
Dale’s Law Meets Geometric Brownian Motion: Multiplicative Updates for Sampling Supplementary Material

Anonymous Author(s)

Affiliation

Address

email

1 Notation

Random variables are denoted in uppercase, and random vectors are denoted by boldface uppercase. Their realizations are denoted using corresponding lowercase letters. The probability density function (p.d.f.) of a random variable X is denoted by $p_X(x)$ and for the random vector \mathbf{X} , it is denoted by $p_{\mathbf{X}}(\mathbf{x})$. The Stein score of the random vector \mathbf{X} evaluated at \mathbf{x} is denoted by $\nabla \log p_{\mathbf{X}}(\mathbf{x})$.

2 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

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

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 $\text{Var}(W) = \exp(\sigma^2 - 1) \exp(2\mu + \sigma^2)$, respectively.

The multivariate log-normal random vector is defined as $\mathbf{W} = \exp(\boldsymbol{\mu} + \sigma \mathbf{Z})$ where $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ and the exponentiation is applied element-wise. Effectively, the entries of \mathbf{W} are independent and identically distributed according to Eq. (1). The corresponding density is denoted as $\mathcal{LN}(\boldsymbol{\mu}, \sigma^2 \mathbb{I})$.

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

Recall from Sec. 5 of the main document that the multiplicative explicit score-matching loss is given by

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

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

$$\mathcal{L}_{\text{M-DSM}}(\boldsymbol{\theta}) = \mathbb{E}_{\substack{\mathbf{X}_0 \sim p_{\mathbf{X}_0} \\ \mathbf{X}_t \sim p_{\mathbf{X}_t | \mathbf{X}_0}}} \left[\frac{1}{2} \left\| \mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t | \mathbf{X}_0}(\mathbf{X}_t | \mathbf{X}_0) - \mathbf{X}_t \circ s_{\boldsymbol{\theta}}(\mathbf{X}_t, t) \right\|_2^2 \right]. \quad (3)$$

21 In the following result, we establish the equivalence between multiplicative explicit score-matching
 22 and multiplicative denoising score-matching loss.

23 **Theorem 3.1** (Multiplicative Denoising Score-Matching). *Under standard assumptions on the*
 24 *density and the score function [Hyvärinen, 2005, Song et al., 2019] over the positive orthant*
 25 *\mathbb{R}_+^d , the multiplicative explicit score-matching (M-ESM) loss given in Eq. (2) and multiplicative*
 26 *denoising score-matching (M-DSM) loss given in Eq. (3) are equivalent up to a constant, i.e.,*
 27 $\mathcal{L}_{M-DSM}(\theta) = \mathcal{L}_{M-ESM}(\theta) + C$, *where C is independent of θ .*

28 *Proof.* We assume that the densities $p_{\mathbf{X}_t}$ and $p_{\mathbf{X}_t|\mathbf{X}_0}$ (defined in Sec. 4 of the main document) are
 29 supported over \mathbb{R}_+^d , and zero elsewhere. Further, we assume that $p_{\mathbf{X}_t}(\mathbf{x}_t) > 0, p_{\mathbf{X}_t|\mathbf{X}_0}(\mathbf{x}_t | \mathbf{x}_0) >$
 30 $0, \forall \mathbf{x}_t \in \mathbb{R}_+^d$ for $t \in [0, 1]$. The expectations are evaluated over the support \mathbb{R}_+^d . We expand
 31 $\mathcal{L}_{M-ESM}(\theta)$ to get

$$\begin{aligned} \mathcal{L}_{M-ESM}(\theta) = \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[\frac{1}{2} \left\| \mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t) \right\|^2 \right] + \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[\frac{1}{2} \left\| \mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t) \right\|^2 \right] \\ - \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[(\mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t))^{\top} (\mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t)) \right]. \end{aligned} \quad (4)$$

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

$$\begin{aligned} 35 \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[(\mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t))^{\top} (\mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t)) \right] \\ = \int (\mathbf{x}_t \circ \nabla \log p_{\mathbf{X}_t}(\mathbf{x}_t))^{\top} (\mathbf{x}_t \circ s_{\theta}(\mathbf{x}_t, t)) p_{\mathbf{X}_t}(\mathbf{x}_t) d\mathbf{x}_t \\ = \int (\mathbf{x}_t \circ \nabla p_{\mathbf{X}_t}(\mathbf{x}_t))^{\top} (\mathbf{x}_t \circ s_{\theta}(\mathbf{x}_t, t)) d\mathbf{x}_t. \end{aligned} \quad (5)$$

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

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

38 Computing the gradient with respect to \mathbf{x}_t on both sides yields

$$\nabla p_{\mathbf{X}_t}(\mathbf{x}_t) = \int \nabla p_{\mathbf{X}_t|\mathbf{X}_0}(\mathbf{x}_t|\mathbf{x}_0) p_{\mathbf{X}_0}(\mathbf{x}_0) d\mathbf{x}_0. \quad (6)$$

39 Substituting Eq. (6) in Eq. (5), multiplying and dividing by $p_{\mathbf{X}_t|\mathbf{X}_0}(\mathbf{x}_t|\mathbf{x}_0)$, we get

$$\begin{aligned} 40 \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[(\mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t))^{\top} (\mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t)) \right] \\ = \int \left(\mathbf{x}_t \circ \int \nabla p_{\mathbf{X}_t|\mathbf{X}_0}(\mathbf{x}_t|\mathbf{x}_0) p_{\mathbf{X}_0}(\mathbf{x}_0) d\mathbf{x}_0 \right)^{\top} (\mathbf{x}_t \circ s_{\theta}(\mathbf{x}_t, t)) d\mathbf{x}_t \\ = \int \int (\mathbf{x}_t \circ \nabla \log p_{\mathbf{X}_t|\mathbf{X}_0}(\mathbf{x}_t|\mathbf{x}_0))^{\top} (\mathbf{x}_t \circ s_{\theta}(\mathbf{x}_t, t)) p_{\mathbf{X}_t|\mathbf{X}_0}(\mathbf{x}_t|\mathbf{x}_0) p_{\mathbf{X}_0}(\mathbf{x}_0) d\mathbf{x}_0 d\mathbf{x}_t, \\ = \mathbb{E}_{\substack{\mathbf{X}_0 \sim p_{\mathbf{X}_0} \\ \mathbf{X}_t \sim p_{\mathbf{X}_t|\mathbf{X}_0}}} \left[(\mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t|\mathbf{X}_0}(\mathbf{X}_t|\mathbf{X}_0))^{\top} (\mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t)) \right]. \end{aligned} \quad (7)$$

41 Substituting Eq. (7) in Eq. (4) gives the following equivalent expression for the multiplicative explicit
 42 score-matching loss:

$$\begin{aligned}
 \mathcal{L}_{\text{M-ESM}}(\theta) &= \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[\frac{1}{2} \left\| \mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t}(\mathbf{X}_t) \right\|_2^2 \right] + \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[\frac{1}{2} \left\| \mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t) \right\|_2^2 \right] \\
 &\quad - \mathbb{E}_{\substack{\mathbf{X}_0 \sim p_{\mathbf{X}_0} \\ \mathbf{X}_t \sim p_{\mathbf{X}_t | \mathbf{X}_0}}} \left[(\mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t | \mathbf{X}_0}(\mathbf{X}_t | \mathbf{X}_0))^{\top} (\mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t)) \right] \\
 &= \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[\frac{1}{2} \left\| \mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t) \right\|_2^2 \right] \\
 &\quad - \mathbb{E}_{\substack{\mathbf{X}_0 \sim p_{\mathbf{X}_0} \\ \mathbf{X}_t \sim p_{\mathbf{X}_t | \mathbf{X}_0}}} \left[(\mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t | \mathbf{X}_0}(\mathbf{X}_t | \mathbf{X}_0))^{\top} (\mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t)) \right] + C_1, \quad (8)
 \end{aligned}$$

43 where C_1 is a constant that is not dependent on θ .

44 We carry out a similar simplification for the multiplicative denoising score-matching loss:

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

45 or equivalently,

$$\begin{aligned}
 \mathcal{L}_{\text{M-DSM}}(\theta) &= \mathbb{E}_{\mathbf{X}_t \sim p_{\mathbf{X}_t}} \left[\frac{1}{2} \left\| \mathbf{X}_t \circ s_{\theta}(\mathbf{X}_t, t) \right\|_2^2 \right] \\
 &\quad - \mathbb{E}_{\substack{\mathbf{X}_0 \sim p_{\mathbf{X}_0} \\ \mathbf{X}_t \sim p_{\mathbf{X}_t | \mathbf{X}_0}}} \left[(\mathbf{X}_t \circ \nabla \log p_{\mathbf{X}_t | \mathbf{X}_0}(\mathbf{X}_t | \mathbf{X}_0))^{\top} (s_{\theta}(\mathbf{X}_t, t) \circ \mathbf{X}_t) \right] \\
 &\quad + C_2, \quad (9)
 \end{aligned}$$

46 where C_2 is a constant that is not dependent on θ .

47 On comparing Eq. (8) and Eq. (9), we get

$$\mathcal{L}_{\text{M-DSM}}(\theta) = \mathcal{L}_{\text{M-ESM}}(\theta) + C_2 - C_1. \quad (10)$$

48 This concludes the proof. \square

49 The implication of the result is as follows: multiplicative explicit score-matching loss is intractable
 50 since we do not have access to the true marginal scores, and, this equivalence allows us to optimize
 51 the score network parameters by minimizing the multiplicative denoising score-matching loss since
 52 the conditional scores can be tractably computed from the forward SDE (cf. Sec. 4).

53 4 Additional Experimental Results

54 4.1 Architecture of the score network

55 The base architecture is the conditional RefineNet architecture [Song and Ermon, 2019] with dilated
 56 convolutions, specifically designed for image generation tasks. The network follows an encoder-
 57 decoder structure with skip connections and conditioning is done through class labels using condi-
 58 tional normalization layers. We modify it to work for N time-steps because we discretize the
 59 SDEs over N steps. The key components are the encoder and the decoder. The encoder starts with a
 60 convolutional layer (begin_conv), has multiple residual blocks organized in stages (res1-res5),

performs progressive downsampling through the network, and uses conditional residual blocks that incorporate class information. On the other hand, the decoder uses conditional refine blocks (refine1-refine5), incorporates skip connections from encoder layers and performs progressive upsampling and refines features.

4.2 Image datasets for evaluation

As mentioned in the main document, we evaluate the proposed model on the following datasets: MNIST, Fashion-MNIST and Kuzushiji-MNIST. The MNIST dataset consists of 70,000 images of handwritten digits, each of size 28×28 . The Fashion-MNIST dataset contains 70,000 images of clothing items, also of size 28×28 . Kuzushiji MNIST is a dataset of 70,000 images of handwritten Kuzushiji (cursive Japanese) characters, each of size 28×28 . The datasets are split into training and test sets, comprising 60,000 and 10,000 images, respectively.

4.3 Training details

We implemented the proposed model using PyTorch. For MNIST, the model is trained for 300k iterations, and for Fashion MNIST and Kuzushiji MNIST, the model is trained for 200k iterations. The chosen optimizer is AdamW optimizer [Loshchilov and Hutter, 2019]. The checkpoints are saved every 5k iterations as mentioned in [Song and Ermon, 2020]. The models are trained on two NVIDIA RTX 4090 and two NVIDIA A6000 GPUs. The model is trained using the Monte Carlo version of the score-matching loss defined in Eq. (3).

$$\hat{\mathcal{L}}_{\text{M-DSM}}(\theta) = \frac{1}{NM} \sum_{i=1}^M \sum_{k=0}^{N-1} \left[\frac{1}{2} \left\| \mathbf{x}_k^{(i)} \circ \nabla \log p_{\mathbf{X}_k | \mathbf{X}_0} \left(\mathbf{x}_k^{(i)} \mid \mathbf{x}_0^{(i)} \right) - \mathbf{x}_k^{(i)} \circ s_{\theta}(\mathbf{x}_k^{(i)}, k) \right\|_2^2 \right], \quad (11)$$

where $k = 0, \dots, N-1$ denotes the discretized time-step, and $i = 1, \dots, M$ denotes the index of the i^{th} sample. Effectively, we have M samples from the training dataset used in the score estimation over N time-steps.

4.4 Sampling algorithm

We observed that the sampler proposed in Algorithm 1 of the main document obtained by Euler-Maruyama discretization sometimes generates images of suboptimal quality. To mitigate this effect, we propose a slightly modified sampler with a step-size that is annealed by a factor $\chi < 1$ to progressively reduce the effect of noise during sampling, and L repeated sampling steps for each noise level. The modified sampler with the annealed step-size is listed in Algorithm 2. The modification improved the quality of the generated samples. Additionally, the step-size annealing can be viewed as a special case of operator splitting methods used in the discretization of SDEs [MacNamara and Strang, 2016]. For the initialization, we must draw a sample \mathbf{X}_{N-1} from the log-normal density, whose parameters $\hat{\boldsymbol{\mu}}, \hat{\sigma}$ are obtained by fitting a log-normal density to the histogram of pixel intensities of the samples at the end of the forward process.

Algorithm 2 Annealed multiplicative updates for generation using Geometric Brownian Motion.

Require: $\sigma, \delta, \boldsymbol{\mu}, L, \kappa, \chi, \hat{\boldsymbol{\mu}}, \hat{\sigma}$, trained score network s_{θ}

```

 $\kappa = 1$ 
2:  $\mathbf{X}_{N-1} \sim \mathcal{LN}(\hat{\boldsymbol{\mu}}, \hat{\sigma}^2 \mathbb{I})$ 
   for  $k \leftarrow N-1$  to 1 do
4:   for  $j \leftarrow 1$  to  $L$  do
        $\mathbf{Z}_{k,j} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ 
6:    $\mathbf{X}_{k-1} = \mathbf{X}_k \circ \exp \left( -\delta \left( \boldsymbol{\mu} - \frac{\sigma^2}{2} \mathbf{1} \right) + \delta \sigma^2 \mathbf{X}_k \circ s_{\theta}(\mathbf{X}_k, k) + \kappa \sigma \sqrt{\delta} \mathbf{Z}_{k,j} \right)$ 
       end for
8:    $\kappa \leftarrow \kappa \times \chi$ 
   end for
```

92

93 In order to simplify the update, we choose $\boldsymbol{\mu} = \frac{\sigma^2}{2} \mathbf{1}$. We found out empirically that $\sigma = 0.8$,
 94 $\chi = 0.995$ and $L = 3$, $\delta = 2 \times 10^{-4}$ gave the best results.

5 Generated Samples

We present samples generated by the proposed model on MNIST, Fashion MNIST and Kuzushiji MNIST datasets in Figs. 1 to 3. The samples are generated using the trained model and the sampling algorithm described in Algorithm 2. We observe that the generated samples are diverse and resemble the training data. They are also noise-free, which goes to show that the annealed multiplicative sampling update is quite robust. There are some samples that are entirely novel and are not identical to the training data. This effect is more pronounced in MNIST and Kuzushiji MNIST datasets. Samples from the Fashion MNIST dataset are less diverse and seem to have latched on to certain modes of the training data. This is by no means evidence of mode collapse but certain classes are underrepresented in the generation. This is probably because the Fashion MNIST dataset is more complex and has more variability in the images compared to MNIST and Kuzushiji MNIST. Understanding the reason behind this phenomenon requires further investigation.

5.1 MNIST



Figure 1: The samples have high diversity and the model even generates samples that are not present in the training data but have semantic similarity to the training data.



Figure 2: Generated Kuzushiji samples. The generated samples are sufficiently diverse and sharp and distinct from the training data.

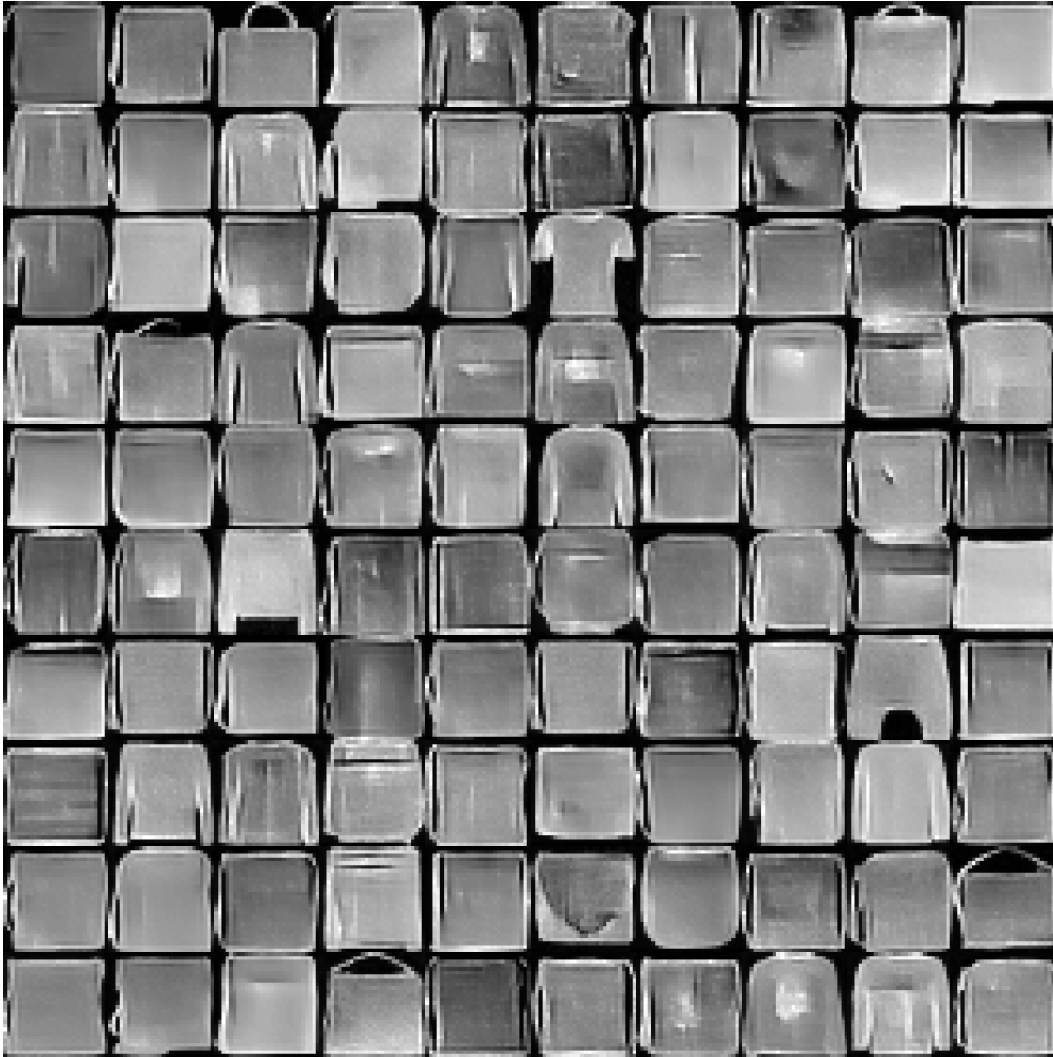


Figure 3: Generated Fashion MNIST samples. We observe less diversity of the generated samples here compared to MNIST and Kuzushiji MNIST possibly due to the complexity of the training data.

110 6 Evaluation Metrics for the Generated Images

111 We use the following metrics to evaluate the quality of the generated images:

- 112 • **Fréchet Inception Distance (FID)** [Heusel et al., 2017], which measures the distance
113 between the distribution of generated images and real images in the feature space of a
114 pre-trained InceptionV3 network [Szegedy et al., 2015]. Lower values indicate better
115 quality.
- 116 • **Kernel Inception Distance (KID)** [Bińkowski et al., 2018], which is similar to FID, but
117 uses a kernel to measure the distance between distributions. It is less sensitive to outliers
118 and is more robust for small sample sizes.
- 119 • **Nearest neighbours from training data**, which is a qualitative measure of how closely the
120 generated samples resemble the training data and to rule out the possibility of memorization
121 of the training samples. The nearest neighbours are identified by measuring the Euclidean

distance between generated samples and images from the training data with distances measured both in the pixel space and InceptionV3 feature space.

6.1 FID and KID

We compute the FID and KID scores using the `torchval` library and `torchmetrics` library for 50k generated samples and 50k real samples from the test set. This is done for grayscale images by repeating the image across the three colour channels and resizing it to 229×229 to match the input dimension expected by the InceptionV3 network. We report the best FID and KID scores obtained in Table 1. We observe that the FID and KID scores are lower for MNIST compared to Kuzushiji MNIST and Fashion MNIST. This is because MNIST is a relatively simpler dataset with less variability compared to Kuzushiji MNIST and Fashion MNIST. The FID and KID scores are higher for Fashion MNIST compared to MNIST, indicating that the generated samples are of lower quality and less diversity as evidenced by the samples in Fig. 3.

Table 1: FID and KID scores for the samples generated using the proposed model. The scores are computed using 50k generated samples and 10k real samples from the test set.

Dataset	FID	KID
MNIST	28.9616	0.0287 ± 0.0015
Fashion MNIST	116.1499	0.4374 ± 0.0044
Kuzushiji MNIST	50.7832	0.0546 ± 0.0021

On an absolute scale, the FID and KID scores obtained are below par that of the state-of-the-art diffusion models, which have evolved significantly over the past decade. However, considering that this is the first-ever model founded on geometric Brownian motion, Dale’s law, and multiplicative updates, the FID and KID scores obtained are definitely encouraging and have a lot of scope for improvement in subsequent work. We have also addressed possible future directions in the main document with respect to applying the proposed model on high-resolution image data.

6.2 Nearest neighbours

We identify the 10 nearest neighbours from the training data using the Euclidean distance between the generated samples and the training samples. The results are displayed in Figs. 4 to 9 of this document. We observe that the generated samples are semantically similar to the training samples, but not identical. This indicates that the model has the capability to generate diverse samples following the underlying distribution and that it does not memorize the training data. The nearest neighbours corresponding to both the pixel space and InceptionV3 feature space are shown in the figures.

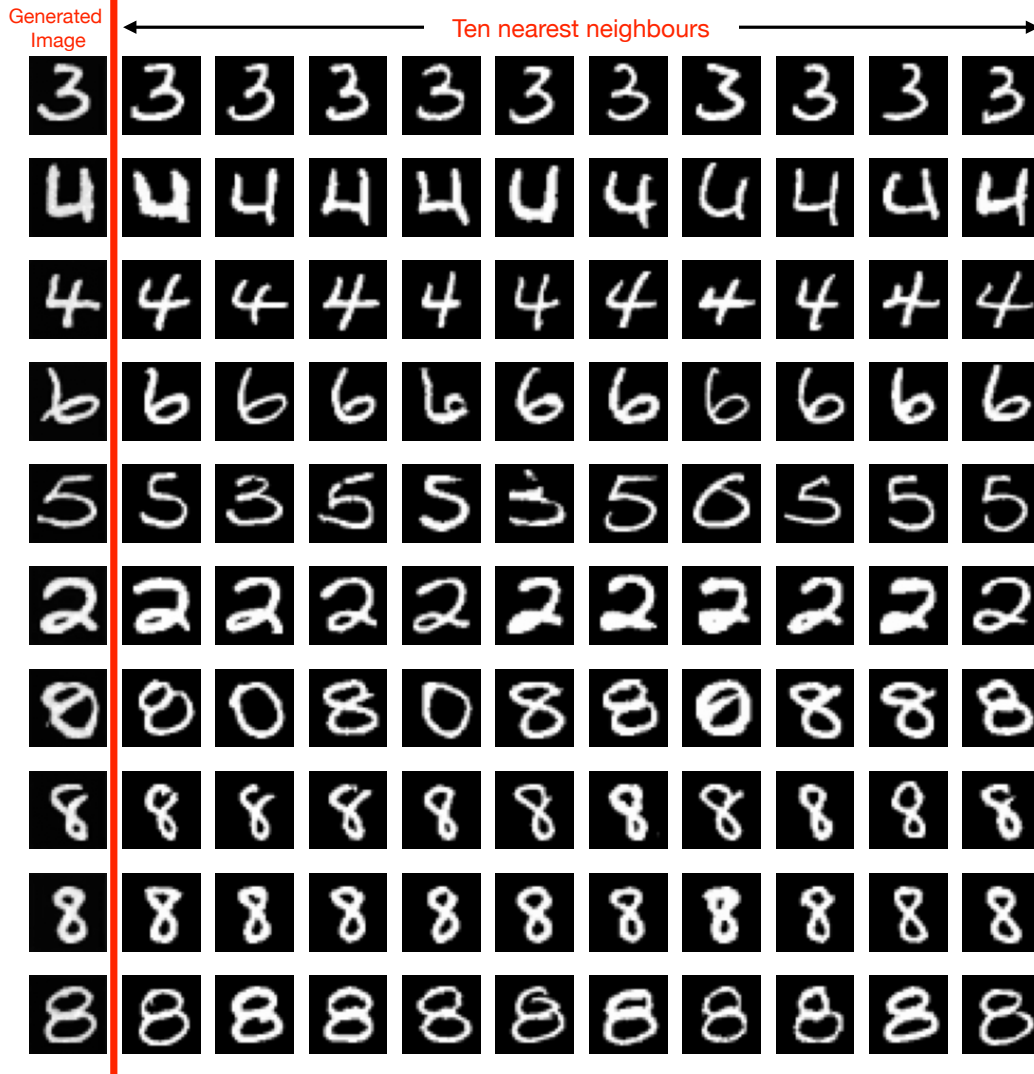


Figure 4: 10 nearest neighbours (calculated using Euclidean distance on raw images) from MNIST training data for samples generated using the proposed model. The last four rows show different instances of the digit 8, which are quite diverse. Similarly, the two instances of the digit 4 generated are visually quite different. These results show that there is enough diversity in the generated samples and no mode collapse whatsoever. This stands testimony to the robustness of the proposed multiplicative denoising score-matching framework.

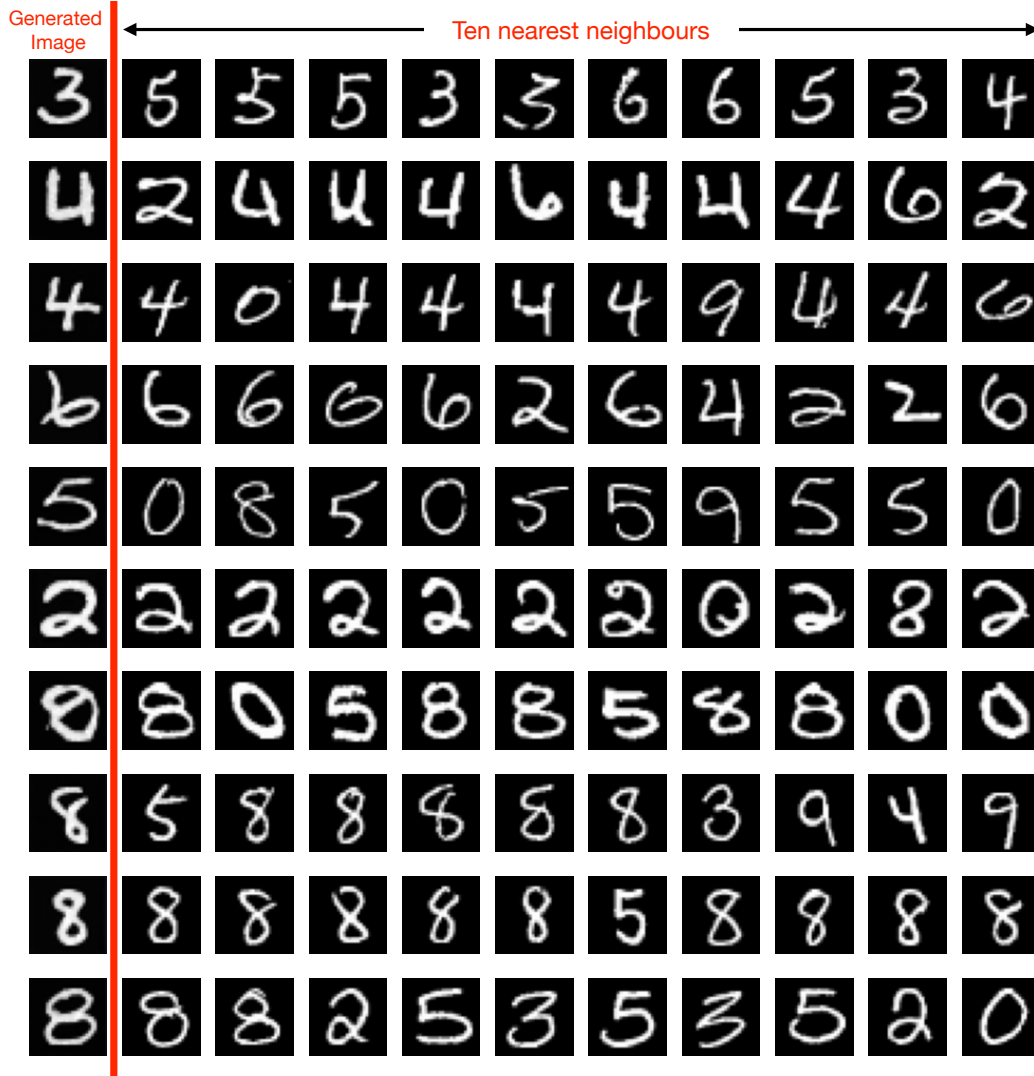


Figure 5: 10 nearest neighbours (calculated using Euclidean distance on InceptionV3 features) from the training data for samples generated. As mentioned in the caption of Fig. 4, there is sufficient diversity in the generated images. The nearest neighbours identified in the InceptionV3 space are not always semantically similar to the generated digit. For example, instances of digits 0 and 6 show up in the ten nearest neighbours of digit 4.



Figure 6: 10 nearest neighbours (calculated using Euclidean distance on raw images) from the training data for samples generated. Here, again, we observe sufficient diversity of the generated characters and semantic similarity with the top 10 nearest neighbours.



Figure 7: 10 nearest neighbours (calculated using Euclidean distance on InceptionV3 features) from the training data for samples generated.

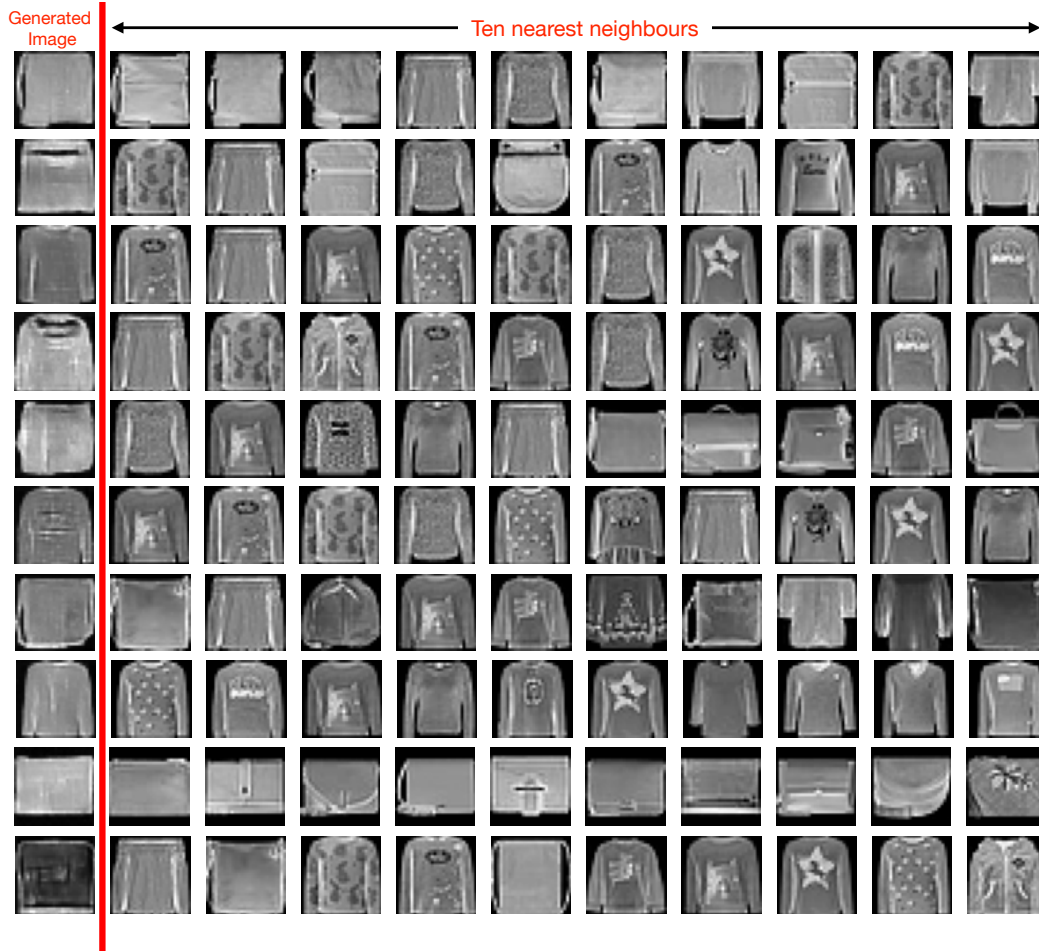


Figure 8: 10 nearest neighbours (calculated using Euclidean distance on raw images) from the training data for samples generated. Compared to MNIST and Kuzushiji MNIST, these samples have less diversity and seem to focus on specific modes (although not collapsing on the mode) in the underlying data distribution.

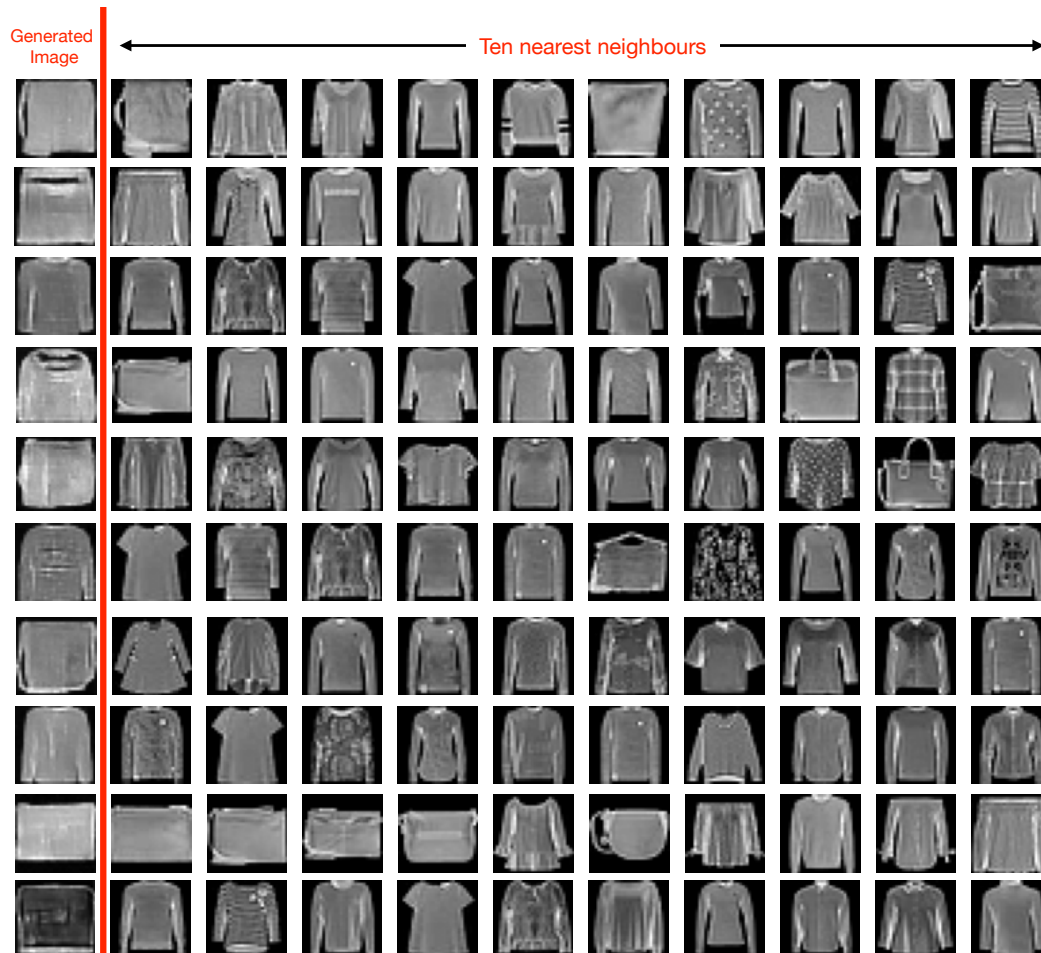


Figure 9: 10 nearest neighbours (calculated using Euclidean distance on InceptionV3 features) from the training data for samples generated.

References

- M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=r1lUOzWCW>.
- M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.
- A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24), 2005. URL <http://jmlr.org/papers/v6/hyvarinen05a.html>.
- I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *9th International Conference on Learning Representations, ICLR*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- S. MacNamara and G. Strang. *Operator Splitting*, pages 95–114. Springer International Publishing, Cham, 2016. ISBN 978-3-319-41589-5. doi: 10.1007/978-3-319-41589-5_3. URL https://doi.org/10.1007/978-3-319-41589-5_3.
- Y. Song and S. 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, editors, *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.
- Y. Song and S. Ermon. Improved techniques for training score-based generative models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12438–12448. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf.
- Y. Song, S. Garg, J. Shi, and S. Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI*, 2019. URL <http://auai.org/uai2019/proceedings/papers/204.pdf>.
- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society. doi: 10.1109/CVPR.2015.7298594. URL <https://doi.ieeeecomputersociety.org/10.1109/CVPR.2015.7298594>.