Dale meets Langevin: A Multiplicative Denoising Diffusion Model

Anonymous authors
Paper under double-blind review

Abstract

Exponentiated gradient descent (EGD), a biologically motivated optimization algorithm that respects Dale's law, results in log-normally distributed synaptic weights, in alignment with experimental observations in neuroscience. Since the marginal distribution of geometric Brownian motion (GBM) at any fixed time is log-normal, there is a natural connection between EGD and GBM-based stochastic processes. We propose a multiplicative score-based generative model with GBM as a forward noising process and derive its corresponding reverse-time SDE in both the ambient space and in the log-transformed space. We derive two multiplicative samplers by discretizing the corresponding reverse-time SDEs: a *sign-agnostic* sampler obtained directly from the ambient-space reverse-time SDE, and a *sign-preserving* sampler, which we refer to as the *Dale-Langevin sampler*, obtained via the Lamperti transform. We further connect the framework to Mirrored Langevin Dynamics, showing that the convex function driving EGD in optimization precisely governs the Dale-Langevin sampler. The *Stein score*, defined as $\nabla \log p_{\boldsymbol{X}}(\boldsymbol{x})$ for a random vector $\boldsymbol{X}$ with density $p_{\boldsymbol{X}}$ evaluated at $\boldsymbol{x}$, comes up naturally in the additive noise based diffusion models. In the multiplicative setting, we encounter $\boldsymbol{x} \circ \nabla \log p_{\boldsymbol{X}}(\boldsymbol{x})$, a modulated version of the Stein score for sampling, which we name the *Hyvärinen score*. In order to estimate the Hyvärinen score, we introduce the multiplicative denoising score-matching loss (M-DSM), the multiplicative explicit score-matching loss (M-ESM), and establish their equivalence. This development subsumes the non-negative score matching loss of Hyvärinen (2007) as a special case. Experimental results on MNIST, Fashion-MNIST, Kuzushiji MNIST, and CIFAR-10 validate the generative capability of the proposed framework.

1 Introduction

Several studies in computational neuroscience (Song et al., 2005; Loewenstein et al., 2011; Buzsáki & Mizuseki, 2014; Melander et al., 2021; Pogodin et al., 2024) have confirmed that synaptic weight distributions in biological neural networks are log-normal, and that neurons obey Dale's law (Eccles et al., 1954), which states that a neuron is either excitatory or inhibitory, and does not flip between the two states during learning. Standard gradient descent does not guarantee sign preservation. Recent efforts to define mathematical structures that respect Dale's law for weight optimization have been fruitful. Cornford et al. (2024) and Nishida et al. (2026) have tackled similar problems in this space; Cornford et al. (2024) showed that exponentiated gradient descent (EGD) applied to neural network training respects Dale's law and produces log-normally distributed weights, while Nishida et al. (2026) utilized log-normal multiplicative dynamics to stabilize low-precision training. EGD is the canonical instance of mirror descent (Bubeck, 2015) with negative entropy as the mirror map, and its natural domain is $\mathbb{R}^d_+$ rather than $\mathbb{R}^d$. We build upon this biological motivation, extending from discriminative neural network optimization to generative modelling via Geometric Brownian Motion.

Sampling from probability density functions and optimization are deeply intertwined and can be viewed as two sides of the same coin. It has been shown that *Langevin dynamics* is gradient flow in the space of probability measures under the Wasserstein-2 metric, and the corresponding discrete-time update matches with *gradient descent* on the Kullback-Leibler (KL) divergence (Wibisono, 2018). This raises the question:

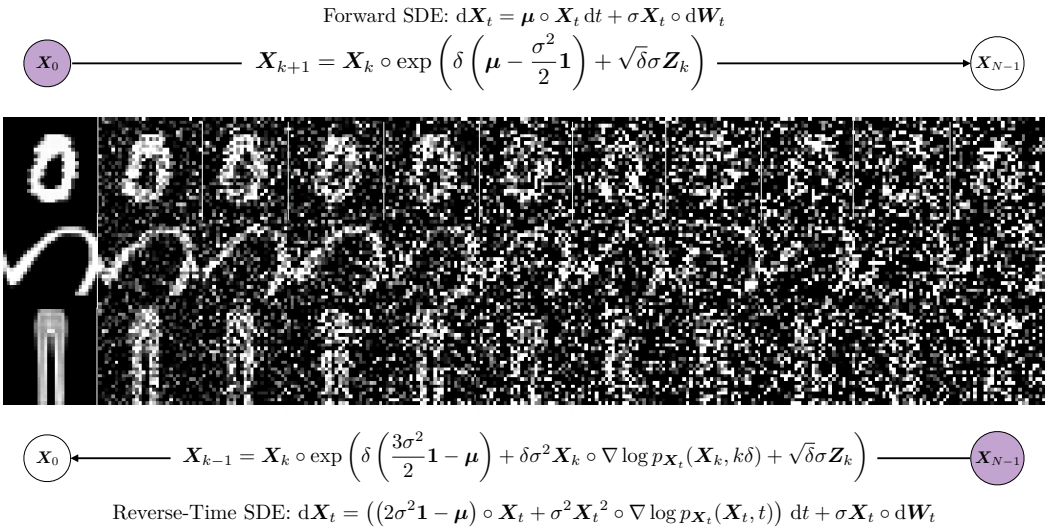


Figure 1: The forward and reverse-time SDEs for Geometric Brownian Motion (GBM). The forward SDE describes the evolution of a clean image sample to a noisy one that eventually becomes log-normally distributed, while the reverse-time SDE generates new samples from the unknown density starting from log-normal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

What is the sampling analogue of EGD, in the same way that Langevin dynamics can be viewed as the sampling analogue of gradient descent? In this paper, we answer that question, and in doing so, introduce a multiplicative generative model and a corresponding multiplicative score-matching loss, establish the link with EGD through Mirrored Langevin dynamics (MLD) (Hsieh et al., 2018) and experimentally demonstrate image generation within the proposed framework.

We start with geometric Brownian motion (GBM), a diffusion process that preserves positivity, and has a marginal distribution that is log-normal (Øksendal, 1985). We derive the reverse-time stochastic differential equation (SDE) in the ambient space and the log-transformed space and derive two multiplicative samplers: a *sign-agnostic* sampler obtained directly from the ambient-space reverse-time SDE, and a *sign-preserving* sampler, which we refer to as the *Dale-Langevin sampler*, obtained via the Lamperti transform and an appropriate change of variables. The Dale-Langevin sampler preserves the sign of every coordinate across all iterations by scaling samples in each update by a score-dependent factor. This is the sampling analogue of EGD, in the same sense that Langevin dynamics is the sampling analogue of gradient descent. Additionally, we show that the convex function driving EGD in optimization precisely matches the one used in the MLD framework to derive the Dale-Langevin sampler.

In standard additive noise based diffusion models (Ho et al., 2020; Song et al., 2021b), the *Stein score* comes up naturally during sampling. In the multiplicative setting, we require a modified version of the Stein score, which we refer to as the *Hyvärinen score*. In order to estimate the Hyvärinen score, we propose multiplicative explicit score matching loss and denoising score matching loss, and prove their equivalence. This is analogous to the equivalence between denoising score-matching loss and the associated explicit score matching loss in the additive setting (Vincent, 2011).

The score-matching loss proposed by Hyvärinen (2007) for non-negative data becomes a special case of the proposed M-ESM corresponding to time-step $t = 0$. We validate the resulting algorithm on MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), Kuzushiji MNIST (Clanuwat et al., 2018), and CIFAR-10 (Krizhevsky et al., 2009). Fig. 1 provides a schematic of the generation and noising process.

1.1 Related Works

Recent advances in generative modelling with score-based generative models (Ho et al., 2020; Song & Ermon, 2019; 2020; Song et al., 2021b) have yielded impressive results across modalities such as images, video, and

audio. The dynamical systems based diffusion probabilistic model of Sohl-Dickstein et al. (2015), inspired by non-equilibrium thermodynamics, provided the foundational framework for these advances. Ho et al. (2020) were the first to demonstrate the generative power of denoising diffusion probabilistic models (DDPMs) by achieving high-quality image synthesis competitive with the then state-of-the-art generative models (Karras et al., 2018; 2020a). Progress in score-matching (Song et al., 2019; Song & Ermon, 2019; 2020; Dhariwal & Nichol, 2021) showed score-based models to outperform Generative Adversarial Networks (GANs) (Goodfellow et al., 2014; Karras et al., 2019; 2020b; 2021). Song et al. (2021c) unified the discrete-time conditional density framework of Ho et al. (2020) with the SDE framework built on standard Brownian motion. Several works have since explored departures from Brownian motion. Bansal et al. (2023) formulate generative models around generic degradation operators such as blurring, masking, etc. and their corresponding restoration maps. Rissanen et al. (2023) view generation as the time-reversal of a heat equation, identify disentanglement of shape and colour, and analyze spectral inductive biases. Santos et al. (2023) develop a discrete state-space diffusion via a pure-death random process.

The Gaussian noise assumption, while analytically convenient, is not compatible with data carrying structural constraints such as positivity, boundedness, or discreteness. Nachmani et al. (2021) factor the positivity constraint by replacing the additive Gaussian noise with additive Gamma noise. Zhou et al. (2023) propose Beta Diffusion for interval-valued data. To handle count data, Chen & Zhou (2023) build a Poisson-process framework. Deviating from the Euclidean paradigm, Fishman et al. (2023) provide a systematic construction of diffusion models on Riemannian manifolds. Vuong & Nguyen (2024) also consider the GBM SDE and solve a multiplicative denoising problem but only by converting it to an additive one using the log transform. The transformation is not optimal from a denoising perspective. They leverage the corresponding reverse-time SDE for sequential denoising but do not explore the potential for image generation.

We explain why the log transform is appropriate, and additionally close the generative loop with reverse-time SDEs and multiplicative samplers for unconditional generation. More recently, Gruhlke et al. (2026) independently proposed multiplicative diffusion motivated by skew-symmetric multiplicative noise working with Stratonovich interpretation of the SDE. Their framework is motivated by physical principles of conservative forward-backward dynamics to preserve the distribution of norm of the data and show that their framework is better suited for heavy-tailed and anisotropic distributions. They train their score model with standard sliced score-matching, which is inherently limited to lower dimensions. Their experiments are focussed on fluid dynamics and not image generation. In contrast, we consider the GBM SDE, where the marginal density of the forward process at any time instant is log-normal. Our score-matching formalism is new and scales beyond two dimensions, as evidenced by results on image generation. Score matching was originally proposed as a technique for estimation of parameters of model densities (Hyvärinen, 2005). More recently, for estimation of graphical models with densities from the exponential family, Yu et al. (2019) generalize Hyvärinen's score-matching loss for non-negative data. Liu et al. (2022) study score matching for densities defined on truncated domains. They do not address the question of generative modelling.

1.2 Main Contributions

We introduce a multiplicative diffusion framework built on GBM and derive reverse-time SDEs both in the ambient space (without the log-transform) and in the log-transformed space resulting in corresponding multiplicative samplers (Section 3). We encounter a variant of Stein score in the multiplicative setting, which we refer to as the Hyvärinen score. We propose multiplicative explicit score matching loss and denoising score matching loss and prove that they are equivalent (Theorem 4.1), thus, generalizing the score-matching loss proposed by Vincent (2011) in the additive noise setting, to a multiplicative one. We also show that the non-negative score-matching loss (Hyvärinen, 2007) emerges as a special case of our formalism (Section 4). We further show that the convex function driving exponential gradient descent in optimization is identical to that resulting in the Dale-Langevin sampler within the Mirrored Langevin Dynamics (Hsieh et al., 2018) framework (Section 5). Finally, we present experimental validation on MNIST, Fashion-MNIST, Kuzushiji MNIST, and CIFAR-10 datasets (Section 6). The GBM forward process, the reverse-time SDEs, the multiplicative samplers, an appropriately defined score-matching loss to train neural networks, demonstration on image generation, together constitute a cogent generative framework within the multiplicative scenario.

1.3 Notation

Random variables are denoted in uppercase (e.g., X, Y), and random vectors in boldface uppercase (e.g., $\boldsymbol{X}, \boldsymbol{Y}$). Their realizations are denoted by the corresponding lowercase letters (e.g., $x, \boldsymbol{x}$). The probability density function (p.d.f.) of a random variable X is written as $p_X(x)$; for a random vector $\boldsymbol{X}$, it is written as $p_{\boldsymbol{X}}(\boldsymbol{x})$. The conditional p.d.f. of $\boldsymbol{X}_t$ given $\boldsymbol{X}_0$ is written $p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$. The domain is $\mathbb{R}^d_+$, the set of all d-dimensional vectors with positive real-valued entries, for all random vectors throughout the paper. The Stein score of the random vector $\boldsymbol{X}$ evaluated at $\boldsymbol{x}$ is $\nabla \log p_{\boldsymbol{X}}(\boldsymbol{x})$. The notation $s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)$ denotes a neural network approximation of the score at time t. The symbol $\circ$ denotes element-wise (Hadamard) multiplication. $\mathbf{1}$ denotes a vector of all ones, and $\mathbb{I}$ denotes the identity matrix of appropriate dimension. $\mathbb{R}^d$ denotes the d-dimensional real Euclidean space and $\|\cdot\|_2$ denotes the Euclidean (ℓ_2) norm. We write $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbb{I})$ for a Gaussian random vector with mean $\boldsymbol{\mu}$ and isotropic covariance $\sigma^2\mathbb{I}$, and $\boldsymbol{X} \sim \mathcal{LN}(\boldsymbol{\mu}, \sigma^2\mathbb{I})$ for the corresponding log-normal random vector. A quick recap of the log-normal distribution and its properties is provided in Section A of the appendix. We write $t \sim \mathcal{U}[0,1]$ for a random variable uniformly distributed over $[0,1]$. $\mathbb{E}$ denotes the expectation operator. $\boldsymbol{W}_t$ denotes a standard Wiener process (Brownian motion). Differentials are written as $\mathrm{d}\boldsymbol{X}_t$ and $\mathrm{d}\boldsymbol{W}_t$. We follow the Itō interpretation of SDEs throughout (Øksendal, 1985). Convergence in distribution of a sequence of random vectors $\boldsymbol{X}_n$ to $\boldsymbol{X}$ is written $\boldsymbol{X}_n \xrightarrow{\mathrm{d}} \boldsymbol{X}$. In the discrete-time samplers, $\delta > 0$ denotes the step-size, k indexes the time-step, $\boldsymbol{X}_k$ denotes the iterate at step k, and $\boldsymbol{Z}_k \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ is an independent noise vector drawn at each step. The annealing coefficient at step k is κ, decayed geometrically by factor $\chi \in (0,1]$ at each outer iteration; L denotes the number of Langevin inner steps per noise level. The parameters $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\sigma > 0$ are the drift and diffusion coefficients, respectively, of the underlying GBM. The Bregman divergence of a strictly convex function $h : \mathbb{R}^d \to \mathbb{R}$ is denoted by $D_h(\boldsymbol{X}, \boldsymbol{X}_k)$. The notation $\nabla_{\boldsymbol{X}} \ell(\boldsymbol{X})\Big|_{\boldsymbol{X}=\boldsymbol{X}_k}$ denotes the gradient of ℓ evaluated at $\boldsymbol{X}_k$.

2 Stochastic Differential Equations and Generative Modelling

Diffusion models (Ho et al., 2020; Song et al., 2021a) and score-based models (Song et al., 2021b) rely on an SDE framework with a forward SDE that transforms data into noise and a corresponding reverse-time SDE that transforms noise into data. These models have been successful in generating realistic samples across different data modalities such as images (Song et al., 2021c), video (Ho et al., 2022) and audio (Richter et al., 2025). 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. Generating new samples requires inversion of the forward process. Because the dynamics are stochastic, inversion goes beyond simple time-reversal and requires matching the marginal densities of the forward and reverse processes at every time instant t. While standard formulations use additive Gaussian noise with an isotropic Gaussian terminal density, the framework admits any forward SDE for which the terminal density is tractable. In the present formulation, the marginal density is log-normal. Anderson (1982) and Castanon (1982) independently showed that, for a forward SDE of the form

$$\mathrm{d}\boldsymbol{X}_t = h(\boldsymbol{X}_t, t)\,\mathrm{d}t + g(\boldsymbol{X}_t, t)\,\mathrm{d}\boldsymbol{W}_t, \tag{1}$$

there exists a corresponding reverse-time SDE, which has immediate relevance for generation. We adopt the multivariate version of the reverse-time SDE formula given by Song et al. (2021b)

$$\mathrm{d}\boldsymbol{X}_t = -\left(h(\boldsymbol{X}_t, t) - \nabla \cdot [g(\boldsymbol{X}_t, t) g(\boldsymbol{X}_t, t)^\top] - g(\boldsymbol{X}_t, t) g(\boldsymbol{X}_t, t)^\top \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)\right) \mathrm{d}t + g(\boldsymbol{X}_t, t)\mathrm{d}\bar{\boldsymbol{W}}_t, \tag{2}$$

where $\bar{\boldsymbol{W}}_t$ is the reverse-time 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 $F(\boldsymbol{x}) := (f^1(\boldsymbol{x}), f^2(\boldsymbol{x}), \cdots, f^d(\boldsymbol{x}))^\top \in \mathbb{R}^{d\times d}$. To generate new samples from Eq. (2), we must have access to the time-dependent score function $\nabla \log p_{\boldsymbol{X}}(\boldsymbol{X}_t, t)$, which, in practice, is approximated by a neural network, $s_{\boldsymbol{\theta}} : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, trained to minimize the denoising score-matching loss (Song et al., 2021c)

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

where $\nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0)$ is determined by the forward SDE (Särkkä & Solin, 2019) and $\lambda(t)$ is designed to stabilise training.

3 Geometric Brownian Motion and its Time-reversal as a Generative Model

Geometric Brownian Motion (GBM) models scenarios where *relative* increments of a stochastic process, rather than the absolute ones, follow a Brownian motion. This is in contrast to standard additive processes such as the Ornstein-Uhlenbeck SDE (Doob, 1942), $\mathrm{d}Y_t = \mu\,\mathrm{d}t + \sigma\,\mathrm{d}W_t$, whose solution $Y_t = Y_0 + \mu t + \sigma W_t$ is Gaussian with mean μ and variance σ^2. GBM was pioneered by Black & Scholes (1973) for modelling stock prices, where proportional rather than absolute changes are of interest. The marginal distribution is log-normal rather than Gaussian, making it a natural SDE of choice for positive-valued data. Formally, a random process X_t is said to follow a GBM if it satisfies the SDE:

$$\mathrm{d}X_t = \mu X_t\,\mathrm{d}t + \sigma X_t\,\mathrm{d}W_t, \tag{3}$$

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 time evolution of X_t follows 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 (Hu, 2000). We consider the element-wise extension of Eq. (3) for image data with the forward SDE for time $t \in [0, 1]$:

$$\mathrm{d}\boldsymbol{X}_t = \boldsymbol{\mu} \circ \boldsymbol{X}_t\,\mathrm{d}t + \sigma \boldsymbol{X}_t \circ \mathrm{d}\boldsymbol{W}_t, \tag{4}$$

where $\circ$ denotes element-wise multiplication, $\boldsymbol{\mu} \in \mathbb{R}^d$ is the drift term, $\sigma > 0$, $\sigma\,\mathrm{diag}(\boldsymbol{X}_t)$ is the diffusion matrix, and $\boldsymbol{W}_t$ denotes the multivariate Wiener process.

The distribution of $\boldsymbol{X}_t$ given $\boldsymbol{X}_0$, as it evolves according to Eq. (4), has i.i.d. entries that are log-normally distributed. Starting from $\boldsymbol{X}_0$ from the unknown density $p_{\boldsymbol{X}_0}$, the solution to Eq. (4) is

$$\boldsymbol{X}_t = \boldsymbol{X}_0 \circ \exp\left(\left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\mathbf{1}\right)t + \sigma \boldsymbol{W}_t\right). \tag{5}$$

The closed-form expression allows us to generate samples from the forward process for any time instant $t \in [0, 1]$. The samples at the end of the forward process are log-normally distributed. We derive a corresponding reverse-time SDE that would enable us to generate samples from the unknown density $p_{\boldsymbol{X}_0}$ starting from samples following the log-normal density. For the forward SDE in Eq. (4), we invoke Eq. (2) to get the following reverse-time SDE

$$\mathrm{d}\boldsymbol{X}_t = \left(\left(2\sigma^2\mathbf{1} - \boldsymbol{\mu}\right) \circ \boldsymbol{X}_t + \sigma^2 {\boldsymbol{X}_t}^2 \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)\right)\,\mathrm{d}t + \sigma \boldsymbol{X}_t \circ \mathrm{d}\boldsymbol{W}_t, \tag{6}$$

where $\nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)$ is the score function corresponding to $\boldsymbol{X}_t$ and $\mathbf{1}$ is a vector of all ones.

3.1 Lamperti Transform for GBM

The forward SDE in Eq. (4) has a state-dependent diffusion term $\sigma\mathbf{X}_t$, which makes the derivation of the reverse-time SDE cumbersome. We show that the state-dependent diffusion is precisely what gives rise to the multiplicative structure of the sampler. The Lamperti transform (Møller & Madsen, 2010), here, the element-wise log map, removes the state-dependence by converting the GBM SDE into an SDE with constant diffusion, to which the standard reverse-time formula in Eq. (2) and the score change-of-variables formula (Robbins, 2024) can be applied. The forward SDE in Eq. (4) can be written equivalently, using the Lamperti transform $\boldsymbol{Y}_t = \log \boldsymbol{X}_t$ invoking Itô's lemma, as follows:

$$\mathrm{d}\log \boldsymbol{X}_t = \left(\boldsymbol{\mu} - \frac{\sigma^2}{2}\mathbf{1}\right)\,\mathrm{d}t + \sigma \mathrm{d}\boldsymbol{W}_t, \tag{7}$$

where log is applied element-wise. The distribution of $\boldsymbol{X}_t$, as it evolves according to Eq. (7), 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 (identical to the solution in Eq. (5)). We can rewrite Eq. (7) as

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

The reverse-time SDE corresponding to the forward SDE in Eq. (8) is given by (cf. Eq. (2))

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

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 leverage the score change-of-variables formula (Robbins, 2024) 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. (9) in terms of $\boldsymbol{X}_t$ as

$$\mathrm{d}\log \boldsymbol{X}_t = -\left(\boldsymbol{\mu} - \frac{3\sigma^2}{2}\mathbf{1} - \sigma^2 \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)\right)\mathrm{d}t + \sigma \mathrm{d}\boldsymbol{W}_t. \tag{10}$$

Eq. (10) is an SDE in the log space with a state-independent diffusion coefficient σ.

3.2 Discretization of the Reverse-Time SDE

The two reverse-time SDEs derived above, Eq. (6), directly in the ambient space, and Eq. (10) via the Lamperti transform in the log-space, yield two distinct samplers upon Euler-Maruyama discretization (Higham, 2001). Both turn out to be multiplicative, yet they differ in a fundamental way: the sampler from Eq. (6) is *sign-agnostic*, i.e., the sign of the entries in $\boldsymbol{X}_t$ may not be preserved over updates, whereas the sampler from Eq. (10) is *sign-preserving*, i.e., the sign of the entries in $\boldsymbol{X}_t$ are preserved over time. The sign-preserving update is reminiscent of Dale's law. The interval $[0, 1]$ is discretized into N steps with step-size $\delta = \frac{1}{N}$. Let $\boldsymbol{X}_{k\delta}$ be denoted as $\boldsymbol{X}_k$ for $k = 1, 2, \ldots, N-1$.

The discretization of Eq. (6) resulting in the **sign-agnostic multiplicative sampler** is given by

$$\boxed{\boldsymbol{X}_{k-1} = \boldsymbol{X}_k \circ \left((1 + 2\delta\sigma^2)\mathbf{1} - \delta\boldsymbol{\mu} + \delta\sigma^2 \boldsymbol{X}_k \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_k, k\delta) + \sqrt{\delta}\sigma \boldsymbol{Z}_k\right),} \tag{11}$$

where $\boldsymbol{Z}_k \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ and $\nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_k, k\delta)$ is the score function evaluated at $\boldsymbol{X}_k$ and $t = k\delta$. The update is multiplicative, $\boldsymbol{X}_{k-1}$ is obtained from $\boldsymbol{X}_k$ after scaling by a data-dependent factor, and because that factor may be negative, the sign of the entries of $\boldsymbol{X}_k$ may not be preserved across updates.

We refer to the discretization of Eq. (10) in the log-space and its point-wise exponentiation as the **Dale-Langevin sampler**:

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

Here, the scaling factor is an exponential, which is strictly positive, and therefore, the sign of every entry of $\boldsymbol{X}_k$ is preserved across updates. This update is also structurally equivalent to exponentiated gradient descent, as we shall make precise in Section 5. In both cases, $\nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)$ must be estimated from the data and Section 4 describes the multiplicative score-matching objective used to train the score network.

4 Multiplicative Score Matching

We identify $\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t)$ as the natural score term in the multiplicative noise setting based on the structure of Eq. (11) and Eq. (12). We define the multiplicative explicit score-matching (M-ESM) and multiplicative denoising score-matching (M-DSM) counterparts of the explicit score matching (ESM) and

denoising score matching (DSM) losses for the additive noise case (Vincent, 2011) to train a neural network $s_{\boldsymbol{\theta}}$, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ (the parameter space), to approximate the score function as follows:

$$\boxed{\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2} \| \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t) - \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t) \|_2^2 \right],} \tag{13}$$

$$\boxed{\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0 \sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t \sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[\frac{1}{2} \| \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0) - \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t) \|_2^2 \right].} \tag{14}$$

Generating samples from the marginal density $p_{\boldsymbol{X}_t}$ is intractable, which makes $\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta})$ impractical to work with directly. We formalize the relationship between $\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta})$ and $\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta})$ in Theorem 4.1. This allows us to train score models with $\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta})$. Before proceeding further with the key result, we state below the assumptions on the density and the data and model score functions over the positive orthant $\mathbb{R}_d^+$:

- **Assumption 1** (Regularity of score functions). The p.d.f. $p_{\boldsymbol{X}_t}$ is differentiable, the expectations $\mathbb{E}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\| \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t) \|_2^2 \right]$ and $\mathbb{E}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\| \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t) \|_2^2 \right]$ are finite for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $t \in [0, 1]$.
- **Assumption 2** (Boundary conditions). The quantity $p_{\boldsymbol{X}_t}(\boldsymbol{X}_t)(\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t))$ vanishes for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $t \in [0, 1]$ as $\| \boldsymbol{X}_t \| \to \infty$.

Theorem 4.1 (Multiplicative Denoising Score-Matching)**.** *Under the assumptions of regularity and appropriate boundary conditions stated above, the M-ESM loss given in Eq. (13) and the M-DSM loss given in Eq. (14) are equivalent up to a constant, i.e., $\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta}) + C$, where C is independent of $\boldsymbol{\theta}$.*

The proof is provided in Section B of the appendix. Since $\mathcal{L}_{\text{M-ESM}}$ is intractable, the true marginal score $\nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t)$ is unknown. The theorem provides a means to optimize $s_{\boldsymbol{\theta}}$ via the conditional score $\nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0)$, which is tractable from the forward SDE. Evaluating at discrete time $t = k\delta$ gives the training target:

$$\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0)\Big|_{t=k\delta} = -\left(\mathbf{1} + \frac{1}{\sigma^2 k\delta} \left(\log \boldsymbol{X}_k - \log \boldsymbol{X}_0 - k\delta \left(\boldsymbol{\mu} - \frac{\sigma^2}{2} \mathbf{1} \right) \right) \right). \tag{15}$$

This is the multiplicative counterpart of the denoising score-matching target proposed by Song et al. (2021c) for additive noise. The score-matching loss proposed by Hyvärinen (2007) for non-negative data ($\mathcal{L}_{\text{NN}}(\boldsymbol{\theta})$) emerges as a special case of $\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta})$ at $t = 0$:

$$\mathcal{L}_{\text{NN}}(\boldsymbol{\theta}) = \frac{1}{2} \mathop{\mathbb{E}}_{\boldsymbol{X}_0 \sim p_{\boldsymbol{X}_0}} \left[\| \boldsymbol{X}_0 \circ \nabla \log p_{\boldsymbol{X}_0}(\boldsymbol{X}_0) - \boldsymbol{X}_0 \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_0) \|_2^2 \right]. \tag{16}$$

The formulation of Hyvärinen (2007) is static as there is no temporal dependency. It was primarily motivated by a need to avoid the singularity at the origin for non-negative data in the context of parameter estimation for model densities. Our framework based on the reverse-time SDE discretization shows that the multiplicative score term pops up naturally. The log-normal structure of GBM implicitly restricts samples to the positive orthant, and the loss in Hyvärinen (2007) is recovered by evaluating Eq. (13) at $t = 0$. The M-DSM framework is thus a generalization of the result of Hyvärinen (2007) to the multiplicative setting.

5 Dale's Law, Exponentiated Gradient Descent, and Mirrored Langevin Dynamics

The Dale-Langevin sampler, exponentiated gradient descent, and Mirrored Langevin Dynamics (MLD) are unified by a single convex function: $h(\boldsymbol{X}) = \sum_{i=1}^d X_i \log X_i - X_i$, where X_i is the ith entry of the vector $\boldsymbol{X}$. In optimization, h defines the Bregman divergence that yields EGD and enforces Dale's law. In sampling, the same h defines the mirror map that yields MLD on the positive orthant. The Dale-Langevin dynamics derived from the GBM reverse-time SDE sits at the intersection between the two perspectives. The difference, however, is that Dale's law acts on the weights of the network, whereas the multiplicative samplers act on the state vector $\boldsymbol{X}_k$.

Gradient Descent	**Exponentiated Gradient Descent**
$\min_{\boldsymbol{w}\in\mathbb{R}^n} f(\boldsymbol{w})$	$\min_{\boldsymbol{w}\in\mathbb{R}^n} \left[\bar{f}(\boldsymbol{w}) + \frac{1}{\eta} D(\boldsymbol{w}, \boldsymbol{w}_k)\right]$
$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \eta\nabla f(\boldsymbol{w}_k)$	$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k \circ \exp\left(-\eta\nabla f(\boldsymbol{w}_k) \circ \mathsf{sign}(\boldsymbol{w}_k)\right)$
Langevin Dynamics	**Multiplicative Update for Sampling**
Sample from $p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{Z}\exp\left(-f(\boldsymbol{x})\right)$	Sample from $p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{Z}\exp\left(-f(\boldsymbol{x})\right)$ with positive support
$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \eta\nabla f(\boldsymbol{X}_k) + \sqrt{2\eta}\boldsymbol{Z}_k$	$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k \circ \exp\left(-\eta\boldsymbol{X}_k \circ \nabla f(\boldsymbol{X}_k) + \sqrt{\eta}\boldsymbol{Z}_k\right)$

Figure 2: The convex function h that produces EGD in optimization (Section 5.1), and the Dale-Langevin sampler derived from the GBM reverse-time SDE share the same multiplicative $\exp(\cdot)$ update structure, establishing a link between the biological model of learning and score-based generative sampling.

5.1 Exponentiated Gradient Descent and Dale's Law

Cornford et al. (2024) demonstrated that standard gradient descent is a suboptimal phenomenological fit to learning experiments in biologically relevant settings: it violates Dale's law (Eccles et al., 1954) by allowing synaptic flips, which leads to weight distributions that are not log-normal, contradicting experimental observations. Additionally, they showed that exponentiated gradient descent (EGD) (Kivinen & Warmuth, 1997) respects Dale's law, produces log-normally distributed weights, and is superior to gradient descent for synaptic pruning. The learning task can be formulated via mirror descent (Bubeck, 2015) as a minimization of a linear combination of task error and synaptic change penalty. For loss $\ell : \mathbb{R}^n \to \mathbb{R}_+$, we obtain the following update rule for synaptic weights $\boldsymbol{X}_k$, $k \in \mathbb{N}$:

$$\boldsymbol{X}_{k+1} = \arg\min_{\boldsymbol{X}} \left[\ell(\boldsymbol{X}) + \frac{1}{\eta} D_h(\boldsymbol{X}, \boldsymbol{X}_k)\right], \tag{17}$$

where $\ell(\boldsymbol{X}) = \ell(\boldsymbol{X}_k) + \nabla\ell(\boldsymbol{X})^\top\Big|_{\boldsymbol{X}=\boldsymbol{X}_k}(\boldsymbol{X} - \boldsymbol{X}_k)$ is the first-order Taylor-series expansion of ℓ about $\boldsymbol{X}_k$, and D_h is the Bregman divergence of a strictly convex function $h : \mathbb{R}^d \to \mathbb{R}$. In particular, $h(\boldsymbol{X}) = \|\boldsymbol{X}\|_2^2$ yields standard gradient descent: $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \eta\nabla\ell(\boldsymbol{X})\Big|_{\boldsymbol{X}=\boldsymbol{X}_k}$, which does not preserve sign and therefore synaptic flips are possible (Cornford et al., 2024). Specifically, Cornford et al. (2024) choose $h(\boldsymbol{X}) = \sum_{i=1}^{d} |X^{(i)}| \log |X^{(i)}| - |X^{(i)}|$, yielding D_h as the unnormalised relative entropy: $D_h(\boldsymbol{X}, \boldsymbol{X}_k) = \sum_{i=1}^{d} X^{(i)} \log \frac{X^{(i)}}{X_k^{(i)}} - X^{(i)} + X_k^{(i)}$. For this choice, Eq. (17) takes the form

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k \circ \exp\left(-\eta\nabla\ell(\boldsymbol{X})\Big|_{\boldsymbol{X}=\boldsymbol{X}_k} \circ \ \mathrm{sign}(\boldsymbol{X}_k)\right) \tag{18}$$

where $\circ$ denotes element-wise multiplication. This update differs from standard gradient descent in three respects: it is multiplicative rather than additive, involves exponentiation, and preserves the sign of every entry of $\boldsymbol{X}_k$ across the iterations. By design, EGD does not allow synaptic flips, automatically respects Dale's law, and produces log-normally distributed weights (Pogodin et al., 2024), which is precisely the terminal distribution of the GBM forward process. Comparing Eq. (18) with the Dale-Langevin sampler in Eq. (12), the structural equivalence is immediate: the multiplicative updates have the same form and are sign-preserving across iterations, with the role of $\nabla\ell$ in EGD played by the multiplicative score $\boldsymbol{X}_k \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_k, k\delta)$ in the sampler. This connection is illustrated in Fig. 2 and formalized next.

(a) Fashion-MNIST (b) Kuzushiji MNIST (c) MNIST

Figure 3: Samples generated via the NCSN-Deeper architecture (Song & Ermon, 2020) using the Annealed Dale-Langevin sampler (Algorithm 1) for: (a) Fashion-MNIST, (b) Kuzushiji MNIST, and (c) MNIST. Class-averaged images are passed through the forward process to result in noisy images, which is then used to initialize the sampling process.

5.2 Connection to Mirrored Langevin Dynamics

Mirrored Langevin Dynamics (MLD) (Hsieh et al., 2018) is the sampling analogue of mirror descent: it circumvents constrained sampling in the primal space by working in the unconstrained dual space. For a target density $p_{\boldsymbol{X}}(\boldsymbol{x}) \propto \exp(-V(\boldsymbol{x}))$, the canonical SDE is

$$\mathrm{d}\boldsymbol{X}_t = -\nabla V(\boldsymbol{X}_t)\,\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{W}_t. \tag{19}$$

Under the mirror map $\boldsymbol{Y}_t = \nabla h(\boldsymbol{X}_t)$ for a convex h satisfying certain assumptions (cf. (Hsieh et al., 2018)), the corresponding SDE in the dual space is given by

$$\mathrm{d}\boldsymbol{Y}_t = -(\nabla^2 h(\boldsymbol{X}_t))^{-1}\left(\nabla V(\boldsymbol{X}_t) + \nabla \log\det \nabla^2 h(\boldsymbol{X}_t)\right)\,\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{W}_t, \tag{20}$$

where $\nabla^2 h(\boldsymbol{X}_t)$ denotes the Hessian of h evaluated at $\boldsymbol{X}_t$. For the convex function $h(\boldsymbol{X}) = \sum_{i=1}^d X_i \log X_i - X_i$, the same h that yields EGD in Section 5.1, we have $\nabla h(\boldsymbol{X}) = \log \boldsymbol{X}$ and $\nabla^2 h(\boldsymbol{X}) = \mathrm{diag}(1/\boldsymbol{X})$. Upon substitution into Eq. (20), we get

$$\mathrm{d}\log \boldsymbol{X}_t = (\boldsymbol{1} + \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t))\,\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{W}_t. \tag{21}$$

Eq. (21) is a sampling SDE on the positive orthant, the multiplicative analogue of the standard Langevin dynamics, and is recovered directly from the GBM's reverse-time SDE (Eq. (10)) via the Lamperti transform. In the appendix (Sec. C), we specialize MLD to a log-normal target and establish convergence guarantees analogous to those in Wibisono (2018).

6 Experiments

We evaluate the generative performance of the proposed framework on three grayscale image datasets: MNIST (handwritten digits), Fashion-MNIST (clothing items), Kuzushiji MNIST (Cornford et al., 2021) (handwritten cursive Japanese characters); and one multi-channel dataset: CIFAR-10 [1]. For the grayscale datasets, each dataset contains 70,000 images of size 28×28, split into 60,000 training and 10,000 test images. The CIFAR-10 (Krizhevsky et al., 2009) dataset contains 60,000 images of size $32 \times 32 \times 3$, split into 50,000 training and 10,000 test images. The pixel values are rescaled to the range $[1, 2]$ to satisfy the positivity constraint. We use $N = 1000$ discretization steps for the forward SDE in Eq. (4), corresponding

[1] Code is available at https://anonymous.4open.science/r/Dale-Meets-Langevin-65F8

Algorithm 1 Annealed Dale-Langevin multiplicative sampler (ADLS)

Require: $\sigma, \delta, \boldsymbol{\mu}, L, \kappa, \chi, n, \hat{\boldsymbol{\mu}}, \hat{\sigma}$, trained score network $s_{\boldsymbol{\theta}}, n$, `initclass_average`
  $\kappa \leftarrow 1$
  if `initclass_average` == `TRUE` **then**
    $\mathbf{X}_{N-1} \leftarrow$ `get_class_average()`
    $\mathbf{X}_{N-1} \leftarrow$ `apply_forward_process(`$\mathbf{X}_{N-1}$`,`μ`,`σ`)`
  else
    $\mathbf{X}_{N-1} \sim \mathcal{LN}(\hat{\mu}\mathbb{1},\, \hat{\sigma}^2\mathbf{I})$
  end if
  for $k \leftarrow N-1$ **to** 1 **do**
    for $j \leftarrow 1$ **to** L **do**
      $\boldsymbol{Z}_{k,j} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$
      $\boldsymbol{X}_{k-1} = \boldsymbol{X}_k \circ \exp\Big(-\delta\big(\boldsymbol{\mu} - \frac{3\sigma^2}{2}\mathbf{1}\big) + \delta\sigma^2 \boldsymbol{X}_k \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_k, k) + \kappa\sigma\sqrt{\delta}\, \boldsymbol{Z}_{k,j}\Big)$
    end for
    $\kappa \leftarrow \kappa\,\chi$
  end for
  $\boldsymbol{X}_1 \leftarrow \boldsymbol{X}_0$
  for $l \leftarrow 1$ **to** n **do**
    $\boldsymbol{X}_{l+1} = \boldsymbol{X}_l \circ \exp\Big(-\delta\big(\boldsymbol{\mu} - \frac{3\sigma^2}{2}\mathbf{1}\big) + \delta\sigma^2 \boldsymbol{X}_l \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_l, l)\Big)$
  end for

Algorithm 2 Annealed sign-agnostic multiplicative sampler (ASAMS)

Require: $\sigma, \delta, \boldsymbol{\mu}, L, \kappa, \chi, n, \hat{\boldsymbol{\mu}}, \hat{\sigma}$, trained score network $s_{\boldsymbol{\theta}}, n$, initclass average
  $\kappa \leftarrow 1$
  if `initclass_average` == `TRUE` **then**
    $\mathbf{X}_{N-1} \leftarrow$ `get_class_average()`
    $\mathbf{X}_{N-1} \leftarrow$ `apply_forward_process(`$\mathbf{X}_{N-1}$`,`μ`,`σ`)`
  else
    $\mathbf{X}_{N-1} \sim \mathcal{LN}(\hat{\mu}\mathbb{1},\, \hat{\sigma}^2\mathbf{I})$
  end if
  for $k \leftarrow N-1$ **to** 0 **do**
    for $j \leftarrow 1$ **to** L **do**
      $\boldsymbol{Z}_{k,j} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$
      $\boldsymbol{X}_{k-1} = \boldsymbol{X}_k \circ \Big((1 + 2\delta\sigma^2)\mathbf{1} - \delta\boldsymbol{\mu} - \delta\sigma^2 \boldsymbol{X}_k \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_k, k) + \kappa\sigma\sqrt{\delta}\, \boldsymbol{Z}_{k,j}\Big)$
    end for
    $\kappa \leftarrow \kappa\,\chi$
  end for
  $\boldsymbol{X}_1 \leftarrow \boldsymbol{X}_0$
  for $l \leftarrow 1$ **to** n **do**
    $\boldsymbol{X}_{l+1} = \boldsymbol{X}_l \circ \Big((1 + 2\delta\sigma^2)\mathbf{1} - \delta\boldsymbol{\mu} - \delta\sigma^2 \boldsymbol{X}_l \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_l, l)\Big)$
  end for

to a step-size of $\delta = 1/N = 0.001$, with $\boldsymbol{\mu} = \frac{\sigma^2}{2}\mathbf{1}$ and $\sigma = 0.8$. We provide additional details in Section D of the appendix. We provide a qualitative assessment to test if the model memorizes data from the training set in Section E of the appendix.

6.1 Score Network Architecture

The score network is based on the NCSNDeeper architecture (Song & Ermon, 2019) with dilated convolutions, adapted for image generation with N discrete time-steps. The network follows an encoder-decoder structure with skip connections and class conditioning is applied via conditional normalization layers. The encoder consists of an initial convolutional layer followed by five stages of residual blocks with progressive downsampling, where each block incorporates class information. The decoder mirrors this structure with five conditional refine blocks that progressively upsample features while incorporating skip connections from the corresponding encoder stages. For CIFAR-10, we use DDPM++ architecture from Song et al. (2021b), which was the additional architectural inclusion from StyleGAN2 (Karras et al., 2020b) with a pre-existing DDPM architecture (Ho et al., 2020).

6.2 Noise Initialization Strategies

We propose two strategies for initializing the reverse process.

Class-averaged initialization. We compute the per-class mean image of the training set and pass it through the forward diffusion process. The resulting terminal state serves as the starting point for sampling.

Log-normal noise initialization. We apply a logarithmic transformation to the terminal samples obtained from the forward process and fit a Gaussian to the resulting pixel-value histograms, estimating mean μ and standard deviation σ. Sampling noise is then drawn from this Gaussian distribution and exponentiated to synthesise the log-normal noise. For CIFAR-10, μ and σ are estimated channel-wise; for the other three datasets, a single-channel estimate is used. The fitted parameters are reported in Table 2 in the appendix.

Table 1: FID and KID scores computed over 50k generated and 10k real test samples. ADLS stands for Annealed Dale-Langevin Sampler (Algorithm 1); ASAMS stands for Annealed sign-agnostic Sampler (Algorithm 2). w/o denotes sampling without the terminal denoising step; (n) denotes n terminal denoising steps. For CIFAR-10, we use exponential moving-averaged checkpoints every 50k iterations.

Dataset	**Sampler**	**Architecture**	**Initialization**	**FID** (↓)	**KID** (↓)
MNIST	ADLS	NCSN-Deeper	class average	11.4858	0.0663 ± 0.0021
MNIST	ADLS	NCSN-Deeper	log-normal	30.0030	0.0385 ± 0.0016
MNIST	ASAMS	NCSN-Deeper	class average	11.3855	0.0859 ± 0.0022
MNIST	ASAMS	NCSN-Deeper	log-normal	14.6595	0.0489 ± 0.0020
Kuzushiji MNIST	ADLS	NCSN-Deeper	class average	16.3796	0.0176 ± 0.0010
Kuzushiji MNIST	ADLS	NCSN-Deeper	log-normal	27.5636	0.0106 ± 0.0008
Kuzushiji MNIST	ASAMS	NCSN-Deeper	class average	26.8998	0.0186 ± 0.0009
Kuzushiji MNIST	ASAMS	NCSN-Deeper	log-normal	21.6120	0.0689 ± 0.0019
Fashion-MNIST	ADLS	NCSN-Deeper	class average	21.2460	0.0233 ± 0.0007
Fashion-MNIST	ADLS	NCSN-Deeper	log-normal	90.7468	0.0654 ± 0.0020
Fashion-MNIST	ASAMS	NCSN-Deeper	class average	21.4585	0.0364 ± 0.0010
Fashion-MNIST	ASAMS	NCSN-Deeper	log-normal	103.2147	0.1091 ± 0.0021
CIFAR-10 w/o	ADLS	DDPM++	class average	88.3223	0.1575 ± 0.0140
CIFAR-10 (25)	ADLS	DDPM++	class average	38.5620	0.0437 ± 0.0071
CIFAR-10 w/o	ADLS	DDPM++	log-normal	86.3112	0.1585 ± 0.0034
CIFAR-10 (25)	ADLS	DDPM++	log-normal	46.1921	0.0520 ± 0.0022
CIFAR-10 w/o	ASAMS	DDPM++	class average	87.9819	0.1629 ± 0.0132
CIFAR-10 (16)	ASAMS	DDPM++	class average	45.0849	0.0700 ± 0.0094
CIFAR-10 w/o	ASAMS	DDPM++	log-normal	100.7164	0.1812 ± 0.0148
CIFAR-10 (25)	ASAMS	DDPM++	log-normal	51.6470	0.0645 ± 0.0086

6.3 Sampling Algorithms

We derived samplers for the reverse-time SDEs via Euler-Maruyama discretization (Higham, 2001): Eq. (11) and Eq. (12). We propose annealed variants in which the noise coefficient κ is decayed geometrically by a factor $\chi \in (0, 1]$ across noise levels. Setting $\chi = 1.0$ recovers standard Euler-Maruyama, while $\chi < 1$ progressively reduces noise injection over the L repeated sampling steps at each noise level, consistently improving sample quality. We also set $\boldsymbol{\mu} = \frac{\sigma^2}{2}\mathbf{1}$ to simplify the update rule. We present the two annealed samplers in Algorithm 1 and Algorithm 2. For single-channel grayscale datasets (MNIST, Fashion-MNIST, Kuzushiji MNIST), additional denoising was not required. For CIFAR-10, additional denoising was found to improve image quality: 25 denoising steps after Algorithm 1 for both log-normal and class-averaged initialization; 25 denoising steps for log-normal initialization and 16 denoising steps for class-averaged initialization after Algorithm 2. Both sampling algorithms (Algorithm 1 and Algorithm 2) can be used with the initialization strategies proposed in Section 6.2. In particular, setting `initclass_average` to `TRUE` initializes sampling with class-averaged images using `get_class_average()` function, which are then passed through the forward process. Otherwise, the initialization is done with log-normal noise with parameters $\hat{\mu}, \hat{\sigma}$ with appropriate broadcasting in the multichannel case. The values of δ, L, and χ are selected empirically via a grid-search, drawing 1024 samples per configuration to minimize FID and KID. The resulting configurations are reported in Tables 3 and 4 in Section D of the appendix. In Fig. 3, we show samples generated with the class-averaged initialization. We also show that the model does not memorize samples (cf. Section E of the appendix). The effect of annealing on generation is analyzed in Section F of the appendix.

6.4 Evaluation Metrics

We evaluate generative quality using Fréchet Inception Distance (FID) (Heusel et al., 2017) and Kernel Inception Distance (KID) (Bińkowski et al., 2018), computed via the `torcheval` and `torchmetrics` libraries (Detlefsen et al., 2022). Scores are computed using 50,000 generated samples and 50,000 real samples from the test set. For grayscale image datasets, the images are replicated across the three channels and resized to 299×299 to match the input requirements of the InceptionV3 network (Szegedy et al., 2016).

We evaluate samples generated by both samplers (Algorithms 1 and 2) under both initialization strategies (log-normal noise and class-averaged), as initialization is crucial for generation. The FID and KID scores for the annealed versions of the Dale-Langevin sampler (ADLS) and the sign-agnostic multiplicative sampler (ASAMS) are reported in Table 1. We observe that, in general, class-averaged initialization gives rise to comparable or better FID and KID compared to the log-normal initialization. Taken together, both metrics establish that multiplicative samplers offer a viable generative strategy.

7 Conclusions and Outlook

We introduced a novel diffusion framework grounded in Geometric Brownian Motion, an SDE for positive-valued data, which preserves strict positivity throughout the trajectory and admits the log-normal as its marginal distribution. A core contribution of this work is the theoretical unification of multiplicative diffusion, score matching, and optimization. We derive two multiplicative samplers — a sign-agnostic one and a sign-preserving one, termed as the Dale-Langevin sampler. We showed that the Dale-Langevin sampler, obtained by discretizing the GBM's reverse-time SDE, is equivalent to standard additive Langevin dynamics on the log-transformed variable via the Lamperti transform, establishing GBM-based diffusion as the multiplicative-space interpretation of score-based diffusion on $\mathbb{R}^d_+$. To match the multiplicative forward process, we derived the multiplicative denoising score-matching (M-DSM) loss and proved its equivalence to the multiplicative explicit score-matching (M-ESM) loss, generalizing the score-matching framework of Vincent (2011) to the multiplicative setting. Non-negative score matching of (Hyvärinen, 2007) naturally emerges as a special case at $t = 0$. Finally, we bridged these sampling dynamics with optimization, showing that the convex function driving exponential gradient descent exactly dictates Dale-Langevin dynamics under the Mirrored Langevin Dynamics (Hsieh et al., 2018) framework. Experiments on MNIST, Fashion-MNIST, Kuzushiji MNIST, and CIFAR-10 demonstrate the generative capability of score-based generative models designed with multiplicative noise. One could also consider a version of the GBM SDE with time-varying drift and diffusion. We provide the corresponding discretization in Section G as a primer and leave the experimental validation for future work.

Beyond unconditional generation, the proposed multiplicative score-matching framework can be adapted for image denoising and restoration tasks where the forward noising process is inherently multiplicative, as opposed to the widely assumed additive noise. While this work focused on log-normal noise, exploring other distributions, such as the gamma distribution, and their associated SDEs remains a compelling future direction. This would broaden the applicability to domains where various types of multiplicative noise are inherent, such as optical coherence tomography (Li et al., 2025) and synthetic aperture RADAR (Fracastoro et al., 2021), enabling more robust and versatile restoration capabilities. Future research directions include high-resolution image synthesis as well as to non-image domains, such as financial time-series, where proportional changes and multiplicative dynamics are intrinsically fundamental.

Limitations

Like most score-based generative modelling paradigms, the proposed model also requires a large amount of training data and computational resources to achieve good performance. Further, the choice of hyperparameters, such as the noise schedule and learning rate, which are carefully tuned, can affect the performance of the model. However, these limitations are true of all deep generative models and not unique to ours.

Broader Impact Statement

This work advances generative modelling by introducing a Geometric Brownian Motion based diffusion framework with multiplicative dynamics substantiated by experiments on standard image datasets. The experiments did not involve human subjects or sensitive personal data. Potential positive impact includes enabling score-based generative modelling and denoising in settings where multiplicative noise and non-negative data are intrinsically fundamental (e.g., optical coherence tomography and synthetic aperture radar). The technique could also be applied to financial time-series data. Risks are broadly consistent with those of generative models: the approach could be misused to generate biased, fake, or misleading content.

References

Brian 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

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41259–41282. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/80fe51a7d8d0c73ff7439c2a2554ed53-Paper-Conference.pdf. 1.1

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1lUOzWCW. 6.4

Fischer Black and Myron 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. 3

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3–4): 231–357, November 2015. ISSN 1935-8237. doi: 10.1561/2200000050. URL https://doi.org/10.1561/2200000050. 1, 5.1

György Buzsáki and Kenji 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. 1

David Castanon. Reverse-time diffusion processes (corresp.). *IEEE Transactions on Information Theory*, 28 (6):953–956, 1982. doi: 10.1109/TIT.1982.1056571. 2

Tianqi Chen and Mingyuan Zhou. Learning to jump: thinning and thickening latent counts for generative modeling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. 1.1

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David 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. 1

Jonathan Cornford, Damjan Kalajdzievski, Marco Leite, Amélie Lamarquette, Dimitri Michael Kullmann, and Blake Aaron 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. 6

Jonathan Cornford, Roman Pogodin, Arna Ghosh, Kaiwen Sheng, Brendan A. Bicknell, Olivier Codol, Beverley A. Clark, Guillaume Lajoie, and Blake 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. 1, 5.1, 5.1

Nicklas Detlefsen, Jiri Borovec, Justus Schock, et al. Torchmetrics - measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101, 2022. doi: 10.21105/joss.04101. URL https://doi.org/10.21105/joss.04101. 6.4

Prafulla Dhariwal and Alex Q. Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt. 1.1

Joseph 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. 3

John C. Eccles, Paul Fatt, and Kyozo 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. 1, 5.1

Nic Fishman, Leo Klarner, Valentin De Bortoli, Emile Mathieu, and Michael John Hutchinson. Diffusion models for constrained domains. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=xuWTFQ4VGO. Expert Certification. 1.1

Giulia Fracastoro, Enrico Magli, Giovanni Poggi, Giuseppe Scarpa, Diego Valsesia, and Luisa 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. 7

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf. 1.1

Robert Gruhlke, Valentin Resseguier, and Merveille Talla. Multiplicative diffusion models: Beyond gaussian latents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=F6w8LcJJFA. 1.1

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf. 6.4

Desmond 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. 3.2, 6.3

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020. 1, 1.1, 2, 6.1

Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. URL https://openreview.net/forum?id=BBelR2NdDZ5. 2

Ya-Ping Hsieh, Ali Kavis, Paul Rolland, and Volkan Cevher. Mirrored langevin dynamics. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/6490791e7abf6b29a381288cc23a8223-Paper.pdf. 1, 1.2, 5.2, 5.2, 7

Yaozhong 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. 3

Aapo 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. 1.1

Aapo 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. (document), 1, 1.2, 4, 4, 7

Tero Karras, Timo Aila, Samuli Laine, and Jaakko 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. 1.1

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019. doi: 10.1109/CVPR.2019.00453. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Karras_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks_CVPR_2019_paper.html. 1.1

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12104–12114. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8d30aa96e72440759f74bd2306c1fa3d-Paper.pdf. 1.1

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jakkko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020b. 1.1, 6.1

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2021. 1.1

Jyrki Kivinen and Manfred 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. 5.1

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 (Canadian Institute for Advanced Research). *Technical Report*, 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html. 1, 6

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. 1

Sanqian Li, Risa Higashita, Huazhu Fu, Bing Yang, and Jiang 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. 7

Song Liu, Takafumi Kanamori, and Daniel J. Williams. Estimating density models with truncation boundaries using score matching. *Journal of Machine Learning Research*, 23(186):1–38, 2022. URL http://jmlr.org/papers/v23/21-0218.html. 1.1

Yonatan Loewenstein, Annerose Kuras, and Simon 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. 1

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *9th International Conference on Learning Representations, ICLR*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. D

Joshua B. Melander, Aran Nayebi, Bart C. Jongbloets, Dale A. Fortin, Maozhen Qin, Surya Ganguli, Tianyi Mao, and Haining 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. 1

Jan Kloppenborg Møller and Henrik Madsen. *From State Dependent Diffusion to Constant Diffusion in Stochastic Differential Equations by the Lamperti Transform*. IMM-Technical Report-2010-16. Technical University of Denmark, DTU Informatics, Building 321, 2010. 3.1

Eliya Nachmani, Robin San Roman, and Lior Wolf. Denoising diffusion gamma models, 2021. URL https://arxiv.org/abs/2110.05948. 1.1

Keigo Nishida, Eren Mehmet Kıral, Kenichi Bannai, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. Log-normal multiplicative dynamics for stable low-precision training of large networks. In *Proceedings of the 43rd International Conference on Machine Learning*. PMLR, 2026. 1

Bernt Øksendal. *Stochastic Differential Equations*, pp. 38–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985. ISBN 978-3-662-13050-6. doi: 10.1007/978-3-662-13050-6_5. URL https://doi.org/10.1007/978-3-662-13050-6_5. 1, 1.3

Roman Pogodin, Jonathan Cornford, Arna Ghosh, Gauthier Gidel, Guillaume Lajoie, and Blake Aaron 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. 1, 5.1

Julius Richter, Danilo De Oliveira, and Timo Gerkmann. Investigating training objectives for generative speech enhancement. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10887784. 2

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol. 1.1

Stephen Robbins. Score change of variables, 2024. URL https://arxiv.org/abs/2412.07904. 3.1, 3.1

Javier E. Santos, Zachary R. Fox, Nicholas Lubbers, and Yen Ting 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. 1.1

Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019. 2

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *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. 1.1

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP. 2

Sen Song, Per Jesper Sjöström, Markus Reigl, Sacha Nelson, and Dmitri 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. 1

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf. 1.1, 6.1

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12438–12448. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf. 1.1, 3, D

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano 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. 1.1

Yang Song, Jascha, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS. 1, 1.1, 2, 2, 6.1

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021c. URL https://openreview.net/forum?id=PxTIG12RRHS. 1.1, 2, 2, 4, G

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 6.4

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23 (7), 2011. doi: 10.1162/NECO_a_00142. 1, 1.2, 4, 7

An Vuong and Thinh Nguyen. Perception-based multiplicative noise removal using sdes, 2024. URL https://arxiv.org/abs/2408.10283. 1.1

Andre 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. 1, 5.2, C

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747. 1

Shiqing Yu, Mathias Drton, and Ali Shojaie. Generalized score matching for non-negative data. *Journal of Machine Learning Research*, 20(76):1–70, 2019. URL http://jmlr.org/papers/v20/18-278.html. 1.1

Mingyuan Zhou, Tianqi Chen, Zhendong Wang, and Huangjie Zheng. Beta diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=zTSlm4nmlH. 1.1

A Log-normal Distribution

A positive random variable W is said to follow the log-normal distribution if $\log W \sim \mathcal{N}(\mu, \sigma^2)$, that is, $\log W$ follows a Gaussian distribution with mean μ and variance σ^2. We denote this as $W \sim \mathcal{LN}(\mu, \sigma^2)$. The log-normal density is given by

$$f_W(w) = \begin{cases} \dfrac{1}{w\sigma\sqrt{2\pi}} \exp\left(-\dfrac{(\log w - \mu)^2}{2\sigma^2}\right), & w > 0, \\ 0, & w \leq 0. \end{cases} \tag{22}$$

Note that μ and σ^2 are **not** the mean and variance of the log-normal random variable. The mean and variance of the log-normal random variable W are $\mathbb{E}[W] = \exp\left(\mu + \frac{\sigma^2}{2}\right)$ and $\mathrm{Var}(W) = \exp\left(\sigma^2 - 1\right)\exp\left(2\mu + \sigma^2\right)$, respectively.

The multivariate log-normal random vector is defined as $\boldsymbol{W} = \exp(\boldsymbol{\mu} + \sigma \boldsymbol{Z})$ where $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I})$ and the exponentiation is applied element-wise. Effectively, the entries of $\boldsymbol{W}$ are independent and identically distributed according to Eq. (22). The corresponding density is denoted as $\mathcal{LN}(\boldsymbol{\mu}, \sigma^2\mathbb{I})$.

B Equivalence Between Multiplicative Denoising Score-Matching and Multiplicative Explicit Score-Matching

Recall from Sec. 4 of the main document that the multiplicative explicit score-matching loss is given by

$$\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2} \left\| \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t) - \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t) \right\|_2^2 \right], \tag{23}$$

and that the multiplicative denoising score-matching loss is given by

$$\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0 \sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t \sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[\frac{1}{2} \left\| \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0) - \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t) \right\|_2^2 \right]. \tag{24}$$

In the following result, we establish the equivalence between multiplicative explicit score-matching and multiplicative denoising score-matching loss. We state below the assumptions on the density and the data and model score functions over the positive orthant $\mathbb{R}_d^+$:

- **Assumption 1** (Regularity of score functions). The p.d.f. $p_{\boldsymbol{X}_t}$ is differentiable, the expectations $\mathbb{E}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\|\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t)\|_2^2 \right]$ and $\mathbb{E}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\|\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\|_2^2 \right]$ are finite for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $t \in [0, 1]$.
- **Assumption 2** (Boundary conditions). The quantity $p_{\boldsymbol{X}_t}(\boldsymbol{X}_t)(\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t))$ vanishes for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $t \in [0, 1]$ as $\|\boldsymbol{X}_t\| \to \infty$.

Theorem B.1 (Multiplicative Denoising Score-Matching)**.** *Under the assumptions of regularity and appropriate boundary conditions stated above, the M-ESM loss given in Eq. (13) and the M-DSM loss given in Eq. (14) are equivalent up to a constant, i.e., $\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta}) + C$, where C is independent of $\boldsymbol{\theta}$.*

Proof. We assume that the densities $p_{\boldsymbol{X}_t}$ and $p_{\boldsymbol{X}_t|\boldsymbol{X}_0}$ (defined in Sec. 3 of the main document) are supported over $\mathbb{R}_+^d$, and zero elsewhere. Further, we assume that $p_{\boldsymbol{X}_t}(\boldsymbol{x}_t) > 0, p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{x}_t \mid \boldsymbol{x}_0) > 0, \ \forall \ \boldsymbol{x}_t \in \mathbb{R}_+^d$ for $t \in [0, 1]$. The expectations are evaluated over the support $\mathbb{R}_+^d$. We expand $\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta})$ to get

$$\begin{aligned}\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2} \left\| \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t) \right\|^2 \right] + \mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2} \left\| \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t) \right\|^2 \right] \\ - \mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t))^\top (\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)) \right].\end{aligned} \tag{25}$$

Consider the cross-term $\mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t))^\top (\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)) \right]$ and express it as an integral over $\mathbb{R}_+^d$. For brevity of notation, we don't explicitly indicate the support $\mathbb{R}_+^d$ in the following integrals. The cross-term is given by

$$\begin{aligned}&\mathop{\mathbb{E}}_{\boldsymbol{X}_t \sim p_{\boldsymbol{X}_t}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t))^\top (\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)) \right] \\ &\qquad = \int (\boldsymbol{x}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{x}_t))^\top (\boldsymbol{x}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)) p_{\boldsymbol{X}_t}(\boldsymbol{x}_t) \, \mathrm{d}\boldsymbol{x}_t \\ &\qquad = \int (\boldsymbol{x}_t \circ \nabla p_{\boldsymbol{X}_t}(\boldsymbol{x}_t))^\top (\boldsymbol{x}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)) \, \mathrm{d}\boldsymbol{x}_t.\end{aligned} \tag{26}$$

We know that the marginal density $p_{\boldsymbol{X}_t}(\boldsymbol{x}_t)$ can be expressed in terms of the conditional density as

$$p_{\boldsymbol{X}_t}(\boldsymbol{x}_t) = \int p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{x}_t|\boldsymbol{x}_0) p_{\boldsymbol{X}_0}(\boldsymbol{x}_0) \, \mathrm{d}\boldsymbol{x}_0.$$

Computing the gradient with respect to $\boldsymbol{x}_t$ on both sides yields

$$\nabla p_{\boldsymbol{X}_t}(\boldsymbol{x}_t) = \int \nabla p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{x}_t|\boldsymbol{x}_0) p_{\boldsymbol{X}_0}(\boldsymbol{x}_0)\, \mathrm{d}\boldsymbol{x}_0. \tag{27}$$

Substituting Eq. (27) in Eq. (26), multiplying and dividing by $p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{x}_t|\boldsymbol{x}_0)$, we get

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{X}_t\sim p_{\boldsymbol{X}_t}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t))^\top (\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t))\right] \\
&= \int \left(\boldsymbol{x}_t \circ \int \nabla p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{x}_t|\boldsymbol{x}_0) p_{\boldsymbol{X}_0}(\boldsymbol{x}_0)\, \mathrm{d}\boldsymbol{x}_0\right)^\top (\boldsymbol{x}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t))\, \mathrm{d}\boldsymbol{x}_t \\
&= \iint (\boldsymbol{x}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{x}_t|\boldsymbol{x}_0))^\top (\boldsymbol{x}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t))\, p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{x}_t|\boldsymbol{x}_0) p_{\boldsymbol{X}_0}(\boldsymbol{x}_0) d\boldsymbol{x}_0\, \mathrm{d}\boldsymbol{x}_t, \\
&= \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0))^\top (\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t))\right].
\end{aligned} \tag{28}$$

Substituting Eq. (28) in Eq. (25) gives the following equivalent expression for the multiplicative explicit score-matching loss:

$$\begin{aligned}
\mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta}) &= \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{X}_t\sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2}\left\|\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t)\right\|^2\right]}_{C_1} + \mathop{\mathbb{E}}_{\boldsymbol{X}_t\sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2}\left\|\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\right\|^2\right] \\
&\quad - \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0))^\top (\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t))\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{X}_t\sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2}\left\|\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\right\|^2\right] \\
&\quad - \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0))^\top (\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t))\right] + C_1,
\end{aligned} \tag{29}$$

where C_1 is a constant that is not dependent on $\boldsymbol{\theta}$.
We carry out a similar simplification for the multiplicative denoising score-matching loss:

$$\begin{aligned}
\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) &= \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[\frac{1}{2}\left\|\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0) - \boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\right\|_2^2\right], \\
&= \underbrace{\mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[\frac{1}{2}\left\|\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0)\right\|_2^2\right]}_{C_2} + \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[\frac{1}{2}\left\|\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\right\|_2^2\right], \\
&\quad - \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0))^\top (s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t) \circ \boldsymbol{X}_t)\right],
\end{aligned}$$

or equivalently,

$$\begin{aligned}
\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) &= \mathop{\mathbb{E}}_{\boldsymbol{X}_t\sim p_{\boldsymbol{X}_t}} \left[\frac{1}{2}\left\|\boldsymbol{X}_t \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t)\right\|_2^2\right] - \mathop{\mathbb{E}}_{\substack{\boldsymbol{X}_0\sim p_{\boldsymbol{X}_0} \\ \boldsymbol{X}_t\sim p_{\boldsymbol{X}_t|\boldsymbol{X}_0}}} \left[(\boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t|\boldsymbol{X}_0}(\boldsymbol{X}_t|\boldsymbol{X}_0))^\top (s_{\boldsymbol{\theta}}(\boldsymbol{X}_t, t) \circ \boldsymbol{X}_t)\right] \\
&\quad + C_2,
\end{aligned} \tag{30}$$

where C_2 is a constant that does not depend on $\boldsymbol{\theta}$.
On comparing Eq. (29) and Eq. (30), we get

$$\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{M-ESM}}(\boldsymbol{\theta}) + C_2 - C_1. \tag{31}$$

This concludes the proof. □

The implication of the result is as follows. Multiplicative explicit score-matching loss is intractable since we do not have access to the true marginal scores. The above equivalence allows us to optimize the score network parameters by minimizing the multiplicative denoising score-matching loss instead since the conditional scores can be tractably computed from the forward SDE (cf. Section 3).

C Mirrored Langevin Dynamics for a Log-normal Target

Wibisono (2018) show that the unadjusted Langevin algorithm for the Ornstein-Uhlenbeck process leads to a bias, i.e., it does not converge exactly to the target distribution. In particular, Wibisono considers a Gaussian target with mean $\boldsymbol{\mu}$ and covariance matrix Σ and the limit measure is a Gaussian with the correct mean but covariance matrix $\hat{\Sigma} = \Sigma(\mathbb{I} - \frac{\epsilon}{2}\Sigma^{-1})^{-1}$, where ϵ is the step-size. In a similar vein, for a log-normal target, we show that the limiting density obtained from GBM does not converge to the correct target density and is biased.

Consider the target $\mathcal{LN}(\boldsymbol{\mu}, \sigma^2\mathbb{I})$, i.e., $\nu(\boldsymbol{x}) = (2\pi\sigma^2)^{-d/2} \prod_{j=1}^{d} x_i^{-1} \exp\left(-\frac{1}{2\sigma^2} \|\log \boldsymbol{x} - \boldsymbol{\mu}\|_2^2\right)$. It can be verified that $\mathbf{1} + \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t) = -\frac{1}{\sigma^2}(\log \boldsymbol{X}_t - \boldsymbol{\mu})$. Substituting this in Eq. (21), we get

$$\mathrm{d}\log \boldsymbol{X}_t = -\frac{1}{\sigma^2}(\log \boldsymbol{X}_t - \boldsymbol{\mu})\,\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{W}_t. \tag{32}$$

Using the Euler-Maruyama discretization with δ such that $0 < \delta < \sigma^2$, we obtain

$$\log \boldsymbol{X}_k = \log \boldsymbol{X}_{k-1} - \frac{\delta}{\sigma^2}(\log \boldsymbol{X}_{k-1} - \boldsymbol{\mu}) + \sqrt{2\delta}\boldsymbol{Z}_k,$$

where $\boldsymbol{Z}_k \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$. Subtracting $\boldsymbol{\mu}$ from both sides and rearranging terms gives

$$\log \boldsymbol{X}_k - \boldsymbol{\mu} = \left(1 - \frac{\delta}{\sigma^2}\right)(\log \boldsymbol{X}_{k-1} - \boldsymbol{\mu}) + \sqrt{2\delta}\boldsymbol{Z}_k.$$

Unrolling this equation till $k = 0$, we get

$$\log \boldsymbol{X}_k - \boldsymbol{\mu} = \left(1 - \frac{\delta}{\sigma^2}\right)^k (\log \boldsymbol{X}_0 - \boldsymbol{\mu}) + \sqrt{\frac{2\sigma^2}{2 - \frac{\delta}{\sigma^2}}\left(1 - \left(1 - \frac{\delta}{\sigma^2}\right)^{2k}\right)}\boldsymbol{Z}_0.$$

This implies the distributional convergence: $\log \boldsymbol{X}_k \xrightarrow{\mathrm{d}} \boldsymbol{\mu} + \sigma\sqrt{\frac{2}{2 - \frac{\delta}{\sigma^2}}}\boldsymbol{Z}_0$, i.e., $\boldsymbol{X}_k \xrightarrow{\mathrm{d}} \exp\left(\boldsymbol{\mu} + \sigma\sqrt{\frac{2}{2 - \frac{\delta}{\sigma^2}}}\boldsymbol{Z}_0\right)$. The corresponding limit measure is $\nu_\delta(\boldsymbol{x}) = \mathcal{LN}\left(\boldsymbol{\mu}, \frac{\sigma^2}{\left(1 - \frac{\delta}{2\sigma^2}\right)}\mathbb{I}\right)$. In this case, integrating the SDE (Eq. (21)) generates samples from the limit measure, which has the right parameter $\boldsymbol{\mu}$, but the covariance matrix of the underlying Gaussian is biased by a factor of $\frac{1}{\left(1 - \frac{\delta}{2\sigma^2}\right)}$.

D Additional Experimental Details

The models are implemented in PyTorch and optimized using AdamW (Loshchilov & Hutter, 2019). The MNIST model is trained for 300k iterations; Fashion-MNIST and Kuzushiji MNIST models are trained for 200k iterations. The CIFAR-10 model is trained for 1.3 million iterations, with checkpoints saved every 5k iterations. For CIFAR-10, we performed exponential moving-average (EMA) for every 50k, 100k, 150k iterations and found that models with EMA for every 50k steps gave better results, similar to the setting in Song & Ermon (2020) for the additive noise. Training was distributed across two NVIDIA RTX 4090 and

Table 2: Estimated parameters $(\hat{\mu}, \hat{\sigma})$ used for log-normal noise initialization. For CIFAR-10, the $\hat{\mu}$ and $\hat{\sigma}$ values are given channel-wise.

Dataset	$\hat{\mu}$	$\hat{\sigma}$
Fashion-MNIST	0.2771	0.8428
Kuzushiji MNIST	0.2140	0.8290
MNIST	0.1565	0.8353
CIFAR-10	[0.4075, 0.3785, 0.3693]	[0.8016, 0.8004, 0.8005]

Table 3: Experimentally determined L, δ, and χ per dataset for log-normal noise initialization, that minimize FID and KID. ADLS stands for Annealed Dale-Langevin Sampler; ASAMS stands for Annealed sign-agnostic Sampler.

Sampler	**Dataset**	L	δ	χ
ADLS (Algorithm 1)	Fashion-MNIST	1	5.86×10^{-4}	0.9950
	Kuzushiji MNIST	3	2.11×10^{-4}	0.9950
	MNIST	1	6.06×10^{-4}	0.9950
	CIFAR-10	6	1.14×10^{-4}	0.9995
ASAMS (Algorithm 2)	Fashion-MNIST	3	1.46×10^{-4}	0.9995
	Kuzushiji MNIST	3	2.11×10^{-4}	1.0000
	MNIST	4	1.43×10^{-4}	0.9950
	CIFAR-10	5	1.43×10^{-4}	0.9995

Table 4: Experimentally determined L, δ, and χ initialized from class-averaged samples passed through the forward process, that minimize FID and KID. ADLS stands for Annealed Dale-Langevin Sampler; ASAMS stands for Annealed sign-agnostic Sampler.

Sampler	**Dataset**	L	δ	χ
ADLS (Algorithm 1)	Fashion-MNIST	1	5.86×10^{-4}	0.9950
	Kuzushiji MNIST	3	2.11×10^{-4}	0.9950
	MNIST	1	6.06×10^{-4}	0.9950
	CIFAR-10	6	1.14×10^{-4}	0.9995
ASAMS (Algorithm 2)	Fashion-MNIST	3	2.59×10^{-4}	0.9995
	Kuzushiji MNIST	2	3.30×10^{-4}	0.9950
	MNIST	4	1.43×10^{-4}	0.9950
	CIFAR-10	5	1.43×10^{-4}	0.9995

two NVIDIA A6000 GPUs. The loss function for each model is the Monte Carlo estimator of the M-DSM loss (Eq. (14)):

$$\hat{\mathcal{L}}_{\text{M-DSM}}(\boldsymbol{\theta}) = \frac{1}{NM} \sum_{i=1}^{M} \sum_{k=0}^{N-1} \left[\frac{1}{2} \left\| \boldsymbol{X}_k^{(i)} \circ \nabla \log p_{\boldsymbol{X}_k|\boldsymbol{X}_0} \left(\boldsymbol{X}_k^{(i)} \mid \boldsymbol{X}_0^{(i)} \right) - \boldsymbol{X}_k^{(i)} \circ s_{\boldsymbol{\theta}}(\boldsymbol{X}_k^{(i)}, k) \right\|_2^2 \right], \tag{33}$$

where $k = 0, 1, \ldots, N-1$ indexes the discretized time-step and $i = 1, \ldots, M$ indexes the training sample.

We set $\boldsymbol{\mu} = \frac{\sigma^2}{2}\mathbf{1}$ to simplify the update rule. The best-case values for δ, L, and χ are selected via an empirical grid search, drawing 1,024 samples per configuration and selecting the setting that gives the best FID and KID. The resulting parameter values for log-normal initialization and class-averaged initialization are reported in Table 3 and Table 4, respectively.

E The GBM Model Does not Memorize the Training Data

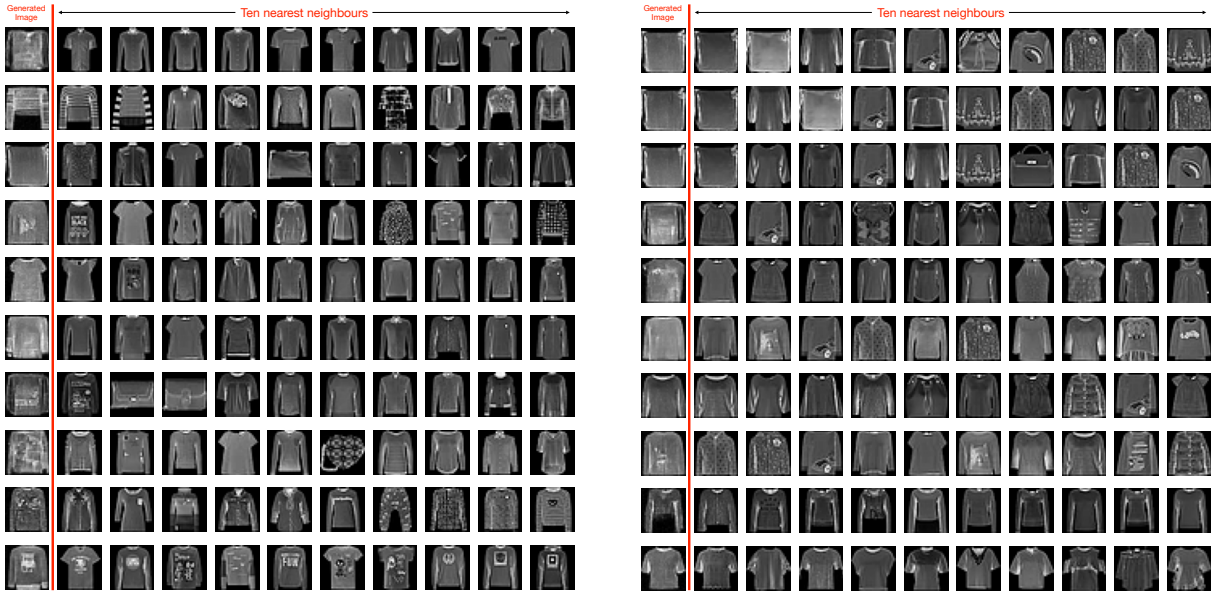
To verify that the model generalizes rather than memorizes, we retrieve the 10 nearest neighbours of each generated sample from the training set using the Euclidean (ℓ_2) distance measured in two complementary

(a) InceptionV3 features

(b) Raw Images as features

Figure 4: Top ten nearest neighbors of the generated CIFAR-10 samples (shown in the first column) with log-normal noise initialization and Dale-Langevin sampler (Algorithm 1). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).

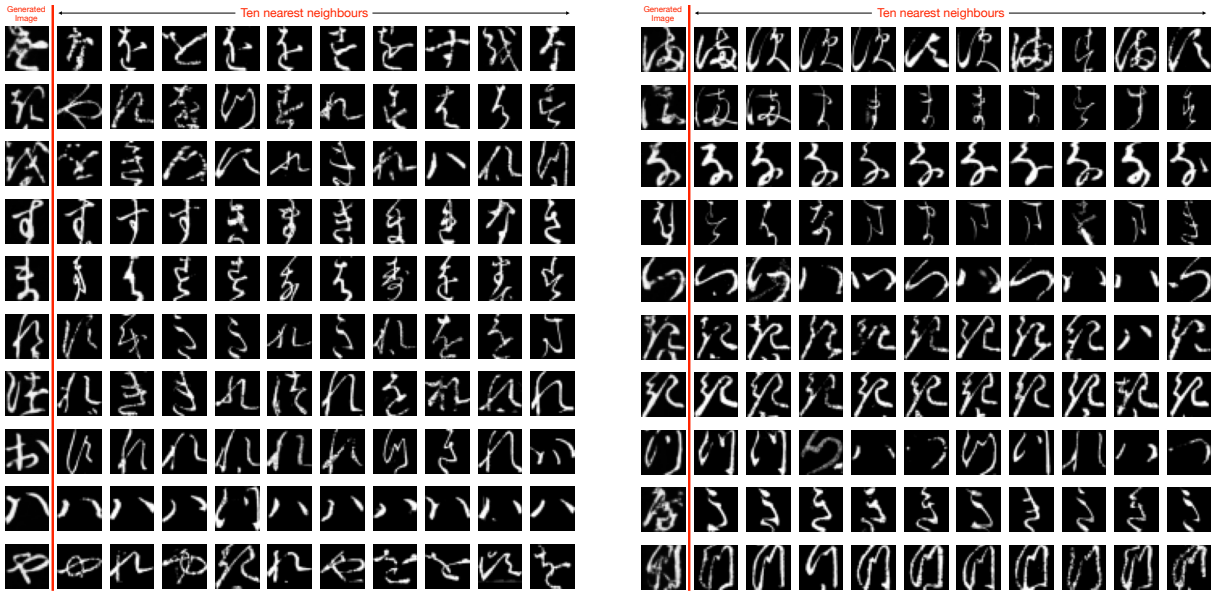
spaces: directly in pixel space (raw image intensities) and in the InceptionV3 feature space (semantic embeddings extracted from a pretrained InceptionV3 network). Pixel-space retrieval checks for low-level visual similarity, while feature-space retrieval probes semantic similarity independent of pixel-level variation. We considered various generative settings (two samplers, two initializations, four datasets) and elicited the nearest neighbours in each case. The nearest neighbours for each setting are shown in Fig. 4 – Fig. 15. In both scenarios, the retrieved neighbours are semantically coherent with the generated samples, but not identical, which confirms that the model captures the diversity of the underlying data distribution and does not merely reduce to training set memorization.



(a) InceptionV3 features

(b) Raw image as features

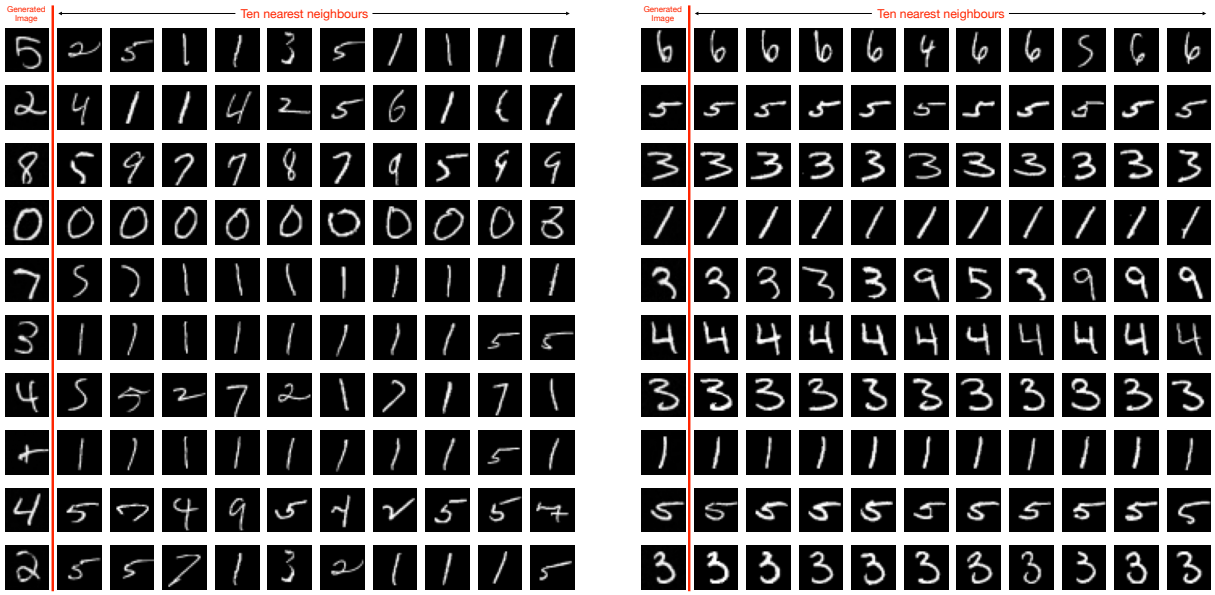
Figure 5: Top ten nearest neighbors of the generated Fashion-MNIST samples (shown in the first column) with log-normal noise initialization and Dale-Langevin sampler (Algorithm 1). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

(b) Raw image as features

Figure 6: Top ten nearest neighbors of the generated Kuzushiji MNIST samples (shown in the first column) with log-normal noise initialization and Dale-Langevin sampler (Algorithm 1). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

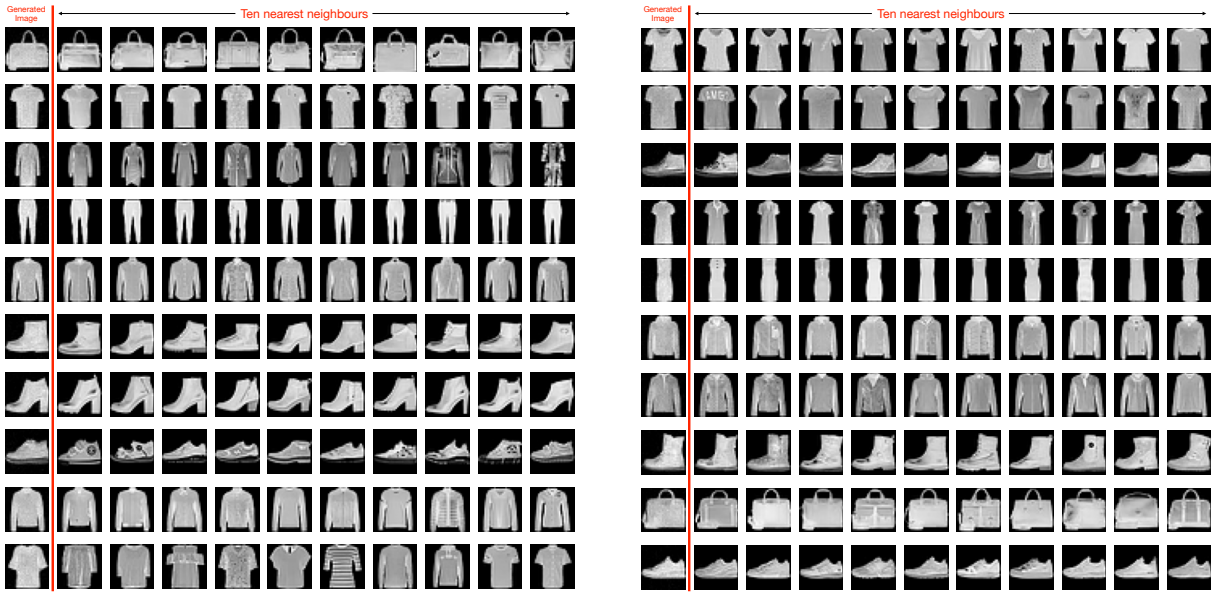
(b) Raw image as features

Figure 7: Top ten nearest neighbors of the generated MNIST samples (shown in the first column) with log-normal noise initialization and Dale-Langevin sampler (Algorithm 1). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).

(a) InceptionV3 features

(b) Raw image as features

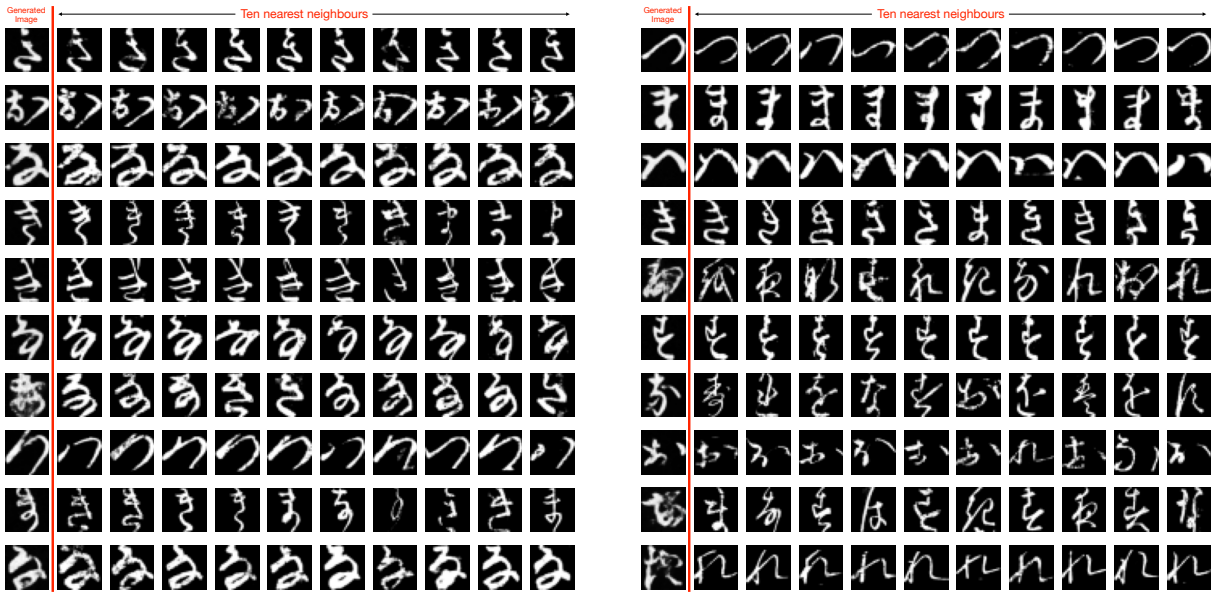
Figure 8: Top ten nearest neighbors of the generated CIFAR-10 samples (shown in the first column) with class-averaged image passed through forward process as initialization and Dale-Langevin sampler (Algorithm 1). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

(b) Raw image as features

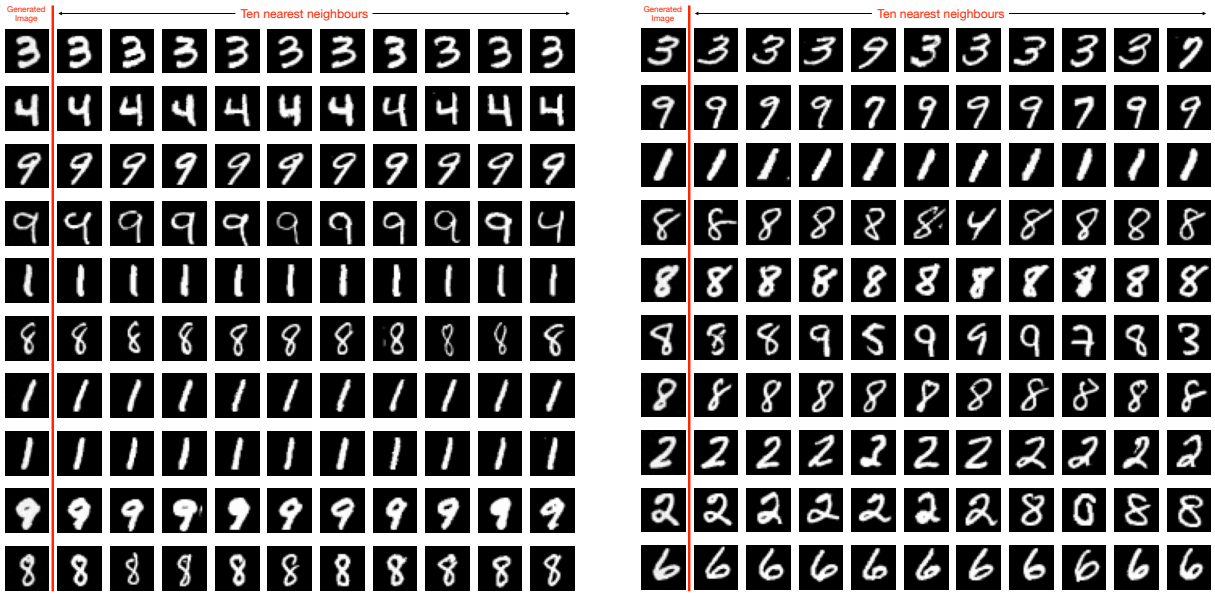
Figure 9: Top ten nearest neighbors of the generated Fashion-MNIST samples (shown in the first column) with class-averaged image passed through forward process as initialization and Dale-Langevin sampler (Algorithm 1). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

(b) Raw image as features

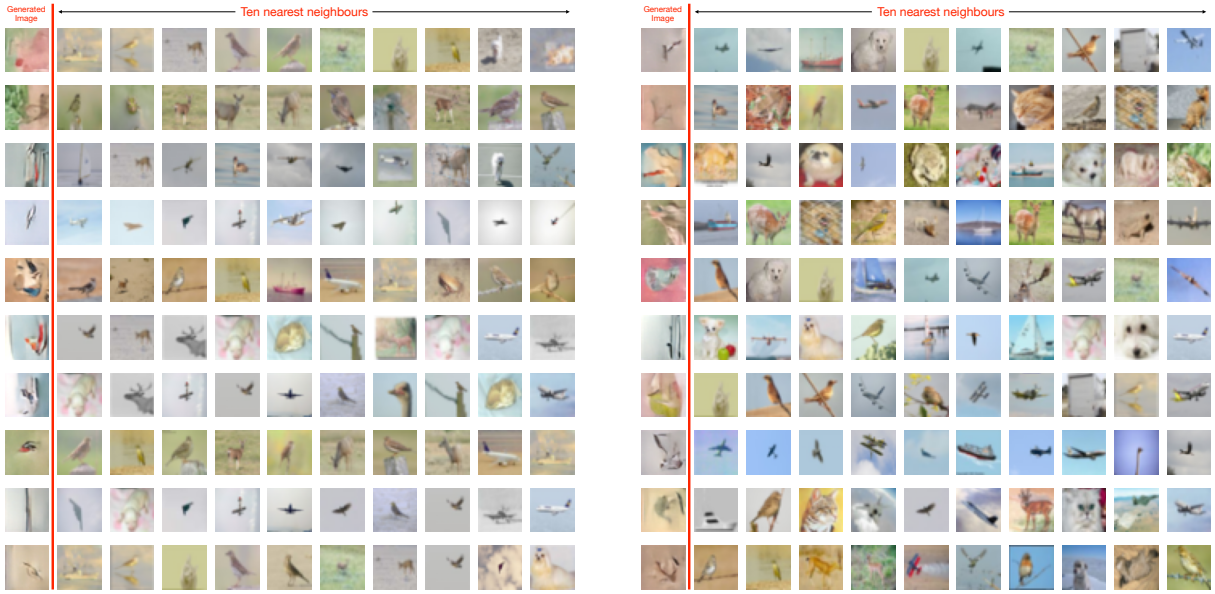
Figure 10: Top ten nearest neighbors of the generated Kuzushiji MNIST samples (shown in the first column) with class-averaged image passed through forward process as initialization and Dale-Langevin sampler (Algorithm 1). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

(b) Raw image as features

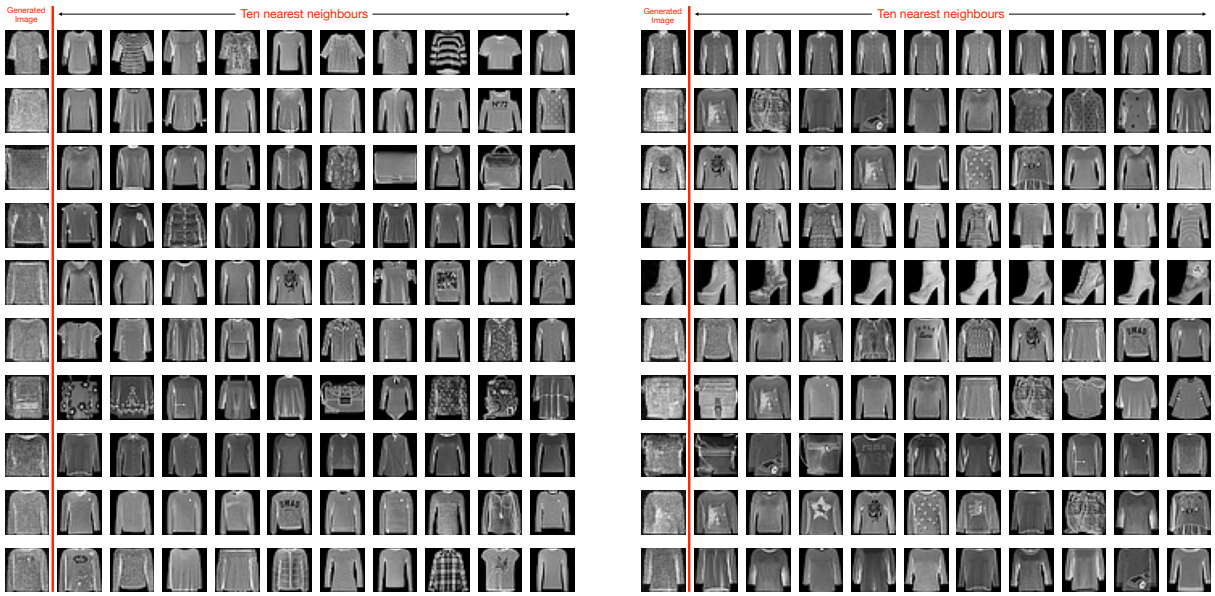
Figure 11: Top ten nearest neighbors of the generated MNIST samples (shown in the first column) with class-averaged image passed through forward process as initialization and Dale-Langevin sampler (Algorithm 1). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

(b) Raw image as features

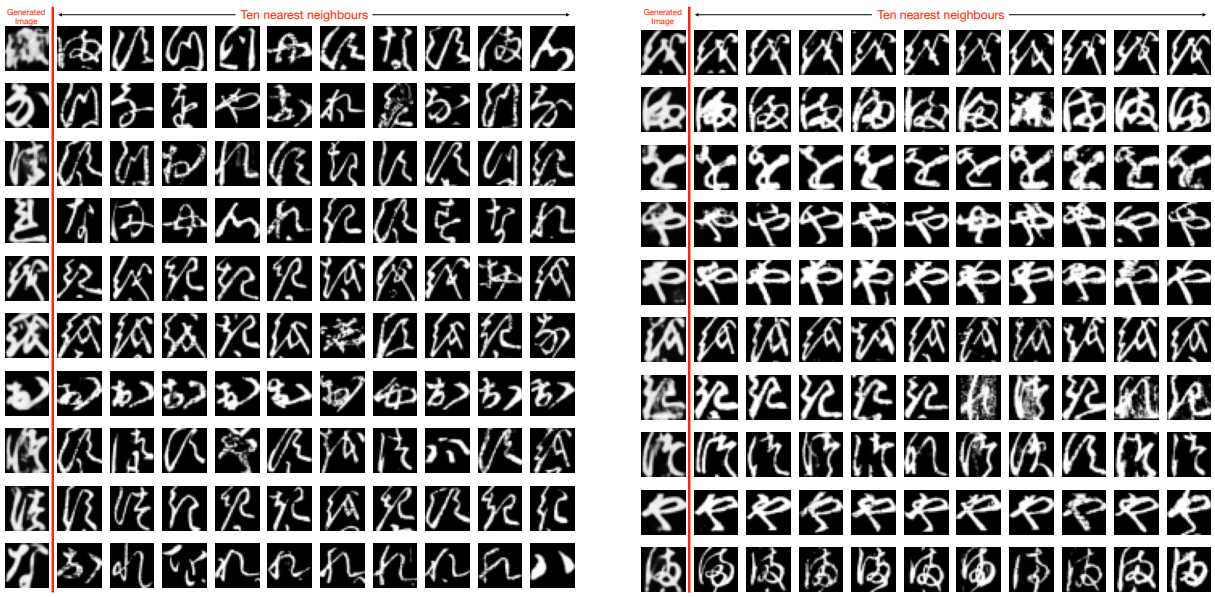
Figure 12: Top ten nearest neighbors of the generated CIFAR-10 samples (shown in the first column) with log-normal initialization and sign-agnostic multiplicative sampler (Algorithm 2). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

(b) Raw image as features

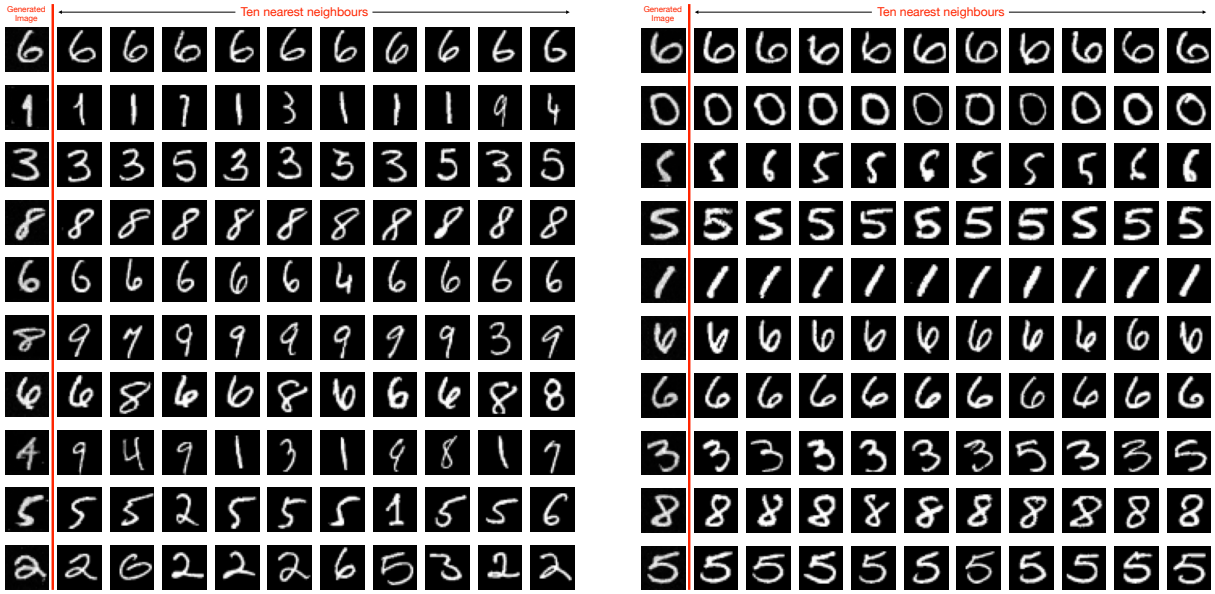
Figure 13: Top ten nearest neighbors of the generated Fashion-MNIST samples (shown in the first column) with log-normal initialization and sign-agnostic multiplicative sampler (Algorithm 2). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

(b) Raw image as features

Figure 14: Top ten nearest neighbors of the generated Kuzushiji MNIST samples (shown in the first column) with log-normal initialization and sign-agnostic multiplicative sampler (Algorithm 2). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

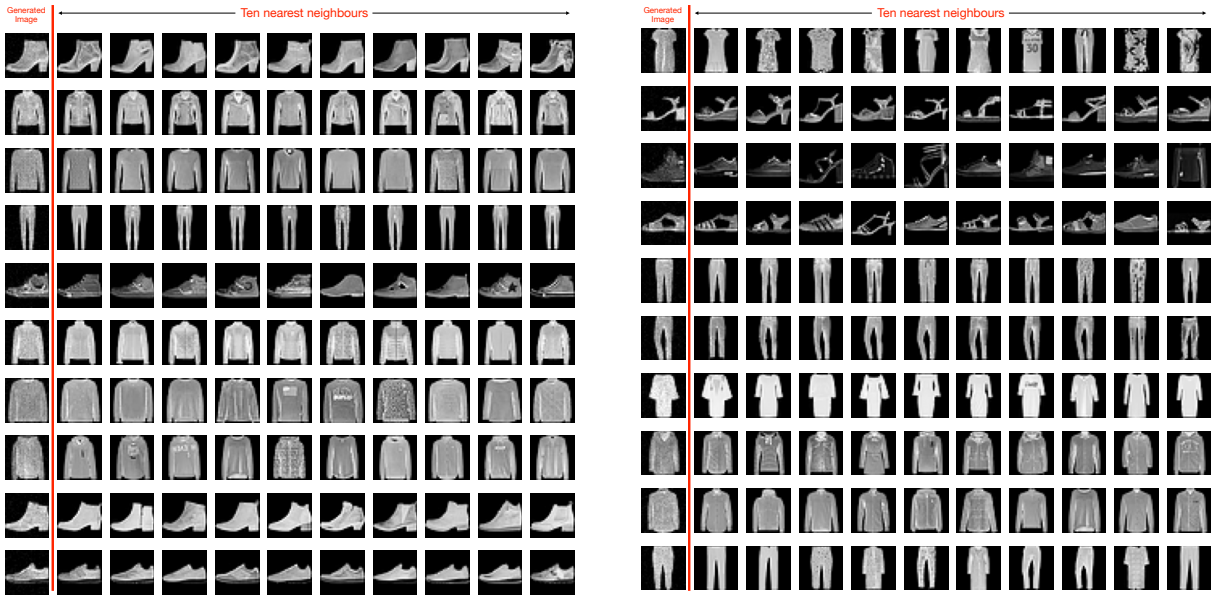
(b) Raw image as features

Figure 15: Top ten nearest neighbors of the generated MNIST samples (shown in the first column) with log-normal initialization and sign-agnostic multiplicative sampler (Algorithm 2). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).

(a) InceptionV3 features

(b) Raw image as features

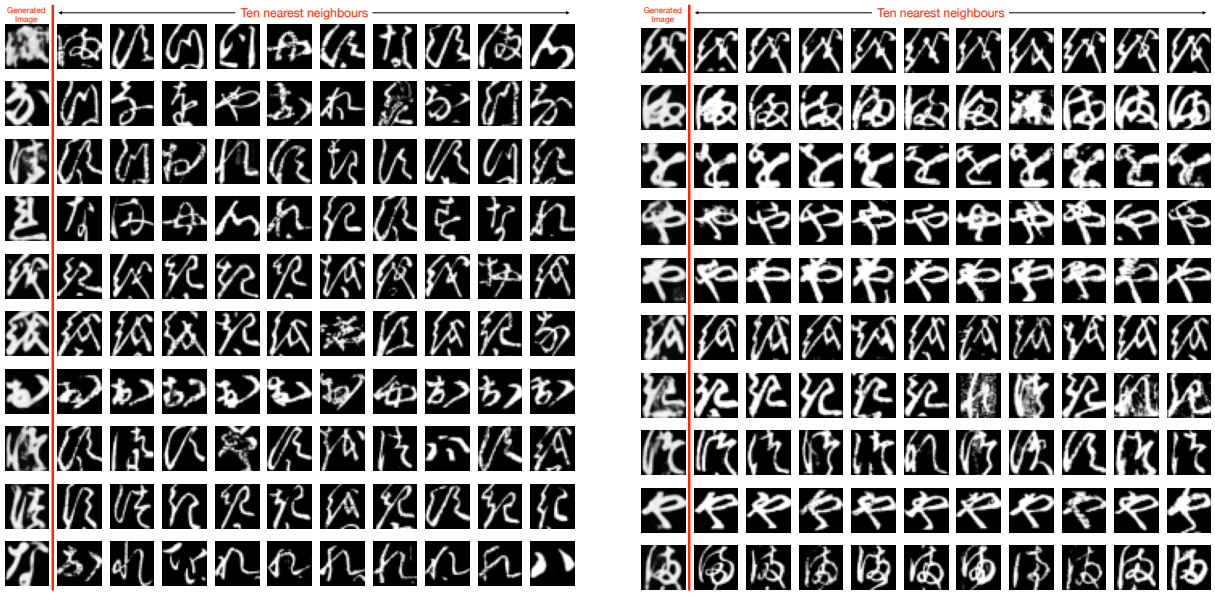
Figure 16: Top ten nearest neighbors of the generated CIFAR-10 samples (shown in the first column) with class-averaged image passed through forward process as initialization and sign-agnostic multiplicative sampler (Algorithm 2). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

(b) Raw image as features

Figure 17: Top ten nearest neighbors of the generated Fashion-MNIST samples (shown in the first column) with class-averaged image passed through forward process as initialization and sign-agnostic multiplicative sampler (Algorithm 2). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).



(a) InceptionV3 features

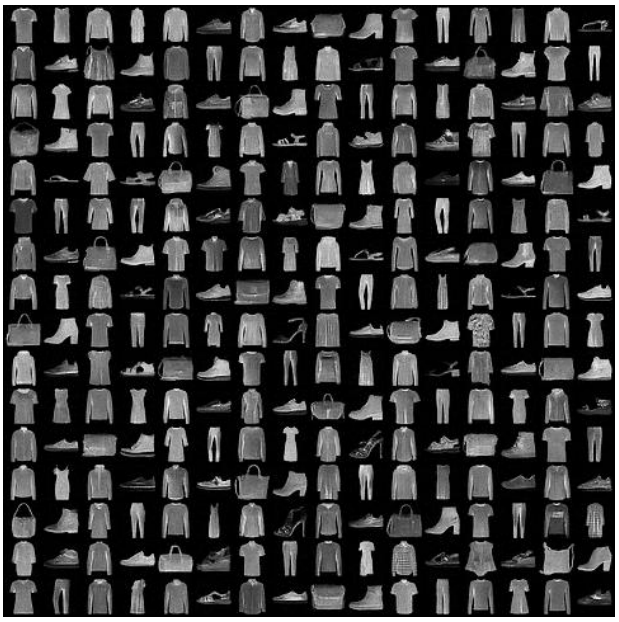
(b) Raw image as features

Figure 18: Top ten nearest neighbors of the generated Kuzushiji MNIST samples (shown in the first column) with class-averaged image passed through forward process as initialization and sign-agnostic multiplicative sampler (Algorithm 2). The nearest neighbors are identified using Euclidean distance on InceptionV3 features in (a) and raw images as features in (b).

F Effect of Annealing

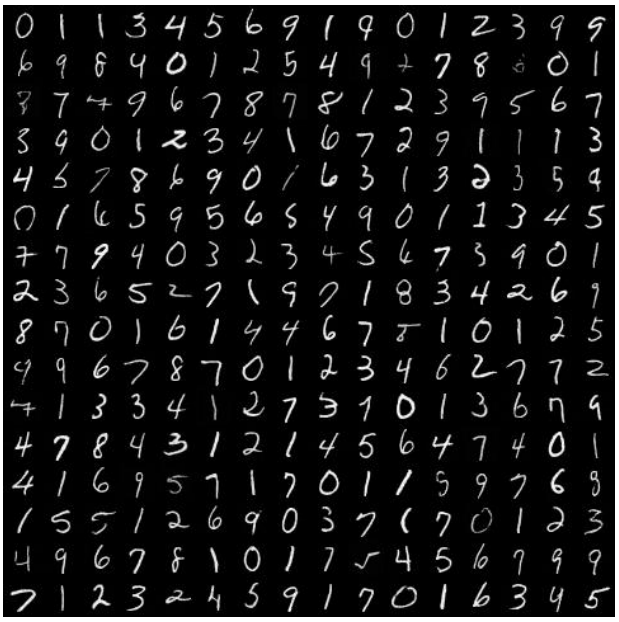
We investigate how the annealing factor χ influences the quality of generated samples for different initialization strategies. For both log-normal and class-average initializations, we conducted an ablation study analyzing the interplay between the annealing factor and the sampler architecture while holding L and the step-size δ constant. This impact is tested across CIFAR-10, Fashion-MNIST, Kuzushiji MNIST, and MNIST datasets using factors of $\chi = 0.995$, 0.9995, and 1.0.

Fig. 19 - Fig. 30 demonstrate the effect of annealing on generated sample quality. We observe a trade-off between sample quality and diversity as a function of the parameter χ. While $\chi = 1.0$ produces diverse, but mildly noisy samples, $\chi = 0.995$ results in relatively clean samples, but with mode collapse on some datasets. The choice of $\chi = 0.9995$ resulted in samples with a right balance of noise and diversity.

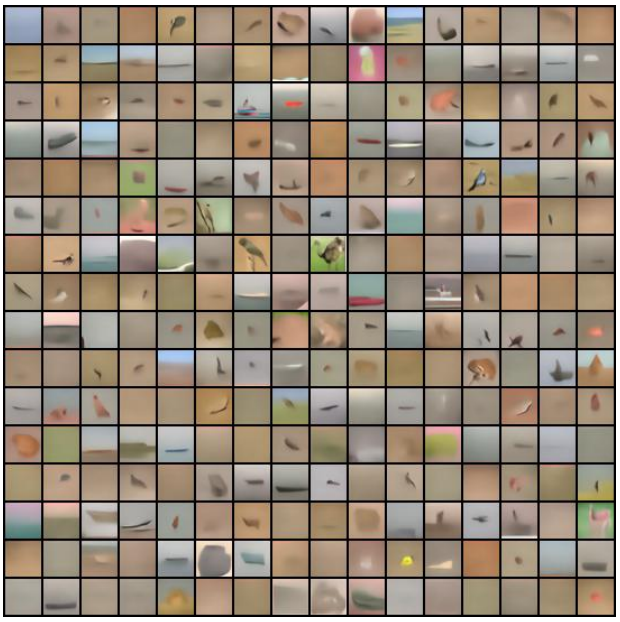


(a) Fashion-MNIST

(b) Kuzushiji MNIST

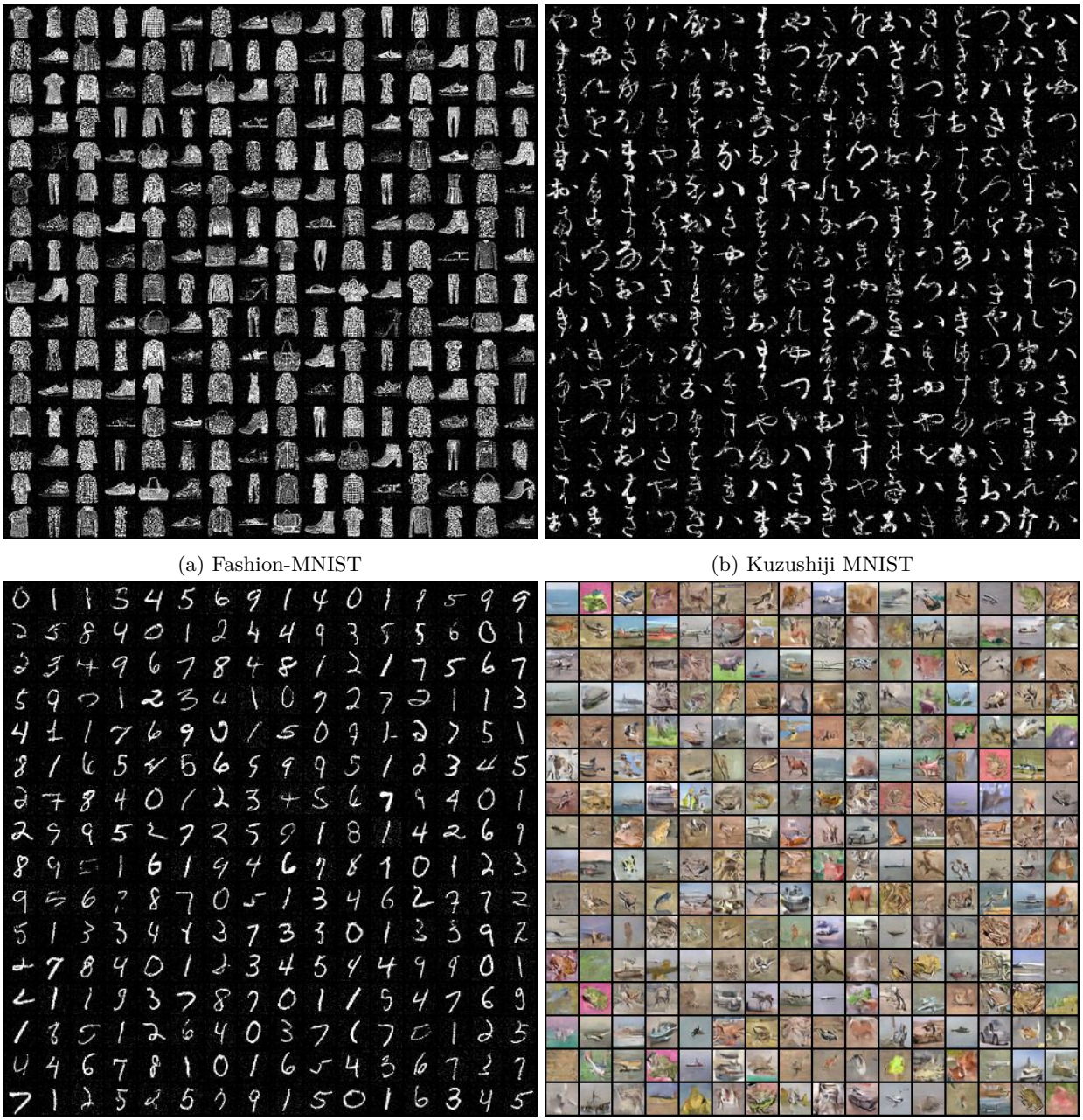


(c) MNIST



(d) CIFAR-10

Figure 19: Samples produced by the Dale-Langevin sampler (Algorithm 1) with annealing factor $\chi = 0.995$, and class-averaged initialization.



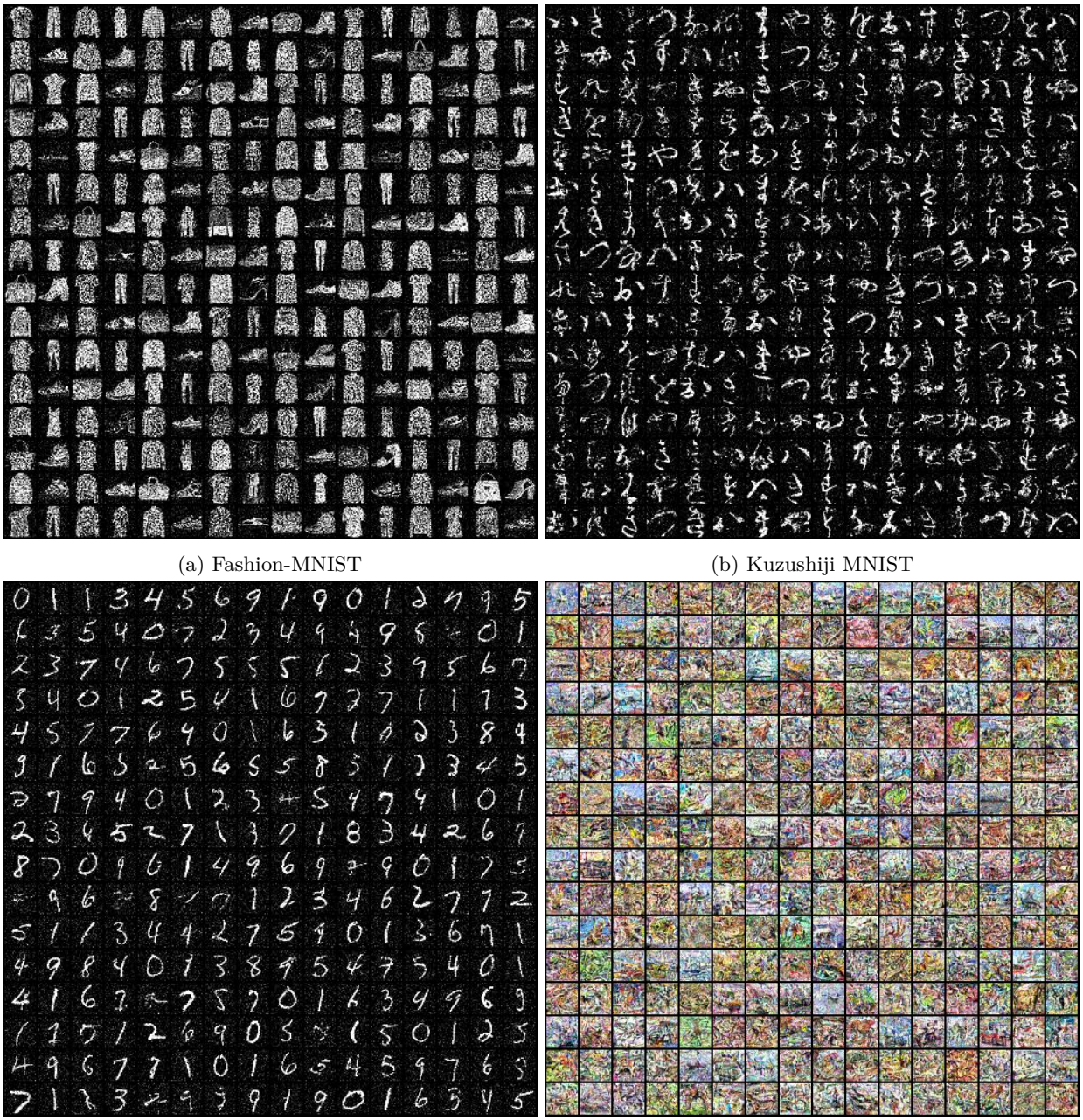
(a) Fashion-MNIST

(b) Kuzushiji MNIST

(c) MNIST

(d) CIFAR-10

Figure 20: Samples produced by the Dale-Langevin sampler (Algorithm 1) with annealing factor $\chi = 0.9995$, and class-averaged initialization.



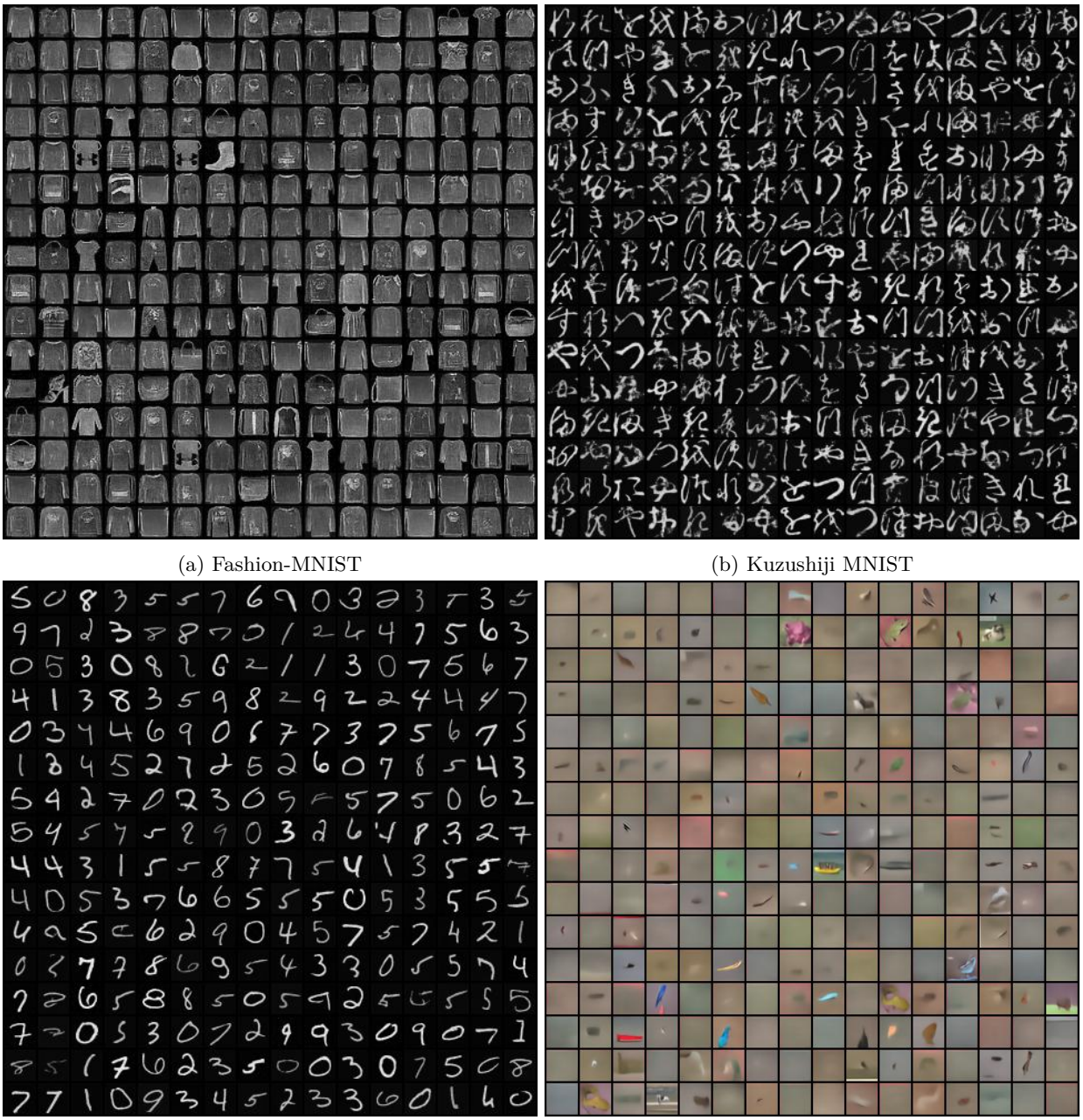
(a) Fashion-MNIST

(b) Kuzushiji MNIST

(c) MNIST

(d) CIFAR-10

Figure 21: Samples produced by the Dale-Langevin sampler (Algorithm 1) with annealing factor $\chi = 1.0$, and class-averaged initialization.



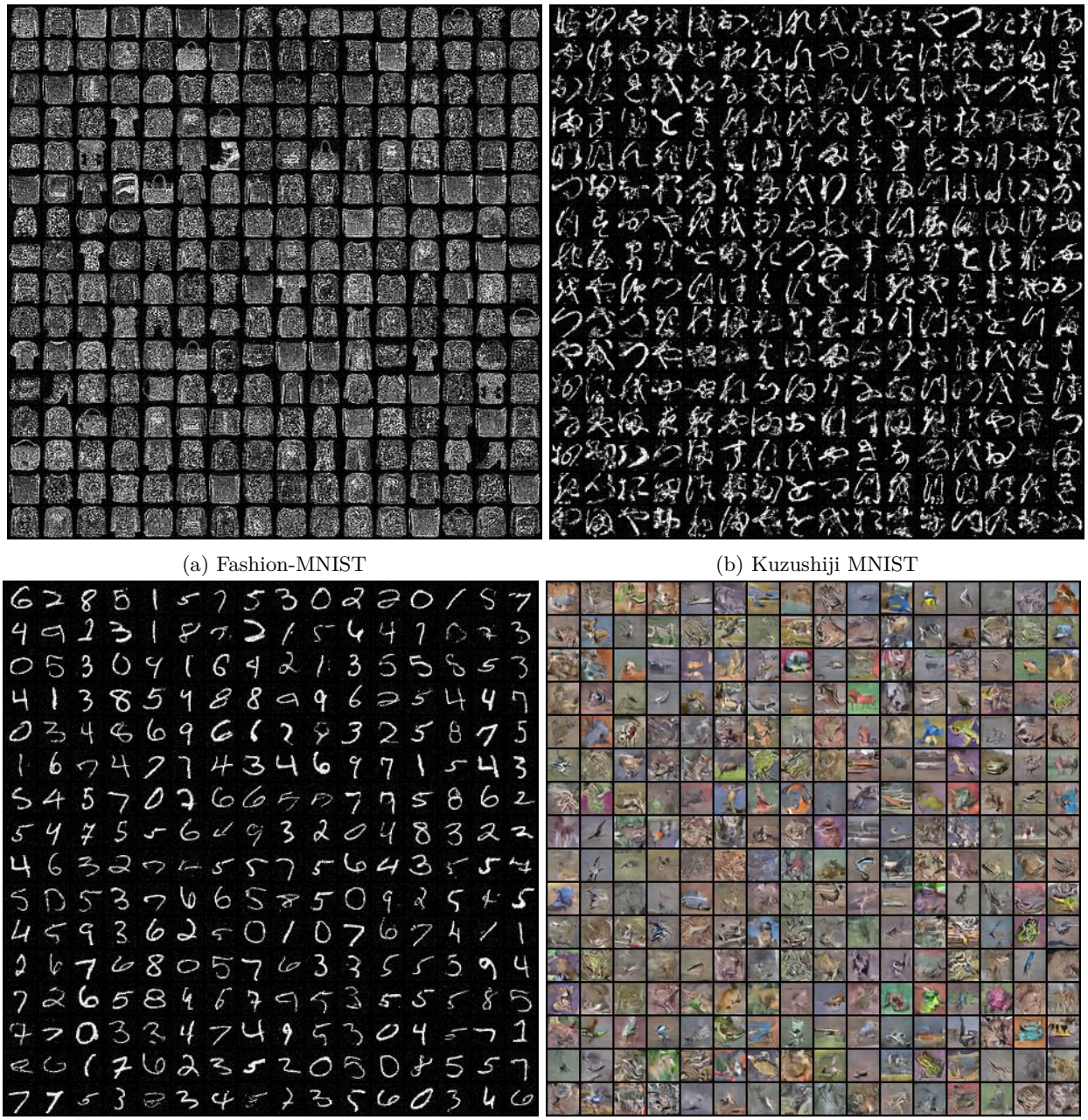
(a) Fashion-MNIST

(b) Kuzushiji MNIST

(c) MNIST

(d) CIFAR-10

Figure 22: Samples generated by the Dale-Langevin sampler (Algorithm 1) with annealing factor $\chi = 0.995$, and log-normal initialization.



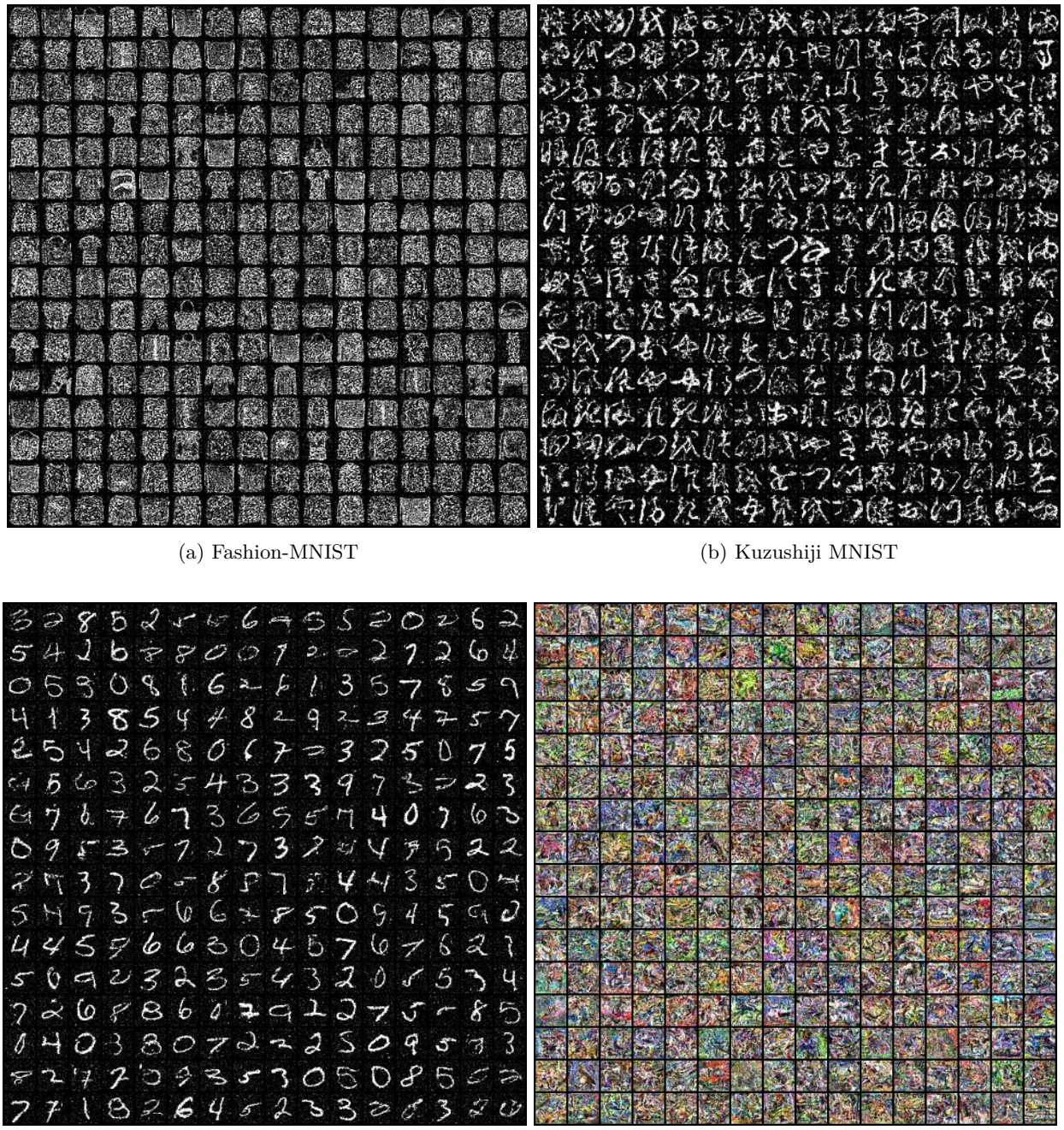
(a) Fashion-MNIST

(b) Kuzushiji MNIST

(c) MNIST

(d) CIFAR-10

Figure 23: Samples generated by the Dale-Langevin sampler (Algorithm 1) with annealing factor $\chi = 0.9995$, and log-normal initialization.



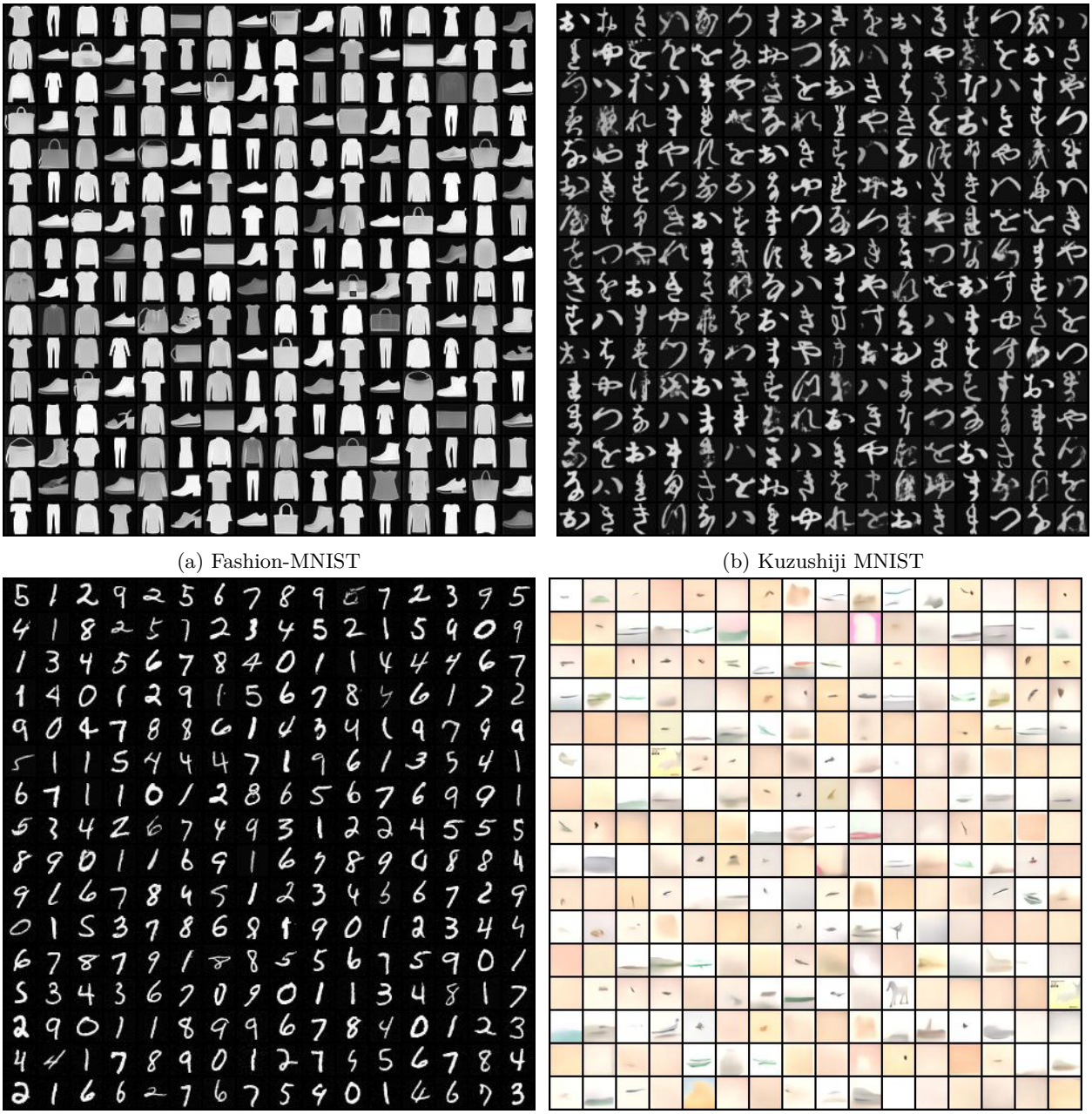
(a) Fashion-MNIST

(b) Kuzushiji MNIST

(c) MNIST

(d) CIFAR-10

Figure 24: Samples generated by the Dale-Langevin sampler (Algorithm 1) with annealing factor $\chi = 1.0$, and log-normal initialization.



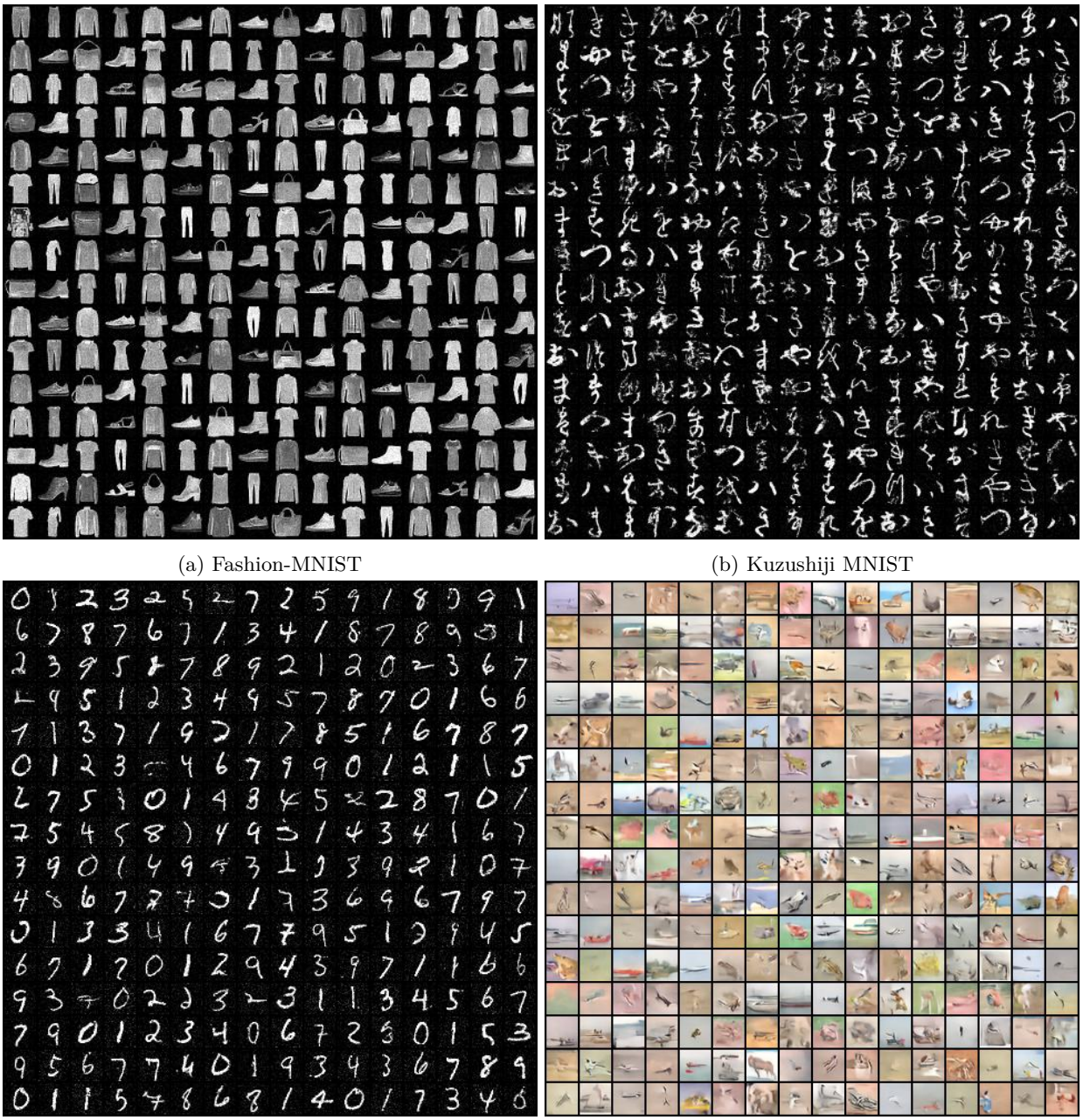
(a) Fashion-MNIST

(b) Kuzushiji MNIST

(c) MNIST

(d) CIFAR-10

Figure 25: Samples generated by the sign-agnostic multiplicative sampler Eq. (11) with annealing factor $\chi = 0.995$, and class-averaged initialization.



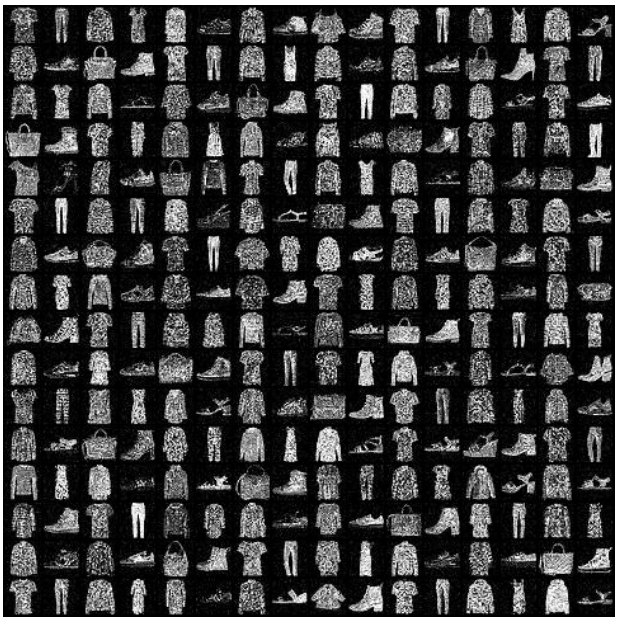
(a) Fashion-MNIST

(b) Kuzushiji MNIST

(c) MNIST

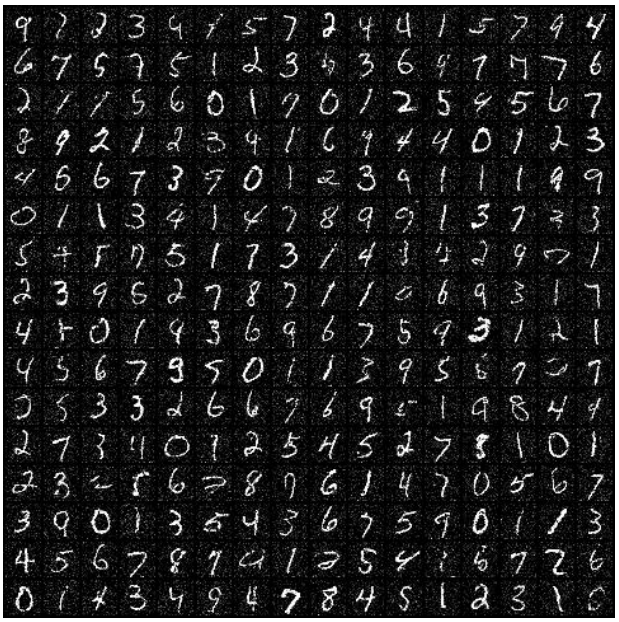
(d) CIFAR-10

Figure 26: Samples generated by the annealed sign-agnostic multiplicative sampler (Algorithm 2) with annealing factor $\chi = 0.9995$, and class-averaged initialization.

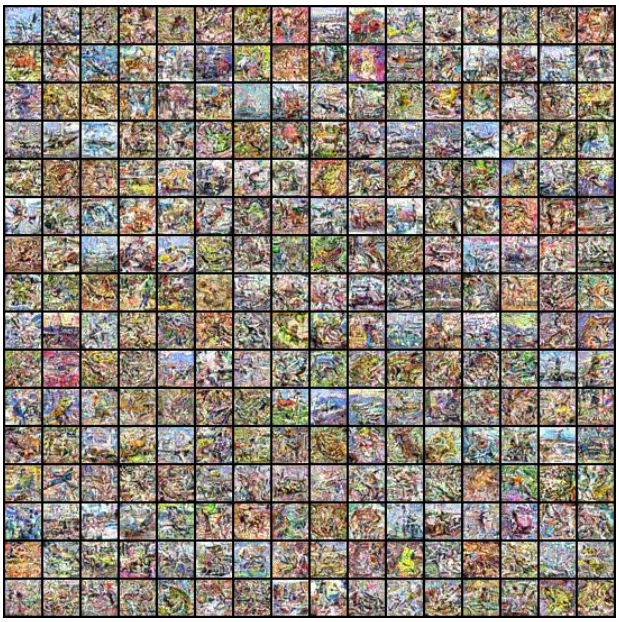


(a) Fashion-MNIST

(b) Kuzushiji MNIST

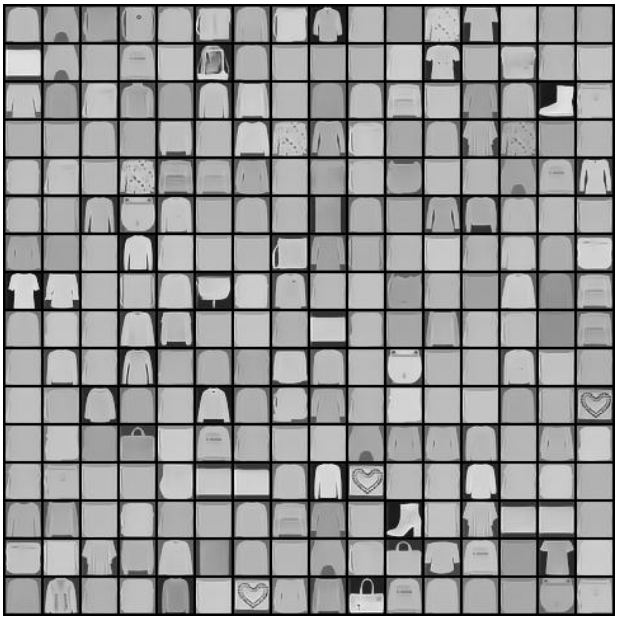


(c) MNIST



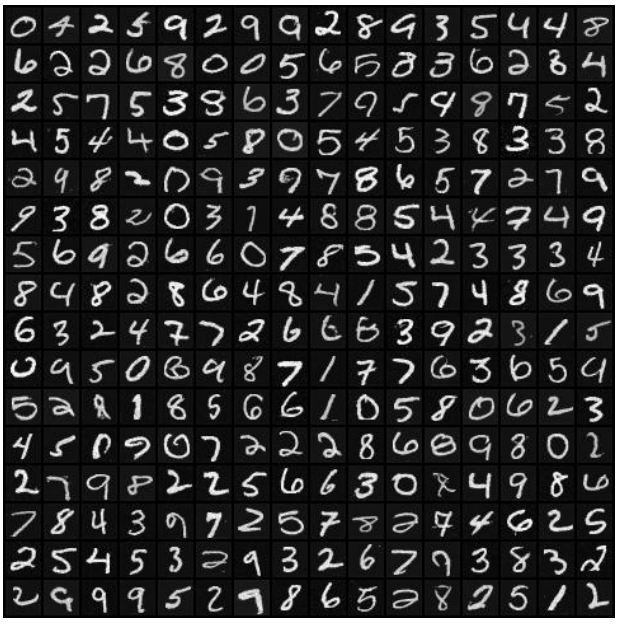
(d) CIFAR-10

Figure 27: Samples generated by the annealed sign-agnostic multiplicative sampler (Algorithm 2) with annealing factor $\chi = 1.0$, and class-averaged initialization.

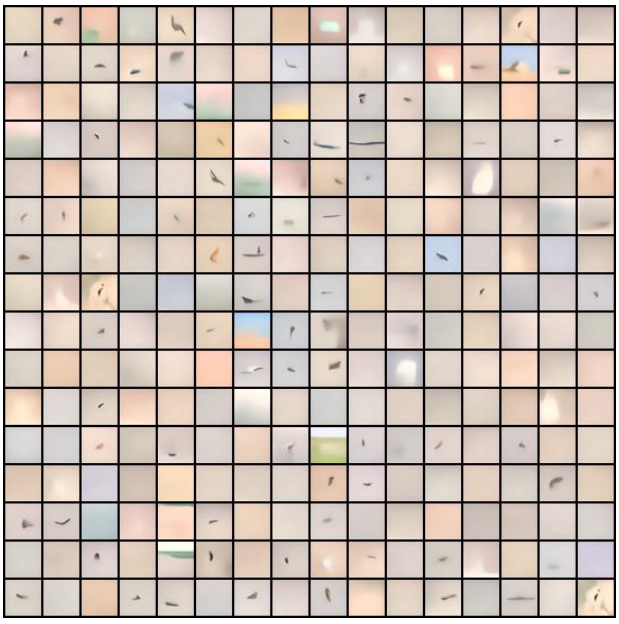


(a) Fashion-MNIST

(b) Kuzushiji MNIST

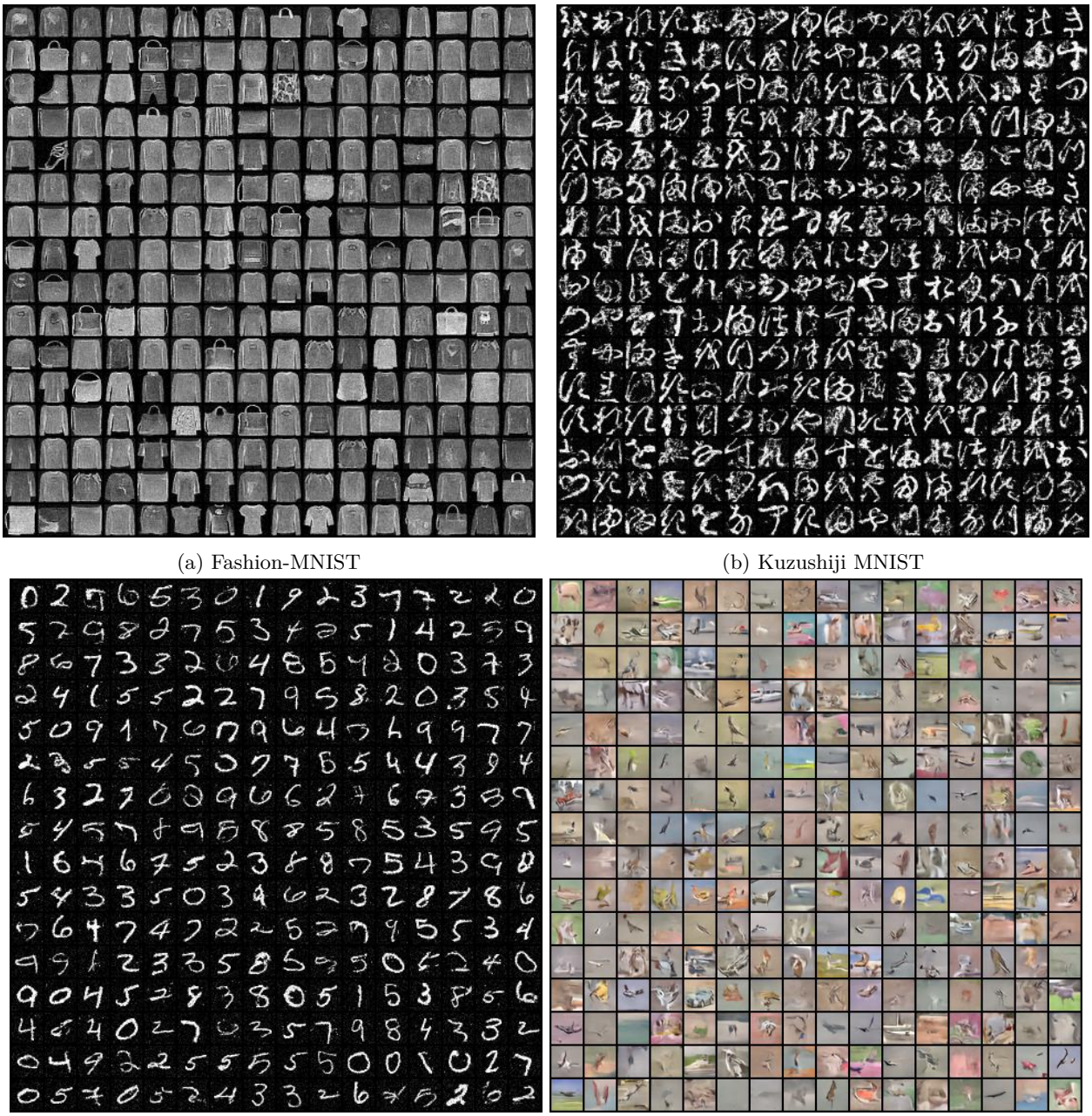


(c) MNIST



(d) CIFAR-10

Figure 28: Samples generated by the annealed sign-agnostic multiplicative sampler (Algorithm 2) with annealing factor $\chi = 0.995$, and log-normal initialization.



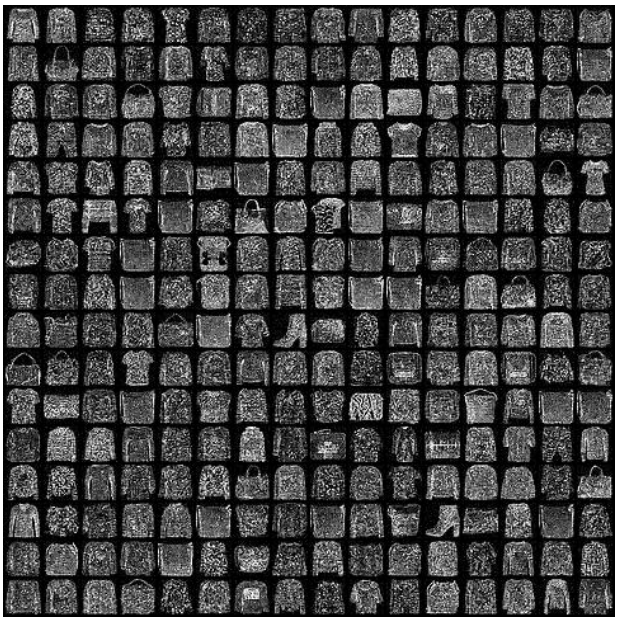
(a) Fashion-MNIST

(b) Kuzushiji MNIST

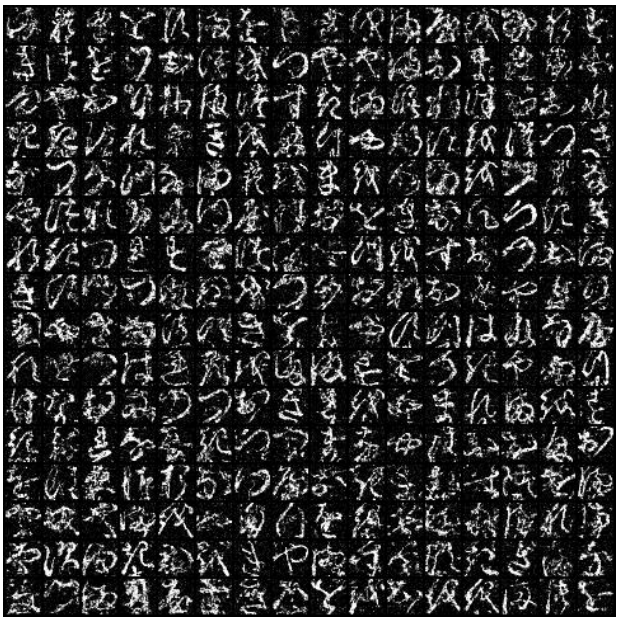
(c) MNIST

(d) CIFAR-10

Figure 29: Samples generated by the annealed sign-agnostic multiplicative sampler (Algorithm 2) with annealing factor $\chi = 0.9995$, and log-normal initialization.

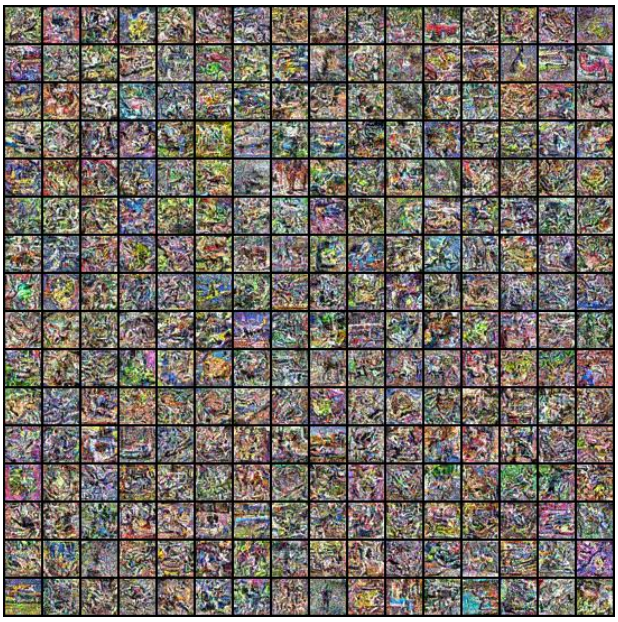


(a) Fashion-MNIST



(b) Kuzushiji MNIST

(c) MNIST



(d) CIFAR-10

Figure 30: Samples generated by the annealed sign-agnostic multiplicative sampler (Algorithm 2) with annealing factor $\chi = 1.0$, and log-normal initialization.

G Time-Varying Drift and Diffusion SDE

One could develop a generalization of the GBM framework with time-varying drift and diffusion terms. Consider the SDE

$$\mathrm{d}\log \boldsymbol{X}_t = \left(\boldsymbol{\mu}_t - \frac{\sigma_t^2}{2}\mathbf{1}\right)\mathrm{d}t + \sigma_t \mathrm{d}\boldsymbol{W}_t, \tag{34}$$

where the log operator is applied element-wise, $\boldsymbol{\mu}_t$ and σ_t are both functions of time. Euler-Maruyama discretization of Eq. 34 yields

$$\log \boldsymbol{X}_{k+1} = \log \boldsymbol{X}_k + \delta\left(\boldsymbol{\mu}_k - \frac{\sigma_k^2}{2}\mathbf{1}\right) + \sqrt{\delta}\sigma_k \boldsymbol{Z}_k, \tag{35}$$

where $\boldsymbol{Z}_k \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$, $\boldsymbol{\mu}_k \triangleq \boldsymbol{\mu}_{k\delta}$ and $\sigma_k \triangleq \sigma_{k\delta}$ for $k = 0, \ldots, N-1$. Unrolling Eq. 35, we get the following expression for $\boldsymbol{X}_k$ in terms of $\boldsymbol{X}_0$

$$\boldsymbol{X}_k = \boldsymbol{X}_0 \circ \exp\left(\delta\boldsymbol{\alpha}_k + \sqrt{\delta}\beta_k \boldsymbol{Z}_0\right), \tag{36}$$

where $\boldsymbol{\alpha}_k = \sum_{j=0}^{k}\left(\boldsymbol{\mu}_j - \frac{\sigma_j^2}{2}\mathbf{1}\right)$, $\beta_k = \sqrt{\sum_{j=0}^{k}\sigma_j^2}$, and $\boldsymbol{Z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$. This allows us to generate samples from the forward process used to train the score model that approximates the true score. Setting $\sigma_t = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t$ for $t \in [0, 1]$, or, equivalently, in the discretized version, $\sigma_j = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{\frac{j-1}{N-1}}$ for $j = 1, \ldots, N$, results in a variance-exploding SDE, which is the multiplicative counterpart of the additive version proposed by Song et al. (2021c). From Eq. (36), we see that $\boldsymbol{X}_k|\boldsymbol{X}_0 \sim \mathcal{LN}(\log \boldsymbol{X}_0 + \delta\boldsymbol{\alpha}_k, \delta\beta_k^2\mathbb{I})$, which serves as the target in the loss (Eq. (14)):

$$\boldsymbol{x}_k \circ \nabla \log p_{\boldsymbol{X}_k|\boldsymbol{X}_0}(\boldsymbol{x}_k|\boldsymbol{x}_0) = -\left(\mathbb{1} + \frac{1}{\delta\beta_k^2}\left(\log \boldsymbol{x}_k - \boldsymbol{x}_0\right)\right).$$

A corresponding reverse-time SDE for Eq. (34) is

$$\mathrm{d}\log \boldsymbol{X}_t = -\left(\boldsymbol{\mu}_t - \frac{3\sigma_t^2}{2}\mathbf{1} - \sigma_t^2 \boldsymbol{X}_t \circ \nabla \log p_{\boldsymbol{X}_t}(\boldsymbol{X}_t, t)\right)\mathrm{d}t + \sigma_t \mathrm{d}\boldsymbol{W}_t. \tag{37}$$

In principle, one could train a neural network to approximate the score, and generate new samples from the discretized version of the reverse-time SDE (Eq. (37)) with the learnt score function.