Sparsely Supervised Diffusion

Wenshuai Zhao[1,4] **Zhiyuan Li**[1] **Yi Zhao**[2] **Mohammad Hassan Vali**[1,4]
Martin Trapp[3] **Joni Pajarinen**[1] **Juho Kannala**[1,5] **Arno Solin**[1,4]

[1]Aalto University, Finland [2]Imperial College London, UK
[3]KTH Royal Institute of Technology, Sweden [4]ELLIS Institute Finland
[5]University of Oulu, Finland

Project website: https://aaltoml.github.io/ssd/

Abstract

Diffusion models have shown remarkable success across a wide range of generative tasks. However, they often suffer from spatially inconsistent generation, arguably due to excessive correlations learned by the model. This can produce samples that are locally plausible but globally inconsistent. We propose a principled method to mitigate this issue. Our method, sparsely supervised diffusion (SSD), is a simple yet effective masking framework that can be implemented with only a few lines of code. We analytically show that SSD fundamentally alters diffusion model training by modifying the spectrum of the data covariance and that it suppresses correlations in the data covariance matrix. Experiments show that our method more accurately approximates the underlying population score function, reduces memorization on small datasets, and promotes the use of essential contextual information during generation. Moreover, even when up to 98% of pixels are masked, it achieves competitive FID scores across a range of datasets and, importantly, avoids the training instability commonly observed on small datasets.

1 Introduction

Generative modeling aims to approximate and sample from typically unknown data distributions [1]. Among the various frameworks proposed [2–5], diffusion models [6, 7] including flow matching [8–10, 1] have achieved remarkable success across diverse domains [11–15], largely attributable to their simple regression-based training objective, such as to regress the injected noise or score function [16, 17, 6, 18]. However, diffusion models suffer from generating unrealistic samples even when only trained on natural data [19], such as the well-known six-finger hand generation problem [20]. This might be problematic in critical domains such as health [21] and robotics [22].

This spatially inconsistent generation can arguably be attributed to spurious correlations learned from finite training sets [23]. Specifically, the score-matching objective used to train diffusion models relies on the empirical score function derived from limited data, whereas the ideal objective would target the underlying continuous population score function [24–26]. This ob-

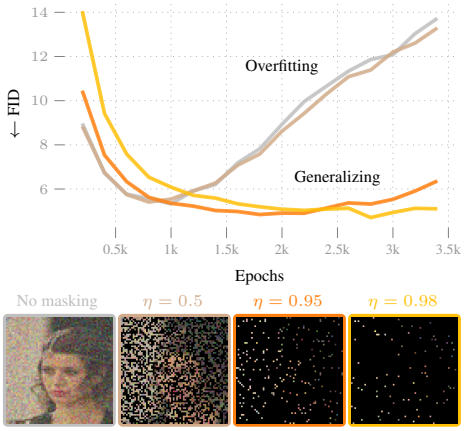


Figure 1: **Masking up to 98% of pixels improves training**. SSD learns exclusively from unmasked pixels while being encouraged to generalize over the masked regions.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

jective mismatch leads the model prone to memorization [27, 23], causing it to generate samples that merely recombine training samples rather than faithfully sampling from the true target distribution. While prior studies have investigated this phenomenon [28] through the lenses of locality [29, 25], spectral bias [30, 31], and implicit dynamical regularization [23, 32], existing solutions remain largely heuristic. Although domain-specific workarounds have been proposed [20], to the best of our knowledge, a principled methodology for mitigating spatial inconsistencies in generation has yet to be established.

We introduce *Sparsely Supervised Diffusion* (SSD), a principled framework designed to fundamentally alter the standard training dynamics of diffusion models. SSD mitigates spatial inconsistency by suppressing spurious correlations in the data covariance matrix. During training, we apply random spatial masks when computing the regression loss, which forces the model to learn solely from unmasked pixels and inherently generalize to the masked regions. Notably, SSD is entirely model-agnostic and can be integrated with just a few lines of code.

Our primary contributions are summarized as follows: *(i)* We analytically show that SSD modifies the spectrum of the data covariance matrix and functions as an *Occam's razor* over data correlations. *(ii)* Because the widely used Fréchet Inception Distance (FID) score cannot adequately reflect the spatial consistency of the generated samples, we design a targeted set of experiments. These evaluations show that SSD generates samples with better spatial consistency, more accurately approximates the underlying population score function, and better captures essential contextual information. *(iii)* Large-scale experiments also show that SSD achieves competitive FID scores even when up to 98% pixels are masked. Furthermore, it mitigates memorization and training instability on small datasets.

2 Related work

We first discuss the underexplored spatially inconsistent generation problem in diffusion models, then survey masking strategies employed in diffusion models for different targets, and finally introduce work on analyzing the training dynamics of diffusion models.

Spatial inconsistency and memorization in diffusion models Despite the remarkable success of diffusion [6, 18] and flow matching models [8, 1] in image and video generation [12, 11], they suffer from spatially inconsistent generation problems [28]. Moreover, memorization can emerge when the training dataset is small [24]. These problems come from the fundamental mechanism inherent in diffusion models [24, 26, 25]. When denoising a given pixel, the model relies on certain neighborhood of the input image, due either to network inductive biases [26] or excessive correlations learned by the model from finite samples [29]. Although there are some heuristic works targeting for realistic generation in specific domains [20, 33], no work addresses this problem in a principled way.

Masking in diffusion models We distinguish our method from existing masking techniques in diffusion models as we target for the underexplored spatial inconsistency problem. Discrete diffusion models incorporate masking as a replacement for Gaussian noise in continuous spaces, primarily to adapt diffusion to discrete domains such as text and code [34–36]. Masked Diffusion Transformer (MDT, [37]) exposes the model only to unmasked patches and trains it to predict the missing ones. However, MDT relies on a heavy asymmetric encoder–decoder design, akin to Masked Autoencoders (MAE, [38]), limiting its applicability to general diffusion frameworks. In contrast, our method introduces masking directly into the regression loss, making it architecture-agnostic and straightforward to implement. Ambient Diffusion [39, 40] passively trains diffusion models using corrupted data, similar as ours, but does not investigate the spatial inconsistency problem. Moreover, we provide an theoretical analysis to show how the proposed SSD changes the training dynamics of diffusion models.

Training dynamics of diffusion models Recent studies have sought to characterize the training dynamics of diffusion models to understand how they generalize to unseen data [23, 24]. Prior work has revealed a spectral bias in the training dynamics of diffusion models [30], wherein training progresses from a generalization regime to a memorization regime, corresponding to the sequential learning of generalizable low-frequency patterns followed by overfitting to individual data samples [23]. Within this line of work, the spectrum of the data covariance has emerged as a crucial factor governing diffusion training dynamics [31]. In this work, we analytically show that the proposed SSD fundamentally

modifies the spectrum of the data covariance, thereby altering the training dynamics of diffusion models and mitigating both excessive correlation-induced inconsistencies and memorization.

3 Background

We provide an overview of diffusion and flow matching models [1]. Particularly, we discuss how the empirical denoiser tends to memorize training samples.

3.1 Diffusion models

Given an unknown data distribution, instead of directly estimating the probability density $p(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^d$, diffusion models learn the score function [18] or denoiser [6] from noise-corrupted data to iteratively transform noises from a prior distribution to the target data distribution.

Forward process The forward process of diffusion models can be described via the following stochastic differential equation (SDE, see, *e.g.*, [41]):

$$\mathrm{d}\boldsymbol{z} = f(\boldsymbol{z}, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{w}, \tag{1}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ represents intermediate corrupted samples during the diffusion process, and $f(\boldsymbol{z}, t)$ and $g(t)$ are known as the drift and the diffusion functions. Typically, $\boldsymbol{w}(t)$ is assumed to be the standard Wiener process. For each timestep $t \in (0, T]$, we obtain the marginal distribution $p_t(\boldsymbol{z}) = \int p_t(\boldsymbol{z} \mid \boldsymbol{x}) p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ from Eq. (1). Generally, by setting proper $f(\boldsymbol{z}, t)$ and $g(t)$, we would like to have $p_t(\boldsymbol{z} \mid \boldsymbol{x})$ as a Gaussian distribution with closed-form mean and variance.

Backward process The core objective of diffusion models is to learn the time-reversal of Eq. (1). This reverse process is governed by the corresponding reverse-time SDE [18]:

$$\mathrm{d}\boldsymbol{z} = \left[f(\boldsymbol{z}, t) - g(t)^2 \nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{z})\right] \mathrm{d}t + g(t)\,\mathrm{d}\tilde{\boldsymbol{w}}. \tag{2}$$

The reverse SDE in Eq. (2) requires estimation of the score function $\nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{z})$. For the chosen diffusion process, the score function takes the explicit form [25]:

$$\nabla_{\boldsymbol{z}} \log p_t(\boldsymbol{z}) = \frac{\mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z}, t] - \boldsymbol{z}}{t^2}. \tag{3}$$

Note that the score estimation in Eq. (3) is mathematically equivalent to estimating the posterior mean $\mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z}, t]$, which is also the objective of denoising. Since the true data distribution $p(\boldsymbol{x})$ is generally unknown, exact computation of the posterior $p_t(\boldsymbol{x} \mid \boldsymbol{z})$ and hence $\mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z}, t]$ is intractable. Instead, diffusion models employ neural networks as denoisers to approximate $\mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z}, t]$. These networks are trained using an empirical data distribution $p_{\mathcal{D}}(\boldsymbol{x}) = \frac{1}{N} \sum_{\boldsymbol{x}^{(i)} \in \mathcal{D}} \delta(\boldsymbol{x} - \boldsymbol{x}^i)$, where the dataset is $\mathcal{D} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N \mid \boldsymbol{x}^i \sim p(\boldsymbol{x})\}$. The training objective is:

$$\mathbb{E}_{\boldsymbol{x}^i \sim p_{\mathcal{D}}(\boldsymbol{x}), \boldsymbol{z} \sim p_t(\boldsymbol{z} \mid \boldsymbol{x}^i), t \sim p(t)} \left[\lambda(t) \left\| \boldsymbol{x}^i - D_\theta(\boldsymbol{z}, t) \right\|^2\right], \tag{4}$$

where $\lambda(t)$ is a weighting function and $D_\theta(\boldsymbol{z}, t)$ represents the neural network denoiser.

Optimal denoiser The theoretical minimizer of Eq. (4) and the optimal denoiser for any $(\boldsymbol{z}, t)$ pair is the empirical posterior mean [42]:

$$\mathbb{E}_{\boldsymbol{x} \sim p_{\mathcal{D}}}[\boldsymbol{x} \mid \boldsymbol{z}, t] = \sum_{\boldsymbol{x}^i \in \mathcal{D}} p_t(\boldsymbol{x}^i \mid \boldsymbol{z}) \boldsymbol{x}^i, \tag{5}$$

which is the average over the images of the data set $\mathcal{D}$, weighted by their posterior probability. Note that the empirical optimal denoiser in Eq. (5) can only generate samples in the training dataset $\mathcal{D}$, *i.e.*, *the optimal denoiser memorizes* [26]. How Eqs. (2) and (3) result in *memorization* using the optimal denoiser in Eq. (5) has been well established in Biroli et al. [43].

3.2 Flow matching

Flow matching provides a unified perspective on diffusion models by directly learning a time-dependent vector field that transforms noisy data $\boldsymbol{z}$ into the target data distribution. This transformation admits multiple probability paths, such as diffusion paths and linear conditional paths, with

diffusion models arising as a special case. In this section, we focus on the objective of learning a linear conditional probability path:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,\pi(\boldsymbol{x}_0),p(\boldsymbol{x}_1)} \left\| \boldsymbol{v}_\theta(t, \boldsymbol{z} \,|\, \boldsymbol{x}_1) - (\boldsymbol{x}_1 - \boldsymbol{x}_0) \right\|^2, \tag{6}$$

where $\boldsymbol{x}_0 \in \pi(\boldsymbol{x}_0)$ denotes noise samples from a prior distribution, $\boldsymbol{x}_1 \in p(\boldsymbol{x}_1)$ are target data samples, and $\boldsymbol{z}$ is the intermediate state. By learning the vector field $\boldsymbol{v}_\theta(t, \boldsymbol{z})$, flow matching iteratively transforms noise samples into data samples through an ordinary differential equation

$$\boldsymbol{z}_{t+h} = \boldsymbol{z}_t + h\boldsymbol{v}_\theta(\boldsymbol{z}_t, t), \tag{7}$$

where $h > 0$ is a user-defined time step.

4 Methods

We first introduce the proposed *Sparsely Supervised Diffusion* (SSD). We then show that the proposed masking strategy alters the spectrum of the data covariance, thereby changing the learning dynamics of diffusion models. Finally, we analyze how the changed spectrum changes the locality mechanism in diffusion models.

4.1 Sparsely supervised diffusion (SSD)

To mitigate the spatially inconsistent generation problem in diffusion models, which arguably arises from the excessive exploitation of data correlations that may be biased by limited samples, we propose SSD, a simple yet effective masking strategy for training diffusion models. The masking strategy introduced in SSD is general and can be applied to both diffusion and flow-matching models. For clarity, we present it under the flow matching formulation:

$$\mathcal{L}_{\text{SSD}}(\boldsymbol{v}_\theta) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{m}\sim p(\boldsymbol{m})}\Big[\sum_{j \,\forall\, \boldsymbol{m}_j=1} \left\| \boldsymbol{v}_\theta(t, \boldsymbol{z}|\boldsymbol{x})[j] - \boldsymbol{v}^*(t, \boldsymbol{z}|\boldsymbol{x})[j] \right\|^2 \Big], \tag{8}$$

where $\boldsymbol{m} \in \{0,1\}^d$ denotes a randomly sampled binary mask with each entry $\boldsymbol{m}_j \sim \text{Bernoulli}(1-\eta)$, and d is the dimensionality of the image $\boldsymbol{x}$. $\boldsymbol{v}^*$ is the target velocity field. The mask ratio $\eta \in (0,1)$ is used to control the proportion of visible elements. Through masking, the objective $\mathcal{L}_{\text{SSD}}$ is optimized only over the unmasked pixel positions.

Remark. *By masking pixels in the target signal, SSD forces the model to rely on essential contextual information to generate complete pixels, as illustrated in Fig. 2. At the same time, it implicitly generates multiple variants of each sample, thereby mitigating memorization.*

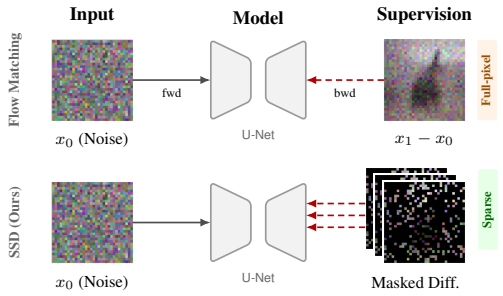


Figure 2: **Overview of the proposed method** in comparison with standard flow matching (FM).

4.2 Masking changes covariance spectrum

The spectrum of data covariance, *i.e.*, the set of eigenvalues, has been shown as the critical factor that affects the diffusion model training, such as convergence speed, separating generalization and memorization regimes and locality [31, 23, 29]. Here we analytically show that the proposed masking strategy alters the spectrum and thus changes the learning dynamics as evidenced in Sec. 5.

To simplify the analysis, we compare the covariance matrices $\boldsymbol{\Sigma}$ and $\tilde{\boldsymbol{\Sigma}}$, computed from original data $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ and masked data $\tilde{\boldsymbol{x}} = \boldsymbol{m} \odot \boldsymbol{x}$, where $\odot$ denotes element-wise multiplication. Denote the eigendecomposition of $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top, \qquad \boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d), \tag{9}$$

where $\mathbf{U}$ is the matrix of eigenvectors $\boldsymbol{u}_i$ and $\text{diag}(\lambda_1, \ldots, \lambda_d)$ represents the diagonal matrix formed by eigenvalues. We thus can compute the new covariance of masked data as

$$\tilde{\boldsymbol{\Sigma}} = (1-\eta)^2\boldsymbol{\Sigma} + \eta(1-\eta)\,\mathbf{D}, \tag{10}$$

where $\mathbf{D} = \text{diag}(\mathbf{\Sigma})$ represents the diagonal matrix formed from the diagonal entries of $\mathbf{\Sigma}$. The formulation of $\tilde{\mathbf{\Sigma}}$ reveals that the masking operation attenuates the off-diagonal elements of the covariance more strongly than the diagonal ones. In other words, **the resulting covariance matrix regularizes the inter-dimensional correlations present in the data**.

We project $\tilde{\mathbf{\Sigma}}$ onto one eigenvector $\boldsymbol{u}_i$ of $\mathbf{\Sigma}$,

$$\tilde{\lambda}_i := \boldsymbol{u}_i^\top \tilde{\mathbf{\Sigma}} \boldsymbol{u}_i = (1-\eta)^2 \lambda_i + \eta(1-\eta) \sum_k \boldsymbol{u}_{ik}^2 \mathbf{\Sigma}_{kk}. \tag{11}$$

The spectrum change can be measured by the ratio:

$$\beta_i := \frac{\tilde{\lambda}_i}{\lambda_i} = (1-\eta)^2 + \eta(1-\eta) \frac{\sum_k \boldsymbol{u}_{ik}^2 \mathbf{\Sigma}_{kk}}{\lambda_i}. \tag{12}$$

Note that $\sum_k \boldsymbol{u}_{ik}^2 \mathbf{\Sigma}_{kk} = \boldsymbol{u}_i^\top \mathbf{D} \boldsymbol{u}_i$, and $\lambda_i = \boldsymbol{u}_i^\top \mathbf{\Sigma} \boldsymbol{u}_i$. Thus,

$$\beta_i \propto \frac{\boldsymbol{u}_i^\top \mathbf{D} \boldsymbol{u}_i}{\boldsymbol{u}_i^\top \mathbf{\Sigma} \boldsymbol{u}_i}. \tag{13}$$

This equation indicates that the masking operation promotes the direction $\boldsymbol{u}_i$ more than the method without masking when β_i is large, and attenuates it when the ratio is small. A full derivation can be found in Sec. A.

Remark. *Intuitively, a large value of β_i implies that the coordinate-wise variations along direction $\boldsymbol{u}_i$ are strong (i.e., $\boldsymbol{u}_i^\top \mathbf{D} \boldsymbol{u}_i$ is large), while the overall data correlations captured by the full covariance matrix lead to a relatively small variance λ_i along this direction.*

4.3 Masking changes locality

As shown by Lukoianov et al. [29], the *locality* observed in diffusion models arises primarily from data statistics—specifically, the *correlations* present in the training dataset—rather than from the inductive bias of the neural network, contrary to earlier claims [26]. Building on this analysis, Eq. (10) shows that the proposed masking strategy suppresses excessive correlations (non-diagonal elements in $\tilde{\mathbf{\Sigma}}$) induced by limited data samples, thereby modifying the underlying mechanism that gives rise to locality in diffusion models.

Locality can be empirically quantified using the gradient sensitivity field during pixel generation, which characterizes the extent to which other pixel locations influence a given pixel in the generative process. Formally, assuming a locally linear denoiser $f(\boldsymbol{x}_t, t) = \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{b}_t$ and Gaussian data with covariance defined in Eq. (9), the gradient sensitivity at output pixel q is given by:

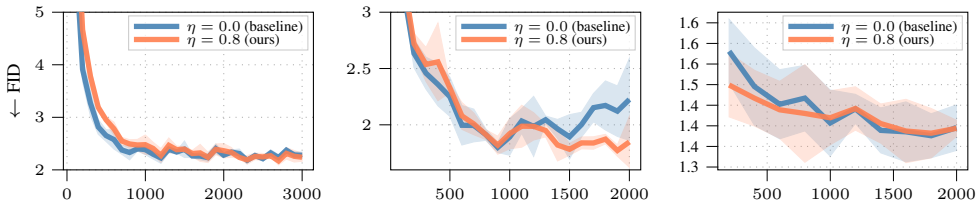
$$S_f^q(\boldsymbol{x}_t, t) = \left[\frac{\partial f(\boldsymbol{x}_t, t)}{\partial \boldsymbol{x}_t} \right]_q = \frac{1}{\sqrt{\alpha_t}} \left[\mathbf{U} \, \text{diag}\left(\frac{\text{SNR}_i}{\text{SNR}_i + 1} \right) \mathbf{U}^\top \right]_q,$$

where α_t is the noise level at timestep t. SNR_i represents the signal-to-noise ratio defined by $\text{SNR}_i = \frac{\lambda_i^2}{\sigma_t^2}$, for $i = \{1, 2, \ldots, d\}$, where σ_t^2 is the noise variance.

From Eq. (13) we can see that the sensitivity field is determined by the principal components of the covariance matrix and can be far from being local. As shown in Eq. (11), our method changes the eigenvalues of the data covariance and thereby changes locality. Intuitively, our method suppresses excessive correlations while promoting critical ones, effectively acting as an *Occam's razor* over data correlations [44]. As a result, the model exhibits stronger yet selective correlation structures, as evidenced by our experiments in Sec. 5.4.

5 Experiments

Overview We design a series of experiments to answer the following questions: Does the proposed SSD masking strategy **(Q1)** affect FID scores? **(Q2)** help to stabilize training on small datasets? **(Q3)** mitigate memorization in diffusion-based generation? **(Q4)** enable the model to capture more essential contextual information during generation? **(Q5)** improve the spatial consistency of generated samples? **(Q6)** facilitate more accurate estimation of the population score function?



(a) FIDs on CIFAR10 (b) FIDs on CelebA-50K (c) FIDs on LSUN Bedroom

Figure 3: **Effect of masking during training.** When $\eta = 0$, the model reduces to the baseline flow matching [49]. Across three datasets, SSD with up to 80% masked pixels can still achieve comparable performance as the baseline. Notably, the baseline in Fig. 3b eventually explodes while SSD remains stable. The shaded regions indicate the 95% confidence interval over four runs.



Figure 4: **Masking leads to more realistic generations.** Non-curated samples from models trained on CelebA. Sample #1 shows finer neck wrinkles at $\eta = 0.8$. In sample #6, an unrealistic bulge on the head at $\eta = 0$ gradually disappears as η increases. A similar artifact in sample #9 of the baseline is also removed by masking.

Data & experimental setup We evaluate the proposed method on four large-scale datasets: CIFAR10 (32×32) [45], CelebA-50K (64×64) [46], LSUN Bedroom (32×32) [47], and ImageNet (256×256) [48]. For ImageNet, SSD is implemented on DiT-XL/2 [7] at 256×256 resolution with the Stable Diffusion VAE (ft-MSE). We evaluate at a classifier-free guidance (CFG) scale of 1.5. Detailed training and evaluation settings can be found in Sec. C. For the remaining datasets, we adopt the flow matching framework (FM, [49]) with a convolutional U-Net architecture [50], using a fixed number of function evaluations (NFE) of 50 and Heun's second-order method [51] as the ODE solver, combined with the sampling strategy of Karras et al. [52]. Fréchet Inception Distance (FID) [53] is computed over 50k generated samples. In all experiments, we do not tune the masking ratio; instead, we evaluate a range of masking ratios when computational resources allow. Determining an optimal masking ratio is left for future work.

5.1 Masking achieves competitive FID scores (Q1)

We plot the evaluation set FID scores across training epochs to visualize the learning dynamics on three datasets, with curves averaged over four runs. Fig. 3 shows that models with an 80% masking ratio generally achieve lower comparable scores to the unmasked baselines. Notably, in Fig. 3b, the baseline FM model diverges after 1500 epochs, whereas our method with $\eta = 0.8$ remains stable. This phenomenon is further val-

Table 1: FID (σ) (↓) on different datasets.

Dataset	Baseline (FM/DiT)	SSD (ours)
CIFAR10	2.28 (0.04)	**2.24 (0.05)**
LSUN	1.40 (0.04)	**1.39 (0.01)**
CelebA-50K	2.23 (0.23)	**1.85 (0.14)**
CelebA-10K	13.72 (–)	**5.10 (–)**
ImageNet	4.98 (–)	**4.83 (–)**

idated in our CelebA-10K experiments as shown in Fig. 1, where the baseline exhibits even more severe divergence. We also report the final evaluation FID scores on all datasets in Table 1. For ImageNet, we set SSD with $\eta = 0.98$ and train the models for 180k steps. At CFG=1.5, SSD slightly improves full-ImageNet FID from 4.98 to 4.83. On a 100K-image subset, the improvement is larger: after 200K steps, FID decreases from 25.99 to 15.58 with $\eta = 0.98$ (Table 8). These results suggest that the benefit depends on both dataset size and training budget.

We visualize non-curated generated samples from different models in Fig. 4. The same noises are fed into each model to observe how the generated images look like over different mask ratios. While the generated images from the same noise generally appear similar, we observe improved spatial consistency as η increases. For instance, the unrealistic bulge on the head in sample #6 gradually diminishes as η increases. Moreover, sample #1 shows more detailed neck wrinkles with $\eta = 0.8$ compared to the image with $\eta = 0$.

Table 2: **Masking mitigates memorization.** FID (↓) and average ℓ_2 distance (↑) to the nearest training samples over 10k generated images on CelebA-10K. Larger ℓ_2 indicates less memorization.

Method	Baseline ($\eta = 0.0$)	Ours ($\eta = 0.98$)
FID (↓)	13.72	**5.10**
ℓ_2 distance (↑)	42.32	**46.02**

5.2 Masking mitigates training divergence (Q2)

We further compare SSD with the baseline on CelebA-10K, which contains 10k images. As shown in Fig. 1, SSD performs well even with masking ratios as high as 98%, and it effectively mitigates the severe divergence observed in baseline models trained for a long time without masking or with lower masking ratios such as $\eta = 0.5$.

5.3 Masking mitigates memorization (Q3)

We also observe that the proposed masking strategy can mitigate the memorization problem in diffusion model training, which is particularly critical for small datasets. We generate 10k images using models trained on CelebA-10K and compute the average of ℓ_2 distances to their nearest neighbors in the training set. The results in Table 2 show that, with $\eta = 0.98$, SSD achieves larger ℓ_2 distances compared to the baseline without masking, indicating that SSD generates images that are more distinct from the training samples, while maintaining comparable or even lower FID, as illustrated in Fig. 1.

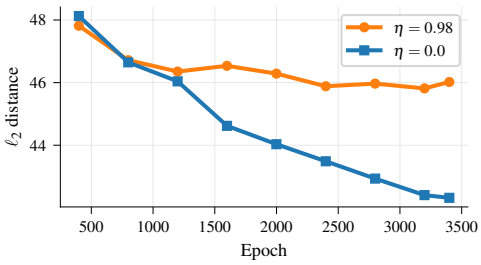


Figure 5: Average ℓ_2 distance to the nearest training samples over training iterations on CelebA-10K. Larger values indicate less memorization.

Fig. 5 shows the ℓ_2 distance evolution during training, demonstrating that SSD effectively mitigates memorization. Following Bonnaire et al. [23], we compute the memorization distance as $d_{\text{mem}} = |x_\tau - a^\mu|_2$, where x_τ denotes a generated sample and a^μ its nearest neighbor in the training set. Additional comparisons with dropout and MDT [37] are provided in Table 7 in the appendix.

5.4 Masking promotes essential contextual usage (Q4)

The gradient sensitivity [25] reflects how pixel positions are correlated when diffusion models generate pixels. For each timestep t, the gradient sensitivity heatmap is computed as

$$S(q,t) = \mathbb{E}_{\boldsymbol{z}\sim p_t(\boldsymbol{x}^{(i)},\boldsymbol{z})}\left[\sum_{c=1}^{3} |\nabla_{\boldsymbol{z}_c} \boldsymbol{v}_\theta(\boldsymbol{z},t)_{q,c}|\right], \tag{14}$$

where $\boldsymbol{v}_\theta(\boldsymbol{z},t)_{q,c}$ denotes the output of the vector field network at pixel position q and image channel c. In our experiments, we compute gradients at the central point of each image. A higher gradient usually means a stronger influence from the location when generating a pixel.

To assess how different pixel locations contribute to the generation of a target pixel, we compute the average gradient sensitivity over 10k CIFAR10 images. Specifically, we evaluate the ℓ_1 norm

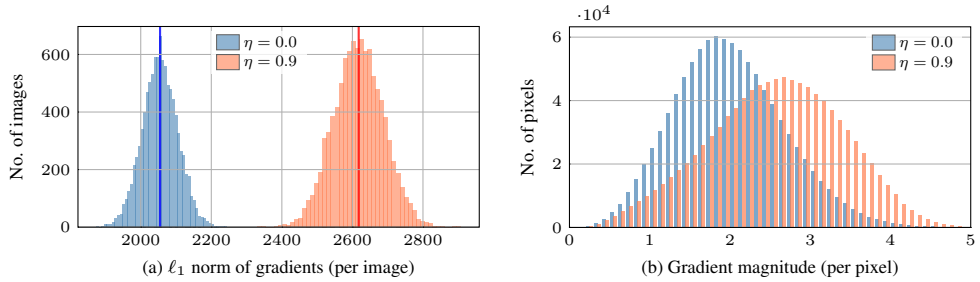


Figure 6: **Masking helps capturing essential contextual information.** (a) Distribution of ℓ_1 norm of gradients computed per image and, (b) distribution of gradient magnitude per pixel over 10k images for CIFAR10 dataset.

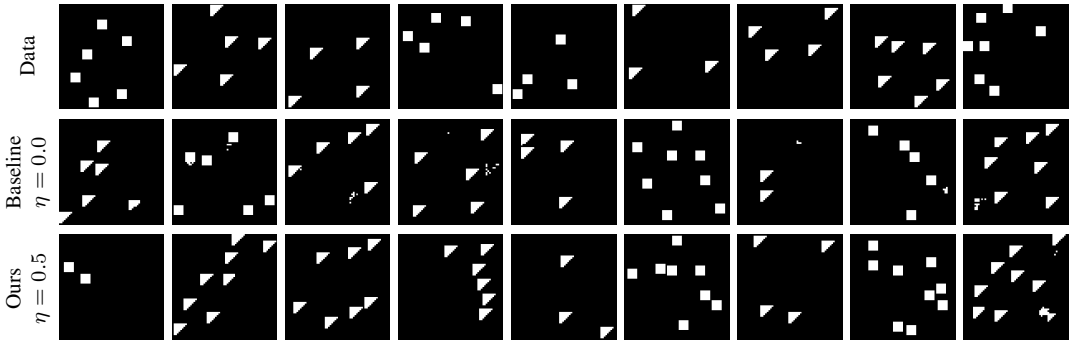


Figure 7: **Demonstration of improved spatial consistency.** After training for 5k epochs on a binary image dataset of triangles (◤) and squares (■) (*1st row*), our model generates images with less scattered dots (*3rd row*), *i.e.* enhanced structural integrity, compared to the standard diffusion model without masking (*2nd row*). Quantitative results are reported in Table 3.

of the gradients, $\|S(q,t)\|_1$, which measures the overall strength of correlations from all other pixel positions to a central pixel in an image. We set q as the central pixel location and $t = 0.789$ as a low noise level. To further examine how individual pixel positions contribute differently, we analyze the distribution of per-pixel gradient magnitudes.

As shown in Fig. 6 (b), the baseline method exhibits a narrow gradient-magnitude distribution, indicating that most pixels exert a similar level of influence. In contrast, our method produces a wider distribution, reflecting more diverse and differentiated contributions across pixel locations. This suggests that masking enables the model to distinguish between pixels and focus on essential correlations, whereas the baseline relies on broadly distributed correlations across most pixels, which can lead to unrealistic artifacts. Although our method captures correlations more selectively, Fig. 6 (a) shows that it exhibits substantially stronger correlations (higher ℓ_1 norm) with specific pixels. Overall, these results demonstrate that our approach effectively captures essential correlations while suppressing unnecessary ones, consistent with prior observations on locality [29].

5.5 Masking improves spatial consistency (Q5)

We construct a toy dataset comprising 500 binary images containing randomly positioned squares and triangles. Fig. 7 presents generated samples from SSD ($\eta = 0.5$) and baseline FM ($\eta = 0$), alongside samples from the training dataset. The results show that, with masking, SSD produces fewer unstructured shapes, whereas the baseline FM model occasionally generates scattered dots in some images.

To quantify spatial consistency, we count the scattered pixel clusters, defined as pixel clusters

Table 3: **Masking improves spatial consistency**. The table shows the number of scattered pixel clusters and number of images containing such clusters over 5k generated images for the illustrative dataset shown in Fig. 7.

Method	No. of scatters (↓)	No. of inconsistent images (↓)
Baseline	11,208	3,899
Ours	**1,014**	**792**

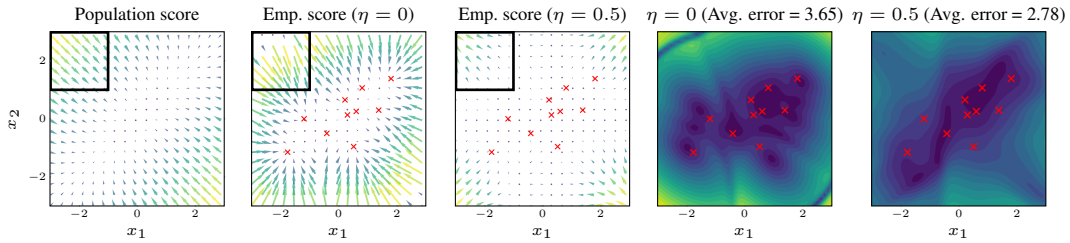


Figure 8: **Masking improves estimating the population score function**. Visualization of population scores (ground truth) and empirical scores estimated with $\eta = 0$ (baseline) and $\eta = 0.5$ (ours), along with a comparison of score estimation errors (right two figures) such that the error is color coded by 0 10.63. The baseline model only accurately estimates scores near observed data points (shown as ×), whereas the proposed masking enables more accurate estimation over broader regions (squared areas) and yields lower overall errors.

smaller than both the triangle and square sizes, across 5k generated images for each method. Table 3 reports both the total number of scattered pixel clusters and the number of images containing such clusters. The results show that, by masking, SSD substantially improves spatial consistency, reducing the number of spatially inconsistent images from 3,899 to 792. Furthermore, the total number of scattered clusters decreases from 11,208 to 1,014, corresponding to a 11× reduction.

Spatial consistency on natural images We additionally evaluate on 11kHands [54] at 64×64 resolution. We train the baseline ($\eta = 0$) and SSD ($\eta = 0.8$) for 400 epochs and generate 1,000 samples per model using the same random seed. Counting four- and six-finger cases, SSD reduces the total from 35 to 22 as shown in Table 4. We also observe that the baseline model frequently generates hands with six fingers, while SSD mitigates this issue but occasionally produces hands with only four fingers. Qualitative samples are provided in Fig. 9. We hypothesize that by tuning the masking ratio, we can further improve the spatial consistency.

Table 4: Comparison of finger-count errors among 1,000 generated 11kHands samples.

Error type (↓)	Baseline ($\eta = 0$)	SSD ($\eta = 0.8$)
Six fingers	26	**8**
Four fingers	**9**	14
Total	35	**22**

5.6 Masking improves score function estimation (Q6)

FID is limited in its ability to assess how closely generated samples match the true data distribution. This limitation helps to explain why diffusion models with excellent FID scores can still produce unrealistic images, even when trained exclusively on natural images [23]. Fundamentally, this issue arises because the training objective estimates only the empirical score function rather than the true underlying score function, thereby leading to both memorization and unrealistic generations.

Table 5: Comparison of empirical score estimation errors on a 2D Gaussian data distribution. Lower (↓) is better.

Metric	Baseline	W/ Masking (ours)
Mean error (↓)	3.65	**2.78**
Max error (↓)	10.63	**6.18**

Since we generally do not know the ground truth score function of real datasets [6], we design experiments on synthesized 2D Gaussian data, where we can analytically compute the score function. We compare the approximated empirical score functions obtained by the baseline and our method under different masking ratios. Fig. 8 visualizes the estimated scores on a grid over the square, based on the given ten data points. By applying masking ($\eta = 0.5$), which generates multiple partial views of the data points, the estimation becomes more accurate over a broader region. The two figures on the right in Fig. 8 presents the estimation errors, defined as the absolute difference from the population score, with and without masking. As reported in Table 5, the masked estimation achieves an average error of 2.78, which is lower than the baseline's average error of 3.65. The improved accuracy arises from the mechanism whereby masking introduces an independence prior across data dimensions, thereby reducing the model's overreliance on misleading correlations induced by limited training samples. The detailed description of the numerical experiment is in Sec. B.

6 Discussion and conclusion

Although diffusion models can generate high-quality images, spatial inconsistency remains a significant challenge. We hypothesize that such unrealistic artifacts arise because excessive correlations present in training data are indiscriminately encoded by diffusion models. Motivated by this observation, we propose *sparsely supervised diffusion* that encourages the model to capture critical, but selective, contextual information when predicting unseen pixel locations. The analytical results show that masking fundamentally alters diffusion model training by modifying the spectrum of the data covariance and suppressing correlations in the data covariance matrix. Extensive experiments show that the proposed SSD offers a *free lunch* for diffusion model training, with several advantages: *(1)* it achieves comparable FID scores across datasets while avoiding long-training instability; *(2)* it mitigates memorization; *(3)* it promotes essential contextual awareness during sample generation; and *(4)* it reduces spatially inconsistent artifacts and yields a more accurate approximation of the population score function. We leave the investigation of the optimal masking ratio for future work.

Acknowledgments

We acknowledge support through Research Council of Finland grants 352788, 362407, 373999, 373780, 362408, and 339730. This research also received support through the Council's Flagship programme, the Finnish Center for Artificial Intelligence (FCAI). Computing resources were supplied by the Aalto Science-IT project. We acknowledge CSC – IT Center for Science, Finland, for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through CSC. We thank the anonymous reviewers for their constructive feedback and helpful suggestions. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[1] Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

[2] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NeurIPS)*, 2014.

[3] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[4] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1747–1756. PMLR, 2016.

[5] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 6840–6851, 2020.

[7] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023.

[8] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[9] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[10] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[11] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 23–40. Springer, 2024.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[13] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 36479–36494, 2022.

[14] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[15] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[16] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2256–2265. pmlr, 2015.

[17] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

[18] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[19] Dazhong Shen, Guanglu Song, Zeyue Xue, Fu-Yun Wang, and Yu Liu. Rethinking the spatial inconsistency in classifier-free diffusion guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9370–9379, 2024.

[20] Supreeth Narasimhaswamy, Uttaran Bhattacharya, Xiang Chen, Ishita Dasgupta, Saayan Mitra, and Minh Hoai. Handiffuser: Text-to-image generation with realistic hand appearances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2468–2479, 2024.

[21] Vijil Chenthamarakshan, Payel Das, Samuel Hoffman, Hendrik Strobelt, Inkit Padhi, Kar Wai Lim, Benjamin Hoover, Matteo Manica, Jannis Born, Teodoro Laino, et al. CogMol: Target-specific and selective drug design for covid-19 using deep generative models. *Advances in Neural Information Processing Systems*, 33:4320–4332, 2020.

[22] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[23] Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mezard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, 2025.

[24] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *International Conference on Learning Representations*, 2024.

[25] Matthew Niedoba, Berend Zwartsenberg, Kevin Patrick Murphy, and Frank Wood. Towards a mechanistic explanation of diffusion model generalization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

[26] Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2025.

[27] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

[28] Sumukh K Aithal, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation. *Advances in neural information processing systems*, 37: 134614–134644, 2024.

[29] Artem Lukoianov, Chenyang Yuan, Justin Solomon, and Vincent Sitzmann. Locality in image diffusion models emerges from data statistics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[30] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *International Conference on Learning Representations*, 2023.

[31] Binxu Wang and Cengiz Pehlevan. An analytical theory of spectral bias in the learning dynamics of diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[32] Ricardo Baptista, Agnimitra Dasgupta, Nikola B Kovachki, Assad Oberai, and Andrew M Stuart. Memorization and regularization in generative diffusion models. *arXiv preprint arXiv:2501.15785*, 2025.

[33] Haoran Lu, Shang Wu, Jianshu Zhang, Maojiang Su, Guo Ye, Chenwei Xu, Lie Lu, Pranav Maneriker, Fan Du, Manling Li, et al. Phys4d: Fine-grained physics-consistent 4d modeling from video diffusion. *arXiv preprint arXiv:2603.03485*, 2026.

[34] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 17981–17993, 2021.

[35] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pages 103131–103167, 2024.

[36] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pages 133345–133385, 2024.

[37] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23164–23173, 2023.

[38] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.

[39] Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alex Dimakis, and Adam Klivans. Ambient diffusion: Learning clean distributions from corrupted data. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pages 288–313, 2023.

[40] Giannis Daras, Adrian Rodriguez-Munoz, Adam Klivans, Antonio Torralba, and Constantinos Daskalakis. Ambient diffusion omni: Training good models with bad data. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, 2025.

[41] Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Cambridge University Press, 2019.

[42] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(110):3371–3408, 2010.

[43] Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

[44] Carl Rasmussen and Zoubin Ghahramani. Occam's razor. In *Advances in Neural Information Processing Systems 13 (NIPS)*, pages 294–300, 2000.

[45] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. MSc thesis, University of Toronto, 2009.

[46] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[47] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[48] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[49] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[51] Uri M Ascher and Linda R Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, 1998.

[52] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 26565–26577, 2022.

[53] 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 30 (NeurIPS)*, 2017.

[54] Mahmoud Afifi. 11K Hands: Gender recognition and biometric identification using a large dataset of hand images. *Multimedia Tools and Applications*, 78(15):20835–20854, 2019.

A Random masking changes spectrum of covariance

A.1 Setup

Consider data $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with covariance matrix $\boldsymbol{\Sigma}$. We apply random binary mask $\boldsymbol{m} \in \{0,1\}^d$ where each $\boldsymbol{m}_j \sim$ Bernoulli$(1-\eta)$ independently, with η being the probability of being masked (*i.e.*, $\mathbb{P}(\boldsymbol{m}_j = 0) = \eta$ and $\mathbb{P}(\boldsymbol{m}_j = 1) = 1 - \eta$). The masked data is:

$$\tilde{\boldsymbol{x}} = \boldsymbol{m} \odot \boldsymbol{x}, \tag{15}$$

where $\odot$ denotes element-wise multiplication.

A.2 Covariance of masked data

The covariance matrix is:

$$\tilde{\boldsymbol{\Sigma}} = \mathbb{E}[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top] - \mathbb{E}[\tilde{\boldsymbol{x}}]\mathbb{E}[\tilde{\boldsymbol{x}}]^\top = \mathbb{E}[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top].$$

The (j,k)-th element is:

$$[\tilde{\boldsymbol{\Sigma}}]_{jk} = \mathbb{E}[\tilde{\boldsymbol{x}}_j\tilde{\boldsymbol{x}}_k] = \mathbb{E}[\boldsymbol{m}_j\boldsymbol{m}_k\boldsymbol{x}_j\boldsymbol{x}_k].$$

Since $\boldsymbol{m}$ and $\boldsymbol{x}$ are independent:

$$[\tilde{\boldsymbol{\Sigma}}]_{jk} = \mathbb{E}[\boldsymbol{m}_j\boldsymbol{m}_k] \cdot \mathbb{E}[\boldsymbol{x}_j\boldsymbol{x}_k] = \mathbb{E}[\boldsymbol{m}_j\boldsymbol{m}_k] \cdot \boldsymbol{\Sigma}_{jk}.$$

For diagonal elements ($j = k$)

$$\mathbb{E}[\boldsymbol{m}_j^2] = \mathbb{E}[\boldsymbol{m}_j] = 1 - \eta.$$

Therefore:

$$[\tilde{\boldsymbol{\Sigma}}]_{jj} = (1-\eta)\boldsymbol{\Sigma}_{jj}. \tag{16}$$

For off-diagonal elements ($j \neq k$)

Since $\boldsymbol{m}_j$ and $\boldsymbol{m}_k$ are independent:

$$\mathbb{E}[\boldsymbol{m}_j\boldsymbol{m}_k] = \mathbb{E}[\boldsymbol{m}_j]\mathbb{E}[\boldsymbol{m}_k] = (1-\eta)^2.$$

Therefore:

$$[\tilde{\boldsymbol{\Sigma}}]_{jk} = (1-\eta)^2\boldsymbol{\Sigma}_{jk}.$$

Combining both cases:

$$\tilde{\boldsymbol{\Sigma}} = (1-\eta)\,\mathrm{diag}(\boldsymbol{\Sigma}) + (1-\eta)^2(\boldsymbol{\Sigma} - \mathrm{diag}(\boldsymbol{\Sigma})).$$

where $\mathrm{diag}(\boldsymbol{\Sigma})$ means the diagonal matrix formed from the diagonal entries of $\boldsymbol{\Sigma}$.

Or equivalently:

$$\boxed{\tilde{\boldsymbol{\Sigma}} = (1-\eta)^2\boldsymbol{\Sigma} + \eta(1-\eta)\,\mathrm{diag}(\boldsymbol{\Sigma})} \tag{17}$$

A.3 Masking affects spectrum of covariance

Let eigendecomposition of $\boldsymbol{\Sigma}$ be

$$\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top, \qquad \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d).$$

Project $\tilde{\boldsymbol{\Sigma}}$ onto one eigenvector $\boldsymbol{u}_i$ of $\boldsymbol{\Sigma}$,

$$\tilde{\lambda}_i := \boldsymbol{u}_i^\top\tilde{\boldsymbol{\Sigma}}\boldsymbol{u}_i = (1-\eta)^2\lambda_i + \eta(1-\eta)\sum_k \boldsymbol{u}_{ik}^2\boldsymbol{\Sigma}_{kk}. \tag{18}$$

Therefore, we can compute the eigenvalue shrinkage ratio:

$$\frac{\tilde{\lambda}_i}{\lambda_i} = (1-\eta)^2 + \eta(1-\eta)\frac{\sum_k \boldsymbol{u}_{ik}^2\boldsymbol{\Sigma}_{kk}}{\lambda_i}. \tag{19}$$

If $\mathbf{\Sigma}$ is *diagonal*, then $\boldsymbol{u}_i$ is a standard basis vector, so

$$\sum_k \boldsymbol{u}_{ik}^2 \mathbf{\Sigma}_{kk} = \lambda_i,$$

for the i-th component, giving

$$\frac{\tilde{\lambda}_i}{\lambda_i} = 1 - \eta,$$

which means that all eigenvalues are equally scaled down.

If $\mathbf{\Sigma}$ is *non-diagonal*, the term

$$\sum_k \boldsymbol{u}_{ik}^2 \mathbf{\Sigma}_{kk} = u_i^\top \mathbf{D} \boldsymbol{u}_i,$$

where $\mathbf{D}$ is $\text{diag}(\mathbf{\Sigma})$, is **not equal** to λ_i in general. This term mixes the diagonal entries of $\mathbf{\Sigma}$ weighted by the components of the eigenvector u_i.

We can thus simplify the ratio as

$$\boxed{\frac{\tilde{\lambda}_i}{\lambda_i} \propto \frac{u_i^\top \mathbf{D} u_i}{\lambda_i}.} \tag{20}$$

This expression indicates that the masking operation promotes the direction u_i when $\frac{u_i^\top \mathbf{D} u_i}{\lambda_i}$ is large, and attenuates it when the ratio is small. Intuitively, a large value of $\frac{u_i^\top \mathbf{D} u_i}{\lambda_i}$ implies that the coordinate-wise variations along direction u_i are strong (*i.e.*, $u_i^\top \mathbf{D} u_i$ is large), while the overall data correlations captured by the full covariance matrix lead to a relatively small variance λ_i along this direction.

B Description of 2D Gaussian experiment

This section describes the detailed setting of the score estimation experiment in Sec. 5.6, including the 2D Gaussian distribution we use, how we compute the population score, and how we compute the empirical scores with and without masking.

2D Gaussian distribution We consider a 2D Gaussian distribution as our data distribution:

$$p_0(x) = \mathcal{N}(x; 0, \mathbf{\Sigma}), \tag{21}$$

where $\boldsymbol{x} = (x_1, x_2)^T \in \mathbb{R}^2$ and the covariance matrix is

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}. \tag{22}$$

Here, $\rho \in [-1, 1]$ is the correlation coefficient between the two dimensions: $\rho = 0$ corresponds to independent dimensions, $\rho > 0$ to positive correlation, and $\rho < 0$ to negative correlation. In our experiment, we set $\rho = 0.7$ to account for the fact that in many high-dimensional datasets, such as images, the dimensions exhibit strong correlations. The probability density function is

$$p_0(\boldsymbol{x}) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)} \left(x_1^2 - 2\rho x_1 x_2 + x_2^2 \right) \right). \tag{23}$$

Forward diffusion process Given data $\boldsymbol{x}_0 \sim p_0(\boldsymbol{x})$, the forward noising process is defined as

$$\boldsymbol{x}_t = e^{-t} \boldsymbol{x}_0 + \sqrt{1 - e^{-2t}}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, \boldsymbol{I}). \tag{24}$$

We set $t = 0.1$ to reflect low-level noises and $\delta_t = 1 - e^{-2t}$ denotes the noise variance at time t.

Population score function For a Gaussian distribution with covariance $\mathbf{\Sigma}$, the score function at time t is

$$\nabla_x \log p_t(\boldsymbol{x}) = -\mathbf{\Sigma}_t^{-1} x, \tag{25}$$

where the time-dependent covariance is

$$\boldsymbol{\Sigma}_t = e^{-2t}\boldsymbol{\Sigma} + \delta_t \boldsymbol{I}. \tag{26}$$

Empirical score estimation Given training data $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^n$, the empirical score at a query point $\boldsymbol{x}$ is estimated using kernel density estimation:

$$\nabla_{\boldsymbol{x}} \log \hat{p}_t(\boldsymbol{x}) = \sum_{i=1}^{n} w_i(\boldsymbol{x}) \cdot \frac{\boldsymbol{x}_t^{(i)} - \boldsymbol{x}}{\delta_t}, \tag{27}$$

with

$$\boldsymbol{x}_t^{(i)} = e^{-t}\boldsymbol{x}_0^{(i)}, \quad d_i(\boldsymbol{x}) = \frac{\|\boldsymbol{x}_t^{(i)} - \boldsymbol{x}\|^2}{\delta_t}, \quad w_i(\boldsymbol{x}) = \frac{\exp(-\frac{1}{2}d_i(\boldsymbol{x}))}{\sum_{j=1}^n \exp(-\frac{1}{2}d_j(\boldsymbol{x}))}.$$

Masked score estimation When a mask $\boldsymbol{m}^{(i)} \in \{0,1\}^d$ indicates observed dimensions for each sample, the masked score is

$$\nabla_{\boldsymbol{x}} \log \hat{p}_t^{\text{mask}}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i^{\text{mask}}(\boldsymbol{x}) \cdot \frac{\boldsymbol{m}^{(i)} \odot (\boldsymbol{x}_t^{(i)} - \boldsymbol{x})}{\delta_t}, \tag{28}$$

where

$$w_i^{\text{mask}}(\boldsymbol{x}) = \frac{\exp\Big(-\frac{1}{2}d_i^{\text{mask}}(\boldsymbol{x})\Big)}{\sum_{j=1}^n \exp\Big(-\frac{1}{2}d_j^{\text{mask}}(\boldsymbol{x})\Big)}, \tag{29}$$

and

$$d_i^{\text{mask}}(\boldsymbol{x}) = \frac{\|\boldsymbol{m}^{(i)} \odot (\boldsymbol{x}_t^{(i)} - \boldsymbol{x})\|^2}{\delta_t}.$$

The final masked score is obtained by averaging over multiple random masks, where each dimension is observed independently with probability $1 - \eta$.

C Implementation details

We report the implementation details for experiments on CelebA-50K, CIFAR10, and LSUN Bedroom datasets in Table 6. For ImageNet, we use DiT-XL/2 at 256×256 resolution with the Stable Diffusion VAE (ft-MSE) [7]. Baseline and SSD use the same training budget: 180K steps with global batch size 4,096 on full ImageNet, and 200K steps with global batch size 2,048 on the 100K-image subset. Sampling uses 250 DDPM steps with CFG=1.5. We compute FID on 50K generated images against the ImageNet validation set using the ADM evaluation suite. For the 11kHands experiment, we evaluate the 400-epoch checkpoints at 64×64 resolution, sampling 1,000 images per model with seed 0, EMA weights, and Heun's second-order solver with NFE=50.

Table 6: Key hyperparameters for experiments on three datasets

Parameter	**CelebA-50K**	**CIFAR10**	**LSUN Bedroom**
Training Configuration			
Epochs	2000	3000	2000
Effective Batch Size	128	256	1024
Learning Rate	0.0001	0.0001	0.0001
Optimizer Betas	[0.9, 0.95]	[0.9, 0.95]	[0.9, 0.95]
Use EMA	True	True	True
Resolution	64×64	32×32	32×32
Flow Matching Configuration			
Skewed Timesteps	True	True	True
EDM Schedule	True	True	True
Sampling Configuration			
ODE Method	Heun2	Heun2	Heun2
Number of Function Evaluations	50	50	50
ODE Tolerance (atol/rtol)	–	–	–
Dataset & Evaluation			
Number of Images	50,000	50,000	303,125
Number of Classes	–	10	–
FID Samples	50,000	50,000	50,000
Evaluation Frequency	100	100	200
CFG Scale	–	–	–
Class Drop Prob	–	–	–
Model Architecture			
Input/Output Channels	3	3	3
Model Channels	128	128	128
Number of ResBlocks	4	4	4
Attention Resolutions	[2, 4]	[2]	[2]
Dropout	0.2	0.3	0.3
Channel Multipliers	[1, 2, 2, 4]	[2, 2, 2]	[2, 2, 2]
Convolution Resample	True	False	False
Number of Heads	2	1	1
Head Channels	64	-1	-1
Scale Shift Norm	True	True	True
ResBlock Up/Down	True	False	False
New Attention Order	True	True	True
System Configuration			
Number of GPUs	8	8	8

D More experiments

D.1 Comparison with other regularization methods

We further compare our method against the commonly used regularization technique of dropout and the MDT [37] baseline in Table 7.

Table 7: Comparison with dropout and MDT baselines. Both dropout and MDT fail to prevent divergence during training. All methods on CelebA-10K are trained for 2000 epochs.

Method	CelebA-10K FID (↓)	CIFAR10 FID (↓)
MDT	24.72	–
Dropout ($p = 0.30$)	13.72	2.28
Dropout ($p = 0.98$)	13.38	9.37
SSD (ours, $\eta = 0.98$)	**5.10**	**2.20**

D.2 ImageNet under data constraints

On ImageNet-100K, SSD achieves a lower FID than the baseline after 200K training steps, whereas the improvement is less pronounced on the full ImageNet dataset. This is expected, as prolonged training on smaller datasets can lead to overfitting and memorization, degrading generation quality.

Table 8: ImageNet FID at CFG=1.5 using DiT-XL/2 at 256×256.

Training data	Training steps	Baseline	SSD ($\eta = 0.8$)	SSD ($\eta = 0.98$)
ImageNet-100K	200K	25.99	18.45	15.58
Full ImageNet	180K	4.98	–	4.83

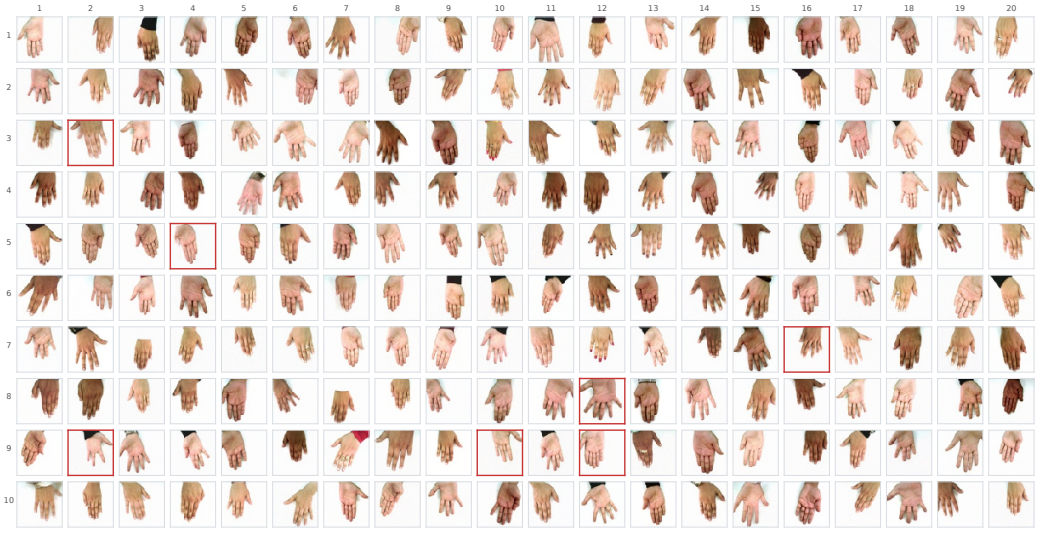
D.3 Patch-wise masking

We compare element-wise loss masking with 8×8 patch-wise loss masking on CelebA-10K at 64×64 resolution in Table 9. The results show that the proposed sparse supervision also works well with different masking strategies.

Table 9: Masking-granularity ablation on CelebA-10K. Results are reported at 3,500 epochs.

Masking strategy	Mask ratio η	Final FID (↓)
None (baseline)	0	13.72
Element-wise	0.5	13.29
Element-wise	0.95	6.36
Element-wise	0.98	5.10
Patch-wise (8×8)	0.6	7.61
Patch-wise (8×8)	0.9	5.14

E Generated hand samples



(a) Baseline ($\eta = 0$)

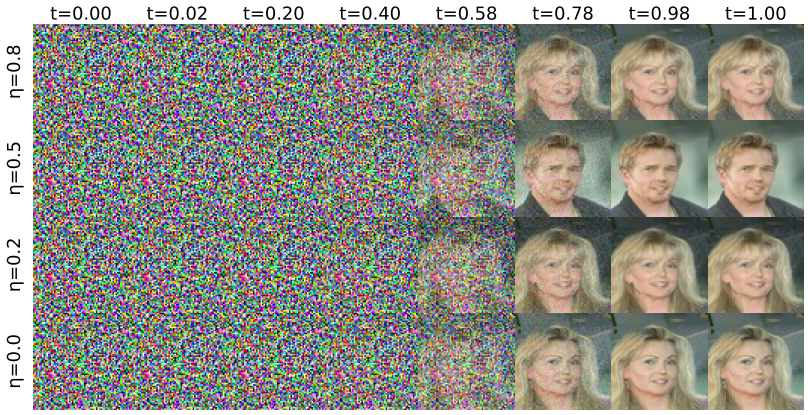


(b) SSD ($\eta = 0.8$)

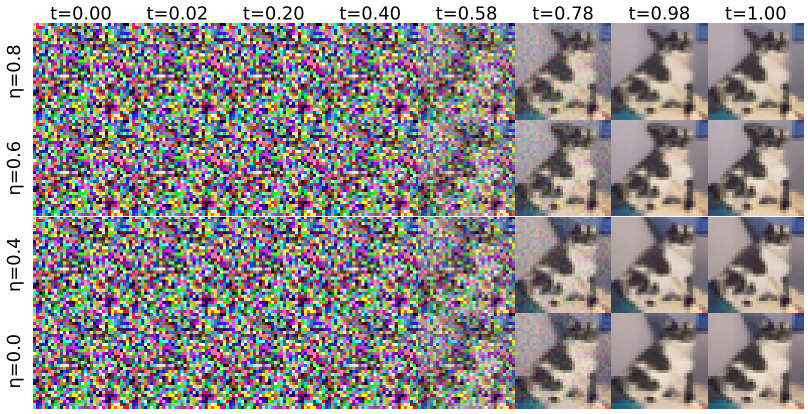
Figure 9: Comparison of generated samples on the 11kHands dataset. Each panel presents 200 samples generated using the same random seed. Thin red outlines highlight samples exhibiting spatial inconsistencies, specifically an incorrect number of fingers. (a) The baseline model frequently generates hands with six fingers, (b) whereas SSD mitigates this issue but occasionally produces hands with only four fingers. The full evaluation results are reported in Table 4.

F Generation visualization

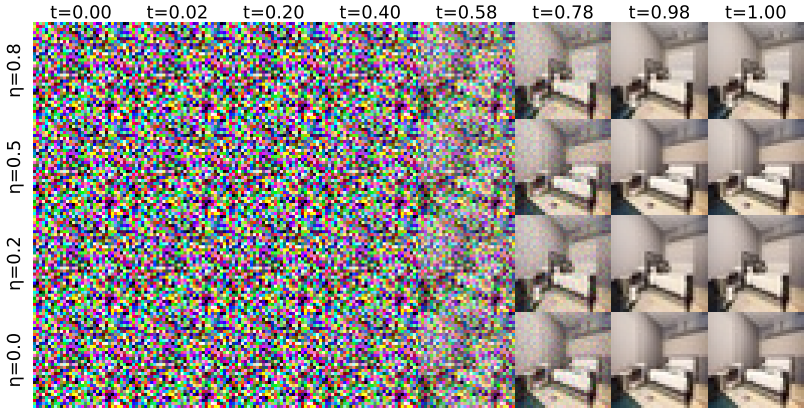
We visualize the sample paths from the same initial noise using models trained with different mask ratios. The sampling is based on the method proposed in Karras et al. [52] and we use Heun's second-order method as the ODE solver [51]. We set NFE as 50. The sample paths in Fig. 10 show that the models may diverge from around $t = 0.58$. More generated samples are shown in 11, 12 and 13.



(a) CelebA-50K 64×64



(b) CIFAR10 32×32



(c) LSUN Bedroom 32×32

Figure 10: Sample paths from same initial noises with models trained with different η across three datasets.



Figure 11: More samples generated by models trained on CelebA-50K with different mask ratios.

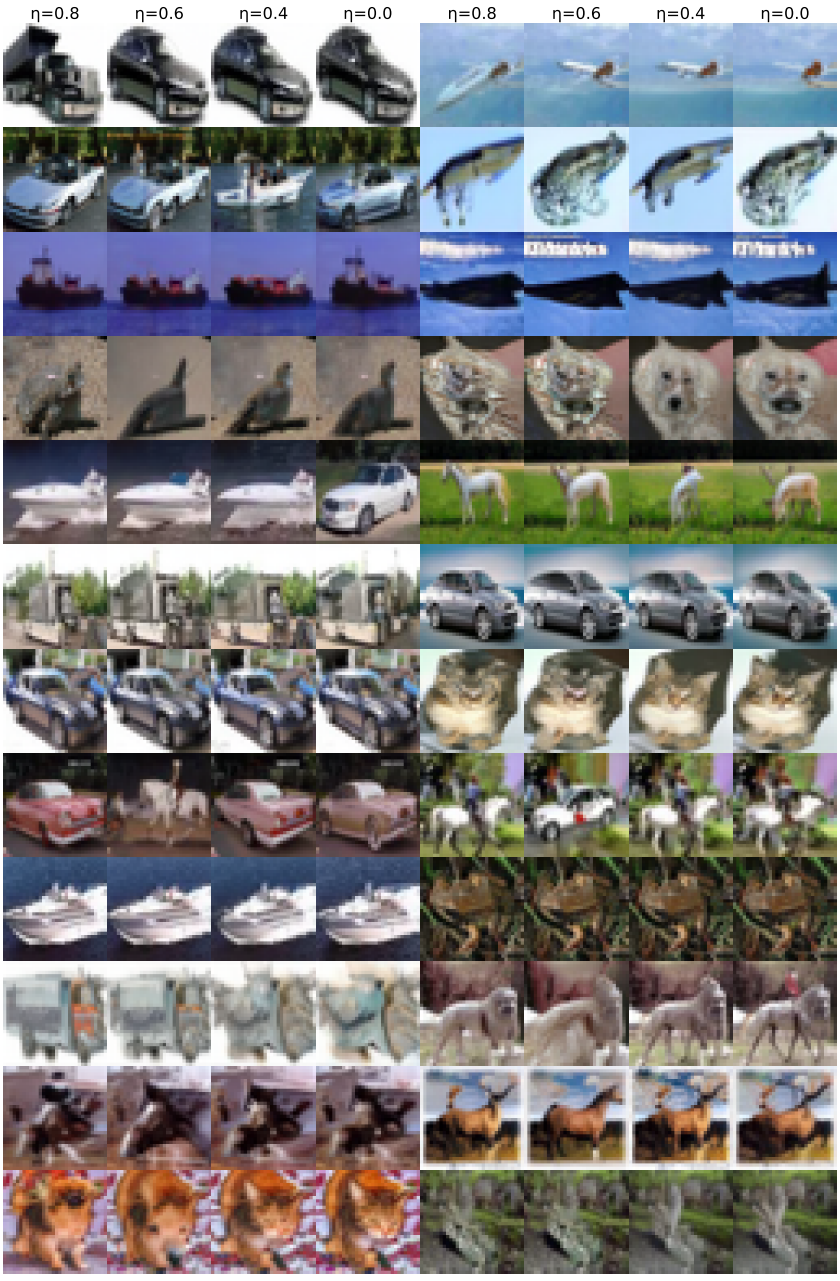


Figure 12: More samples generated by models trained on CIFAR10 with different mask ratios.

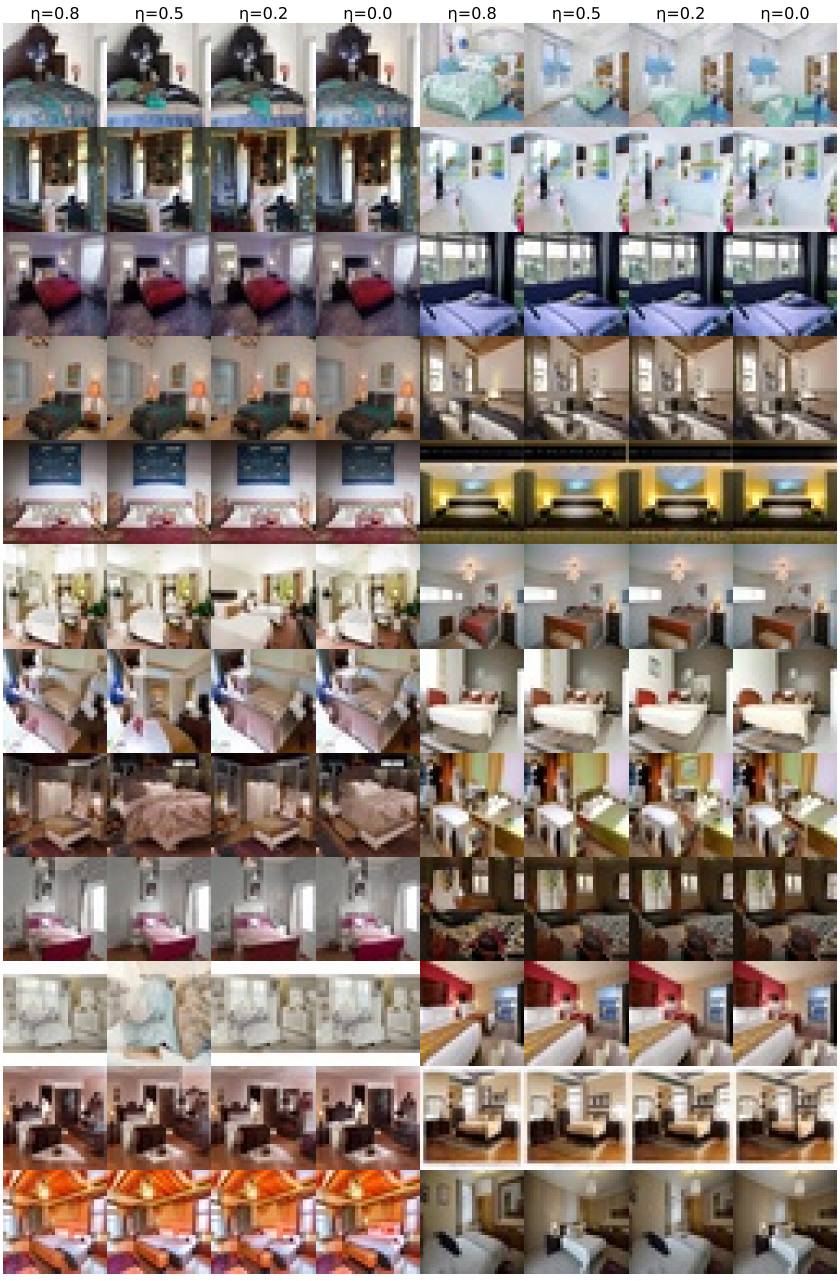


Figure 13: More samples generated by models trained on LSUN Bedroom with different mask ratios.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Sec. 1 (Introduction) clearly state our three main contributions: (i) the analytical result that SSD modifies the spectrum of the data covariance (developed in Sec. 4.2 and Appendix A); (ii) targeted experiments showing improved spatial consistency, more accurate population score estimation, and better contextual usage (Secs. 5.4–5.6); and (iii) competitive FID scores at high masking ratios together with mitigation of memorization and training instability on small datasets (Secs. 5.1–5.3, Table 1).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitation regarding the choice of masking ratio is acknowledged in Sec. 5 ("In all experiments, we do not tune the masking ratio... Determining an optimal masking ratio is left for future work") and reiterated in Sec. 6 (Discussion and conclusion).

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The theoretical results on how masking changes the covariance spectrum are stated in Sec. 4.2 (Eqs. 9–13) and Sec. 4.3 (locality analysis). The full derivation, including all assumptions (zero-mean Gaussian data, independent Bernoulli masking) and the complete proof, is provided in Appendix A (Secs. A.1–A.3, Eqs. 15–20).

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The training objective and method are fully specified in Sec. 4.1 (Eq. 8). Datasets, model architectures (U-Net, DiT), ODE solver (Heun's second-order), NFE, and FID evaluation protocol are described in Sec. 5 (Data & experimental setup). Complete hyperparameters for CelebA-50K, CIFAR-10, and LSUN Bedroom are reported in Appendix C, Table 6, and ImageNet settings follow DiT defaults as noted in Appendix C. The 2D Gaussian score-estimation experiment is fully specified in Appendix B (Eqs. 21–29).

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Our method is model-agnostic and can be implemented with only a few lines of code. We have released the code in the project website. We are only using public datasets. Please refer to their downloading instructions.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Training and evaluation details (epochs, batch sizes, optimizer, learning rate, EMA, resolution, FM configuration, sampling configuration, and model architecture) are reported in Sec. 5 and comprehensively in Appendix C, Table 6. The 2D Gaussian experiment setup is detailed in Appendix B.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: FID curves in Fig. 3 (Sec. 5.1) are averaged over four runs, with shaded regions denoting 95% confidence intervals. Standard deviations across four runs are also reported in Table 1 (Sec. 5.1) for CIFAR-10, LSUN, and CelebA-50K.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: Compute resources are reported in Appendix C, Table 6 (8 GPUs for CelebA-50K, CIFAR-10, and LSUN Bedroom). For ImageNet, the global batch size (4096) and number of training steps (180k) are specified in Appendix C.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The research uses standard public benchmark datasets (CIFAR-10, CelebA, LSUN Bedroom, ImageNet) cited in Sec. 5 and conforms to the NeurIPS Code of Ethics. No human subjects, crowdsourcing, or sensitive data are involved.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All existing assets used — datasets (CIFAR-10 [46], CelebA [47], LSUN [48], ImageNet [49]) and code/methods (DiT [7], Flow Matching [50], EDM sampler [53], U-Net [51], Heun's method [52], FID [54]) — are properly cited in Sec. 5 and the References. Their use is consistent with their respective licenses and intended research use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.