SFBD: A METHOD FOR TRAINING DIFFUSION MODELS WITH NOISY DATA

Haoye Lu*, **Qifan Wu & Yaoliang Yu***
David R. Cheriton School of Computer Science
University of Waterloo
ON, N2L 5Z5, Canada
{haoye.lu, ryan.wu1, yaoliang.yu}@uwaterloo.ca

ABSTRACT

Recent diffusion-based generative models achieve remarkable results by training on massive datasets, yet this practice raises concerns about memorization and copyright infringement. A proposed remedy is to train exclusively on noisy data with potential copyright issues, ensuring the model never observes original content. However, through the lens of deconvolution theory, we show that although it is theoretically feasible to learn the data distribution from noisy samples, the practical challenge of collecting sufficient samples makes successful learning nearly unattainable. To overcome this limitation, we propose to pretrain the model with a small fraction of clean data to guide the deconvolution process. Combined with our Stochastic Forward–Backward Deconvolution (SFBD) method, we attain an FID of 6.31 on CIFAR-10 with just 4% clean images (and 3.58 with 10%). Theoretically, we prove that SFBD guides the model to learn the true data distribution. The result also highlights the importance of pretraining on limited but clean data or the alternative from similar datasets. Empirical studies further support these findings and offer additional insights.

1 INTRODUCTION

Diffusion-based generative models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021a;b; 2023) have gained increasing attention. Nowadays, it is considered one of the most powerful frameworks for learning high-dimensional distributions and we have witnessed many impressive breakthroughs (Croitoru et al., 2023) in generating images (Ho et al., 2020; Song et al., 2021a;b; Rombach et al., 2022), audios (Kong et al., 2021; Yang et al., 2023) and videos (Ho et al., 2022).

Due to some inherent properties, diffusion models are relatively easier to train. This unlocks the possibility of training very large models on web-scale data, which has been shown to be critical to train powerful models. This paradigm has recently led to impressive advances in image generation, as demonstrated by cutting-edge models like Stable Diffusion (-XL) (Rombach et al., 2022; Podell et al., 2024) and DALL-E (2, 3) (Betker et al., 2023). However, despite their success, the reliance on extensive web-scale data introduces challenges. The complexities of the datasets at such a scale often result in the inclusion of copyrighted content. Furthermore, diffusion models exhibit a greater tendency than earlier generative approaches, such as Generative Adversarial Networks (GANs) (Goodfellow et al., 2014; 2020), to memorize training examples. This can lead to the replication of parts or even entire images from their training sets (Carlini et al., 2023; Somepalli et al., 2023).

A recent approach to mitigating memorization and copyright concerns trains diffusion models on corrupted samples (Daras et al., 2023b; Somepalli et al., 2023; Daras & Dimakis, 2023; Daras et al., 2024). In this framework, models never see original data; instead, samples undergo a non-invertible corruption process, like adding Gaussian noise, preventing memorization and reproduction. Interestingly, under mild assumptions, certain non-invertible corruption processes, such as Gaussian noise injection, create a mathematical bijection between the noisy and original distributions. Thus, in theory, a generative model can learn the original distribution using only noisy samples (Bora et al.,

*Haoye Lu and Yaoliang Yu are also affiliated with the Vector Institute in Toronto, Canada.

2018). Building on this concept, Daras et al. (2024) demonstrated that when an image is corrupted via a forward diffusion up to a specific noise level σ, diffusion models can recover distributions at noise levels below σ by enforcing consistency constraints (Daras et al., 2023a).

While Daras et al. (2024) empirically showed that their approach could be used to fine-tune Stable Diffusion XL (Podell et al., 2024) using noisy images with a heuristic consistency loss, they did not explore whether a diffusion model can be successfully trained solely with noisy images. Moreover, the effectiveness of the consistency loss in such scenarios remains an open question.

In this paper, we address these questions by connecting the task of estimating the original distribution from noisy samples to the well-studied density deconvolution problem (Meister, 2009). Through the lens of deconvolution theory, we establish that the optimal convergence rate for estimating the data density is $\mathcal{O}(\log n)^{-2}$ when n noisy samples are generated via a forward diffusion process. This pessimistic rate suggests that while it is theoretically feasible to learn the data distribution from noisy samples, the practical challenge of collecting sufficient samples makes successful learning nearly unattainable. Our empirical studies further validate this theoretical insight and suggest the inefficiency of the current consistency loss outside the regime of fine-tuning latent diffusion models.

To address the poor convergence rate in training diffusion models with noisy data, we propose pretraining models on a small subset of copyright-free clean data as an effective solution. Since the current consistency loss remains ineffective even with pretraining, we propose a new deconvolution method, Stochastic Forward–Backward Deconvolution (SFBD, pronounced `sofabed`), that is fully compatible with the existing diffusion training framework. Experimentally, we achieve an FID of 6.31 with just 4% clean images on CIFAR-10 and 3.58 with 10% clean images. Our theoretical results ensure that the learnt distribution converges to true data distribution and justifies the necessity of pretraining. Furthermore, our results suggest that models can be pretrained using datasets with similar features when clean, copyright-free data are unavailable. Ablation studies provide additional evidence supporting our claims.

A very recent study by Daras et al. (2025), using Gaussian Mixture Models, also highlights the challenge of training diffusion models with only noisy samples and shows that adding a few clean samples can significantly improve performance. The convergence of conclusions from fundamentally different approaches reinforces the findings of both works.

2 PRELIMILARIES

In this section, we recall diffusion models, the density deconvolution problem and the consistency constraints.

2.1 DIFFUSION MODELS

Diffusion models generate data by progressively adding Gaussian noise to input data and then reversing this process through sequential denoising steps to sample from noise. Given distribution p_0 on $\mathbb{R}^d$, the forward perturbation is specified by a stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x}_t = g(t)\,\mathrm{d}\mathbf{w}_t, \quad t \in [0, T], \tag{1}$$

$\mathbf{x}_0 \sim p_0$, T is a fixed positive constant and $g(t)$ is a scalar function. $\{\mathbf{w}_t\}_{t\in[0,T]}$ is the standard Brownian motion.

Eq (1) induces a transition kernel $p_{t|s}(\mathbf{x}_t|\mathbf{x}_s)$ for $0 \le s \le t \le T$, which is Gaussian and its mean and covariance matrix can be computed in closed form (Särkkä & Solin, 2019, Eqs 4.23 and 5.51). In particular, for $s = 0$, we write

$$p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0, \sigma_t^2\mathbf{I}), \tag{2}$$

for all $t \in [0, T]$, where we set $g(t) = (\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t})^{1/2}$. When σ_T^2 is very large, $\mathbf{x}_T$ can be approximately regarded as a sample from $\mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$. Let $p_t(\mathbf{x}_t) = \int p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\, p_0(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0$ denote the marginal distribution of $\mathbf{x}_t$, where we have $p_T \approx \mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$. Anderson (1982) showed that backward SDE

$$\mathrm{d}\mathbf{x}_t = -g(t)^2\,\nabla \log p_t(\mathbf{x}_t)\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t,\ \mathbf{x}_T \sim p_T \tag{3}$$

has a transition kernel that matches the posterior distribution of the forward process, $p_{s|t}(\mathbf{x}_s|\mathbf{x}_t) = \frac{p_{t|s}(\mathbf{x}_t|\mathbf{x}_s)p_s(\mathbf{x}_s)}{p_t(\mathbf{x}_t)}$ for $s \leq t$ in $[0, T]$. Thus, the backward SDE preserves the same marginal distributions as the forward process. Here, $\bar{\mathbf{w}}_t$ represents a standard Wiener process with time flowing backward from T to 0, while $\nabla \log p_t(\mathbf{x}_t)$ denotes the score function of the distribution $p_t(\mathbf{x}_t)$. With a well-trained network $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) \approx \nabla \log p_t(\mathbf{x}_t)$, we substitute it into Eq (3) and solve the SDE backward from $\tilde{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$. The resulting $\tilde{\mathbf{x}}_0$ then serves as an approximate sample of p_0.

To train $\mathbf{s}_{\boldsymbol{\phi}}$ to estimate the score, let $\mathcal{T}$ be a sampler of $t \in [0, T]$ and $w(t)$ a weight function. The network $\mathbf{s}_{\boldsymbol{\phi}}$ is then trained via the conditional score-matching loss (Song et al., 2021b):

$$\mathcal{L}_s(\boldsymbol{\phi}) = \mathop{\mathbb{E}}_{t\sim\mathcal{T}} \mathop{\mathbb{E}}_{p_0} \mathop{\mathbb{E}}_{p_{t|0}} \left[w(t) \|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \nabla \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\|^2 \right].$$

Instead, we may train a denoiser $D_{\boldsymbol{\phi}}(\mathbf{x}, t)$ to estimate $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ by minimizing (Karras et al., 2022)

$$\mathcal{L}_d(\boldsymbol{\phi}) = \mathop{\mathbb{E}}_{t\sim\mathcal{T}} \mathop{\mathbb{E}}_{p_0} \mathop{\mathbb{E}}_{p_{t|0}} \left[w(t) \|D_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \mathbf{x}_0\|^2 \right] \tag{4}$$

then estimate

$$\nabla \log p_t(\mathbf{x}_t) = \big(\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] - \mathbf{x}_t\big)/\sigma_t^2 \approx \big(D_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \mathbf{x}_t\big)/\sigma_t^2. \tag{5}$$

2.2 Density Deconvolution Problems

Classical deconvolution problems arise in scenarios where data are corrupted due to significant measurement errors, and the goal is to estimate the underlying data distribution. Specifically, let the corrupted samples $\mathcal{Y} = \{\mathbf{y}^{(i)}\}_{i=1}^n$ be generated by the process:

$$\mathbf{y}^{(i)} = \mathbf{x}^{(i)} + \boldsymbol{\epsilon}^{(i)}, \tag{6}$$

where $\mathbf{x}^{(i)}$ and $\boldsymbol{\epsilon}^{(i)}$ are independent random variables. Here, $\mathbf{x}^{(i)}$ is drawn from an unknown distribution with density p_{data}, and $\boldsymbol{\epsilon}^{(i)}$ is sampled from a *known* error distribution with density h. It can be shown that the corrupted samples $\mathbf{y}^{(i)}$ follow a distribution with density $p_{\text{data}} * h$, where $*$ denotes the convolution operator. We provide more details in Appx A.

The objective of the (density) deconvolution problem is to estimate the density of p_{data} using the observed data $\mathcal{Y}$, which is sampled from the convoluted distribution $p_{\text{data}} * h$. In essence, deconvolution reverses the density convolution process, hence the name of the problem.

To assess the quality of an estimator $\hat{p}(\cdot; \mathcal{Y})$ of p_{data} based on $\mathcal{Y}$, the mean integrated squared error (MISE) is commonly used. MISE is defined as:

$$\text{MISE}(\hat{p}, p_{\text{data}}) = \mathbb{E}_{\mathcal{Y}} \int_{\mathbb{R}^d} \left| \hat{p}(\mathbf{x}; \mathcal{Y}) - p_{\text{data}}(\mathbf{x}) \right|^2 \mathrm{d}\mathbf{x}. \tag{7}$$

In this paper, we focus on a corruption process implemented via forward diffusion as described in Eq (1). Consequently, unless otherwise stated, in the rest of this work, we assume the error distribution h is Gaussian $\mathcal{N}(\mathbf{0}, \sigma_\zeta^2 \mathbf{I})$ with a given and fixed $\zeta \in (0, T)$.

To see why we could identify an original distribution p through $p * h$, let $\Phi_p(\mathbf{u}) = \mathbb{E}_p[\exp(i\, \mathbf{u}^\top \mathbf{x})]$ for $\mathbf{u} \in \mathbb{R}^d$ be the characteristic function of p. Then,

Proposition 1. *Let p and q be two distributions defined on $\mathbb{R}^d$. For all $\mathbf{u} \in \mathbb{R}^d$,*

$$|\Phi_p(\mathbf{u}) - \Phi_q(\mathbf{u})| \leq \exp\big(\frac{\sigma_\zeta^2}{2}\|\mathbf{u}\|^2\big) \sqrt{2\, D_{\text{KL}}(p * h \| q * h)}.$$

(All proofs are deferred to the appendix.) This result shows if two distributions p and q are similar after being convoluted with h, they must have similar characteristic functions and thus similar distribution. In particular, when $p * h = q * h$, then $p = q$, the case also discussed in Wang et al. (2023, Thm 2). As a result, whenever we could find q satisfying $p_{\text{data}} * h = q * h$, we can conclude $p_{\text{data}} = q$.

2.3 Deconvolution through the Consistency Constraints

While Prop 1 shows it is possible to train a generative model using noisy samples, it remains a difficult question of how to use noisy samples to train a diffusion model to generate clean samples *effectively*.

The question was partially addressed by Daras et al. (2024) through the consistency property (Daras et al., 2023a). In particular, since we have access to the noisy samples $\mathbf{x}_\zeta$ from $p_{\text{data}} * h$, we can use them to train a network $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ to approximate $\nabla \log p_t(\mathbf{x}_t)$ for $t > \zeta$ through a modified score matching loss, which is referred as ambient score matching (ASM), denoted by $\mathcal{L}_{\text{ASM}}(\boldsymbol{\phi})$. In their implementation, $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ is parameterized by $\frac{D_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t^2}$, where $D_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ is trained to approximate $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$. In contrast, for $t \leq \zeta$, score-matching is no longer applicable. Instead, Daras et al. (2024) propose that $D_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ should obey the consistency property:

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_s] = \mathbb{E}_{p_{r|s}}\big[\mathbb{E}[\mathbf{x}_0|\mathbf{x}_r]\big], \text{ for } 0 \leq r \leq s \leq T \tag{8}$$

by jointly minimizing the *consistency loss*:

$$\mathcal{L}_{\text{con}}(\boldsymbol{\phi}, r, s) = \mathbb{E}_{p_s}\big\|D_{\boldsymbol{\phi}}(\mathbf{x}_s, s) - \mathbb{E}_{p_{r|s}}[D_{\boldsymbol{\phi}}(\mathbf{x}_r, r)]\big\|^2, \tag{9}$$

where r and s are sampled from predefined distributions. Sampling from $p_{r|s}$ is implemented by solving Eq (3) backward from $\mathbf{x}_s$, replacing the score function with the network-estimated one $D_{\boldsymbol{\phi}}$ via Eq (5). For sampling from p_s, we first sample $\mathbf{x}_\tau$ for $\tau > s$ and $\tau > \zeta$, then sample from $p_{s|\tau}$ in a manner analogous to sampling from $p_{r|s}$.

It can be shown that if $D_{\boldsymbol{\phi}}$ minimizes the consistency loss for all r, s and perfectly learns the score function for $t > \zeta$, then $\frac{D_{\boldsymbol{\phi}}(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t^2}$ becomes an exact estimator of the score function for all $t \in [0, T]$. Consequently, the distribution $p_0 = p_{\text{data}}$ can be sampled by solving Eq (3).

Daras et al. (2024) demonstrated the effectiveness of this framework only in fine-tuning latent diffusion models, leaving its efficacy when training from scratch unreported. Moreover, as sampling from $p_{r|s}$ depends on the model's approximation of the score (which is particularly challenging to estimate accurately for $t < \zeta$) rather than the ground truth, there remains a gap between the theoretical framework and its practical implementation. This gap limits the extent to which the algorithm's effectiveness is supported by their theoretical results.

3 Theoretical Limit of Deconvolution

In this section, we analyze the complexity of deconvolution when data corruption follows a forward diffusion process. Using deconvolution theory, we show that while Daras et al. (2024) demonstrated that diffusion models can be trained on noisy samples, obtaining enough samples for high-quality training is practically infeasible.

The following two theorems establish that the optimal convergence rate for estimating the data density is $\mathcal{O}(\log n)^{-2}$. These results, derived using standard deconvolution theory (Meister, 2009) under a Gaussian noise assumption, highlight the inherent difficulty of the problem. We present the result for $d = 1$, which suffices to illustrate the challenge.

Theorem 1. *Assume $\mathcal{Y}$ is generated according to Eq (6) with $\epsilon \sim \mathcal{N}(0, \sigma_\zeta^2)$ and p_{data} is a univariate distribution. Under some weak assumptions on p_{data}, for a sufficiently large sample size n, there exists an estimator $\hat{p}(\cdot; \mathcal{Y})$ such that*

$$\text{MISE}(\hat{p}, p_{data}) \leq C\ \sigma_\zeta^4 \cdot (\log n)^{-2}, \tag{10}$$

where C is determined by p_{data}.

Theorem 2. *In the same setting as Thm 1, for an arbitrary estimator $\hat{p}(\cdot; \mathcal{Y})$ of p_{data} based on $\mathcal{Y}$,*

$$\text{MISE}(\hat{p}, p_{data}) \geq K \cdot (\log n)^{-2}, \tag{11}$$

where $K > 0$ is determined by p_{data} and error distribution h.

The optimal convergence rate $\mathcal{O}(\log n)^{-2}$ indicates that reducing the MISE to one-fourth of its current value requires an additional $n^2 - n$ samples. In contrast, under the error-free scenario, the optimal convergence rate is known to be $\mathcal{O}(n^{-4/5})$ (Wand, 1998), where reducing the MISE to one-fourth of its current value would only necessitate approximately $4.657n$ additional samples.

The pessimistic rate indicates that effectively training a generative model using only corrupted samples with Gaussian noise is nearly impossible. Thus, this implies that training from scratch, using

only noisy images, with the consistency loss discussed in Sec 2.3, is infeasible. Notably, as indicated by Eq (10), this difficulty is significantly more severe with larger σ_ζ^2, while a large σ_ζ^2 is typically required to alter the original samples significantly to address copyright and privacy concerns.

To mitigate the pessimistic statistical rate, we propose pretraining diffusion models on a small set of copyright-free samples. Although limited, this data offers valuable priors, initializing the model closer to the true distribution than random weights. In image generation, for instance, it helps the model learn common structures like continuity, smoothness, edges, and typical object appearances.

Unfortunately, our empirical study in Sec 5 will show that the consistency loss-based method discussed in Sec 2.3 cannot deliver promising results even after pretraining. We suspect that this is caused by the gap between their theoretical framework and the practical implementation. As a result, we propose SFBD in Sec 4 to bridge such a gap.

4 Stochastic Forward–Backward Deconvolution

In this section, we introduce a novel method for solving the deconvolution problem that integrates seamlessly with the existing diffusion model framework. As our approach involves iteratively applying the forward diffusion process described in Eq (1), followed by a backward step with an optimized drift, we refer to this method as Stochastic Forward-Backward Deconvolution (SFBD), as described in Alg 1.

Algorithm 1 Stochastic Forward–Backward Deconvolution. (Given sample set $\mathcal{D}$, $p_{\mathcal{D}}$ denotes the corresponding empirical distribution.)

Input: clean data: $\mathcal{D}_{\text{clean}} = \{\mathbf{x}^{(i)}\}_{i=1}^{M}$, noisy data: $\mathcal{D}_{\text{noisy}} = \{\mathbf{y}_\tau^{(i)}\}_{i=1}^{N}$, number of iterations: K.

// Initialize Denoiser

$\boldsymbol{\phi}_0 \leftarrow$ Pretrain $D_{\boldsymbol{\phi}}$ using Eq (4) with $p_0 = p_{\mathcal{D}_{\text{clean}}}$

for $k = 1$ *to* K **do**

  // Backward Sampling

  $\mathcal{E}_k \leftarrow \{\mathbf{y}_0^{(i)} : \forall \mathbf{y}_\tau^{(i)} \in \mathcal{D}_{\text{noisy}}$, solve backward SDE Eq (3) from τ to 0, starting from $\mathbf{y}_\tau^{(i)}$, where the score function is estimated as $\frac{D_{\boldsymbol{\phi}_{k-1}}(\mathbf{x}_t,t)-\mathbf{x}_t}{\sigma_t^2}\}$

  // Denoiser Update

  $\boldsymbol{\phi}_k \leftarrow$ Train $D_{\boldsymbol{\phi}}$ by minimizing Eq (4) with $p_0 = p_{\mathcal{E}_k}$

Output: Final denoiser $D_{\boldsymbol{\phi}_K}$

The proposed algorithm begins with a small set of clean data, $\mathcal{D}_{\text{clean}}$, for pretraining, followed by iterative optimization using a large set of noisy samples. As demonstrated in Sec 5, decent quality images can be achieved on datasets such as CIFAR-10 (Krizhevsky & Hinton, 2009) and CelebA (Liu et al., 2015) using as few as 50 clean images. During pretraining, the algorithm produces a neural network denoiser, $D_{\boldsymbol{\phi}_0}$, which serves as the initialization for the subsequent iterative optimization process. Specifically, the algorithm alternates between the following two steps: for $k = 1, 2, \dots K$,

1. (Backward Sampling) This step can be intuitively seen as a denoising process for samples in $\mathcal{D}_{\text{noisy}}$ using the backward SDE Eq (3). In each iteration, we use the best estimation of the score function so far induced by $D_{\boldsymbol{\phi}_{k-1}}$ through Eq (5).
2. (Denoiser Update) Fine-tune denoiser $D_{\boldsymbol{\phi}_{k-1}}$ to obtain $D_{\boldsymbol{\phi}_k}$ by minimizing Eq (4) with the denoised samples obtained in the previous step.

The following proposition shows that when $\mathcal{D}_{\text{noisy}}$ contains sufficiently many samples to characterize the true noisy distribution $p_{\text{data}} * h$, when $K \to \infty$, the diffusion model implemented by denoiser $D_{\boldsymbol{\phi}_K}$ has the sample distribution converging to the true p_{data}.

Proposition 2. *Let p_t^* be the density of $\mathbf{x}_t$ obtained by solving the forward diffusion process Eq (1) with $\mathbf{x}_0 \sim p_{data}$, where we have $p_\zeta^* = p_{\text{data}} * h$. Consider a modified Alg 1, where the empirical distribution $P_{\mathcal{D}_{noisy}}$ is replaced with the ground truth p_ζ^*. Correspondingly, $p_{\mathcal{E}_k}$ becomes $p_0^{(k)}$, the distribution of $\mathbf{x}_0$ induced by solving:*

$$\mathrm{d}\mathbf{x}_t = -g(t)^2\, \mathbf{s}_{\boldsymbol{\phi}_{k-1}}(\mathbf{x}_t, t)\, \mathrm{d}t + g(t)\, \mathrm{d}\bar{\mathbf{w}}_t,\ \mathbf{x}_\zeta \sim p_\zeta^* \tag{12}$$

from ζ to 0, where $\mathbf{s}_{\boldsymbol{\phi}_k}(\mathbf{x}_t, t) = \frac{D_{\boldsymbol{\phi}_k}(\mathbf{x}_t,t)-\mathbf{x}_t}{\sigma_t^2}$, $g(t) = (\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t})^{1/2}$ and $D_{\boldsymbol{\phi}_k}$ is obtained by minimizing (4) according to Alg 1. Assume $D_{\boldsymbol{\phi}_k}$ reaches the optimal for all k. Under mild assumptions,

$$D_{\mathrm{KL}}(p_{data} \parallel p_0^{(k)}) \geq D_{\mathrm{KL}}(p_{data} \parallel p_0^{(k+1)}). \tag{13}$$

for all $k \geq 0$. In addition, for all $K \geq 1$ and $\mathbf{u} \in \mathbb{R}^d$, we have

$$\min_{k=1,\dots K} \left| \Phi_{p_{data}}(\mathbf{u}) - \Phi_{p_0^{(k)}}(\mathbf{u}) \right| \leq \exp\left(\frac{\sigma_\zeta^2}{2}\|\mathbf{u}\|^2\right) \sqrt{\frac{2M_0}{K}},$$

where $M_0 = \frac{1}{2}\int_0^\zeta g(t)^2 \mathbb{E}_{p_t^*} \left\| \nabla \log p_t^*(\mathbf{x}_t) - \mathbf{s}_{\boldsymbol{\phi}_0}(\mathbf{x}_t, t) \right\|^2 \mathrm{d}t.$

Prop 2 shows that after sufficient iterations of backward sampling and denoiser updates, the denoised sample distribution converges to the true data distribution at a rate of $\mathcal{O}(1/\sqrt{K})$. Thus, after fine-tuning the denoiser on these denoised samples during the Denoiser Update step, the diffusion model is expected to generate samples that approximately follow the data distribution, solving the deconvolution problem. Note that this result describes the convergence rate of SFBD under the assumption of an infinite number of noisy samples, which is distinct from the optimal sample efficiency rate discussed in Sec 3.

The importance of pretraining. Prop 2 also highlights the critical role of pretraining, as it allows the algorithm to begin fine-tuning from a point much closer to the true data distribution. Specifically, effective pretraining ensures that $\mathbf{s}_{\boldsymbol{\phi}_0}$ closely approximates the ground-truth score, leading to a smaller M_0 in Prop 2. This, in turn, reduces the number of iterations K required for the diffusion model to generate high-quality samples.

The practical limits of increasing K. While Prop 2 suggests that increasing the number of iterations K can continuously improve sample quality, practical limitations come into play. Sampling errors introduced during the backward sampling process, as well as imperfections in the denoiser updates, accumulate over time. These errors eventually offset the benefits of additional iterations, as demonstrated in Sec 5. This observation further highlights the importance of pretraining to mitigate the impact of such errors and achieve high-quality samples with fewer iterations.

Alternative methods for backward sampling. While the backward sampling in Alg 1 is presented as a naive solution to the backward SDE in Eq (3), the algorithm is not limited to this approach. Any backward SDE and solver yielding the same marginal distribution as Eq (3) can be employed. Alternatives include PF-ODE, the predictor-corrector sampler (Song et al., 2021b), DEIS (Zhang & Chen, 2023), and the 2nd order Heun method used in EDM (Karras et al., 2022). Compared to the Euler–Maruyama method, these approaches require fewer network evaluations and offer improved error control for imperfect score estimation and step discretization. As the algorithm generates $\mathcal{E}_k$ that contains samples closer to p_{data} with increasing k, clean images used for pretraining can be incorporated into $\mathcal{E}_k$ to accelerate this process. In our empirical study, this technique is applied whenever clean samples and noisy samples (prior to corruption) originate from the same distribution.

Relationship to the consistency loss. SFBD can be seen as an algorithm that enforces the consistency constraint across all positive time steps and time zero. Specifically, we have

Proposition 3. *Assume that the denoising network $D_{\boldsymbol{\phi}}$ is implemented to satisfy $D_{\boldsymbol{\phi}}(\cdot, 0) = Id(\cdot)$. When $r = 0$, the consistency loss in Eq (9) is equivalent to the denoising noise in Eq (4) for $t = s$.*

The requirement that $D_{\boldsymbol{\phi}}(\cdot, 0) = \mathrm{Id}(\cdot)$ is both natural and intuitive, as $D_{\boldsymbol{\phi}}(\mathbf{x}_0, 0)$ approximates $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_0] = \mathbf{x}_0$. This fact is explicitly enforced in the design of the EDM framework (Karras et al., 2022), which has been widely adopted in subsequent research.

A key distinction between SFBD and the original consistency loss implementation is that SFBD does not require sampling from $p_{r|s}$ or access to the ground-truth score function induced by the unknown data distribution p_{data}. This is because, in the original implementation, $p_0 = p_{\text{data}}$, whereas in SFBD, $p_0 = p_0^{(k)}$, as defined in Prop 2, and is obtained iteratively through the backward sampling step. As k increases, $p_0^{(k)}$ converges to p_{data}, ensuring that the same consistency constraints are eventually enforced. Consequently, SFBD bridges the gap between theoretical formulation and practical implementation that exists in the original consistency loss framework.

5 Empirical Study

In this section, we demonstrate the effectiveness of the SFBD framework proposed in Sec 4. Compared to other models trained on noisy datasets, SFBD consistently achieves superior performance across all benchmark settings. Additionally, we conduct ablation studies to validate our theoretical findings and offer practical insights for applying SFBD effectively.

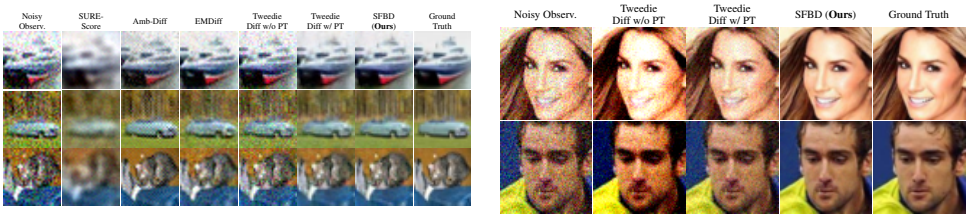
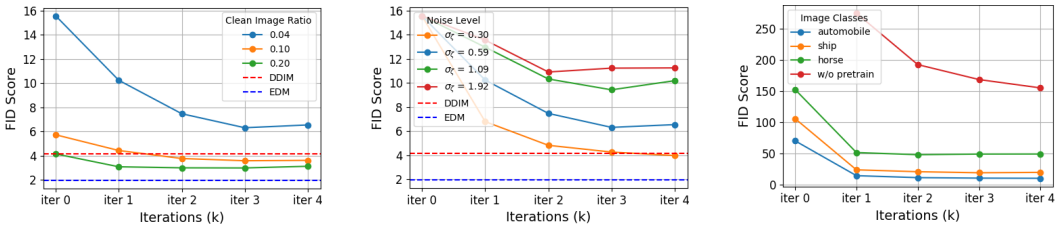


Figure 1: Denoised samples of CIFAR-10 (left) and CelebA (right). (Noise level $\sigma_\zeta = 0.2$)



(a) Clean Image Ratio (b) Noise Level (c) Pretrain on Similar Datasets

Figure 2: SFBD performance on CIFAR-10 under various conditions. Unless specified, the clean image ratio is 0.04 and the noise level σ_ζ is 0.59. In (a) and (b), FID at iteration 0 corresponds to the pretrained model. In (c), models are pretrained on clean images from the "truck" class, with FID at iteration 0 measuring the distance between these clean images and those used for fine-tuning. For the w/o pretraining setting, models are trained on the full CIFAR-10 dataset with $\sigma_\zeta = 0.59$.

Datasets and evaluation metrics. The experiments are conducted on the CIFAR-10 (Krizhevsky & Hinton, 2009) and CelebA (Liu et al., 2022) datasets, with resolutions of 32×32 and 64×64, respectively. CIFAR-10 consists of 50,000 training images and 10,000 test images across 10 classes. CelebA, a dataset of human face images, includes a predefined split of 162,770 training images, 19,867 validation images, and 19,962 test images. For CelebA, images were obtained using the preprocessing tool provided in the DDIM official repository (Song et al., 2021a).

We evaluate image quality using the Frechet Inception Distance (FID), computed between the reference dataset and 50,000 images generated by the models. Generated samples for FID computation are presented in Appx D.

Models and other configurations. We implemented SFBD algorithms using the architectures proposed in EDM (Karras et al., 2022) as well as the optimizers and hyperparameter configurations therein. All models are implemented in an unconditional setting, and we also enabled the non-leaky augmentation technique (Karras et al., 2022) to alleviate the overfitting problem. For the backward sampling step in SFBD, we adopt the 2$^{\text{nd}}$-order Heun method (Karras et al., 2022). More information is provided in Appx E.

Table 1: Model performance comparison. When $\sigma_\zeta > 0$, the models are trained on noisy images. Underscored results are produced by this work.

Method	CIFAR10 (32 x 32)			CelebA (64 x 64)		
	σ_ζ	Pretrain	FID	σ_ζ	Pretrain	FID
DDPM (Ho et al., 2020)	0.0	No	4.04	0.0	No	**3.26**
DDIM (Song et al., 2021a)	0.0	No	4.16	0.0	No	6.53
EDM (Karras et al., 2022)	0.0	No	**1.97**	-	-	-
SURE-Score (Aali et al., 2023)	0.2	Yes	132.61	-	-	-
EMDiff (Bai et al., 2024)	0.2	Yes	86.47	-	-	-
TweedieDiff (Daras et al., 2024)	0.2	No	167.23	0.2	No	246.95
TweedieDiff (Daras et al., 2024)	0.2	Yes	65.21	0.2	Yes	58.52
SFBD (Ours)	0.2	Yes	**13.53**	0.2	Yes	**6.49**

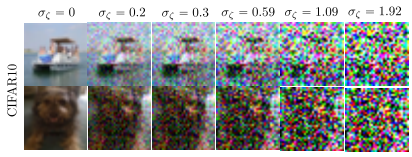


Figure 3: Noisy images with various σ_ζ.

5.1 Performance Comparison

We compare SFBD with representative models for training on noisy images (Table 1). SURE-Score (Aali et al., 2023) and EMDiffusion (Bai et al., 2024) use Stein's unbiased risk estimate and expectation-maximization, respectively, for inverse problems. TweedieDiffusion (Daras et al., 2024) applies the original consistency loss Eq (9). Daras et al. (2025) improved TweedieDiffusion performance through a simplified implementation of the consistency loss. A detailed comparison with the optimized TweedieDiffusion model will be provided in the full version of this work.

Following the experimental setup of Bai et al. (2024), images are corrupted by adding independent Gaussian noise with a standard deviation of $\sigma_\zeta = 0.2$ to each pixel after rescaling pixel values to $[-1, 1]$. For reference, we also include results for models trained on clean images ($\sigma_\zeta = 0$). In cases with pretraining, the models are initially trained on 50 clean images randomly sampled from the training datasets. For all results presented in this work, the same set of 50 sampled images is used.

Table 1 shows SFBD produces images of significantly higher quality than all baselines, as further illustrated by the denoised images in Fig 1 by evaluating the backward SDE starting from a noisy image in the training dataset. Notably, on CelebA, SFBD achieves performance comparable to DDIM, which is trained on clean images. While TweedieDiffusion benefits from pretraining, its results remain inferior to SFBD. In fact, we observe that the original consistency loss Eq (9) offers limited performance improvement after pretraining; the FID begins to degrade soon after its application.

5.2 Ablation Study

In this section, we investigate how SFBD's performance varies with clean image ratios, noise levels, and pretraining on similar datasets. The results align with our discussion in Sec 3 and Sec 4 and provide practical insights. Experiments are conducted on CIFAR-10, with the default $\sigma_\zeta = 0.59$. This noise level significantly alters the original images, aligning with our original motivation to address potential copyright concerns (see Fig 3).

Clean image ratio. Fig 2(a) shows the FID trajectories across fine-tuning iterations k for different clean image ratios. With just 4% clean images, SFBD achieves strong performance (FID: 6.31) and outperforms DDIM with 10% clean images. While higher clean image ratios further improve performance, the gains diminish as a small amount of clean data already provides sufficient high-frequency features (e.g., edges and local details) to capture feature variations. Since these features are shared across images, additional clean data offers limited improvement.

These findings suggest that practitioners with limited clean datasets should focus on collecting more copyright-free data to enhance performance. Notably, when clean images are scarce, the marginal gains from additional fine-tuning iterations k are greater than when more clean data is available. Therefore, in scenarios where acquiring clean data is challenging, increasing fine-tuning iterations can be an effective alternative to improve results.

Noise level. Fig 2(b) shows SFBD's sampling performance across fine-tuning iterations for different noise levels, using the values from 2$^{\text{nd}}$ order Heun sampling in EDM (Karras et al., 2022). The impact of noise on the original images is visualized in Fig 3. As shown in Fig 2(b), increasing σ_ζ significantly degrades SFBD's performance. This is expected, as higher noise levels obscure more features in the original images. Furthermore, as suggested by Thm 1, higher σ_ζ demands substantially more noisy images, which cannot be compensated by pretraining on a small clean image set. Importantly, this performance drop is a mathematical limitation discussed in Sec 3, rather than an issue solvable by better deconvolution algorithms.

Pretraining with clean images from similar datasets. Fig 2(c) evaluates SFBD's performance when fine-tuning on image sets from different classes, with the model initially pretrained on clean truck images. The results show that the closer the noisy dataset is to the truck dataset (as indicated by the FID at iter 0), the better the model performs after fine-tuning. This is expected, as similar datasets share common features that facilitate learning the target data distribution. Interestingly, even when the pretraining dataset differs significantly from the noisy dataset, the model still outperforms the version without pretraining. This is because unrelated datasets often share fundamental features, such as edges and local structures. *Therefore, practitioners should always consider pretraining before fine-tuning on target noisy datasets, while more similar pretraining datasets yield better final sampling performance.*

6 Conclusion

We presented SFBD, a new deconvolution method based on diffusion models. Under mild assumptions, we theoretically showed that our method could guide diffusion models to learn the true data distribution through training on noisy samples. The empirical study corroborates our claims and shows that our model consistently achieves state-of-the-art performance in some benchmark tasks.

ACKNOWLEDGMENTS

We gratefully acknowledge funding support from NSERC and the Canada CIFAR AI Chairs program. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

REFERENCES

Asad Aali, Marius Arvinte, Sidharth Kumar, and Jonathan I. Tamir. Solving inverse problems with score-based generative priors learned from noisy data. In *57th Asilomar Conference on Signals, Systems, and Computers*, pp. 837–843, 2023. URL https://doi.org/10.1109/IEEECONF59524.2023.10477042.

B D O Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. URL https://doi.org/10.1016/0304-4149(82)90051-5.

Weimin Bai, Yifei Wang, Wenzheng Chen, and He Sun. An expectation-maximization algorithm for training clean diffusion models from corrupted observations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=jURBh4V9N4.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. OpenAI, 2023. URL https://cdn.openai.com/papers/dall-e-3.pdf.

Ashish Bora, Eric Price, and Alexandros G. Dimakis. AmbientGAN: Generative models from lossy measurements. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hy7fDog0b.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium*, pp. 5253–5270, 2023. URL https://www.usenix.org/system/files/usenixsecurity23-carlini.pdf.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869, 2023. URL https://doi.org/10.1109/TPAMI.2023.3261988.

Giannis Daras and Alex Dimakis. Solving inverse problems with ambient diffusion. In *NeurIPS 2023 Workshop on Deep Learning and Inverse Problems*, 2023. URL https://openreview.net/forum?id=mGwg10bgHk.

Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. In *Advances in Neural Information Processing Systems*, pp. 42038–42063, 2023a. URL https://openreview.net/forum?id=GfZGdJHj27.

Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alex Dimakis, and Adam Klivans. Ambient diffusion: Learning clean distributions from corrupted data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=wBJBLy9kBY.

Giannis Daras, Alex Dimakis, and Constantinos Costis Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=PlVjIGaFdH.

Giannis Daras, Yeshwanth Cherapanamjeri, and Constantinos Costis Daskalakis. How much is a noisy image worth? data scaling laws for ambient diffusion. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=qZwtPEw2qN.

Ian 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*, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. URL https://doi.org/10.1145/3422622.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=f3zNgKga_ep.

Thomas Kailath. The Structure of Radon-Nikodym Derivatives with Respect to Wiener and Related Measures. *The Annals of Mathematical Statistics*, 42(3):1054–1067, 1971. URL https://doi.org/10.1214/aoms/1177693332.

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*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*. 2015. URL https://arxiv.org/abs/1412.6980.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=XVjTT1nw5z.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. URL http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.

Alexander Meister. *Deconvolution Problems in Nonparametric Statistics*. Springer, 2009. URL https://doi.org/10.1007/978-3-540-87557-4.

Bernt Oksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, 6th edition, 2003. URL https://doi.org/10.1007/978-3-642-14394-6.

Michele Pavon and Anton Wakolbinger. On free energy, stochastic control, and Schrödinger processes. In *Modeling, Estimation and Control of Systems with Uncertainty: Proceedings of a Conference held in Sopron, Hungary, September 1990*, pp. 334–348. 1991. URL https://doi.org/10.1007/978-1-4612-0443-5_22.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

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)*, pp. 10684–10695, 2022. URL https://doi.org/10.1109/CVPR52688.2022.01042.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2256–2265, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6048–6058, 2023. URL https://doi.org/10.1109/CVPR52729.2023.00586.

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

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*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Leonard A. Stefanski and Raymond J. Carroll. Deconvolving kernel density estimators. *Statistics*, 21 (2):169–184, 1990. URL https://doi.org/10.1080/02331889008802238.

Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Cambridge University Press, 2019. URL https://doi.org/10.1017/9781108186735.

Alexandre Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009. URL https://doi.org/10.1007/b13794.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving schrodinger bridges via maximum likelihood. *Entropy*, 23(9), 2021. URL https://www.mdpi.com/1099-4300/23/9/1134.

M.P. Wand. Finite sample performance of deconvolving density estimators. *Statistics & Probability Letters*, 37(2):131–139, 1998. URL https://www.sciencedirect.com/science/article/pii/S0167715297001107.

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-GAN: Training GANs with diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=HZf7UbpWHuA.

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1720–1733, 2023. URL https://doi.org/10.1109/TASLP.2023.3268730.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Loek7hfb46P.

A A BRIEF INTRODUCTION TO THE DENSITY CONVOLUTIONS

In this section, we give a brief discussion on the density convolution and how it is related to our problem.

For simplicity, we stick to the case when $d = 1$. Consider the data generation process in Eq (6). Let p_y denote the density of the distribution of the noisy samples $y^{(i)}$. Then we have

Fact 1. *For $\omega \in \mathbb{R}$,*

$$p_y(\omega) = \int p_{data}(x)\ h(\omega - x)\,\mathrm{d}x = (p_{data} * h)(\omega). \tag{14}$$

Proof. This is because, for all measurable function ψ, we have

$$\begin{aligned}\int \psi(\omega) p_y(\omega)\,\mathrm{d}\omega &= \int\int \psi(x+\epsilon)\ p_{\text{data}}(x) h(\epsilon)\,\mathrm{d}x\,\mathrm{d}\epsilon = \int\int \psi(\omega) p_{\text{data}}(x) h(\omega - x)\,\mathrm{d}x\,\mathrm{d}w \\ &= \int \psi(w) \left[\int p_{\text{data}}(x)\ h(\omega - x)\,\mathrm{d}x\right]\,\mathrm{d}\omega.\end{aligned}$$

As the equality holds for all ψ, we have $p_y(\omega) = \int p_{\text{data}}(x)\ h(\omega - x)\,\mathrm{d}x = (p_{\text{data}} * h)(\omega)$. □

As a result, according to Fact 1, the density convolution is naturally involved in our setting.

Then, we provide an alternative way to show why we can recover p_{data} given p_y and h. (Namely, we need to deconvolute p_y to obtain p_{data}.) Our discussion can be seen a complement of the discussion following Prop 1. Let ϕ_p denote the characteristic function of the random variable with distribution p such that

$$\phi_p(t) = \int \exp(it\omega)\ p(\omega)\,\mathrm{d}\omega. \tag{15}$$

We note that the characteristic function of a density p is its Fourier transform. As a result, through the dual relationship of multiplication and convolution under Fourier transformation (Meister, 2009, Lemma A.5), we have

$$\phi_{p_y}(t) = \phi_{p_{\text{data}}}(t)\ \phi_h(t). \tag{16}$$

As a result, given noisy data distribution p_y and noise distribution h, we have

$$\phi_{p_{\text{data}}}(t) = \frac{\phi_{p_y}(t)}{\phi_h(t)}. \tag{17}$$

Finally, we can recover p_{data} through an inverse Fourier transform:

$$p_{\text{data}}(x) = (2\pi)^{-1} \int \exp(-itx)\ \phi_{p_{\text{data}}}(t)\,\mathrm{d}t = (2\pi)^{-1} \int \exp(-itx)\ \frac{\phi_{p_y}(t)}{\phi_h(t)}\,\mathrm{d}t. \tag{18}$$

We conclude this section by summarizing the relationship between data and noisy sample distributions in Fig 4.

$$\begin{array}{ccc} p_{\text{data}} & \underset{\text{deconvolution}}{\overset{\text{convolution}}{\rightleftarrows}} & p_y = p_{\text{data}} * h \\ \wr & & \wr \\ x^{(i)} & \underset{\text{irreversible}}{\overset{\text{add } \epsilon^{(i)} \sim h}{\rightleftarrows}} & y^{(i)} = x^{(i)} + \epsilon^{(i)} \end{array}$$

Figure 4: While the corruption process is irreversible at the sample level, a bijective relationship exists between the clean and noisy data distributions.

B PROOFS RELATED TO DECONVOLUTION THEORY

We first show the result suggesting it is possible to identify a distribution through its noisy version obtained by corrupting its samples by injecting independent Gaussian noises.

Proposition 1. *Let p and q be two distributions defined on $\mathbb{R}^d$. For all $\mathbf{u} \in \mathbb{R}^d$,*

$$|\Phi_p(\mathbf{u}) - \Phi_q(\mathbf{u})| \leq \exp\big(\frac{\sigma_\zeta^2}{2}\|\mathbf{u}\|^2\big)\sqrt{2\, D_{\mathrm{KL}}(p * h \| q * h)}.$$

Lemma 1. *Given two distributions p and q on $\mathbb{R}^d$. Let $\Phi_p(\mathbf{u})$ and $\Phi_q(\mathbf{u})$ be their characteristic functions. Then for all $\mathbf{u} \in \mathbb{R}^d$, we have*

$$\big|\Phi_p(\mathbf{u}) - \Phi_q(\mathbf{u})\big| \;\leq\; \sqrt{2\, D_{\mathrm{KL}}(p \parallel q)}. \tag{19}$$

Proof. We note that

$$\Phi_p(\mathbf{u}) \;=\; \mathbb{E}_p[\exp(i\mathbf{u}^\top \mathbf{x})], \quad \Phi_q(\mathbf{u}) \;=\; \mathbb{E}_q[\exp(i\mathbf{u}^\top \mathbf{x})].$$

Then for any $\mathbf{u} \in \mathbb{R}^d$, we have

$$\begin{aligned}
\big|\Phi_p(\mathbf{u}) - \Phi_q(\mathbf{u})\big| &\leq \left|\int_{\mathbb{R}^d} \exp(i\mathbf{u}^\top \mathbf{x}) p(\mathbf{x})\, \mathrm{d}\mathbf{x} - \int_{\mathbb{R}^d} \exp(i\mathbf{u}^\top \mathbf{x}) q(\mathbf{x})\, \mathrm{d}\mathbf{x}\right| \\
&= \left|\int_{\mathbb{R}^d} \exp(i\mathbf{u}^\top \mathbf{x})\Big(p(\mathbf{x}) - q(\mathbf{x})\Big)\, \mathrm{d}\mathbf{x}\right| \leq \int_{\mathbb{R}^d} \underbrace{\big|\exp(i\mathbf{u}^\top \mathbf{x})\big|}_{=1} |p(\mathbf{x}) - q(\mathbf{x})|\, \mathrm{d}\mathbf{x} \\
&= \int_{\mathbb{R}^d} |p(\mathbf{x}) - q(\mathbf{x})|\, \mathrm{d}\mathbf{x} \\
&= 2\, \|p - q\|_{\mathrm{TV}},
\end{aligned}$$

where the last equality is due to Scheffe's theorem (Tsybakov, 2009, Lemma 2.1, p. 84)).

Then, by Pinsker's inequality (Tsybakov, 2009, Lemma 2.5, p. 88), we have

$$\big|\Phi_p(\mathbf{u}) - \Phi_q(\mathbf{u})\big| \;\leq\; 2\, \|p - q\|_{\mathrm{TV}} \;\leq\; \sqrt{2\, D_{\mathrm{KL}}(P \parallel Q)}.$$

which completes the proof. □

Proof of Prop 1. Note that, by the convolution theorem (Meister, 2009, A.4), for all $\mathbf{u} \in \mathbb{R}^d$, we have

$$\Phi_{p*h}(\mathbf{u}) = \Phi_p(\mathbf{u})\; \Phi_h(\mathbf{u}) = \Phi_p(\mathbf{u})\; \exp\big(-\frac{\sigma_\zeta^2}{2}\|\mathbf{u}\|^2\big),$$

as $h \sim \mathcal{N}(\mathbf{0}, \sigma_\zeta^2 \mathbf{I})$ having $\Phi_h(\mathbf{u}) = \exp\big(-\frac{\sigma_\zeta^2}{2}\|\mathbf{u}\|^2\big)$. Applying Lem 1, we have

$$\exp\big(-\frac{\sigma_\zeta^2}{2}\|\mathbf{u}\|^2\big)\; \Big|\Phi_p(\mathbf{u}) - \Phi_q(\mathbf{u})\Big| = \big|\Phi_{p*h}(\mathbf{u}) - \Phi_{q*h}(\mathbf{u})\big| \leq \sqrt{2\, D_{\mathrm{KL}}(p * h \| q * h)}. \tag{20}$$

Rearranging the inequality completes the proof. □

We then derive the proofs regarding the sample complexity of the deconvolution problem.

Theorem 1. *Assume $\mathcal{Y}$ is generated according to Eq (6) with $\epsilon \sim \mathcal{N}(0, \sigma_\zeta^2)$ and p_{data} is a univariate distribution. Under some weak assumptions on p_{data}, for a sufficiently large sample size n, there exists an estimator $\hat{p}(\cdot; \mathcal{Y})$ such that*

$$\mathrm{MISE}(\hat{p}, p_{data}) \leq C\, \sigma_\zeta^4 \cdot (\log n)^{-2}, \tag{10}$$

where C is determined by p_{data}.

Proof. The result is constructed based on the work by Stefanski & Carroll (1990). In particular, assuming that p_{data} is continuous, bounded and has two bounded integrable derivatives such that

$$\int p''_{\text{data}}(x)\,\mathrm{d}x < \infty, \tag{21}$$

we can construct a kernel based estimator of p_{data} of rate

$$\frac{\lambda^4}{4}\mu_{K,2}^2 \int p''_{\text{data}}(x)\,\mathrm{d}x, \tag{22}$$

where $\mu_{\kappa,2}^2$ is a constant determined by the selected kernel κ and λ is a function of number of samples n gradually decreasing to zero as $n \to \infty$. It is required that λ satisfies

$$\frac{1}{2\pi n\lambda}\exp(\frac{B^2\sigma_\zeta^2}{\lambda^2}) \to 0 \tag{23}$$

as $n \to \infty$, where $B > 0$ is a constant depending on the picked kernel κ. Here, we assume we picked a kernel with $B < 1$.

To satisfy the constraint, we choose $\lambda(n) = \frac{\sigma_\zeta}{\sqrt{\log n}}$. Plugging it into Eq (23), we have

$$\lim_{n\to\infty}\frac{1}{n\lambda}\exp(\frac{B^2\sigma_\zeta^2}{\lambda^2}) = \lim_{n\to\infty}\frac{\sqrt{\log n}}{n\sigma_\zeta}\exp\left(B^2\log n\right) = \lim_{n\to\infty}\frac{\sqrt{\log n}}{n^{1-B^2}\sigma_\zeta}. \tag{24}$$

To show $\lim_{n\to\infty}\frac{\sqrt{\log n}}{n^{1-B^2}\sigma_\zeta} = 0$, it suffices to show $\lim_{n\to\infty}\frac{\log n}{n^{2-2B^2}\sigma_\zeta^2} = 0$. By L'Hopital's rule, we have

$$\lim_{n\to\infty}\frac{\log n}{n^{2-2B^2}\sigma_\zeta^2} = \lim_{n\to\infty}\frac{1}{(2-2B^2)n^{2-2B^2}\sigma_\zeta^2} = 0 \tag{25}$$

As a result, $\lambda(n) = \frac{\sigma_\zeta}{\sqrt{\log n}}$ is a valid choice, which gives the convergence rate $\frac{\sigma_\zeta^4}{(\log n)^2}$. □

Theorem 2. *In the same setting as Thm 1, for an arbitrary estimator $\hat{p}(\cdot;\mathcal{Y})$ of p_{data} based on $\mathcal{Y}$,*

$$\mathrm{MISE}(\hat{p}, p_{data}) \geq K\cdot(\log n)^{-2}, \tag{11}$$

where $K > 0$ is determined by p_{data} and error distribution h.

Proof. This result is a special case of Theorem 2.14 (b) in (Meister, 2009). When the error density is Gaussian, we have $\gamma = 2$. In addition, in the proof of Thm 1, we assumed that p_{data} has two bounded integrable derivatives, which equivalently assumes p_{data} satisfies the Soblev condition with smoothness degree $\beta = 2$ (see Eq. A.8, Meister 2009). Then the theorem shows $\mathrm{MISE}(\hat{p}, p_{\text{data}}) \geq \text{const}\cdot(\log n)^{-2\beta/\gamma} = \text{const}\cdot(\log n)^{-2}$. □

C Proofs Related to the Results of SFBD

We first prove Prop 2, which we restate below:

Proposition 2. *Let p_t^* be the density of $\mathbf{x}_t$ obtained by solving the forward diffusion process Eq (1) with $\mathbf{x}_0 \sim p_{data}$, where we have $p_\zeta^* = p_{\text{data}} * h$. Consider a modified Alg 1, where the empirical distribution $P_{\mathcal{D}_{noisy}}$ is replaced with the ground truth p_ζ^*. Correspondingly, $p_{\mathcal{E}_k}$ becomes $p_0^{(k)}$, the distribution of $\mathbf{x}_0$ induced by solving:*

$$\mathrm{d}\mathbf{x}_t = -g(t)^2\,\mathbf{s}_{\boldsymbol{\phi}_{k-1}}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t,\ \mathbf{x}_\zeta \sim p_\zeta^* \tag{12}$$

from ζ to 0, where $\mathbf{s}_{\boldsymbol{\phi}_k}(\mathbf{x}_t, t) = \frac{D_{\boldsymbol{\phi}_k}(\mathbf{x}_t,t)-\mathbf{x}_t}{\sigma_t^2}$, $g(t) = (\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t})^{1/2}$ and $D_{\boldsymbol{\phi}_k}$ is obtained by minimizing (4) according to Alg 1. Assume $D_{\boldsymbol{\phi}_k}$ reaches the optimal for all k. Under mild assumptions,

$$D_{\mathrm{KL}}(p_{data} \parallel p_0^{(k)}) \geq D_{\mathrm{KL}}(p_{data} \parallel p_0^{(k+1)}). \tag{13}$$

for all $k \geq 0$. In addition, for all $K \geq 1$ and $\mathbf{u} \in \mathbb{R}^d$, we have

$$\min_{k=1,\dots K}\left|\Phi_{p_{data}}(\mathbf{u}) - \Phi_{p_0^{(k)}}(\mathbf{u})\right| \leq \exp\left(\frac{\sigma_\zeta^2}{2}\|\mathbf{u}\|^2\right)\sqrt{\frac{2M_0}{K}},$$

where $M_0 = \frac{1}{2}\int_0^\zeta g(t)^2\mathbb{E}_{p_t^*}\left\|\nabla\log p_t^*(\mathbf{x}_t) - \mathbf{s}_{\boldsymbol{\phi}_0}(\mathbf{x}_t, t)\right\|^2\mathrm{d}t$.

To facilitate our discussions, let

- $\overleftarrow{Q}^{\boldsymbol{\phi}_{k-1}}_{0:\zeta}$: the path measure induced by the backward process Eq (12). In general, we use $\overleftarrow{Q}^{\boldsymbol{\phi}}_{0:\zeta}$ to denote the path measure when the drift term is parameterized $\boldsymbol{\phi}$.
- $\overrightarrow{P}^{(k)}_{0:\zeta}$: the path measure induced by the forward process Eq (1) with $p_0 = p_0^{(k)}$, defined in Prop 2. The density of its marginal distribution at time t is denoted by $p_t^{(k)}$
- $\overrightarrow{P}^{*}_{0:\zeta}$: the path measure induced by the forward process Eq (1) with $p_0 = p_{\text{data}}$.

We note that, according to Alg 1, the marginal distribution of $\overleftarrow{Q}^{\boldsymbol{\phi}_{k-1}}_{0:\zeta}$ at $t = 0$ has density $p_0^{(k)}$.

The following lemma allows us to show that the training of the diffusion model can be seen as a process of minimizing the KL divergence of two path measures.

Lemma 2 (Pavon & Wakolbinger 1991, Vargas et al. 2021)**.** *Given two SDEs:*

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}_i(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_0^{(i)}(\mathbf{x}) \quad t \in [0, T] \tag{26}$$

for $i = 1, 2$. Let $P^{(i)}_{0:T}$, for $i = 1, 2$, be the path measure induced by them, respectively. Then we have,

$$D_{\mathrm{KL}}(P^{(1)}_{0:T} \parallel P^{(2)}_{0:T}) = D_{\mathrm{KL}}(p_0^{(1)} \parallel p_0^{(2)}) \;+\; \mathbb{E}_{P^{(1)}_{0:T}}\Big[\int_0^T \frac{1}{2\,g(t)^2}\,\|\mathbf{f}_1(\mathbf{x}_t, t) - \mathbf{f}_2(\mathbf{x}_t, t)\|^2\,dt\Big]. \tag{27}$$

In addition, the same result applies to a pair of backward SDEs as well, where $p_0^{(i)}$ is replaced with $p_T^{(i)}$.

Proof. By the disintegration theorem (e.g., see Vargas et al. 2021, Appx B), we have

$$D_{\mathrm{KL}}(P_1 \parallel P_2) = D_{\mathrm{KL}}(p_0^{(1)} \parallel p_0^{(2)}) \;+\; \mathbb{E}_{P^{(1)}_{0:T}}\left[\log \frac{\mathrm{d}P^{(1)}_{0:T}(\cdot|\mathbf{x}_0))}{\mathrm{d}P^{(2)}_{0:T}(\cdot|\mathbf{x}_0)}\right], \tag{28}$$

where $P^{(i)}_{0:T}(\cdot|\mathbf{x}_0)$ is the conditioned path measure of $P^{(i)}_{0:T}$ given the initial point $\mathbf{x}_0$. Then, applying the Girsanov theorem (Kailath, 1971; Oksendal, 2003) on the second term yields the desired result. □

By Lem 2, we can show that the Denoiser Update step in Alg 1 finds $\boldsymbol{\phi}_k$ minimizing $D_{\mathrm{KL}}(\overrightarrow{P}^{(k)}_{0:\zeta} \parallel \overleftarrow{Q}^{\boldsymbol{\phi}}_{0:\zeta})$. To see this, note that

$$\begin{aligned}
\boldsymbol{\phi}_k &= \operatorname*{argmin}_{\boldsymbol{\phi}} D_{\mathrm{KL}}(\overrightarrow{P}^{(k)}_{0:\zeta} \parallel \overleftarrow{Q}^{\boldsymbol{\phi}}_{0:\zeta}) \\
&= \operatorname*{argmin}_{\boldsymbol{\phi}} D_{\mathrm{KL}}(p_\zeta^{(k)} \parallel p_\zeta^{*}) \;+\; \mathbb{E}_{\overrightarrow{P}^{(k)}_{0:\zeta}}\Big[\int_0^\zeta \frac{g(t)^2}{2}\,\|\nabla \log p_t^{(k)}(\mathbf{x}_t) - \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)\|^2\,dt\Big],
\end{aligned} \tag{29}$$

where $p_t^{(k)}$ is the marginal distribution induced by the forward process (1) with the boundary condition $p_0^{(k)}$ at $t = 0$. Note that, we have applied Lem 2 to the backward processes inducing $\overrightarrow{P}^{(k)}_{0:\zeta}$ and $\overleftarrow{Q}^{\boldsymbol{\phi}}_{0:\zeta}$. Thus, the drift term of $\overrightarrow{P}^{(k)}_{0:\zeta}$ is not zero but $-g(t)^2\,\nabla \log p_t^{(k)}(\mathbf{x}_t)$ according to Eq (3). Since the first term of Eq (29) is a constant, the minimization results in

$$\nabla \log p_t^{(k)}(\mathbf{x}_t) = \mathbf{s}_{\boldsymbol{\phi}_k}(\mathbf{x}_t, t) \tag{30}$$

for all $\mathbf{x}_t \in \mathbb{R}^d$ and $t \in (0, \zeta]$. In addition, we note that, the denoising loss in Eq (4) is minimized when $\nabla \log p_t^{(k)}(\mathbf{x}_t) = \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ for all $t > 0$; as a result, $\boldsymbol{\phi}_k$ minimizes $D_{\mathrm{KL}}(\overrightarrow{P}^{(k)}_{0:\zeta} \parallel \overleftarrow{Q}^{\boldsymbol{\phi}}_{0:\zeta})$ as claimed.

Now, we are ready to prove Prop 2.

Proof of Prop 2. Applying Lem 2 to the backward process

$$D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_{k-1}}_{0:\zeta}) = \underbrace{D_{\mathrm{KL}}(p^*_\zeta \parallel p^*_\zeta)}_{=0} + \mathbb{E}_{\overrightarrow{P}^*_{0:\zeta}}\left[\int_0^\zeta \tfrac{g(t)^2}{2}\|\nabla \log p^*_t(\mathbf{x}_t) - \mathbf{s}_{\phi_{k-1}}(\mathbf{x}_t, t)\|^2 \,\mathrm{d}t\right]$$
$$= \mathbb{E}_{\overrightarrow{P}^*_{0:\zeta}}\left[\int_0^\zeta \tfrac{g(t)^2}{2}\|\nabla \log p^*_t(\mathbf{x}_t) - \mathbf{s}_{\phi_{k-1}}(\mathbf{x}_t, t)\|^2 \,\mathrm{d}t\right] \tag{31}$$

Likewise,

$$\begin{aligned} D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overrightarrow{P}^{(k)}_{0:\zeta}) =&\ D_{\mathrm{KL}}(p^*_\zeta \parallel p^{(k)}_\zeta) + \mathbb{E}_{\overrightarrow{P}^*_{0:\zeta}}\left[\int_0^\zeta \tfrac{g(t)^2}{2}\|\nabla \log q^*_t(\mathbf{x}_t) - \nabla \log p^{(k)}_t(\mathbf{x}_t)\|^2 \,\mathrm{d}t\right] \\ =&\ D_{\mathrm{KL}}(p^*_\zeta \parallel p^{(k)}_\zeta) + \mathbb{E}_{\overrightarrow{P}^*_{0:\zeta}}\left[\int_0^\zeta \tfrac{g(t)^2}{2}\|\nabla \log q^*_t(\mathbf{x}_t) - \mathbf{s}_{\phi_k}(\mathbf{x}_t, t)\|^2 \,\mathrm{d}t\right] \\ \overset{(31)}{=}&\ D_{\mathrm{KL}}(p^*_\zeta \parallel p^{(k)}_\zeta) + D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_k}_{0:\zeta}) \end{aligned} \tag{32}$$

where the second equality is due to the discussion on deriving Eq (30).

Lem 2 also implies that

$$D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_{k-1}}_{0:\zeta}) = D_{\mathrm{KL}}(p_{\text{data}} \parallel p^{(k)}_0) + \underbrace{\mathbb{E}_{\overrightarrow{P}^*_{0:\zeta}}\left[\int_0^\zeta \tfrac{1}{2}\|\mathbf{b}^{(k-1)}(\mathbf{x}_t, t)\|^2 \,\mathrm{d}t\right]}_{:=\mathcal{B}_{k-1}}, \tag{33}$$

where $\mathbf{b}^{(k-1)}(\mathbf{x}_t, t)$ is the drift of the forward process inducing $\overleftarrow{Q}^{\phi_{k-1}}_{0:\zeta}$. In addition,

$$D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overrightarrow{P}^{(k)}_{0:\zeta}) = D_{\mathrm{KL}}(p_{\text{data}} \parallel p^{(k)}_0) + \mathbb{E}_{\overrightarrow{P}^*_{0:\zeta}}\left[\int_0^\zeta \tfrac{1}{2}\|\mathbf{0} - \mathbf{0}\|^2 \,\mathrm{d}t\right] = D_{\mathrm{KL}}(p_{\text{data}} \parallel p^{(k)}_0). \tag{34}$$

As a result,

$$\begin{aligned} D_{\mathrm{KL}}(p_{\text{data}} \parallel p^{(k)}_0) \overset{(34)}{=} & D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overrightarrow{P}^{(k)}_{0:\zeta}) \overset{(32)}{=} D_{\mathrm{KL}}(p^*_\zeta \parallel p^{(k)}_\zeta) + D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_k}_{0:\zeta}) \\ \geq & D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_k}_{0:\zeta}) \overset{(33)}{=} D_{\mathrm{KL}}(p_{\text{data}} \parallel p^{(k+1)}_0) + \mathcal{B}_k \\ \geq & D_{\mathrm{KL}}(p_{\text{data}} \parallel p^{(k+1)}_0) \end{aligned}$$

which is (13). In addition, we have

$$\begin{aligned} D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_{k-1}}_{0:\zeta}) \overset{(33)}{=} & D_{\mathrm{KL}}(p_{\text{data}} \parallel p^{(k)}_0) + \mathcal{B}_{k-1} \overset{(34)}{=} D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overrightarrow{P}^{(k)}_{0:\zeta}) + \mathcal{B}_{k-1} \\ \overset{(32)}{=} & D_{\mathrm{KL}}(p^*_\zeta \parallel p^{(k)}_\zeta) + D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_k}_{0:\zeta}) + \mathcal{B}_{k-1} \\ = & D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_k}_{0:\zeta}) + \big[D_{\mathrm{KL}}(p^*_\zeta \parallel p^{(k)}_\zeta) + \mathcal{B}_{k-1}\big]. \end{aligned}$$

As a result, applying this relationship recursively, we have

$$D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_0}_{0:\zeta}) = \sum_{k=1}^K D_{\mathrm{KL}}(p^*_\zeta \parallel p^{(k)}_\zeta) + \sum_{k=1}^K \mathcal{B}_{k-1} + D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_K}_{0:\zeta}). \tag{35}$$

Since $D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \parallel \overleftarrow{Q}^{\phi_0}_{0:\zeta}) = M_0$, we have

$$\sum_{k=1}^K D_{\mathrm{KL}}(p_{\text{data}} * h \parallel p^{(k)} * h) = \sum_{k=1}^K D_{\mathrm{KL}}(p^*_\zeta \parallel p^{(k)}_\zeta) \leq M_0, \tag{36}$$

for all $K \geq 1$. This further implies,

$$\min_{k \in \{1,2,\ldots,K\}} D_{\mathrm{KL}}(p_{\text{data}} * h \parallel p^{(k)} * h) \leq \frac{M_0}{K}. \tag{37}$$

Applying Prop 1, we obtain,

$$\min_{k \in \{1,2,\ldots,K\}} \left| \Phi_{p_{\text{data}}}(\mathbf{u}) - \Phi_{p_0^{(k)}}(\mathbf{u}) \right| \leq \exp\big(\frac{\sigma_\zeta^2}{2}\|\mathbf{u}\|^2\big)\sqrt{\frac{2M_0}{K}}. \tag{38}$$

□

We complete this section by showing the connection between our framework and the original consistency loss.

Proposition 3. *Assume that the denoising network $D_{\boldsymbol{\phi}}$ is implemented to satisfy $D_{\boldsymbol{\phi}}(\cdot, 0) = Id(\cdot)$. When $r = 0$, the consistency loss in Eq (9) is equivalent to the denoising noise in Eq (4) for $t = s$.*

Proof. When $t = s$, denoising noise in Eq (4) becomes

$$\begin{aligned}
&\mathop{\mathbb{E}}_{p_0}\mathop{\mathbb{E}}_{p_{s|0}}\left[\|D_{\boldsymbol{\phi}}(\mathbf{x}_s, s) - \mathbf{x}_0\|^2\right] = \mathbb{E}_{p_s}\mathbb{E}_{p_{0|s}}\left[\|D_{\boldsymbol{\phi}}(\mathbf{x}_s, s) - \mathbf{x}_0\|^2\right] \\
&= \mathbb{E}_{p_s}\mathbb{E}_{p_{0|s}}\left[\|D_{\boldsymbol{\phi}}(\mathbf{x}_s, s) - \mathbb{E}_{p_{0|s}}[\mathbf{x}_0] + \mathbb{E}_{p_{0|s}}[\mathbf{x}_0] - \mathbf{x}_0\|^2\right] \\
&= \mathbb{E}_{p_s}\mathbb{E}_{p_{0|s}}\left[\|D_{\boldsymbol{\phi}}(\mathbf{x}_s, s) - \mathbb{E}_{p_{0|s}}[\mathbf{x}_0]\|^2\right] + \underbrace{\mathbb{E}_{p_s}\mathbb{E}_{p_{0|s}}\left[\|\mathbb{E}_{p_{0|s}}[\mathbf{x}_0] - \mathbf{x}_0\|^2\right]}_{\text{Const.}} \\
&\qquad + 2\underbrace{\mathbb{E}_{p_s}\mathbb{E}_{p_{0|s}}\left[\left\langle D_{\boldsymbol{\phi}}(\mathbf{x}_s, s) - \mathbb{E}_{p_{0|s}}[\mathbf{x}_0], \mathbb{E}_{p_{0|s}}[\mathbf{x}_0] - \mathbf{x}_0\right\rangle\right]}_{=0} \\
&= \mathbb{E}_{p_s}\left[\|D_{\boldsymbol{\phi}}(\mathbf{x}_s, s) - \mathbb{E}_{p_{0|s}}[\mathbf{x}_0]\|^2\right] + \text{Const.} \\
&= \mathbb{E}_{p_s}\left[\|D_{\boldsymbol{\phi}}(\mathbf{x}_s, s) - \mathbb{E}_{p_{0|s}}[D_{\boldsymbol{\phi}}(\mathbf{x}_0, 0)]\|^2\right] + \text{Const.},
\end{aligned}$$

which is the consistency loss in Eq (9) when $r = 0$. □

D ADDITIONAL SAMPLING RESULTS

In this section, we present model-generated samples used for FID computation in Sec 5.2. The samples are taken from the models at their fine-tuning iteration with the lowest FID.

Samples for computing FIDs in Fig 2(a) - Clean Image Ratio

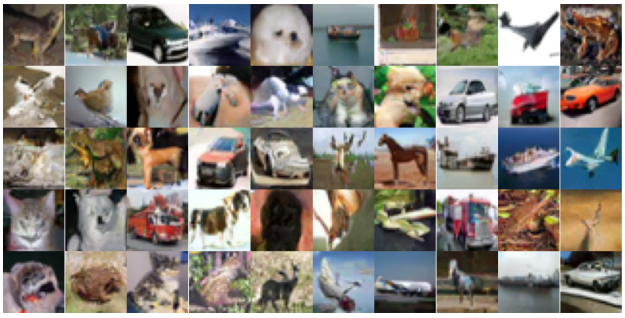


Figure 5: Clean image ratio = 0.04 – FID: 6.31

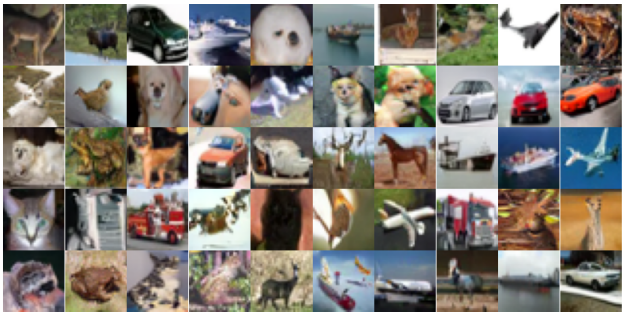


Figure 6: Clean image ratio = 0.1 – FID: 3.58

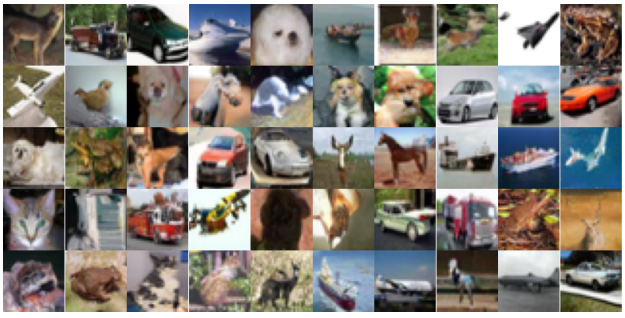


Figure 7: Clean image ratio = 0.2 – FID: 2.98

Samples for computing FIDs in Fig 2(b) - Noise Level

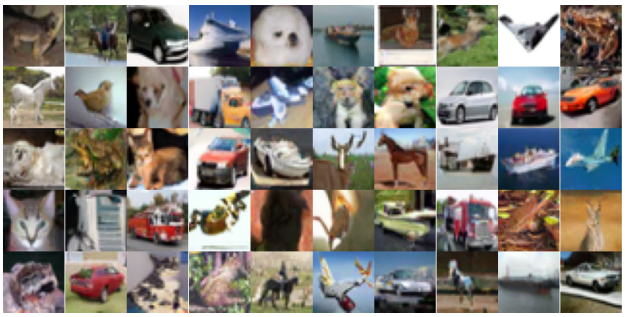


Figure 8: Noise level $\sigma_\zeta = 0.30$ – FID: 3.97

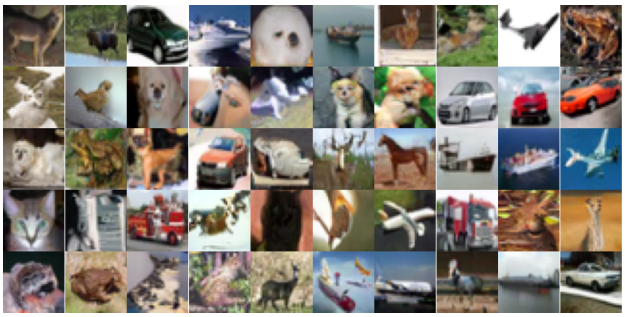


Figure 9: Noise level $\sigma_\zeta = 0.59$ – FID: 6.31

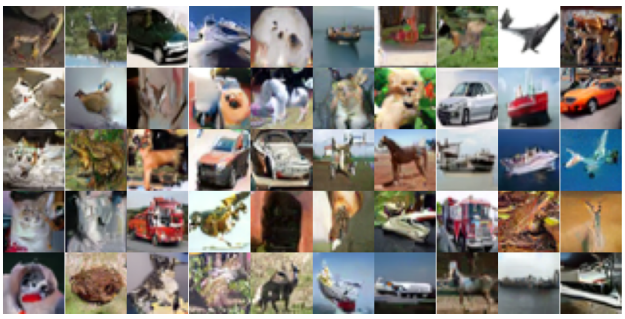


Figure 10: Noise level $\sigma_\zeta = 1.09$ – FID: 9.43

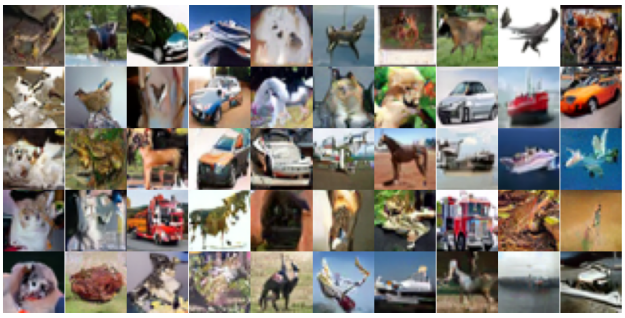


Figure 11: Noise level $\sigma_\zeta = 1.92$ – FID: 10.91

Samples for computing FIDs in Fig 2(c) - Pretraining on Similar Datasets

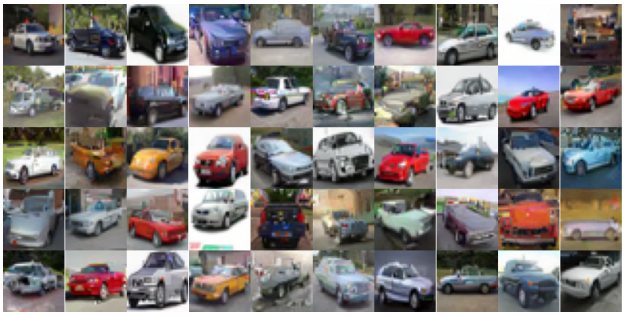


Figure 12: Class for fine-tuning: automobile – FID: 10.39

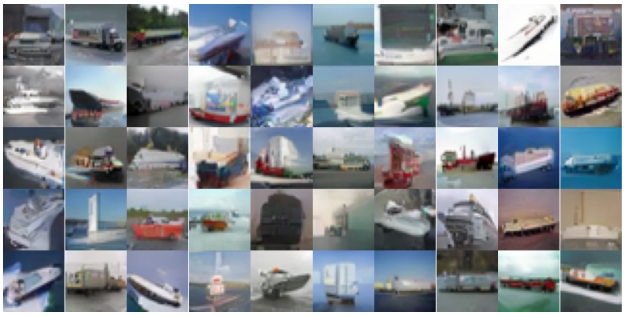


Figure 13: Class for fine-tuning: ship – FID: 19.19

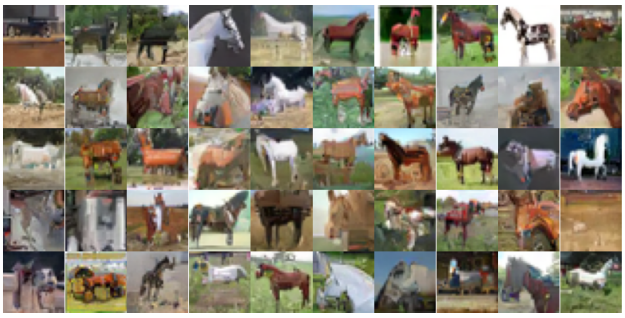


Figure 14: Class for fine-tuning: horse – FID: 48.11

Figure 15: Class for fine-tuning: no pretrain – FID: 155.04

E EXPERIMENT CONFIGURATIONS

E.1 MODEL ARCHITECTURES

We implemented the proposed SFBD algorithm based on the following configurations throughout our empirical studies:

Table 2: Experimental Configuration for CIFAR-10 and CelebA

Parameter	**CIFAR-10**	**CelebA**
General		
Batch Size	512	256
Loss Function	`EDMLoss` (Karras et al., 2022)	`EDMLoss` (Karras et al., 2022)
Sampling Method	2^{nd} order Heun method (EDM) (Karras et al., 2022)	2^{nd} order Heun method (EDM) (Karras et al., 2022)
Sampling steps	18	40
Network Configuration		
Dropout	0.13	0.05
Channel Multipliers	$\{2, 2, 2\}$	$\{1, 2, 2, 2\}$
Model Channels	128	128
Resample Filter	$\{1, 1\}$	$\{1, 3, 3, 1\}$
Channel Mult Noise	1	2
Optimizer Configuration		
Optimizer Class	`Adam` (Kingma & Ba, 2015)	`Adam` (Kingma & Ba, 2015)
Learning Rate	0.001	0.0002
Epsilon	1×10^{-8}	1×10^{-8}
Betas	(0.9, 0.999)	(0.9, 0.999)

E.2 DATASETS

All experiments on CIFAR-10 (Krizhevsky & Hinton, 2009) use only the training set, except for the one presented in Fig 2(c). For this specific test, we merge the training and test sets so that each class contains a total of 6,000 images. At iteration 0, the FID computation measures the distance between clean images of trucks and those from the classes on which the model is fine-tuned. For subsequent iterations, FID is calculated in the same manner as in other experiments. Specifically, the model first generates 50,000 images, and the FID is computed between the sampled images and the images from the fine-tuning classes. All experiments on CelebA (Liu et al., 2015) are performed on its training set.