \sigma \mathrm{d}\boldsymbol{W}_t, \tag{10}$$

where $\nabla \log p_Y(\boldsymbol{Y}_t, t)$ is the score function corresponding to $\boldsymbol{Y}_t$ and $\mathbf{1}$ is a vector of all ones. We invoke the score change-of-variables formula [44] that allows us to represent $\nabla \log p_{\boldsymbol{Y}_t}(\boldsymbol{Y}_t, t)$ in terms of $\nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)$ as $\nabla \log p_{\boldsymbol{Y}_t}(\boldsymbol{Y}_t, t) = \mathbf{1} + \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)$. Thus, we rewrite Eq. (10) in terms of $\boldsymbol{X}_t$ and simplify it to obtain

$$\mathrm{d}\log \boldsymbol{X}_t = \left(\boldsymbol{\mu} - \frac{3\sigma^2}{2}\mathbf{1} - \sigma^2 \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)\right)\,\mathrm{d}t + \sigma \mathrm{d}\boldsymbol{W}_t. \tag{11}$$

To simulate the reverse-time SDE on a computer, it must be discretized in time. We chose the time range $[0, 1]$ with N steps, which results in a step-size of $\delta = \frac{1}{N}$ and for brevity, denote $\boldsymbol{X}_{k\delta}$ as $\boldsymbol{X}_k$, for $k = 0, \ldots, N-1$. In particular, we choose the Euler-Maruyama discretization scheme [20] for Eq. (11) to get

$$\log \boldsymbol{X}_{k-1} = \log \boldsymbol{X}_k - \delta\left(\boldsymbol{\mu} - \frac{3\sigma^2}{2}\mathbf{1} - \sigma^2\left(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)\right)\Big|_{t=k\delta}\right) + \sqrt{\delta}\sigma\boldsymbol{Z}_k, \tag{12}$$

where $\boldsymbol{Z}_k \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ (the standard normal distribution), and since the $\log$ operates element-wise, exponentiating both sides gives

$$\boldsymbol{X}_{k-1} = \boldsymbol{X}_k \circ \exp\left(-\delta\left(\boldsymbol{\mu} - \frac{3\sigma^2}{2}\mathbf{1}\right) + \delta\sigma^2\boldsymbol{X}_k \circ \nabla \log p_{\boldsymbol{X}_k}(\boldsymbol{X}_k, k) + \sqrt{\delta}\sigma\boldsymbol{Z}_k\right). \tag{13}$$

The update rule in Eq. (13) is similar to the EGD update rule in Eq. (2). Consider the optimization problem with a modification of the task error as

$$\boldsymbol{X}_{t+1} = \arg\min_{\boldsymbol{X}}\left[\bar{\ell}(\xi(\boldsymbol{X})) + \frac{1}{\eta}D_\phi(\boldsymbol{X}, \boldsymbol{X}_t)\right], \tag{14}$$

with the choice of $\xi : \mathbb{R}^d \to \mathbb{R}^d$ as $\xi^{(i)}(\boldsymbol{X}) = 0.5\left(X^{(i)}\right)^2$ for $i = 1, 2, \cdots, d$. This leads to the following multiplicative update rule

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k \circ \exp\left(-\eta \boldsymbol{X}_k \circ \nabla_{\boldsymbol{X}} \ell(\boldsymbol{X}) \mid_{\boldsymbol{X}=\boldsymbol{X}_k}\right). \tag{15}$$

Interestingly, if we assume that the density $p_{\boldsymbol{X}_k}(\boldsymbol{X}_k, k)$ is of the form $p_{\boldsymbol{X}_k}(\boldsymbol{X}_k, k) = \frac{1}{Z}\exp\left(-\ell(\boldsymbol{X}_k)\right)$, with $\eta = \delta\sigma^2$ and $\boldsymbol{\mu} = \frac{3\sigma^2}{2}$, then the corresponding sampling step in Eq. (13) is of the form

$$\boldsymbol{X}_{k-1} = \boldsymbol{X}_k \circ \exp\left(-\eta \boldsymbol{X}_k \circ \nabla_{\boldsymbol{X}} \ell(\boldsymbol{X}) \mid_{\boldsymbol{X}=\boldsymbol{X}_k} + \sqrt{\eta}\boldsymbol{Z}_k\right), \tag{16}$$

where $\boldsymbol{Z}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I})$. Therefore, the proposed sampler is structurally equivalent to the modified exponential gradient descent step in Eq. (15).

5 Multiplicative Score Matching

Following the definitions of explicit score-matching (ESM) loss and denoising score-matching (DSM) loss for the additive noise case [56], we propose the multiplicative counterparts $\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta})$ and $\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta})$ as follows:

$$\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2}\|\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t) - \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\|_2^2\right], \quad \text{and} \tag{17}$$

$$\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0 \sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t \sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[\frac{1}{2}\|\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0) - \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\|_2^2\right]. \tag{18}$$

The two types of score-matching loss functions are related as follows.

Theorem 5.1 (Multiplicative Denoising Score-Matching)**.** *Under standard assumptions on the density and the score function [24, 53] over the positive orthant $\mathbb{R}^{+d}$, the multiplicative explicit score-matching (M-ESM) loss given in Eq.* (17) *and multiplicative denoising score-matching (M-DSM) loss given in Eq.* (18) *are equivalent up to a constant, i.e., $\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) + C$, where C is independent of $\boldsymbol{\theta}$.*

The proof is provided in the supplementary material. The usefulness of this result is explained next. We need the marginal score function $\nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t)$ in the reverse-time SDE Eq. (13) but optimizing Eq. (17) is intractable since we do not have access to the "true" marginal score. The theorem provides us with a means to optimize for $s_{\boldsymbol{\theta}}$ in terms of the conditional score $\nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0)$, which can be derived from the forward SDE. The challenge in leveraging Eq. (13) to generate new samples arises from our lack of knowledge of $\nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t)$. This function must be estimated by some form of score-matching. To this end, we propose the following score-matching loss

$$\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0 \sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t \sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[\frac{1}{2}\|\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t \mid \boldsymbol{X}_0) - \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\|_2^2\right]. \tag{19}$$

In practice, this choice of the loss function allows us to train the score network $s_{\boldsymbol{\theta}}$ using samples from the forward SDE in Eq. (7) and the corresponding conditional score $\nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0)$ evaluated at discrete instants of time $t = k\delta$ can be computed using the forward SDE and the expression for the target in the loss function is given by

$$\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0) = -\left(\boldsymbol{1} + \frac{1}{\sigma^2 t\delta}\left(\log \boldsymbol{X}_k - \log \boldsymbol{X}_0 - t\delta\left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\boldsymbol{1}\right)\right)\right). \tag{20}$$

The proposed loss function in Eq. (19) is the multiplicative noise counterpart of the denoising score-matching loss proposed by Song et al. [55] for additive noise. To the best of our knowledge, this formulation of the score-matching loss and its manifestation in the multiplicative noise setting is new. It would be appropriate to remark here that the score term in Eq. (13) also arises in the score-matching loss proposed by Hyvärinen [25] for non-negative real data given by

$$\mathcal{L}_{\text{NN}}(\boldsymbol{\theta}) = \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{X}_0 \sim p_{\boldsymbol{X}_0}} \left[\|\boldsymbol{X}_0 \circ \nabla \log p_{\boldsymbol{X}_0}(\boldsymbol{X}_0) - \boldsymbol{X}_0 \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_0)\|_2^2\right], \tag{21}$$

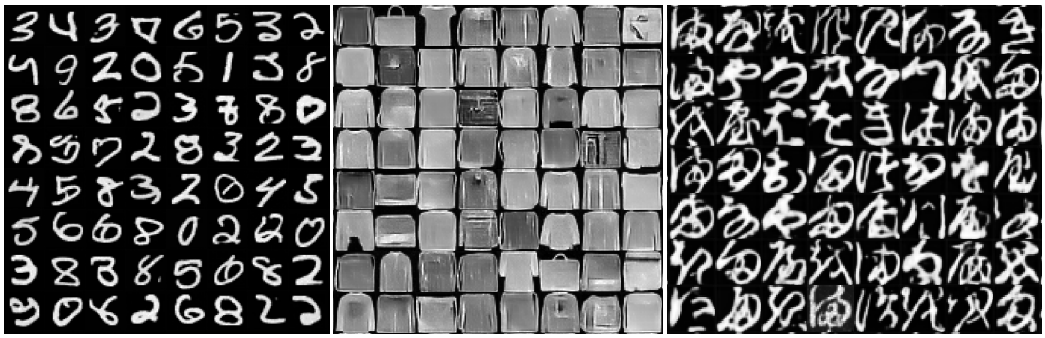

Figure 2: Uncurated sample images generated from MNIST, Fashion-MNIST and Kuzushiji MNIST datasets, corresponding to the score model with minimum score-matching loss during training.

where $\nabla \log p_{\boldsymbol{X}_0}(\boldsymbol{X}_0)$ is the true score. Hyvärinen [25]'s formulation is static in the sense that it does not leverage the SDE, whereas we do. Hyvärinen [25]'s score-matching loss can also be seen as an instance of the multiplicative explicit score-matching loss (M-ESM) for $t = 0$. Hyvärinen [25]'s motivation for introducing this loss function is to avoid the singularity at the origin for non-negative data. Our framework encapsulates this variant of the score-matching loss as a special case. This is primarily due to the structure of GBM that assumes the log-normal distribution which implicitly restricts the samples to be positive. Thus, our framework generalizes the score-matching loss proposed by Hyvärinen [25] to the case of multiplicative noise.

Algorithm 1 Multiplicative updates for generation using Geometric Brownian Motion (GBM).

Require: $\sigma, \delta, \boldsymbol{\mu}$, trained score network $s_{\boldsymbol{\theta}}$

for $k \leftarrow 0$ to $N-1$ **do**

  $\boldsymbol{Z}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I})$

  $\boldsymbol{X}_{k-1} = \boldsymbol{X}_k \circ \exp\left(-\delta\left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\boldsymbol{1}\right) + \delta\sigma^2 \boldsymbol{X}_k \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_k, k) + \sigma\sqrt{\delta}\boldsymbol{Z}_k\right)$

end for

5.1 Image Generation using Multiplicative Score Matching

The goal in diffusion-based image generative modeling is to construct two stochastic processes, as illustrated in Fig. 1 – the forward process to generate a noisy version of a clean image and the reverse process to enable us to sample from the unknown density $p_{\boldsymbol{X}_0}$. For the forward model, starting from an image $\boldsymbol{X}_0$ coming from the unknown density, the forward SDE in Eq. (8) can be used to generate noisy versions of $\boldsymbol{X}_0$ as follows

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k \circ \exp\left(\delta\left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\boldsymbol{1}\right) + \sqrt{\delta}\sigma\boldsymbol{Z}_k\right), \tag{22}$$

for $k = 0, \ldots, N-2$, and $\boldsymbol{X}_{N-1}$ is log-normally distributed and $\boldsymbol{Z}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I})$. For the reverse process, i.e., generation, we can generate samples from the reverse-time SDE in Eq. (11) using the discretized version of the reverse-time SDE in Eq. (13) and the score model $s_{\boldsymbol{\theta}}(\cdot)$ trained with the loss defined in Eq. (19) in place of the true score function $\nabla \log p_{\boldsymbol{X}_t}(\cdot)$. The new generation/sampling procedure is summarized in Algorithm 1. The algorithm takes as input the parameters $\sigma, \delta, \boldsymbol{\mu}$ and the trained score network $s_{\boldsymbol{\theta}}$ and generates samples from the unknown density $p_{\boldsymbol{X}_0}$ by iterating over N steps. The algorithm starts with a sample $\boldsymbol{X}_{N-1}$ from the log-normal distribution and iteratively updates the sample using the reverse-time SDE in Eq. (13). The final output should be a sample $\boldsymbol{X}_0 \sim p_{\boldsymbol{X}_0}$.

6 Experiments

We evaluate the generative performance of the proposed model[1] by training the score model on standard datasets such as MNIST, Fashion-MNIST and Kuzushiji MNIST dataset used by Cornford et al. [11]. The datasets are split as $60,000$ images for training and $10,000$ images for testing. All images are rescaled to have pixel values in the range [1, 2]. Note that the proposed framework requires a non-negative dynamic range of pixel values. We choose $N = 1000$ discretization levels for the forward SDE (7) and leads to the step size $\delta = 1/N$. During sampling, we observed that the same step-size did not always work and we had to work with smaller step-sizes for each of the three datasets. The model is trained using the M-DSM loss defined in Eq. (19). The hyperparameters $\boldsymbol{\mu} = \frac{\sigma^2}{2}\mathbf{1}$, σ and δ are set to 0.8 and 0.001, respectively. The model is trained for 200000 iterations and the checkpoints are saved every 5000 iterations as mentioned in [52] on two NVIDIA RTX 4090 and two A6000 GPUs. We perform exponential moving average for the saved checkpoints every 50000 iterations. The generated samples are shown in Figure 2, from where we observe that the visual quality of the generated images matches is on par with that of the ground truth. For quantitative assessment, we use Fréchet Inception Distance (FID) [19] and Kernel Inception Distance (KID) [5] measured between $10,000$ images from the test dataset and the same number of generated images. Lower FID and KID values indicate superior generative performance. While both these metrics are not commonly used to quantify the generative performance for grayscale images, we follow Xu et al. [61] and report these numbers for transparency and reproducibility (cf. Supplementary Material).

7 Conclusions

We proposed a novel generative model based on Geometric Brownian Motion (GBM) and a new technique for score-matching. We showed that the GBM framework is a natural setting for modeling non-negative data and that the new multiplicative score-matching loss can be used effectively to train the model. The model is capable of generating new samples from image datasets like MNIST, Fashion MNIST and Kuzushiji MNIST. The results are promising from a generative modeling perspective. The multiplicative score matching framework can also be suitably adapted for image denoising and restoration tasks where the forward model has multiplicative noise as opposed to the widely assumed additive noise. While this work focuses on log-normal noise, other distributions such as the gamma distribution, could also be considered with associated SDEs. This would broaden the applicability of the model to datasets and domains where various types of multiplicative noise are prevalent such as optical coherence tomography [33] and synthetic aperture radar [17], enabling more robust and versatile generative and restoration capabilities. Starting off with the results shown in the paper, one could also extend applicability of the proposed model to high-resolution images. Application to non-image data, such as financial time-series, is another potential direction for further research.

Limitations

The proposed generative model requires a large amount of training data and computational resources to achieve good performance, which can be a constraint in some applications. In the true spirit of data-driven generation, some of the generated images do not have the same *semantic* meaning as samples from the source dataset. Incorporating semantics into generative modeling is a research direction by itself. Instead of cherry-picking the results, we reported them as obtained to highlight both the strengths and limitations of the proposed approach. The choice of hyperparameters, such as the noise schedule and learning rate, which are carefully tuned, can affect the performance of the model. However, this limitation is true of all deep generative models and not unique to ours.

Broader Impact: The proposed approach of leveraging the GBM and multiplicative score-matching is novel and has the potential to advance the field of generative modeling along new lines. The model may find natural applicability in financial time-series modeling, forecasting, and generation. Ethical concerns pertaining to the use of generative models and the potential for misuse by generating biased, fake, or misleading content are all pervasive and the proposed framework is no exception.

[1] Code for this paper is available at `https://anonymous.4open.science/r/gbm_dale-CC20`

References

[1] B. D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. ISSN 0304-4149. doi: https://doi.org/10.1016/0304-4149(82)90051-5. URL https://www.sciencedirect.com/science/article/pii/0304414982900515.

[2] A. Bansal, E. Borgnia, H.-M. Chu, J. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 41259–41282. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/80fe51a7d8d0c73ff7439c2a2554ed53-Paper-Conference.pdf.

[3] H. C. Barron, T. P. Vogels, T. E. Behrens, and M. Ramaswami. Inhibitory engrams in perception and memory. *Proceedings of the National Academy of Sciences*, 114(26):6666–6674, 2017. doi: 10.1073/pnas.1701812114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1701812114.

[4] S. Bartunov, A. Santoro, B. A. Richards, L. Marris, G. E. Hinton, and T. P. Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9390–9400, Red Hook, NY, USA, 2018. Curran Associates Inc.

[5] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

[6] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973. ISSN 00223808, 1537534X. URL http://www.jstor.org/stable/1831029.

[7] S. Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3–4):231–357, Nov. 2015. ISSN 1935-8237. doi: 10.1561/2200000050. URL https://doi.org/10.1561/2200000050.

[8] G. Buzsáki and K. Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264–278, 2014. doi: 10.1038/nrn3687. URL https://doi.org/10.1038/nrn3687.

[9] D. Castanon. Reverse-time diffusion processes (corresp.). *IEEE Transactions on Information Theory*, 28(6):953–956, 1982. doi: 10.1109/TIT.1982.1056571.

[10] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical japanese literature, 2018. URL https://nips2018creativity.github.io/doc/deep_learning_for_classical_japanese_literature.pdf. Presented at the Machine Learning for Creativity and Design Workshop, NeurIPS 2018, Montreal, Canada.

[11] J. Cornford, D. Kalajdzievski, M. Leite, A. Lamarquette, D. M. Kullmann, and B. A. Richards. Learning to live with dale's principle: {ANN}s with separate excitatory and inhibitory units. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=eU776ZYxEpz.

[12] J. Cornford, R. Pogodin, A. Ghosh, K. Sheng, B. A. Bicknell, O. Codol, B. A. Clark, G. Lajoie, and B. A. Richards. Brain-like learning with exponentiated gradients. *bioRxiv*, 2024. doi: 10.1101/2024.10.25.620272. URL https://www.biorxiv.org/content/early/2024/10/26/2024.10.25.620272.

[13] J. L. Doob. The brownian movement and stochastic equations. *Annals of Mathematics*, 43(2):351–369, 1942. ISSN 0003486X, 19398980. URL http://www.jstor.org/stable/1968873.

[14] S. Dorkenwald, N. L. Turner, T. Macrina, K. Lee, R. Lu, J. Wu, A. L. Bodor, A. A. Bleckert, D. Brittain, N. Kemnitz, W. M. Silversmith, D. Ih, J. Zung, A. Zlateski, I. Tartavull, S.-C. Yu, S. Popovych, W. Wong, M. Castro, C. S. Jordan, A. M. Wilson, E. Froudarakis, J. Buchanan, M. M. Takeno, R. Torres, G. Mahalingam, F. Collman, C. M. Schneider-Mizell, D. J. Bumbarger, Y. Li, L. Becker, S. Suckow, J. Reimer, A. S. Tolias, N. Macarico da Costa, R. C. Reid, and H. S. Seung. Binary and analog variation of synapses between cortical pyramidal neurons. *eLife*, 11:e76120, nov 2022. ISSN 2050-084X. doi: 10.7554/eLife.76120. URL https://doi.org/10.7554/eLife.76120.

[15] J. C. Eccles, P. Fatt, and K. Koketsu. Cholinergic and inhibitory synapses in a pathway from motor-axon collaterals to motoneurones. *The Journal of Physiology*, 126(3):524–562, 1954. doi: https://doi.org/10.1113/jphysiol.1954.sp005226. URL https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1954.sp005226.

[16] R. Feynman, R. Leighton, M. Sands, and E. Hafner. *The Feynman Lectures on Physics; Vol. I*, volume 33. AAPT, 1965.

[17] G. Fracastoro, E. Magli, G. Poggi, G. Scarpa, D. Valsesia, and L. Verdoliva. Deep learning methods for synthetic aperture radar image despeckling: An overview of trends and perspectives. *IEEE Geoscience and Remote Sensing Magazine*, 9(2):29–51, 2021. doi: 10.1109/MGRS.2021.3070956.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. 2014.

[19] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

[20] D. J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001. doi: 10.1137/S0036144500378302. URL https://doi.org/10.1137/S0036144500378302.

[21] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

[22] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[23] Y. Hu. Multi-dimensional geometric brownian motions, onsager-machlup functions, and applications to mathematical finance. *Acta Mathematica Scientia*, 20(3):341–358, 2000. ISSN 0252-9602. doi: https://doi.org/10.1016/S0252-9602(17)30641-0. URL https://www.sciencedirect.com/science/article/pii/S0252960217306410.

[24] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24), 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

[25] A. Hyvärinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5):2499–2512, 2007. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2006.09.003. URL https://www.sciencedirect.com/science/article/pii/S0167947306003264.

[26] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

[27] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12104–12114. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8d30aa96e72440759f74bd2306c1fa3d-Paper.pdf.

[28] M. Khona, S. Chandra, J. J. Ma, and I. R. Fiete. Winning the lottery with neural connectivity constraints: Faster learning across cognitive tasks with spatially constrained sparse rnns. *Neural Computation*, 35(11):1850–1869, 2023. doi: 10.1162/neco_a_01613.

[29] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.1996.2612. URL `https://www.sciencedirect.com/science/article/pii/S0890540196926127`.

[30] Q. V. Le, N. Jaitly, and G. E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *CoRR*, abs/1504.00941, 2015. URL `http://arxiv.org/abs/1504.00941`.

[31] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

[32] P. Li, J. Cornford, A. Ghosh, and B. Richards. Learning better with dale's law: A spectral perspective. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 944–956. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/02dd0db10c40092de3d9ec2508d12f60-Paper-Conference.pdf`.

[33] S. Li, R. Higashita, H. Fu, B. Yang, and J. Liu. Score prior guided iterative solver for speckles removal in optical coherent tomography images. *IEEE Journal of Biomedical and Health Informatics*, 29(1):248–258, 2025. doi: 10.1109/JBHI.2024.3480928.

[34] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020. doi: 10.1038/s41583-020-0277-3. URL `https://doi.org/10.1038/s41583-020-0277-3`.

[35] Y. Loewenstein, A. Kuras, and S. Rumpel. Multiplicative dynamics underlie the emergence of the log-normal distribution of spine sizes in the neocortex in vivo. *Journal of Neuroscience*, 31(26):9481–9488, 2011. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.6130-10.2011. URL `https://www.jneurosci.org/content/31/26/9481`.

[36] Y. Loewenstein, A. Kuras, and S. Rumpel. Multiplicative dynamics underlie the emergence of the log-normal distribution of spine sizes in the neocortex in vivo. *The Journal of Neuroscience*, 31(26):9481–9488, June 2011. doi: 10.1523/JNEUROSCI.6130-10.2011. URL `https://www.jneurosci.org/content/31/26/9481`.

[37] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL `https://aclanthology.org/J93-2004/`.

[38] J. B. Melander, A. Nayebi, B. C. Jongbloets, D. A. Fortin, M. Qin, S. Ganguli, T. Mao, and H. Zhong. Distinct in vivo dynamics of excitatory synapses onto cortical pyramidal neurons and parvalbumin-positive interneurons. *Cell Reports*, 37(6):109972, 2021. ISSN 2211-1247. doi: https://doi.org/10.1016/j.celrep.2021.109972. URL `https://www.sciencedirect.com/science/article/pii/S2211124721014510`.

[39] A. S. Nemirovsky and D. B. Yudin. Problem complexity and method efficiency in optimization. *SIAM Review*, 27(2):264–265, 1985. doi: 10.1137/1027074. URL `https://doi.org/10.1137/1027074`.

[40] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL `http://jmlr.org/papers/v22/19-1028.html`.

[41] R. Pogodin, J. Cornford, A. Ghosh, G. Gidel, G. Lajoie, and B. A. Richards. Synaptic weight distributions depend on the geometry of plasticity. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=x5txICnnjC`.

[42] J. Richter, D. De Oliveira, and T. Gerkmann. Investigating training objectives for generative speech enhancement. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10887784.

[43] S. Rissanen, M. Heinonen, and A. Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol.

[44] S. Robbins. Score change of variables, 2024. URL https://arxiv.org/abs/2412.07904.

[45] J. E. Santos, Z. R. Fox, N. Lubbers, and Y. T. Lin. Blackout diffusion: generative diffusion models in discrete-state spaces. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[46] S. Särkkä and A. Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

[47] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proc. of the 32nd Intl. Conf. on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[48] H. F. Song, G. R. Yang, and X.-J. Wang. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: A simple and flexible framework. *PLoS Comput. Biol.*, 12(2):e1004792, Feb. 2016.

[49] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[50] S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biology*, 3(3):e68, 2005. doi: 10.1371/journal.pbio.0030068. URL https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.0030068.

[51] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

[52] Y. Song and S. Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

[53] Y. Song, S. Garg, J. Shi, and S. Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI*, 2019. URL http://auai.org/uai2019/proceedings/papers/204.pdf.

[54] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[55] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[56] P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7), 2011. doi: 10.1162/NECO_a_00142.

[57] A. Vuong and T. Nguyen. Perception-based multiplicative noise removal using sdes, 2024. URL https://arxiv.org/abs/2408.10283.

[58] J. C. R. Whittington and R. Bogacz. Theories of error Back-Propagation in the brain. *Trends Cogn Sci*, 23(3):235–250, Jan. 2019.

[59] A. Wibisono. Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem. In *Proceedings of the 31st Conference On Learning Theory*, 2018. URL `https://proceedings.mlr.press/v75/wibisono18a.html`.

[60] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[61] C. Xu, X. Cheng, and Y. Xie. Normalizing flow neural networks by jko scheme. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 47379–47405. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/93fce71def4e3cf418918805455d436f-Paper-Conference.pdf`.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes. Our contributions are clearly stated in the abstract and introduction in Sec. 1. The claims made in the paper are supported by theoretical results and experimental evaluations in Sec. 4 and Sec. 6.

   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: Yes. Please refer to page 9 for a discussion of the limitations of the proposed model.

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: Yes, please refer to the supplementary material for the proof of Theorem 5.1.

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: Yes. We use publicly-accessible datasets such as MNIST [31], Fashion MNIST [60] and Kuzushiji MNIST [10]. The code is available at `https://anonymous.4open.science/r/gbm_dale-CC20` and includes instructions to reproduce the results.

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: [Yes]

Justification: Yes. The code is available at `https://anonymous.4open.science/r/gbm_dale-CC20` and includes instructions to reproduce the results. The datasets used in the experiments are publicly available.

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: Yes, please refer to Sec. 6 for the details of the experimental setup.

   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: We cannot afford to report error bars for experiments since they are computationally extremely expensive.

   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: Please refer to the supplementary material.

   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: Yes. We adhere to 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: [Yes]

Justification: Yes. Please see page 9.

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: [Yes]

Justification: The code is publicly available and is subject to the discretion of the user, also see page 9.

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: [Yes]

Justification: All datasets, code and models used are cited.

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: [Yes]

Justification: Yes. The link to code to reproduce our experimental observations has been provided in the footnote on page 9.

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: No human subjects were involved.

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: No human subjects were involved.

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: LLM has been used only for minor editing and formatting purposes.

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.