EDGE-PRESERVING NOISE FOR DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Classical diffusion models typically rely on isotropic Gaussian noise, treating all regions uniformly and overlooking structural information that may be vital for high-quality generation. We introduce an edge-preserving diffusion process that generalizes isotropic models through a hybrid noise scheme. At its core is an edge-aware scheduler that transitions smoothly from edge-preserving to isotropic noise, allowing the model to capture fine structural details while generally maintaining global performance. To measure the impact of structure-aware noise on the generative process, we analyze and evaluate our edge-preserving process against isotropic models in both diffusion and flow-matching frameworks. Importantly, we show that existing isotropic models can be efficiently fine-tuned with edge-preserving noise, making our approach practical for adapting pre-trained systems. Beyond improvements in unconditional generation, it offers significant benefits in structure-guided tasks such as stroke-to-image synthesis, improving robustness, fidelity, and perceptual quality. Extensive evaluations (FID, KID, CLIP-score) show consistent improvements of up to 30%, highlighting edge-preserving noise as a simple yet powerful advance for generative diffusion, particularly in structure-guided settings.

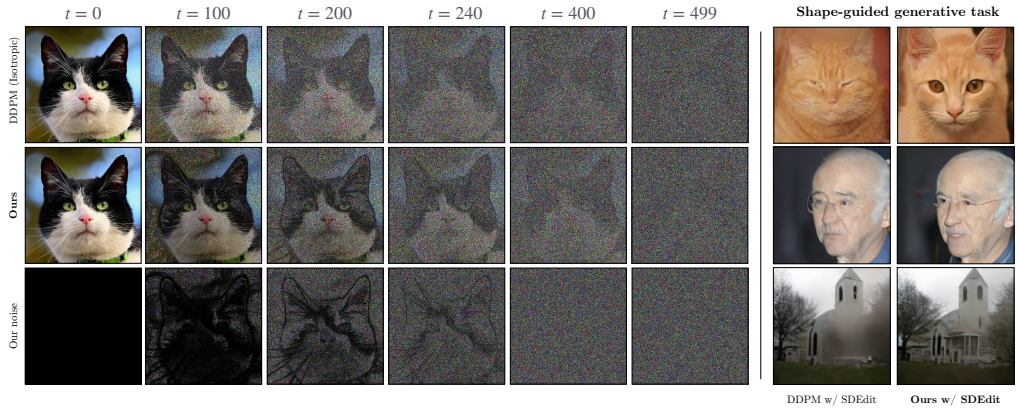


Figure 1: A classic isotropic diffusion process (top row) is compared to our hybrid edge-aware diffusion process (middle row) on the left side. We propose a hybrid noise schedule (bottom row) that smoothly transitions from anisotropic ($t \in [0; 250[$) to isotropic noise ($t \in [250; 499]$) . We use our edge-aware noise for training and inference. On the right, we compare both noise schemes on the SDEdit framework (Meng et al., 2022) for stroke-based image generation. Our model consistently outperforms DDPM's isotropic noise scheme, is more robust against visual artifacts and produces sharper outputs without missing structural details.

1 INTRODUCTION

Previous work on diffusion models mostly uses isotropic Gaussian noise to transform an unknown data distribution into a known distribution (e.g., normal distribution), from which samples can be efficiently drawn (Song & Ermon, 2019; Song et al., 2021; Ho et al., 2020; Kingma et al., 2021). Due to the isotropic nature of the noise, all regions in the data samples $\mathbf{x}_0$ are uniformly corrupted,

regardless of the underlying structural content, which is typically distributed in a non-isotropic manner. In the generative backward process, the model learns an isotropic denoising function, but in doing so, it ignores potentially valuable non-isotropic information in the data that it was trained on. Denoising has been a central topic in image processing research (Elad et al., 2023). The seminal work by Perona & Malik (1990) showed that accounting for image structure enables substantial gains in denoising performance. Since generative diffusion models can also be seen as *denoisers*, we ask ourselves: *Can incorporating structural information from data samples improve the effectiveness of a generative diffusion process?*

To explore our question, we introduce a new class of diffusion models that generalizes over existing isotropic models and explicitly learns a content-aware noise scheme. We call our noise scheme *edge-preserving noise*.

To summarize, we make the following contributions:

- We introduce a novel class of content-aware diffusion models and show how it is a generalization of existing isotropic diffusion models (Section 4.3). We also demonstrate that our noise framework can be applied in the more general setting of flow matching (Section 4.4).
- We run extensive qualitative and quantitative experiments across a variety of datasets to validate the positive impact of using edge-preserving noise over isotropic noise (Section 5 and Appendix F).
- We analyze our model's generative process, and demonstrate that it converges more rapidly to sharper, less noisy predictions (Fig. 2). In addition, we conduct a frequency analysis, suggesting that our edge-preserving model better learns the low-to-mid frequencies of the target data (Appendix C).
- We observe consistent quantitive/qualitative improvements for unconditional image generation. In particular, our noise framework demonstrates strong potential for shape-guided generative tasks, showing greater robustness and significantly improved quality on these tasks (Fig. 3).

2 Related work

Most existing diffusion-based generative models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Song et al., 2021; Ho et al., 2020) corrupt data samples by adding noise with the same variance across all pixels. Generative models tend to have the ability to produce more diverse and novel content when the noise variance is higher, whereas lower variance noise is better at preserving the underlying structure of the data. Various efforts have explored diffusion processes beyond those driven solely by isotropic noise. Rissanen et al. (2023) introduced an inverse heat dissipation model (IHDM), which applies isotropic Gaussian blurring to corrupt images, which they show is equivalent to introducing non-isotropic noise in the frequency domain. One line of work (Bansal et al., 2023; Daras et al., 2023) investigates arbitrary forward diffusion processes with mixed components such as blurring, noise, masking... Hoogeboom & Salimans (2023) propose a generalized form of heat dissipation and diffusion by combining isotropic noise and blurring.

Another line of work has explored non-isotropic forms of noise in diffusion models. Dockhorn et al. (2022) proposed to use critically-damped Langevin diffusion where the data variable at any time is augmented with an additional "velocity" variable. Noise is only injected in the velocity variable. Voleti et al. (2022) performed a limited study on the impact of isotropic vs non-isotropic Gaussian noise for a score-based model. The idea behind non-isotropic Gaussian noise is to use noise with different variance across image pixels. They use a non-diagonal covariance matrix to generate non-isotropic Gaussian noise, but their sample quality did not improve in comparison to the isotropic case. Yu et al. (2024) developed this idea further and proposed a Gaussian noise model that adds noise with non-isotropic variance to pixels. The variance is chosen based on how much a pixel or region needs to be edited. They demonstrated a positive impact on editing tasks. More recently, Huang et al. (2024) proposed a blue noise diffusion model (BNDM), using negatively correlated noise for enhanced visual quality and FID scores. While IHDM and BNDM also consider a form of non-isotropic noise, they do not explicitly account for structures present in the signal.

Our definition of non-isotropy is inspired by the seminal work of Perona & Malik (1990) on anisotropic diffusion for edge-preserving image filtering (removing noise from images). We apply a non-isotropic variance to pixels in an edge-aware manner, meaning that we suppress noise on edges.

3 PRELIMINARIES

Generative diffusion processes. A generative diffusion model consists of two processes: the forward process transforms data samples $\mathbf{x}_0$ into samples $\mathbf{x}_T$ that are distributed according to a well-known prior distribution, such as a normal distribution $\mathcal{N}(0, I)$. The corresponding backward process does exactly the opposite: it transforms samples $\mathbf{x}_T$ into $\hat{\mathbf{x}}_0$, distributed according to the target distribution $p_0(\mathbf{x})$. Sampling from this backward process involves predicting a vector quantity, interpretable as either noise or the gradient of the data distribution, which is precisely the task for which the generative diffusion model is trained. Previous works (Song & Ermon, 2019; Song et al., 2021; Ho et al., 2020; Kingma et al., 2021; Rissanen et al., 2023; Hoogeboom & Salimans, 2023) typically formulate the forward process as the following linear equation:

$$\mathbf{x}_t = \gamma_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}_t \tag{1}$$

here, $\mathbf{x}_t$ is the data sample diffused up to time t, $\mathbf{x}_0$ stands for the original data sample, $\boldsymbol{\epsilon}_t$ is a standard normal Gaussian noise, and the *signal coefficient* γ_t and *noise coefficient* σ_t determine the signal-to-noise ratio (SNR) (γ_t/σ_t). The SNR refers to the proportion of signal retained relative to the amount of injected noise. Note that γ_t and σ_t are both scalars. Previous works have made several different choices for γ_t and σ_t respectively, leading to different variants, each with their own advantages and limitations.

Denoising probabilistic model. Following the probabilistic paradigm of Ho et al. (2020), we would like to introduce the posterior probability distributions of the general diffusion process described by Eq. (1). We will show the exact form that our forward and backward processes take in Section 4.1 and Section 4.3 respectively. For details and full derivations of the equations provided in this paragraph, we would like to refer to the appendix of Kingma et al. (2021). The isotropic diffusion process formulated in Eq. (1) has the following marginal distribution:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\gamma \mathbf{x}_0, \sigma_t^2 \boldsymbol{I}) \tag{2}$$

Moreover, it has the following Markovian transition probabilities:

$$q(\mathbf{x}_t|\mathbf{x}_s) = \mathcal{N}(\gamma_{t|s}\mathbf{x}_s, \sigma_{t|s}^2 \boldsymbol{I}) \tag{3}$$

with the forward posterior signal coefficient $\gamma_{t|s} = \frac{\gamma_t}{\gamma_s}$ and the forward posterior variance (or square of the noise coefficient) $\sigma_{t|s}^2 = \sigma_t^2 - \gamma_{t|s}^2 \sigma_s^2$ and $0 < s < t < T$. For a Gaussian diffusion process, given that $q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0) \propto q(\mathbf{x}_t|\mathbf{x}_s)q(\mathbf{x}_s|\mathbf{x}_0)$, one can analytically derive a *backward process* that is also Gaussian, and has the following marginal distribution:

$$q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\boldsymbol{\mu}_{t\to s}, \sigma_{t\to s}^2 \boldsymbol{I}). \tag{4}$$

The backward posterior variance $\sigma_{t\to s}^2$ has the following form:

$$\sigma_{t\to s}^2 = \left(\frac{1}{\sigma_s^2} + \frac{\gamma_{t|s}^2}{\sigma_{t|s}^2}\right)^{-1} \tag{5}$$

and the backward posterior mean $\boldsymbol{\mu}_{t\to s}$ is formulated as:

$$\boldsymbol{\mu}_{t\to s} = \sigma_{t\to s}^2 \left(\frac{\gamma_{t|s}}{\sigma_{t|s}^2}\mathbf{x}_t + \frac{\gamma_s}{\sigma_s^2}\mathbf{x}_0\right). \tag{6}$$

Samples can be generated by simulating the reverse Gaussian process with the posteriors in Eq. (5) and Eq. (6). A practical issue is that Eq. (6) itself depends on the unknown $\mathbf{x}_0$, the sample we are trying to generate. To overcome this, one can instead approximate the analytic reverse process in which $\mathbf{x}_0$ is replaced by its approximator $\hat{\mathbf{x}}_0$, learned by a deep neural network $f_\theta(\boldsymbol{x}_t, t)$. The network

can learn to directly predict $\mathbf{x}_0$ given an $\mathbf{x}_t$ (a sample with a level of noise that corresponds to time t), but previous work (Ho et al., 2020) has shown that it is beneficial to instead optimize the network to learn the approximator $\hat{\boldsymbol{\epsilon}}_t$. $\hat{\boldsymbol{\epsilon}}_t$ predicts the unscaled Gaussian white noise that was injected at time t. $\hat{\mathbf{x}}_0$ can then be obtained via Eq. (7), which follows from Eq. (1).

$$\hat{\mathbf{x}}_0 = \frac{1}{\gamma_t}\mathbf{x}_t - \frac{\sigma_t}{\gamma_t}\hat{\boldsymbol{\epsilon}}_t \tag{7}$$

Edge-preserving filters in image processing. In this work, we aim to choose γ_t and σ_t such that we obtain a diffusion process that injects noise in a content-aware manner. To do this, we are inspired by the field of image processing, where a classic and effective technique for denoising is edge-preserved filtering via *anisotropic diffusion* (Weickert, 1998). To overcome the problem of destroying relevant structural information in the image when applying an isotropic filter, Perona & Malik (1990) instead propose an anisotropic diffusion process of the form:

$$\mathbf{x}_t = \mathbf{x}_0 + \int_0^t \mathbf{c}(\mathbf{x}_s, s)\Delta\mathbf{x}_s \, ds \tag{8}$$

where the diffusion coefficient $\mathbf{c}(\mathbf{x}_s, s)$ takes the following form:

$$\mathbf{c}(\mathbf{x}, t) = \frac{1}{\sqrt{1 + \frac{||\nabla \mathbf{x}_t||}{\lambda}}} \tag{9}$$

where $||\nabla \mathbf{x}||$ is the gradient magnitude image, and λ is the *edge sensitivity*. Intuitively, in the regions of the image where the gradient response is high (on edges), the diffusion coefficient will be smaller, and therefore the signal gets less distorted there. The edge sensitivity λ determines how sensitive the diffusion coefficient is to the image gradient response.

Inspired by the anisotropic diffusion coefficient presented in Eq. (9), we aim to design a *linear diffusion process* that incorporates edge-preserving noise. Our hope is that by doing this, the generative diffusion model will better learn the underlying geometrical structures of the target distribution, leading to a more effective generative denoising process. To obtain our content-aware linear diffusion process, we apply the idea of edge-preserved filtering to the noise term of Eq. (1). We cannot directly use (Perona & Malik, 1990)'s formulation because their time-dependent diffusion coefficient makes the process nonlinear. Instead, we make the coefficient depend only on $\mathbf{x}_0$:

$$\mathbf{x}_t = \gamma_t \mathbf{x}_0 + \frac{b}{\sqrt{1 + \frac{||\nabla \mathbf{x}_0||}{\lambda(t)}}}\boldsymbol{\epsilon}_t \tag{10}$$

Where b is the noise coefficient's numerator and can be chosen as desired. To investigate the mere impact of non-isotropic edge-preserving noise on the generative diffusion process, we chose our parameters $\gamma_t = \sqrt{\bar{\alpha}_t}$ and $b = \sqrt{1 - \bar{\alpha}_t}$ such that it closely matches the well-studied forward process of (Ho et al., 2020), but nothing prevents us from making different choices for γ_t and b. Note that the noise coefficient in Eq. (1) becomes a tensor $\boldsymbol{\sigma}_t$ instead of a scalar σ_t for our process. Intuitively, we preserve edges by reducing noise based on the edges in the *original* image. In our formulation, we also consider λ to be time-varying (more details in Section 4.2).

4 An edge-preserving generative process

4.1 Forward process with hybrid noise scheme

The forward *edge-preserving* process described in Eq. (10) in its pure form is not very meaningful in a generative setting. This is because if the edges are preserved all the way up to time $t = T$, we end up with a rather complex *prior* distribution $p_T(x)$ that we cannot efficiently take samples from. Instead, we would like to end up with a well-known distribution at time $t = T$, such as the standard normal distribution. To achieve this, we instead consider the following hybrid forward process:

$$\mathbf{x}_t = \gamma_t \mathbf{x}_0 + \frac{b}{(1 - \tau(t))\sqrt{1 + \frac{||\nabla \mathbf{x}_0||}{\lambda(t)}} + \tau(t)}\boldsymbol{\epsilon}_t \tag{11}$$

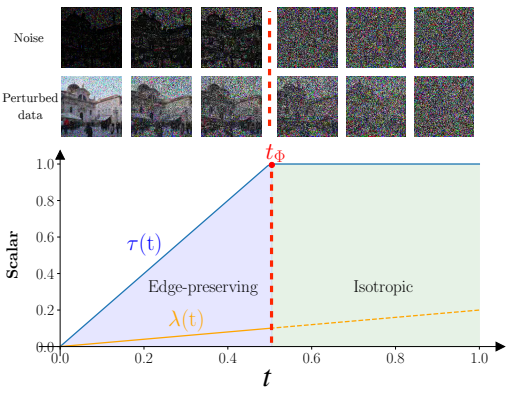
The function $\tau(t)$ now appearing in the denominator of the diffusion coefficient is the *transition function*. When $\tau(t) < 1$, we obtain edge-preserving noise (the edge-preservation is strongest when $\tau(t) \approx 0$). The turning point where $\tau(t) = 1$ is called the *transition point* t_Φ. At the transition point, we switch over to isotropic noise with scalar noise coefficient $\sigma_t = b$ (note that we chose $\gamma_t = \sqrt{\bar{\alpha}_t}$ and $b = \sqrt{1 - \bar{\alpha}_t}$).

This approach allows us to flexibly design noise schedulers that start off with edge-preserving noise and towards the end of the forward process fall back to an isotropic diffusion coefficient. Practically, one can choose any function for $\tau(t)$, as long as it maps to $[0; 1]$ and $\tau(t) = 1$ for t in proximity to T. We performed an ablation for different transition functions in Appendix A.

Observe how our diffusion process generalizes over existing isotropic processes: by setting $\tau(t) = 1$ constant, we simply obtain an isotropic process with signal coefficient γ_t and noise coefficient $\sigma_t = b$. Choosing any other non-constant function for $\tau(t)$ leads to a hybrid diffusion process that consists of an edge-preserving stage and an isotropic stage (starting at $\tau(t) = 1$).

4.2 Time-varying edge sensitivity $\lambda(t)$

The edge sensitivity parameter λ controls the level of detail preserved along image edges. Very low values of (e.g. $\lambda = 1e - 5$) will retain almost all fine details. The more we increase λ, the less details will be preserved. When λ becomes very high (e.g. $\lambda = 1$), the process becomes nearly isotropic. Our ablation study (Appendix A) explores impact of this parameter in more detail. We found that constant λ-values harm sample quality: too low values results in unrealistic, "cartoonish" images, while too high values diminish the effectiveness of the edge-preserving diffusion model, making the model behave almost like an isotropic process.



To address this, we instead consider a time-varying edge sensitivity $\lambda(t)$. We set an interval $[\lambda_{min}; \lambda_{max}]$ that bounds the possible values for the time-varying edge sensitivity. The function that governs $\lambda(t)$ within this interval can in theory again be chosen freely. We have so far experimented with a linear function and a sigmoid function. We experienced that a linear function for $\lambda(t)$ resulted in higher sample quality and therefore used this function for our experiments. Additionally, we have attempted to optimize the interval $[\lambda_{min}; \lambda_{max}]$, but this led to unstable behaviour.

4.3 Edge-Aware Generative Process in Diffusion Models

Given the forward hybrid diffusion process introduced in Section 4.1, we construct the corresponding generative backward process within the denoising diffusion framework (for the edge-preserving flow-matching variant, see Section 4.4). Specifically, we derive explicit expressions for the posterior mean $\boldsymbol{\mu}_{t\to s}$ and variance $\boldsymbol{\sigma}^2_{t\to s}$ of the backward process by substituting our chosen signal coefficient γ_t and variance $\boldsymbol{\sigma}^2_t$ into Eq. (6) and Eq. (5).Recall that we chose $\boldsymbol{\sigma}^2_t$ to be a tensor, which is why the backward posterior variance $\boldsymbol{\sigma}^2_{t\to s}$ is again a tensor, contrary to isotropic diffusion processes considered in previous works. Regardless, we can use the same equations and the algebra still works.

We first introduce an auxiliary variable $\boldsymbol{\sigma}^2(t)$, which represents the variance of our forward process at a given time t. This is simply the square of our choice for the noise coefficient $\boldsymbol{\sigma}_t$ formulated in Eq. (11):

$$\boldsymbol{\sigma}^2(t) = \frac{1 - \bar{\alpha}_t}{(1 - \tau(t))^2 \left(1 + \frac{||\nabla \mathbf{x}_0||}{\lambda(t)}\right) + 2\left((1 - \tau(t))\sqrt{1 + \frac{||\nabla \mathbf{x}_0||}{\lambda(t)}\tau(t)}\right) + \tau(t)^2} \tag{12}$$

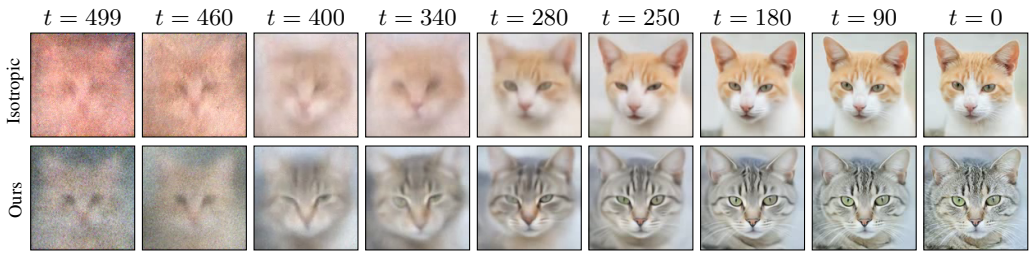


Figure 2: We visually compare the impact of our edge-preserving noise on the generative process. In each column, we show predictions $\hat{\mathbf{x}}_0$ at selected time steps. Our method converges significantly faster to a sharper and less noisy image than its isotropic counterpart. This is evident by the earlier emergence (from $t = 400$) of structural details like the pattern on the cat's head, eyes, and whiskers with our approach.

Here $\bar{\alpha}_t$ has the same meaning as earlier described in Section 3. We now have the backward posterior variance $\boldsymbol{\sigma}^2_{t\to s}$:

$$\boldsymbol{\sigma}^2_{t\to s} = \left(\frac{1}{\boldsymbol{\sigma}^2(t)} + \frac{\frac{\bar{\alpha}_t}{\bar{\alpha}_s}}{\boldsymbol{\sigma}^2(t) - \frac{\bar{\alpha}_t}{\bar{\alpha}_s}\boldsymbol{\sigma}^2(s)} \right)^{-1} \tag{13}$$

and the backward posterior mean $\boldsymbol{\mu}_{t\to s}$:

$$\boldsymbol{\mu}_{t\to s} = \sigma^2_{t\to s} \left(\frac{\frac{\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_s}}}{\boldsymbol{\sigma}^2(t) - \frac{\bar{\alpha}_t}{\bar{\alpha}_s}\boldsymbol{\sigma}^2(s)} \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_s}}{\boldsymbol{\sigma}^2(s)} \mathbf{x}_0 \right) \tag{14}$$

Given Eq. (13) and Eq. (14), the only unknown preventing us from simulating the Gaussian backward process is $\mathbf{x}_0$. Note that $\mathbf{x}_0$ in our case depends on a non-isotropic noise. Therefore, we cannot just use an isotropic approximator $\hat{\boldsymbol{\epsilon}}_t$ for the isotropic noise $\boldsymbol{\epsilon}_t$ to predict $\hat{\mathbf{x}}_0$ via Eq. (7). Instead, we need a model that can predict the non-isotropic noise $\boldsymbol{\sigma}_t\boldsymbol{\epsilon}_t$.

We introduce the loss function that trains such an approximator:

$$\mathcal{L} = ||f_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\sigma}_t\boldsymbol{\epsilon}_t||^2. \tag{15}$$

It is very similar to the simplified loss function derived in DDPM, **with the difference that our model explicitly learns to predict the non-isotropic edge-preserving noise ($\boldsymbol{\sigma}_t\boldsymbol{\epsilon}_t$)**. Note that we apply no weighting to our loss function. In Appendix B, we show that this is a heuristic, and we also show how our loss formulation can be derived from a negative log-likelihood perspective, with the accurate theoretically-founded weighting.

$f_\theta(\boldsymbol{x}_t, t)$ stands for the time-conditioned U-Net used to approximate the time-varying noise function. The visual difference between the backward process of an isotropic diffusion model (DDPM) and ours is shown in Fig. 2. Our formulation introduces a negligible overhead. The only additional computation that needs to be performed is the image gradient $||\nabla \mathbf{x}_0||$, which can be done very efficiently on modern GPUs. We have not noticed any significant difference in training time between vanilla DDPM and our method.

4.4 Edge-Aware Generative Process in Flow Matching

The general framework of flow matching allows users to design probability paths that on their turn will correspond to some probability flow vector field. Our goal is to construct a such path that leads to flows that are aware of the geometric structures in the target dataset. Motivated by its simple formulation and the impressive results Lipman et al. (2022) achieved with it, we choose to build upon the optimal transport variant of flow matching (OT-FM). Theorem 3 derived by Lipman et al. (2022) provides an elegant and flexible design framework for probability flows, where the user only has to specify differentiable functions $\mu_t(x_1)$ and $\boldsymbol{\sigma_t}(x_1)$. These functions correspond to the signal coefficient γ_t and noise coefficient σ_t that we introduced in Section 3. The OT-FM formulation chooses $\mu_t(x_1) = t$ and $\boldsymbol{\sigma_t}(x_1) = 1 - t$. Similar to what we did for isotropic diffusion (Sections 4

and 4.3), we make this probability path "edge-preserving" by leaving $\mu_t(x_1)$ unchanged, and only operate on $\boldsymbol{\sigma_t}(x_1)$:

$$\boldsymbol{\sigma_t}(x_1) = \frac{1-t}{(1-\tau(t))\sqrt{1+\frac{||\nabla \mathbf{x}_0||}{\lambda(t)}}+\tau(t)} \tag{16}$$

To use this formulation in the framework of flow matching, we also need to find its corresponding time derivative:

$$\boldsymbol{\sigma_t}'(x_1) = \frac{gf' - fg'}{g^2} \tag{17}$$

Which follows from the quotient rule for derivatives, where f is the numerator of $\boldsymbol{\sigma_t}(x_1)$, g is the denominator of $\boldsymbol{\sigma_t}(x_1)$, and f' and g' are its respective time derivatives: $f' = -1$, and g' has the following form:

$$g' = \tau'(t) + \left((1-\tau(t))\left(\frac{\lambda(t)-\lambda'(t)}{2\lambda(t)^2\sqrt{1+\frac{||\nabla \mathbf{x}_0||}{\lambda(t)}}}\right) + \left(-\tau'(t)\sqrt{1+\frac{||\nabla \mathbf{x}_0||}{\lambda(t)}}\right)\right) \tag{18}$$

However, note that because $\boldsymbol{\sigma_t}(x_1)$ has the requirement to be differentiable, f and g should also be differentiable. For f there are trivially no issues, but g contains two nested functions $\tau(t)$ and $\lambda(t)$, which are not necessarily differentiable. For example, the linear choice we made for $\tau(t)$ (see inline figure in Section 4) is only piecewise differentiable. We experimentally found that using this formulation leads to an unstable optimization objective, preventing the model from convergence. To overcome this, we tried to simplify our choice for $\boldsymbol{\sigma_t}(x_1)$. In particular, we removed the time dependency on the edge sensitivity λ, and changed $\tau(t)$ to be piecewise constant instead of a piecewise linear function:

$$\tau(t) = \begin{cases} 0 & t \leq t_\Phi \\ 1 & t > t_\Phi \end{cases} \tag{19}$$

This results to the time-derivative g' of the denominator of $\boldsymbol{\sigma_t}(x_1)$ now becoming:

$$g' = (\frac{\partial\sqrt{1+\frac{||\nabla \mathbf{x}_0||}{\lambda}}}{\partial t} - \sqrt{1+\frac{||\nabla \mathbf{x}_0||}{\lambda}}\tau'(t)) + \tau'(t) = 0 \tag{20}$$

This is the case because λ no longer depends on time t and the time derivative of $\tau(t)$ is now $\tau'(t) = 0$ over the whole domain. As a consequence, $\boldsymbol{\sigma_t}'(x_1)$ now becomes:

$$\boldsymbol{\sigma_t}'(x_1) = \frac{gf' - fg'}{g^2} = \frac{gf'}{g^2} = \frac{f'}{g} = \frac{-1}{(1-\tau(t))\sqrt{1+\frac{||\nabla \mathbf{x}_0||}{\lambda}}+\tau(t)} \tag{21}$$

With this slightly simplified formulation, we experienced a much more stable training objective. This is the final formulation that we used to generate the results in Fig. 5.

5 Experiments

Implementation details We provide the implementation details for our experiments in Appendix E. Please also find our training performance analysis on different frequency bands in Appendix C.

Unconditional image generation We evaluate unconditional generation with edge-preserving noise in the diffusion framework (Fig. 4, Appendix F), with FID scores reported in Table 1. Across datasets, edge-preserving (non-isotropic) noise improves both metrics and visual quality compared to isotropic noise (Ho et al., 2020). While gains over DDPM can be subtle, our method reduces artifacts and shows clearer advantages in structure-guided generation (Fig. 3).

FID (↓)	Blue	Isotr.	Ours
CelebA(128^2)	68.0	45.80	**39.08**
Church(128^2)	93.81	72.54	**56.14**
Cat(128^2)	51.05	27.61	**23.50**

Figure 3: **Left:** Impact of different types of noise to the SDEdit framework (Meng et al., 2022) for shape-guided generation. The leftmost column displays the stroke-based guide (created via k-means clustering applied to an image), with the other three columns showing the model outputs. Overall, using our noise franework results in sharper details and less distortions compared to other noises, leading to a better visual and quantitative performance. The corresponding FID scores are shown in the top right column. **Right:** Our noise also works effectively with human-drawn paintings as shape guides, showing particularly precise adherence to details, such as the orange patches on the cat's fur.

Figure 4: Comparison of unconditional samples generated using the isotropic noise model from DDPM (Ho et al., 2020) and our proposed edge-preserving noise model. While qualitative differences can be subtle, the quantitative metrics reported in Table 1 indicate that the edge-preserving noise model enhances the generative process. Additional results are provided in Appendix F.

Table 1: Quantitative FID and KID score comparison (lower is better) for unconditional image generation for DDPM (Ho et al., 2020) (isotropic noise) and our method across different datasets.

FID / KID (↓)	CelebA(128^2)	LSUN-Church(128^2)	AFHQ-Cat(128^2)
DDPM	31.60 / 0.031	31.01 / 0.024	12.51 / 0.007
Ours	**26.15 / 0.022**	**23.16 / 0.018**	**9.53 / 0.005**

To show that our noise scheduler also works in practice within the framework of flow matching, we compare OT-FM (Lipman et al., 2022) against an edge-preserving variant (EP-OT-FM) (see Section 4.4 for details). Although FID/KID are similar, results in Fig. 5 show consistent visual improvements. In a user study with 30 participants, our EP-OT-FM was consistently preferred over OT-FM in terms of perceived quality for AFHQ-Cat, CelebA, and CIFAR-10 samples (see Table 2).

Edge-preserving noise in the latent space We also tested edge-preserving noise in the latent space, with results shown in Table 3 and Fig. 13. We would like to clarify that it makes sense to do this, given that in the latent space, a lot of geometric structure and shape of the original image is actually preserved (see Fig. 6).

Stroke-guided image generation (SDEdit) We applied our edge-preserving diffusion noise to SDEdit (Meng et al., 2022) for stroke-based generation.

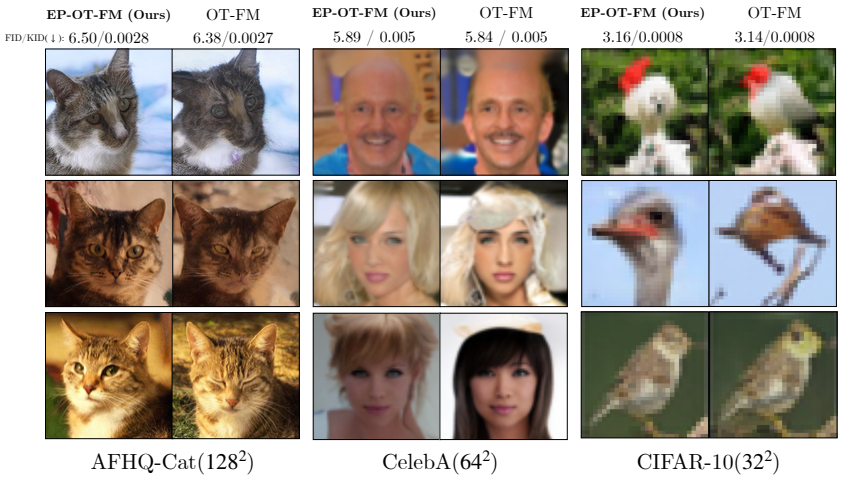


Figure 5: Visual and quantitative comparison between **our edge-preserving variant (EP-OT-FM)**, displayed on the left side of each column, and the standard Optimal Transport Flow Matching (OT-FM) (Lipman et al., 2022), displayed on the right. While FID and KID scores (lower is better) are closely matching, we observe that in the majority of time, our method delivers visual improvements in samples generated with the same seed.

Using k-means clustering, 1000 images were converted into stroke paintings and reconstructed with backbones trained on different noise types, including blue noise (Huang et al., 2024) and isotropic noise (Ho et al., 2020), at a hijack point of $0.55T$. **Our method better adheres to guiding priors, reduces artifacts, and achieves superior FID scores (Fig. 3)**.

Table 2: User study results for perceived quality on samples generated by EP-OT-FM vs. OT-FM. Score can range from 1 (worst) to 5 (best).

	EP-OT-FM (Ours)	OT-FM
Mean score	**3.72**	2.80
Score std. dev.	**0.24**	0.75

Additional results (Appendix F) and further evaluations on precision/recall and CLIP confirm that it maintains diversity while enhancing semantic preservation compared to the isotropic backbone. These findings highlight the usefulness of edge-preserving noise in editing tasks that rely on geometric fidelity.

Fine-tuning with edge-preserving noise We found that a model pre-trained with isotropic noise can be efficiently fine-tuned using edge-preserving noise. After fewer than 5k fine-tuning iterations on a model pre-trained for 150,000 steps (2000 epochs), it already shows clear evidence of learning the non-isotropic noise patterns in the data (see Fig. 14). This improvement is reflected in the FID score, which drops from 16.03 for the pre-trained model to 12.59 after fine-tuning.

6 Limitations and Conclusion

We introduced a new class of edge-preserving generative diffusion models that generalize isotropic models and can be applied in both the frameworks of diffusion and flow matching. Our hybrid process consists of an edge-preserving phase, which maintains structural details, followed by an isotropic phase to ensure convergence to a known prior. This decoupled approach better captures low-to-mid frequencies and accelerates convergence to sharper, less noisy predictions. It outperforms its isotropic counterparts on both unconditional and shape-guided generative tasks. In addition, our framework offers a large hyperparameter space that remains open for further exploration. We do not claim that the parameters we currently used are optimal. Future work could explore more applications that can benefit from accurate structure-guided generation, as well as experiment with our non-isotropic noise framework in video generation (e.g. for better temporal consistency).

7 REPRODUCIBILITY STATEMENT

We provide theoretical details on our proposed noise framework in Sections 3, 4.3 and 4.4. Additionally, we provide further implementation details of our experiments and training parameters in Appendix E. The source code for training and inference along with the checkpoints to produce the reported results will be released upon publication of the paper.

8 ETHICS STATEMENT

Generative diffusion models, while capable of generating high-quality and realistic images, pose several significant risks. One major concern is the creation of deepfakes, which can spread misinformation and deceive the public, undermining trust in digital media. Additionally, the replication of artistic works without proper attribution raises intellectual property issues. Nonetheless, we believe that transparency and the development of new insights into how these models function can also support the cybersecurity community in advancing methods to more effectively secure generative models. We should not forget to mention that generative diffusion models also impact the world in a positive manner, from synthetic data generation, artistic content generation to scientific breakthrough such as drug discovery.

REFERENCES

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36, 2023.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8188–8197, 2020.

Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alex Dimakis, and Peyman Milanfar. Soft diffusion: Score matching with general corruptions. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *International Conference on Learning Representations*, 2022.

Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):44:1–44:10, 2012.

Michael Elad, Bahjat Kawar, and Gregory Vaksman. Image denoising: The deep learning revolution and beyond—a survey paper. *SIAM Journal on Imaging Sciences*, 16(3):1594–1654, 2023.

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. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Emiel Hoogeboom and Tim Salimans. Blurring diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

Xingchang Huang, Corentin Salaun, Cristina Vasconcelos, Christian Theobalt, Cengiz Oztireli, and Gurprit Singh. Blue noise for diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

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

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*, pp. 10684–10695, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Vikram Voleti, Christopher Pal, and Adam M Oberman. Score-based denoising diffusion with non-isotropic gaussian noise models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

Joachim Weickert. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

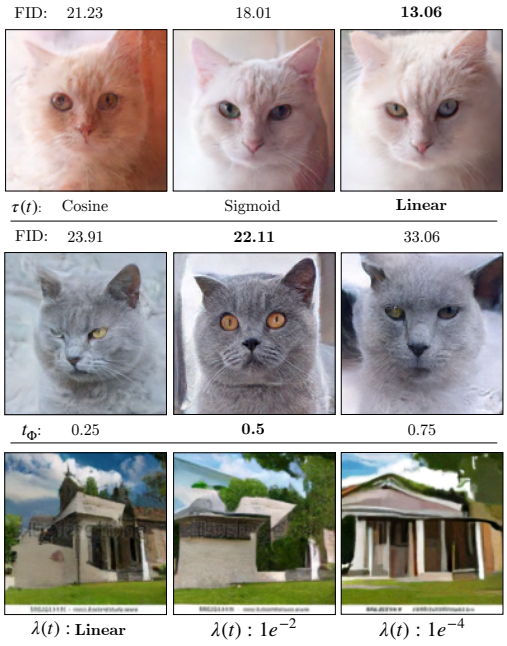
Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Xi Yu, Xiang Gu, Haozhi Liu, and Jian Sun. Constructing non-isotropic gaussian diffusion model using isotropic gaussian diffusion model for image editing. *Advances in Neural Information Processing Systems*, 36, 2024.

A Ablation Study

Impact of transition function $\tau(t)$. We have experimented with three different choices for the transition function $\tau(t)$: linear, cosine and sigmoid. While cosine and sigmoid show similar performance, we found that having a smooth linear transition function significantly improves the performance of the model. A qualititative and quantitative comparison between the choices is presented in the inline figure below.

Impact of transition points t_Φ. We have investigated the impact of the transition point t_Φ on our method's performance by considering 3 different diffusion schemes: 25% edge-preserving - 75% isotropic, 50% isotropic - 50% edge-preserving and 75% edge-preserving - 25% isotropic. A visual example for AFHQ-Cat (128^2) is presented in the inline figure on the right. We have experienced that there are limits to how far the transition point can be placed without sacrificing sample quality. Visually, we observe that the further the transition point is placed, the less details the model generates. The core shapes however stay intact. This is illustrated well by Fig. 9 in Appendix F. For the datasets we tested on, we found that the 50%-50% diffusion scheme works best in terms of FID metric and visual sharpness. This again becomes apparent in Fig. 9: although the samples for $t_\Phi = 0.25$ contain slightly more details, the samples for $t_\Phi = 0.5$ are significantly sharper.



Impact of edge sensitivity $\lambda(t)$. As shown in the inline figure on the right, lower constant $\lambda(t)$ values lead to less detailed, more flat, "water-painting-style" samples. Intuitively, a lower constant $\lambda(t)$ corresponds to stronger edge-preservation in the noise and our model is explicitly trained accordingly to better learn the core structural shapes instead of the high-frequency details that we typically find in interior regions. Our time-varying choice for $\lambda(t)$ works better than other settings in our experiments, by effectively balancing the preservation of structural information across different granularities of detail.

B Relation to ELBO objective in diffusion literature

In this section we explain how the loss derivation from a perspective of minimizing the negative log-likelihood can be done for our formulation, similar to what is discussed in the original DDPM Ho et al. (2020) paper.

The denoising probabilistic model paradigm defined in the DDPM paper defines the loss by minimizing a variational upper bound on the negative log likelihood. Because our noise is still Gaussian, the derivation they make in Eq. (3) to (5) of their paper still holds for us. The difference however is that we are non-isotropically scaling our noise based on the image content. As a result, our methods differ on Eq. (8) in their paper. Instead, we end up with the following form of this equation:

$$L_{t-1} = \mathbb{E}_q[\langle \Sigma^{-1}(\tilde{\mu_{\mathbf{t}}}(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)), (\tilde{\mu_{\mathbf{t}}}(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t))\rangle] \tag{22}$$

In essence, for our formulation that considers non-isotropic Gaussian noise, we need to apply a different loss scaling for each pixel. A theoretically-founded *weighted* version of our loss function (introduced in Eq. (15)) would then be the following:

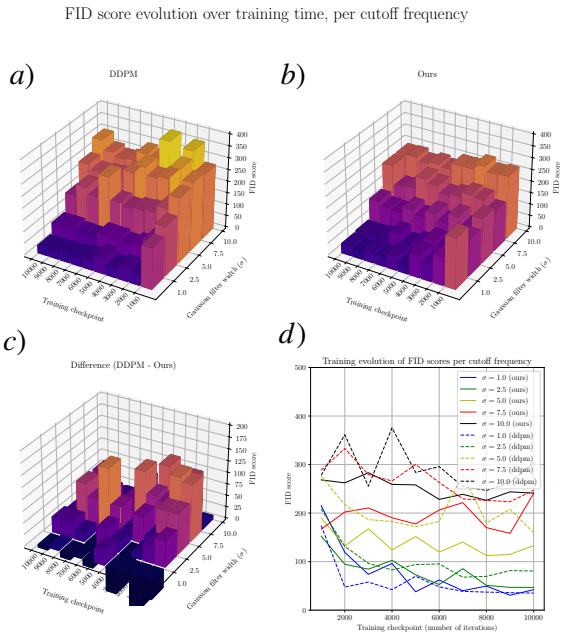
$$\mathcal{L} = \frac{1}{2}\langle \Sigma^{-1}(f_\theta(\mathbf{x}_t, t) - \sigma_t \epsilon_t)), (f_\theta(\mathbf{x}_t, t) - \sigma_t \epsilon_t)\rangle \tag{23}$$

However, note that in the original DDPM paper, the loss is also simplified by removing the weighting, which they call the simplified or *reweighted loss* (Eq. (14) in their paper). The authors argue that this reweighting leads to improved sample quality. To save computational resources, we follow a similar heuristic in our loss function where we remove the weighting. While our heuristic loss function already proved effective, a more theoretically accurate loss would include the scaling discussed above.

In our non-isotropic case, Σ is dependent on both the clean data $\mathbf{x}_0$ and t. Therefore, the scaling could be approximated by choosing $\hat{\Sigma}_t$ such that $c_1||\hat{\Sigma}_t|| \leq ||\Sigma(\mathbf{x}_0)_t|| \leq c_2||\hat{\Sigma}_t||$, for some $c_1, c_2 > 0$ and for all $\mathbf{x}_0$, where $||.||$ is an appropriate norm.

C FREQUENCY ANALYSIS OF TRAINING PERFORMANCE

To better understand our model's capacity of modeling the target distribution, we conducted an analysis on its training performance for different frequency bands. Our setup is as follows, we create 5 versions of the AFHQ-Cat128 dataset, each with a different cutoff frequency. This corresponds to convoluting each image in the dataset with a Gaussian kernel of a specific standard deviation σ, representing a frequency band. For each frequency band, we then trained our model for a fixed amount of 10000 training iterations. We place a model checkpoint at every 1000 iterations, so we can also investigate the evolution of the performance over this training time. We measure the performance by computing the FID score between 1000 generated samples (for that specific checkpoint) and the original dataset of the corresponding frequency band. A visualization of the analyzed results is presented in the inline figure on the right. We found that our model is able to learn the low-to-mid frequencies of the dataset significantly better than the isotropic model (DDPM). The figure shows the evolution of FID score over the first 10,000 training iterations per frequency band (larger σ values correspond to lower frequency bands). $a)$ and $b)$ show performance in terms of FID score of DDPM and our model, respectively. $c)$ shows their difference (positive favors our method). $d)$ visualizes the information in 2D for a more accurate comparison. Our model significantly outperforms in low-to-mid frequency bands (lower FID is better).



D VISUALIZATION OF IMAGE LATENTS

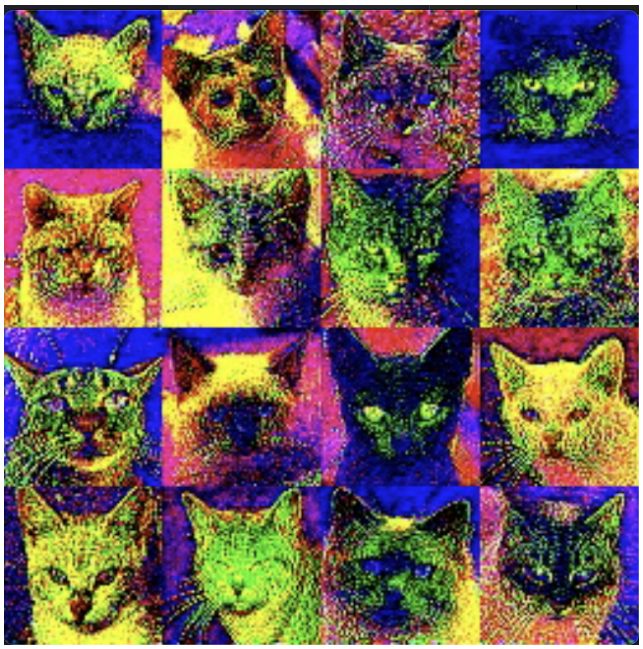


Figure 6: We visualize image latents for the AFHQ-Cat(512^2) dataset. Notice that most of the structural content of the image remains preserved in the latent space. Therefore, it makes sense to also apply edge-preserving noise in the latent space (also see Table 3 and Fig. 13).

E IMPLEMENTATION DETAILS OF EXPERIMENTS

We compare our method against two baselines that use an isotropic form of noise, namely DDPM (Ho et al., 2020) and Optimal Transport Flow Matching (OT-FM) (Lipman et al., 2022).

We perform unconditional generation experiments on two settings: pixel-space diffusion following the setting of Ho et al. (2020); Rissanen et al. (2023) and latent-space diffusion following (Rombach et al., 2022) noted as LDM in Table 3, where the diffusion process runs in the latent space. Besides this, we perform an experiment on shape-guided generation, as summarized in Fig. 3, and an analysis on the capabilities of our model to learn different frequency bands of the data, further explained in Appendix C. We used the following datasets: CIFAR-10(32^2, 50,000 training images) (Krizhevsky et al., 2009), CelebA (128^2, 30,000 training images) (Lee et al., 2020), AFHQ-Cat (128^2, 5,153 training images) (Choi et al., 2020), Human-Sketch (128^2, 20,000 training images) (Eitz et al., 2012) (see Fig. 7) and LSUN-Church (128^2, 126,227 training images) (Yu et al., 2015) for pixel-space diffusion. For latent-space diffusion (Rombach et al., 2022), we tested on AFHQ-Cat (512^2).

We used a batch size of 64 for all experiments in image space, and a batch size of 128 for all experiments in latent space. We trained CIFAR-10(32^2) and AFHQ-Cat (128^2) for 1000 epochs, AFHQ-Cat (512^2) (latent diffusion) for 1750 epochs, CelebA(128^2) for 475 epochs and LSUN-Church(128^2) for 90 epochs for our method and the baselines we compare to. Our framework is implemented in Pytorch (Paszke et al., 2017). For the network architecture we adopt the 2D U-Net from Rissanen et al. (2023). We use T = 500 discrete time steps for both training and inference, except for AFHQ-Cat (128^2), where we used T = 750. To optimize the network parameters, we use Adam optimizer (Kingma & Ba, 2014) with learning rate $1e^{-4}$ for latent-space diffusion models and $2e^{-5}$ for pixel-space diffusion models. We trained all datasets on 2x NVIDIA Tesla A40.

For our final results, we used a linear scheme for $\lambda(t)$ that linearly interpolates between $\lambda_{min} = 1e^{-4}$ and $\lambda_{max} = 1e^{-1}$. We used a transition point $t_\Phi = 0.5$ and a linear transition function $\tau(t)$.

To evaluate the quality of generated samples, we consider FID (Heusel et al., 2017). using the implementation from Stein et al. (2024), with Inception-v3 network (Szegedy et al., 2016) as backbone. We generated 30k images to compute FID scores for unconditional generation and shape-guided generation, for all datasets.

F ADDITIONAL RESULTS

In this section, we provide additional results and ablations.

Table 3 shows quantitative FID comparisons using latent diffusion (Rombach et al., 2022) models on all the baselines.

Figure 10, Figure 11, Figure 12, Figure 13 show more generated samples and comparisons with DDPM on all previously introduced datasets. In Fig. 7 we show samples for the Human-Sketch (128^2) data set specifically. This dataset was of particular interest to us, given the images only consist of high-frequency, edge content. Although we observed that this data is remarkably challenging for all methods, our model is able to consistently deliver visually better results.

Figure 9 shows an additional visualization of the impact t_Φ for the LSUN-Church (128^2) dataset. $t_\Phi = 0.5$ works best in terms of FID metric, consistent to the results shown in Appendix A.

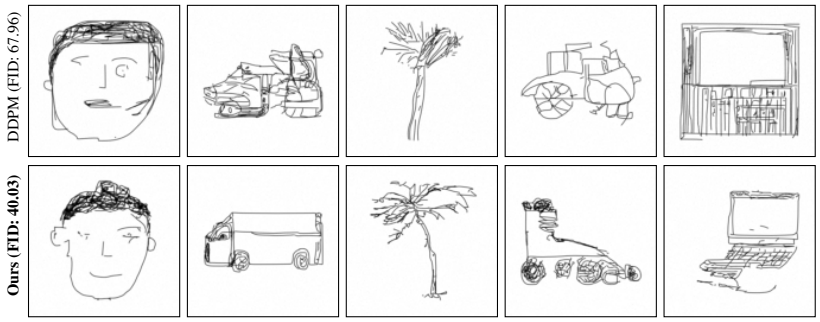


Figure 7: Generated unconditional samples for the Human Sketch (128^2) dataset (Eitz et al., 2012). Both models were trained for an equal amount of 575 epochs.

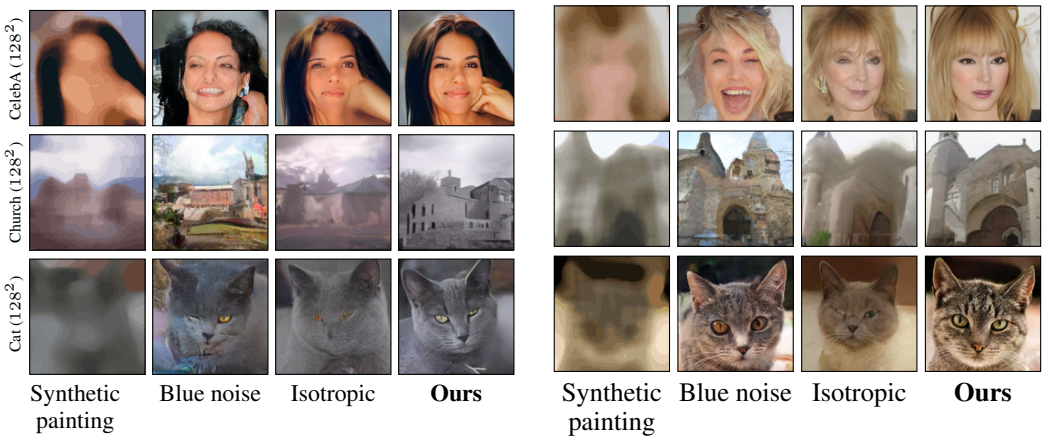


Figure 8: More samples for our model and other baselines applied to SDEdit (Meng et al., 2022). Note how our model is able to generate sharper results that suffer less from artifacts. Although BNDM can generate satisfactory results in certain cases (e.g., cat and church), it often deviates from the stroke painting guide, potentially producing outcomes that differ significantly from the user's original intent. In contrast, our method closely follows the stroke painting guide, accurately preserving both shape and color.

Figure 9: Impact of location of transition point t_Φ on sample quality, shown for the LSUN-Church (128^2) dataset. If we place t_Φ too far, the model happens to learn only the lowest frequencies and generates no details at all. Placing it too early leads to results that are less sharp. We found that by placing t_Φ at 50%, we strike a good balance between the two, leading to better quantitative and qualitative results.

Table 3: Quantitative FID score comparison on latent diffusion models (Rombach et al., 2022) between DDPM (Ho et al., 2020) and our method.

Unconditional FID (↓)	AFHQ-Cat(512^2, latent)
DDPM	22.86
Ours	**18.91**

Table 4: Shape-guided image generation (based on SDEdit (Meng et al., 2022)): precision (metric for realism) and recall (metric for diversity) scores (Kynkäänniemi et al., 2019) for isotropic model DDPM, and our edge-preserving model. We consistently outperform in terms of precision, and closely match in terms of recall.

	Ours		Isotropic noise	
Shape-guided image generation	Precision (↑)	Recall (↑)	Precision (↑)	Recall (↑)
AFHQ-Cat(128^2)	**0.93**	**0.80**	0.92	0.66
CelebA(128^2)	**0.65**	0.46	0.53	**0.53**
LSUN-Church(128^2)	**0.87**	0.46	0.84	**0.50**

Table 5: Unconditional image generation: precision (metric for realism) and recall (metric for diversity) scores for isotropic model DDPM, and our edge-preserving model. While our model slightly get outperformed, we find that our edge-preserving model closely matches DDPM on both metrics. We would therefore argue that edge-preserving noise minimally impacts diversity.

	Ours		Isotropic noise	
Unconditional image generation	Precision (↑)	Recall (↑)	Precision (↑)	Recall (↑)
AFHQ-Cat(128^2)	0.76	0.20	**0.77**	**0.21**
CelebA(128^2)	0.90	0.16	**0.92**	**0.17**
LSUN-Church(128^2)	**0.65**	0.33	0.47	**0.38**

Table 6: Additional CLIP-based comparisons (Radford et al., 2021) for stroke-guided generation (Meng et al., 2022) show that our method consistently outperforms the isotropic baseline, producing images that are more semantically aligned with the originals.

CLIP	Ours	DDPM
AFHQ-Cat(128^2)	**88.97**	88.78
CelebA(128^2)	**61.15**	61.02
LSUN-Church(128^2)	**64.32**	62.57

DDPM Ours

Figure 10: More unconditional samples for DDPM (isotropic noise) and our edge-preserving noise on the AFHQ-Cat (128^2) dataset.

DDPM Ours

Figure 11: More unconditional samples for DDPM (isotropic noise) and our edge-preserving noise on the CelebA (128^2) dataset.

DDPM Ours

Figure 12: More unconditional samples for DDPM (isotropic noise) and our edge-preserving noise on the LSUN-Church (128^2) dataset. Although our results appear similar to DDPM's, our method more effectively captures the geometric details of buildings and exhibits fewer artifacts, such as blurry regions, compared to DDPM.

DDPM Ours

Figure 13: More unconditional samples for DDPM (isotropic noise) and our edge-preserving noise on the AFHQ-Cat (512^2, LDM) dataset. All samples are generated via diffusion in latent space. While difference in visual quality is subtle, Table 3 shows that our noise framework improved the FID score.

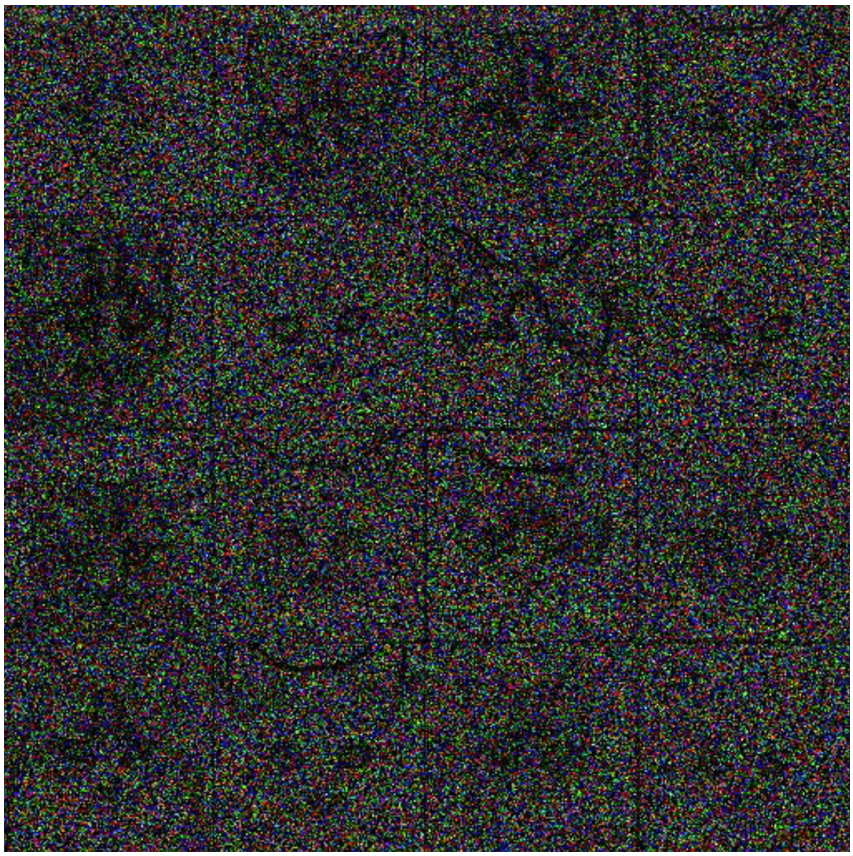


Figure 14: Grid of predicted noises for a batch of 16 samples after fine-tuning an isotropic model pre-trained for 2000 epochs on the AFHQ-Cat (128^2) dataset. After fewer than 5k fine-tuning iterations with edge-preserving noise, the model has already learned the non-isotropic variance corresponding to the structures in the data.