Stochastic Forward–Backward Deconvolution: Training Diffusion Models with Finite Noisy Datasets

Haoye Lu [1 2] Qifan Wu [1] Yaoliang Yu [1 2]

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 FID 6.31 on CIFAR-10 with just 4% clean images (and 3.58 with 10%). We theoretically show 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; Song et al., 2023), 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 recently proposed approach to address memorization and copyright concerns involves training (or fine-tuning) diffusion models using corrupted samples (Daras et al., 2023b; Somepalli et al., 2023; Daras & Dimakis, 2023; Daras et al., 2024). In this framework, the model is never exposed to the original samples during training. Instead, these samples undergo a known non-invertible corruption process, such as adding independent Gaussian noise to each pixel in image datasets. This ensures that the model cannot memorize or reproduce the original content, as the corruption process is irreversible for individual samples.

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., 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

[1] Cheriton School of Computer Science, University of Waterloo, Canada [2] Vector Institute, Canada. Correspondence to: Haoye Lu <haoye.lu@uwaterloo.ca>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

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 the true data distribution and justify 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. Code for the empirical study is available at: github.com/watml/SFBD.

2. Related Work

The rise of large diffusion models trained on massive datasets has sparked growing concerns about copyright infringement and memorization of training data (Carlini et al., 2023; Somepalli et al., 2023). While differential privacy (DP) has been explored as a mitigation strategy (Abadi et al., 2016; Xie et al., 2018; Dockhorn et al., 2023), it often presents practical challenges. Notably, DP can require users to share their original data with a central server for training unless local devices have sufficient computational power for backpropagation.

In contrast, training on corrupted data provides a compelling alternative, allowing users to contribute without exposing their original data. By sharing only non-invertible, corrupted versions, sensitive information remains on users' devices, eliminating the need to transmit original data.

Learning generative models from corrupted data poses a significant challenge, as the model must reconstruct the underlying data distribution from incomplete or noisy information. In their work on AmbientGAN, Bora et al. (2018) showed that it is empirically feasible to train GANs using corrupted images. They also provided a theoretical guarantee that, with a sufficient number of corrupted samples generated by randomly blacking out pixels, the learned distribution converges to the true data distribution. Building on this, Wang et al. (2023) demonstrated a closely related result: under certain weak assumptions, if the model-generated fake samples and the corrupted true samples share the same distribution after undergoing identical corruption, then the fake data distribution aligns perfectly with the true data distribution. Their analysis applies to scenarios where corruption is implemented via a forward diffusion process but does not address cases where the two corrupted distributions are similar but not identical – a case we explore in Prop 1 below.

Inspired by the success of training GANs using corrupted data, Daras et al. (2023b); Aali et al. (2023); Daras & Dimakis (2023); Bai et al. (2024); Daras et al. (2024) demonstrated the feasibility of training diffusion models with corrupted data. Notably, Daras et al. (2024) showed that when corruption is performed by a forward diffusion process, the marginal distribution at one time step determines the distributions at other time steps, all of which must satisfy certain consistency constraints. Building on this, they showed that if a model learns distributions above the corruption noise level, it can infer those below the noise level by adhering to these constraints. To enforce this, they introduced a consistency loss to improve compliance with the constraints, though its effectiveness was demonstrated only in fine-tuning latent diffusion models.

Outside the field of machine learning, the problem of estimating the original distribution from noisy samples has traditionally been addressed through density deconvolution (Meister, 2009). This research area aims to recover the distribution of error-free data from noise-contaminated observations. Most existing deconvolution methods are limited to the univariate setting (Carroll & Hall, 1988; Zhang, 1990; Fan, 1991; Cordy & Thomas, 1997; Delaigle & Hall, 2008; Meister & Neumann, 2010; Lounici & Nickl, 2011; Guan, 2021), with only a few approaches extending to the multivariate case. These multivariate techniques typically rely on normal mixture models (Bovy et al., 2011; Sarkar et al., 2018) or kernel smoothing methods (Masry, 1993; Lepski & Willer, 2019). Integrating these theoretical insights into modern generative model frameworks remains a significant challenge. However, by reinterpreting generative models trained on noisy data through the lens of deconvolution theory, we can gain a deeper understanding of their fundamental limitations and capabilities, as they inherently address the deconvolution problem.

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. Methodologically, Daras et al. (2025) apply Tweedie's formula (consistency constraint) to recover the clean distribution. In contrast, ours introduces a novel forward-backward deconvolution strategy, offering a fresh perspective without the heavy computational cost of enforcing consistency.

Beyond diffusion model training, Bie et al. (2022) and Ben-David et al. (2023) demonstrate that even a small amount of clean public data can substantially reduce the sample complexity in differentially private (DP) estimation when learning from sensitive data. Although motivated by a different goal, Nie et al. (2022) propose DiffPure, an algorithm that leverages a pretrained diffusion model to remove adversarial perturbations via a forward–backward diffusion process. While DiffPure and SFBD share a similar structure, they serve distinct purposes: DiffPure assumes a well-trained model to purify data, whereas SFBD is designed to train the diffusion model itself.

3. Prelimilaries

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

3.1. 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 $s \le t$ 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 predefined sampler of $t \in [0, T]$ and $w(t)$ be a weight function. The network $\mathbf{s}_{\boldsymbol{\phi}}$ can be effectively 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 first 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

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

3.2. 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:

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

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 $\mathbf{u} \in \mathbb{R}^d$,*

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

(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 $p_{\mathrm{data}} * h = q * h$, we can conclude $p_{\mathrm{data}} = q$.

3.3. 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 $p_{\mathrm{data}} * h$, we can use them to train a network $\mathbf{s}_\phi(\mathbf{x}_t, t)$ 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 $\mathcal{L}_{\mathrm{ASM}}(\phi)$. In their implementation, $\mathbf{s}_\phi(\mathbf{x}_t, t)$ is parameterized by $\frac{D_\phi(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t^2}$, where $D_\phi(\mathbf{x}_t, t)$ 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 $D_\phi(\mathbf{x}_t, t)$ should obey the consistency property:

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_s] = \mathbb{E}_{p_{r|s}}\left[\mathbb{E}[\mathbf{x}_0|\mathbf{x}_r]\right], \text{ for } 0 \leq r \leq s \leq T \quad (8)$$

by jointly minimizing the *consistency loss*:

$$\mathcal{L}_{\mathrm{con}}(\phi, r, s) = \mathbb{E}_{p_s}\left\| D_\phi(\mathbf{x}_s, s) - \mathbb{E}_{p_{r|s}}[D_\phi(\mathbf{x}_r, r)] \right\|^2, \quad (9)$$

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_ϕ 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_ϕ minimizes the consistency loss for all r, s and perfectly learns the score function for $t > \zeta$, then $\frac{D_\phi(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t^2}$ becomes an exact estimator of the score function for all $t \in [0, T]$. Consequently, the distribution $p_0 = p_{\mathrm{data}}$ 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.

4. Theoretical Limit of Deconvolution

In this section, we evaluate the complexity of a deconvolution problem when the data corruption process is modelled using a forward diffusion process. Through the framework of deconvolution theory, we demonstrate that while Daras et al. (2024) showed that diffusion models can be trained using noisy samples, obtaining a sufficient number of samples to train high-quality models 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 (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*

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

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 $\mathcal{Y}$,*

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

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

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. Consequently, this implies that training from scratch, using only noisy images, with the consistency loss discussed in Sec 3.3, is infeasible. Notably, as indicated by Eq (10), this difficulty becomes 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 address the pessimistic statistical rate, we propose pretraining diffusion models on a small set of copyright-free samples. While this limited dataset can only capture a subset of the features and variations of the full true data distribution, we argue that it provides valuable prior information, enabling the model to start from a point much closer to the ground distribution compared to random weight initialization. For example, for image generation, pretraining allows the model to learn common features and structures shared among samples, such as continuity, smoothness, edges, and general appearance of typical object types.

Unfortunately, our empirical study in Sec 6 will show that the consistency loss-based method discussed in Sec 3.3 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 5 to bridge such a gap.

5. 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 $\tau$ 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 6, 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 $\mathbf{s}_{\phi_k}(\mathbf{x}_t, t) = \frac{D_{\phi_k}(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t^2}$, $g(t) = (\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t})^{1/2}$ and D_{ϕ_k} is obtained by minimizing (4) according to Alg 1. Assume D_{ϕ_k} reaches the optimal for all k. Under mild assumptions, for $k \geq 0$, we have

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

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

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

where

$$M_0 = \tfrac{1}{2} \int_0^\zeta g(t)^2 \mathbb{E}_{p_t^*} \left\| \nabla \log p_t^*(\mathbf{x}_t) - \mathbf{s}_{\phi_0}(\mathbf{x}_t, t) \right\|^2 \mathrm{d}t.$$

Prop 2 shows that, after sufficiently many iterations of backward sampling and denoiser updates, the distribution of denoised samples produced by the backward sampling step converges to the true data distribution at a rate of $\mathcal{O}(1/\sqrt{K})$ in terms of the characteristic function. This convergence implies that the corresponding densities—and thus the distributions—also become close. Consequently, fine-tuning the denoiser on these denoised samples during the Denoiser Update step enables the diffusion model to generate samples that approximately follow the true data distribution, thereby solving the deconvolution problem.

One might argue that the norm in the convergence bound depends not only on the iteration count K but also on the norm of $\mathbf{u}$, suggesting the existence of nontrivial approximation gaps when $|\mathbf{u}|$ is large, regardless of how large K is. We clarify that in practice, the contribution of the characteristic function at large $|\mathbf{u}|$ is typically negligible. For distributions with smooth and bounded densities, characteristic functions decay rapidly—often at an exponential or super-polynomial rate. We elaborate further on this point in Appx C, following the proof of the proposition.

Lastly, we note that this convergence result assumes access to infinitely many noisy samples and should be distinguished from the sample efficiency bounds discussed in Sec 4.

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 $\mathbf{s}_{\phi_0}$ 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 6. This observation further highlights the importance of pretraining to mitigate the impact of such errors and achieve high-quality samples with fewer iterations.

Table 1: Performance comparison of generative models. When $\sigma_\zeta > 0$, the models are trained on noisy images corrupted by Gaussian noise $\mathcal{N}(\mathbf{0}, \sigma_\zeta^2 \mathbf{I})$ after rescaling pixel values to $[-1, 1]$. For pretrained models, 50 clean images are randomly sampled from the training datasets for pretraining. $\underline{\text{Underscored}}$ results are produced by this work. **Bolded** values indicate the best performance.

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	$\underline{167.23}$	0.2	No	$\underline{246.95}$
TweedieDiff (Daras et al., 2024)	0.2	Yes	$\underline{65.21}$	0.2	Yes	$\underline{58.52}$
TweedieDiff+ (Daras et al., 2025)	0.2	Yes	8.05	0.2	Yes	-
SFBD (Ours)	0.2	Yes	$\underline{\mathbf{13.53}}$	0.2	Yes	$\underline{\mathbf{6.49}}$

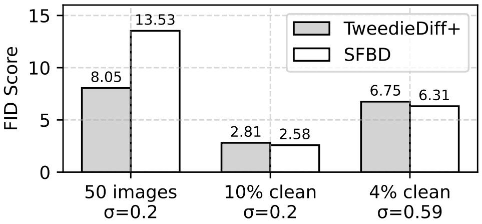


Figure 1: TweedieDiff+ vs SFBD on CIFAR10

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.

6. Empirical Study

In this section, we demonstrate the effectiveness of the SFBD framework proposed in Sec 5. 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.

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.

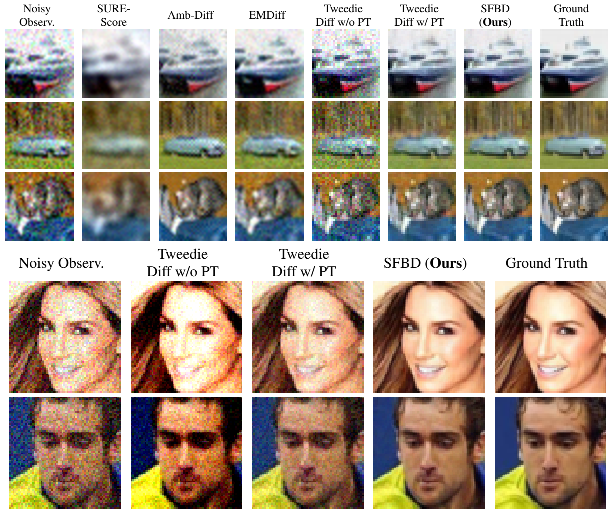


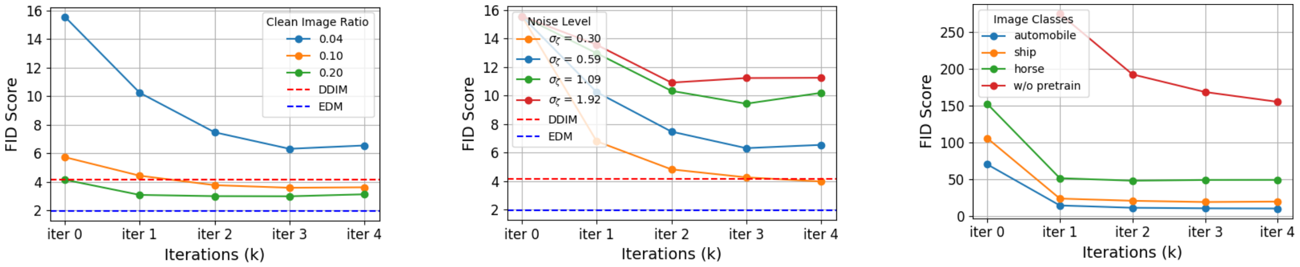
Figure 2: Denoised samples of CIFAR-10 (up) and CelebA (down). (Noise level $\sigma_\zeta = 0.2$)

6.1. Performance Comparison

In Table 1, we compare SFBD with representative models for training on noisy images. SURE-Score (Aali et al., 2023) and EMDiff (Bai et al., 2024) tackle general inverse problems using Stein's unbiased risk estimate and expectation-maximization, respectively. TweedieDiffusion (Daras et al., 2024) applies the original consistency loss from Eq (9), while Daras et al. (2025) introduce TweedieDiff+ with a simplified implementation that improves performance.

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.

As shown in Table 1, SFBD consistently produces higher-quality images than all baselines except TweedieDiff+, with further visual evidence provided in Fig 2, where denoised samples are generated by evaluating the backward SDE from noisy training images. Notably, on CelebA, SFBD achieves performance comparable to DDIM trained entirely on clean data. While TweedieDiff benefits from pretrain-



(a) Clean Image Ratio

(b) Noise Level

(c) Pretraining on Similar Datasets

Figure 3: 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 $\sigma_\zeta = 0.59$.

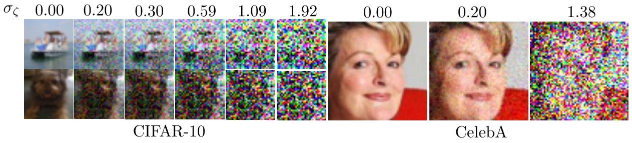


Figure 4: Noisy images with different σ_ζ.

ing, its performance remains inferior to SFBD. In fact, we observe that the original consistency loss (9) yields only limited improvement post-pretraining, with FID scores deteriorating soon after the loss is applied. TweedieDiff+, which adopts a simplified version of the consistency loss, outperforms SFBD when only a very limited amount of clean data is available—likely due to overfitting during our model's pretraining. However, this effect diminishes as more clean data is introduced, allowing SFBD to surpass TweedieDiff+, as shown in Fig 1. We attribute this to SFBD's more stable fine-tuning via the score-matching loss, which avoids the additional constraints imposed by consistency-based methods.

6.2. 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 4 and Sec 5 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 4).

Clean image ratio. Fig 3(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 3(b) shows SFBD's sampling performance across fine-tuning iterations for different noise levels, using the values from 2nd order Heun sampling in EDM (Karras et al., 2022). The impact of noise on the original images is visualized in Fig 4. As shown in Fig 3(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 4, rather than an issue solvable by better deconvolution algorithms. In Sec 6.3, we show that slightly increasing pretraining clean image set can yield strong results, even at reasonably high noise levels on CelebA.

Pretraining with clean images from similar datasets. Fig 3(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

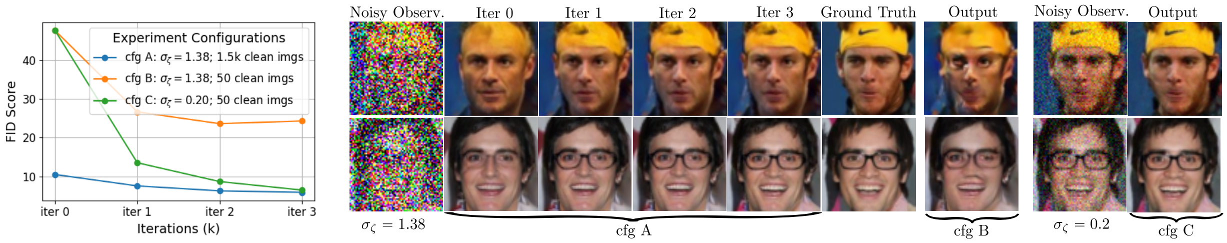


Figure 5: (Left) SFBD performance on CelebA under three configurations, with FID at iteration 0 for the pretrained model. (Right) Denoised samples generated by the backward SDE, starting from a noisy image in the training dataset. For cfg A, results are shown after each fine-tuning iteration, while cfg B and cfg C are shown at their minimum FID iterations.

Table 2: Comparison of SFBD and Restormer (Zamir et al., 2022) on denoising tasks.

CIFAR-10 (4% clean images)						CelebA					
σ	Model	Iter 1	Iter 2	Iter 3	Iter 4	Setting	Model	Iter 1	Iter 2	Iter 3	Iter 4
0.30	SFBD	6.16	3.42	2.68	2.35	50 clean imgs	SFBD	47.69	10.05	5.63	3.93
	Restormer	53.87				$\sigma = 0.2$	Restormer	18.90			
0.59	SFBD	10.23	7.47	6.31	6.54	1.5k clean imgs	SFBD	9.05	5.76	4.56	3.98
	Restormer	99.99				$\sigma = 1.38$	Restormer	227.91			
1.09	SFBD	12.68	9.39	9.08	10.14	—					
	Restormer	132.69									

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.*

6.3. Further Discussions

Additional results on CelebA. Fig 5 presents SFBD performance trajectories on CelebA under three configurations. While Table 1 reports results using configuration (cfg) C to align with benchmarks, this setup is impractical due to its low noise level, which fails to address copyright and privacy concerns. As illustrated in Fig 5 (right), the low noise level allows human observers to identify individuals and recover image details, with model-denoised images nearly identical to the originals. To address this, we report results for cfgs A and B with $\sigma_\zeta = 1.38$, concealing most original image information. While pretraining on 50 clean images performs poorly, increasing the size to 1.5k (still $< 1\%$ of the training dataset) achieves impressive results. At iteration 3, the model reaches FID 5.91, outperforming DDIM trained on clean images. This supports our discussion in Sec 6.2: collecting more clean data significantly boosts performance when the clean dataset is small.

Features learned from noisy images. As shown in Fig 5, when $\sigma_\zeta = 1.38$, almost all information from the original images is obscured, prompting the question: can the model learn from such noisy inputs, and how does this happen? In Fig 5, we plot the model's denoised outputs in cfg A after each fine-tuning iteration. These outputs serve as samples for the next iteration, revealing what the model learns and adapts to in the process. For the first row, the pretrained model (iter 0) produces a face very different from the original, failing to recover features like a headband. This occurs because the clean dataset for pretraining lacks similar faces with headbands. Instead of random guesses, the model combines local features (e.g., face shapes, eyes) learned from the clean data with the global structure from the noisy images. This process combines previously learned features in new ways, helps the model better generalize, and gradually improves its ability to approximate the true distribution, as supported by Prop 2. Likewise, the model learns how to attach the glasses to the individual's face in the second row.

How SFBD differs from standard denoising algorithms. While SFBD is intuitively described as alternating between denoising and fine-tuning, it does not function like a traditional denoiser that reconstructs exact clean samples. Instead, it learns to match the full data distribution, allowing greater flexibility and producing more realistic outputs. In Table 2, we compare the FID of SFBD-denoised samples at each iteration ($\mathcal{E}_k$ in Alg 1) with that of the training set and Restormer (Zamir et al., 2022), a strong off-the-shelf denoiser. SFBD consistently yields significantly lower FIDs—even after the first iteration—and the sample quality improves steadily with more updates.

These results also caution against replacing SFBD's denoising process with a classical denoiser. Since models trained on denoised samples cannot surpass their targets in FID, the values in Table 2 represent upper bounds. Notably, final SFBD models achieve lower generative FIDs (Fig 3(b)) than those from Restormer-denoised data, making it unlikely that a competitive model could be trained on such samples.

7. Conclusion

In this paper, 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.

Acknowledgement

We gratefully acknowledge funding support from NSERC, the Canada CIFAR AI Chairs program and the Ontario Early Researcher 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.

Impact Statement

This paper introduces SFBD, a framework for effectively training diffusion models primarily using noisy samples. Our approach enables data sharing for generative model training while safeguarding sensitive information.

For organizations utilizing personal or copyrighted data to train their models, SFBD offers a practical solution to mitigate copyright concerns, as the model never directly accesses the original samples. This mathematically guaranteed framework can promote data-sharing by providing a secure and privacy-preserving training method.

However, improper implementation poses a risk of sensitive information leakage. A false sense of security could further exacerbate this issue, underscoring the importance of rigorous validation and responsible deployment.

References

Aali, A., Arvinte, M., Kumar, S., and Tamir, J. I. 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.

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016. URL https://doi.org/10.1145/2976749.2978318.

Anderson, B. D. O. 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.

Bai, W., Wang, Y., Chen, W., and Sun, H. 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.

Ben-David, S., Bie, A., Canonne, C. L., Kamath, G., and Singhal, V. Private distribution learning with public data: The view from sample compression. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=nDIrJmKPd5.

Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al. Improving image generation with better captions. OpenAI, 2023. URL https://cdn.openai.com/papers/dall-e-3.pdf.

Bie, A., Kamath, G., and Singhal, V. Private estimation with public data. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

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

Bovy, J., Hogg, D. W., and Roweis, S. T. Extreme deconvolution: Inferring complete distribution functions from noisy, heterogeneous and incomplete observations. *The Annals of Applied Statistics*, 5(2B):1657–1677, 2011. URL https://doi.org/10.1214/10-AOAS439.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. 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.

Carroll, R. J. and Hall, P. Optimal rates of convergence for deconvolving a density. *Journal of the American Statistical Association*, 83(404):1184–1186, 1988. URL https://doi.org/10.1080/01621459.1988.10478718.

Cordy, C. B. and Thomas, D. R. Deconvolution of a distribution function. *Journal of the American Statistical Association*, 92(440):1459–1465, 1997. URL https://doi.org/10.2307/2965416.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. 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.

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

Daras, G., Dagan, Y., Dimakis, A., and Daskalakis, C. 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.

Daras, G., Shah, K., Dagan, Y., Gollakota, A., Dimakis, A., and Klivans, A. 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.

Daras, G., Dimakis, A., and Daskalakis, C. C. 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.

Daras, G., Cherapanamjeri, Y., and Daskalakis, C. C. 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.

Delaigle, A. and Hall, P. Using SIMEX for smoothing-parameter choice in errors-in-variables problems. *Journal of the American Statistical Association*, 103(481): 280–287, 2008. URL ttps://doi.org/10.1198/016214507000001355.

Dockhorn, T., Cao, T., Vahdat, A., and Kreis, K. Differentially private diffusion models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=ZPpQk7FJXF.

Fan, J. On the optimal rates of convergence for nonparametric deconvolution problems. *The Annals of Statistics*, 19(3):1257–1272, 1991. URL https://www.jstor.org/stable/2241949.

Feller, W. *An Introduction to Probability Theory and Its Applications, Volume 2*. Wiley, 1971.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. URL https://doi.org/10.1145/3422622.

Guan, Z. Fast nonparametric maximum likelihood density deconvolution using bernstein polynomials. *Statistica Sinica*, 31(2):891–908, 2021. URL https://doi.org/10.5705/ss.202018.0173.

Ho, J., Jain, A., and Abbeel, P. 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.

Ho, J., Salimans, T., Gritsenko, A. A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=f3zNgKga_ep.

Kailath, T. 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.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

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

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

Krizhevsky, A. and Hinton, G. 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.

Lepski, O. and Willer, T. Oracle inequalities and adaptive estimation in the convolution structure density model. *The Annals of Statistics*, 47(1):233–287, 2019. URL https://doi.org/10.1214/18-AOS1687.

Liu, X., Gong, C., and Liu, Q. 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.

Liu, Z., Luo, P., Wang, X., and Tang, X. 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.

Lounici, K. and Nickl, R. Global uniform risk bounds for wavelet deconvolution estimators. *The Annals of Statistics*, 39(1):201–231, 2011. URL https://doi.org/10.1214/10-AOS836.

Masry, E. Strong consistency and rates for deconvolution of multivariate densities of stationary processes. *Stochastic Processes and Their Applications*, 47(1): 53–74, 1993. URL https://doi.org/10.1016/0304-4149(93)90094-K.

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

Meister, A. and Neumann, M. H. Deconvolution from non-standard error densities under replicated measurements. *Statistica Sinica*, 20:1609–1636, 2010. URL https://www3.stat.sinica.edu.tw/sstest/j20n4/j20n412/j20n412.html.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. Diffusion models for adversarial purification. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 16805–16827, 2022. URL https://proceedings.mlr.press/v162/nie22a.html.

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

Pavon, M. and Wakolbinger, A. 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.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. 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.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. 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.

Sarkar, A., Pati, D., Chakraborty, A., Mallick, B. K., and Carroll, R. J. Bayesian semiparametric multivariate density deconvolution. *Journal of the American Statistical Association*, 113(521):401–416, 2018. URL https://doi.org/10.1080/01621459.2016.1260467.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. 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.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. 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.

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

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

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

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

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

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

Vargas, F., Thodoroff, P., Lamacraft, A., and Lawrence, N. Solving schrodinger bridges via maximum likelihood. *Entropy*, 23(9), 2021. URL https://www.mdpi.com/1099-4300/23/9/1134.

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

Wang, Z., Zheng, H., He, P., Chen, W., and Zhou, M. Diffusion-GAN: Training GANs with diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=HZf7UbpWHuA.

Xie, L., Lin, K., Wang, S., Wang, F., and Ren, J. Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739, 2018. URL https://arxiv.org/abs/1802.06739.

Yang, D., Yu, J., Wang, H., Wang, W., Weng, C., Zou, Y., and Yu, D. 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.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., and Yang, M.-H. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5728–5739, June 2022.

Zhang, C.-H. Fourier methods for estimating mixing densities and distributions. *The Annals of Statistics*, 18 (2):806–831, 1990. URL https://doi.org/10.1214/aos/1176347627.

Zhang, Q. and Chen, Y. 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 6.

$$\begin{tikzcd}
p_{\text{data}} \arrow[r, shift left, "\text{convolution}"] \arrow[d, phantom, "\wr"] & p_y = p_{\text{data}} * h \arrow[l, shift left, "\text{deconvolution}"] \arrow[d, phantom, "\wr"] \\
x^{(i)} \arrow[r, shift left, "\text{add } \epsilon^{(i)} \sim h"] & y^{(i)} = x^{(i)} + \epsilon^{(i)} \arrow[l, shift left, "\text{irreversible}", "/\!/" marking]
\end{tikzcd}$$

Figure 6: 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 $\mathbf{u} \in \mathbb{R}^d$,*

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

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*

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

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

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

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

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

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

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

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 (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*

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

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

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

where $\mu^2_{\kappa,2}$ 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 $\mathcal{Y}$,*

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

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

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 $\text{MISE}(\hat{p}, p_{\text{data}}) \geq \text{const}\cdot(\log n)^{-2\beta/\gamma} = \text{const}\cdot(\log n)^{-2}$. □

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, for $k \geq 0$, we have

$$D_{\text{KL}}(p_{data}\,\|\,p_0^{(k)}) \geq D_{\text{KL}}(p_{data}\,\|\,p_0^{(k+1)}). \tag{13}$$

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

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

where

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

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 $D_{\mathrm{KL}}(\overrightarrow{P}^{(k)}_{0:\zeta} \,\|\, \overleftarrow{Q}^{\boldsymbol{\phi}}_{0:\zeta})$ as claimed.

Now, we are ready to prove Prop 2.

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

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

Likewise,

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

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

Lem 2 also implies that

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

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

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

As a result,

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

which is (13). In addition, we have

$$\begin{aligned} D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \,\|\, \overleftarrow{Q}^{\boldsymbol{\phi}_{k-1}}_{0:\zeta}) &\overset{(33)}{=} D_{\mathrm{KL}}(p_{\mathrm{data}} \,\|\, p^{(k)}_0) + \mathcal{B}_{k-1} \overset{(34)}{=} D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \,\|\, \overrightarrow{P}^{(k)}_{0:\zeta}) + \mathcal{B}_{k-1} \\ &\overset{(32)}{=} D_{\mathrm{KL}}(p^*_\zeta \,\|\, p^{(k)}_\zeta) + D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \,\|\, \overleftarrow{Q}^{\boldsymbol{\phi}_k}_{0:\zeta}) + \mathcal{B}_{k-1} \\ &= D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \,\|\, \overleftarrow{Q}^{\boldsymbol{\phi}_k}_{0:\zeta}) + \left[D_{\mathrm{KL}}(p^*_\zeta \,\|\, p^{(k)}_\zeta) + \mathcal{B}_{k-1}\right]. \end{aligned}$$

As a result, applying this relationship recursively, we have

$$D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \,\|\, \overleftarrow{Q}^{\boldsymbol{\phi}_0}_{0:\zeta}) = \sum_{k=1}^{K} D_{\mathrm{KL}}(p^*_\zeta \,\|\, p^{(k)}_\zeta) + \sum_{k=1}^{K} \mathcal{B}_{k-1} + D_{\mathrm{KL}}(\overrightarrow{P}^*_{0:\zeta} \,\|\, \overleftarrow{Q}^{\boldsymbol{\phi}_K}_{0:\zeta}). \tag{35}$$

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

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

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

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

Applying Prop 1, we obtain,

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

□

Additional Comments on the Convergence Guarantee of Prop 2. In the main text, we noted that although the bound appears to grow with $\|\mathbf{u}\|$, the behavior of characteristic functions at large $\|\mathbf{u}\|$ is typically negligible in practice. Characteristic functions of distributions with smooth, bounded densities tend to decay rapidly – often exponentially or at a super-polynomial rate.

To make this precise, consider the 1D case: the characteristic function $\phi(u)$ is the Fourier transform of the density. If the density is k-times differentiable, then it is well known (e.g., Lemma 4, p. 514, Feller 1971) that $|\phi(u)| = o(|u|^{-k})$. This implies that for sufficiently large $|u|$, the characteristic function becomes negligible in magnitude.

Thus, assuming both p_{data} and $p_0^{(k)}$ are smooth with bounded support, it suffices to match their characteristic functions over a compact domain $|u| < U$ for some $U > 0$. Such local agreement in the Fourier domain implies closeness of the corresponding densities, and hence the distributions.

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 6. The samples are taken from the models at their fine-tuning iteration with the lowest FID.

D.1. CIFAR-10

Samples for computing FIDs in Fig 3(a) - Clean Image Ratio

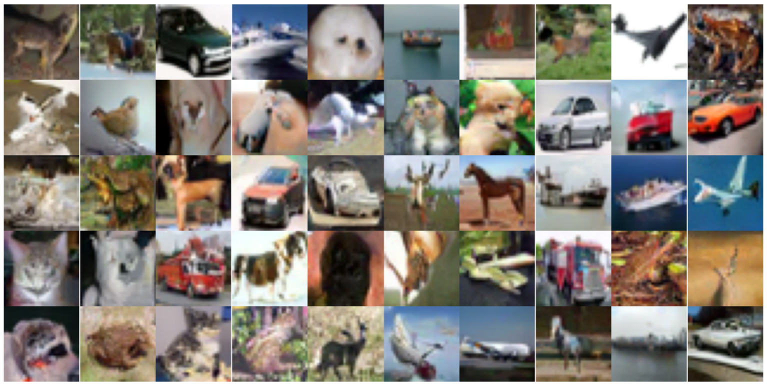


Figure 7: Clean image ratio = 0.04 – FID: 6.31

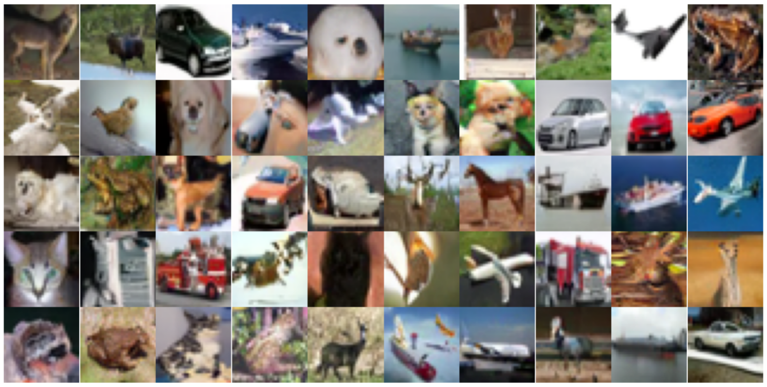


Figure 8: Clean image ratio = 0.1 – FID: 3.58

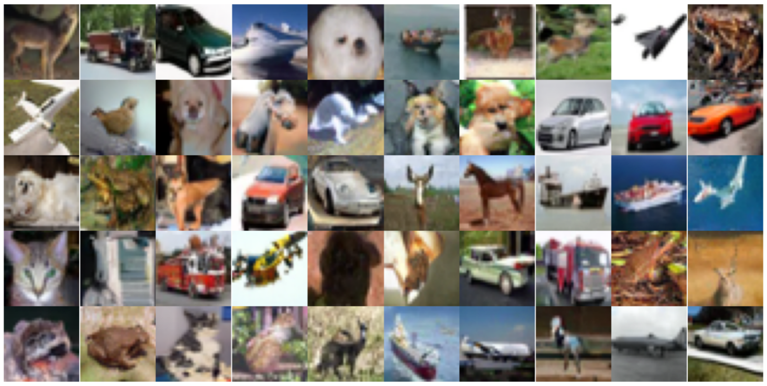


Figure 9: Clean image ratio = 0.2 – FID: 2.98

Samples for computing FIDs in Fig 3(b) - Noise Level

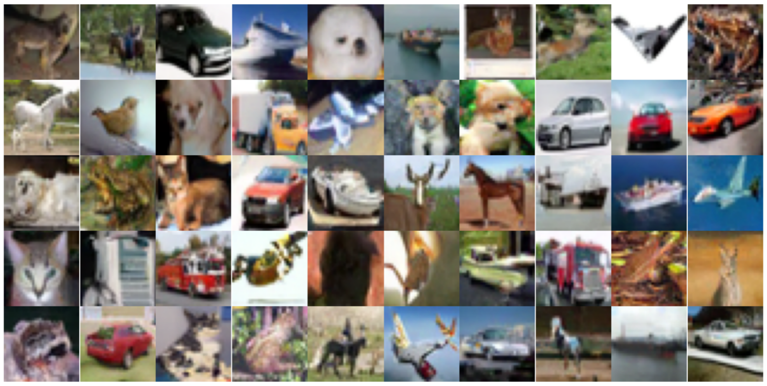


Figure 10: Noise level $\sigma_\zeta = 0.30$ – FID: 3.97

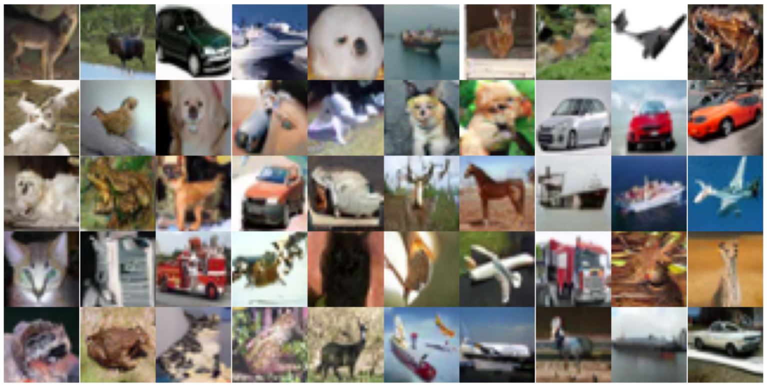


Figure 11: Noise level $\sigma_\zeta = 0.59$ – FID: 6.31

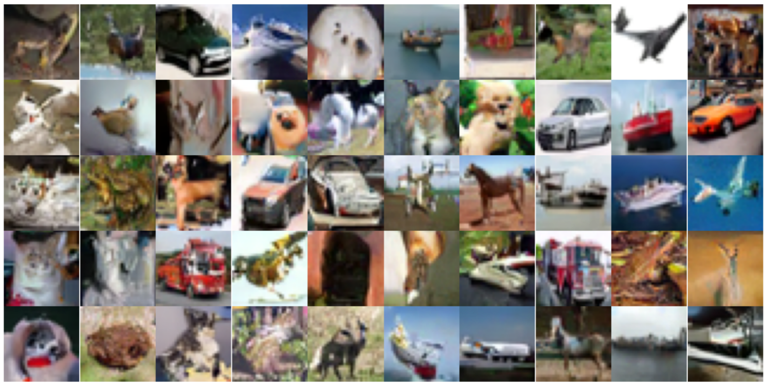


Figure 12: Noise level $\sigma_\zeta = 1.09$ – FID: 9.43

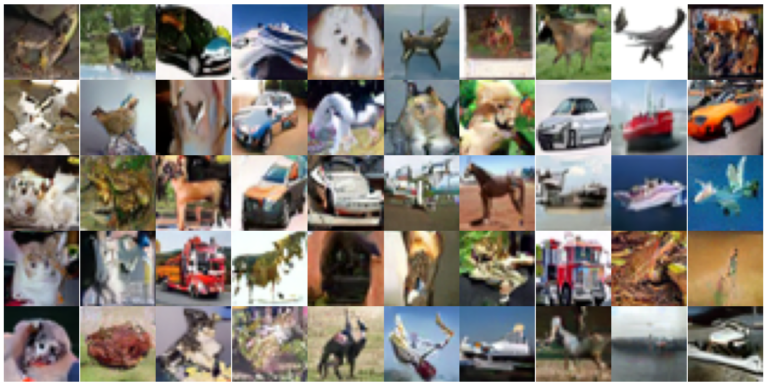


Figure 13: Noise level $\sigma_\zeta = 1.92$ – FID: 10.91

Samples for computing FIDs in Fig 3(c) - Pretraining on Similar Datasets

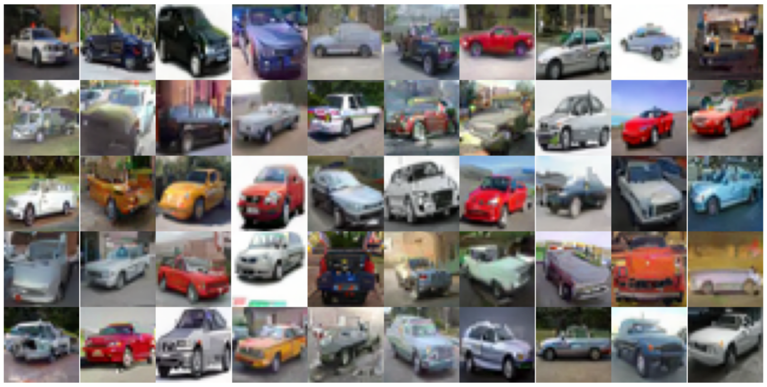


Figure 14: Class for fine-tuning: automobile – FID: 10.39

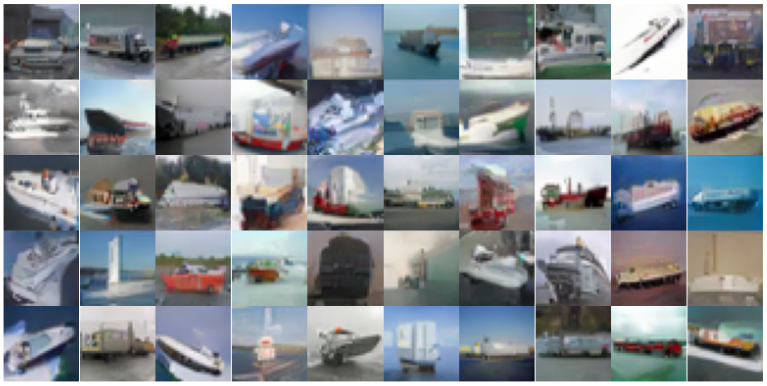


Figure 15: Class for fine-tuning: ship – FID: 19.19

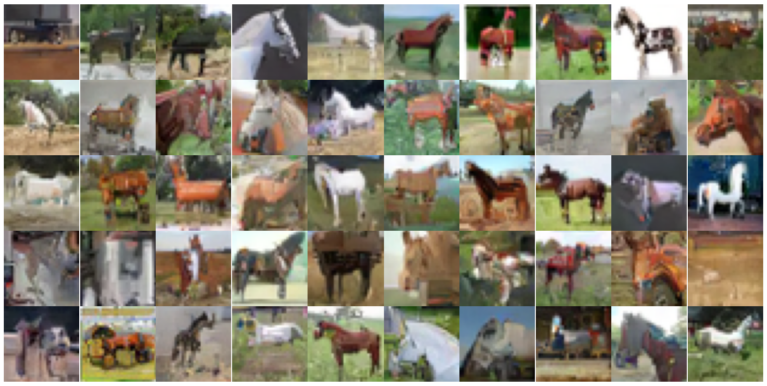


Figure 16: Class for fine-tuning: horse – FID: 48.11

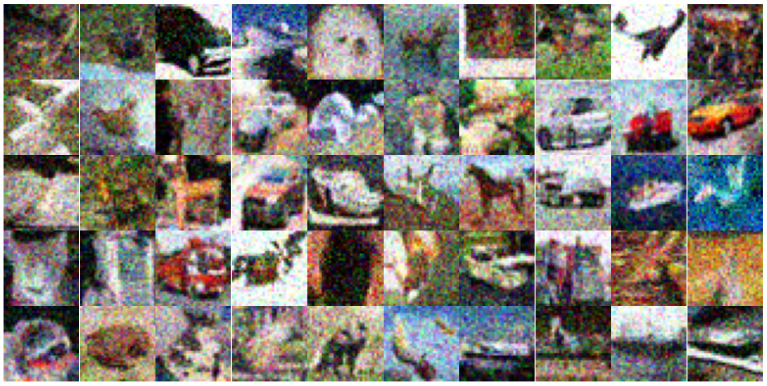


Figure 17: Class for fine-tuning: no pretrain – FID: 155.04

D.2. CelebA

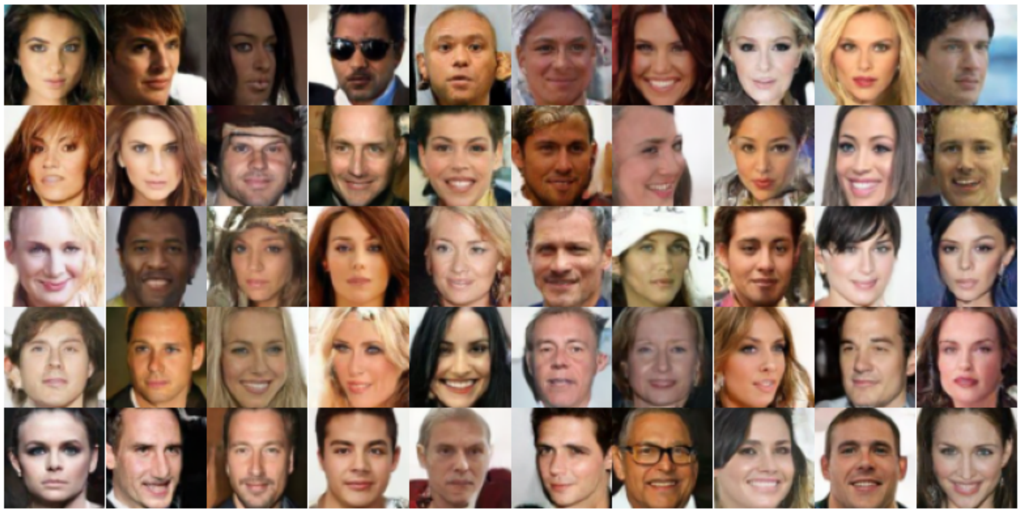


Figure 18: cfg A: $\sigma_\zeta = 1.38$; 1,500 clean images for pretraining – FID: 5.91

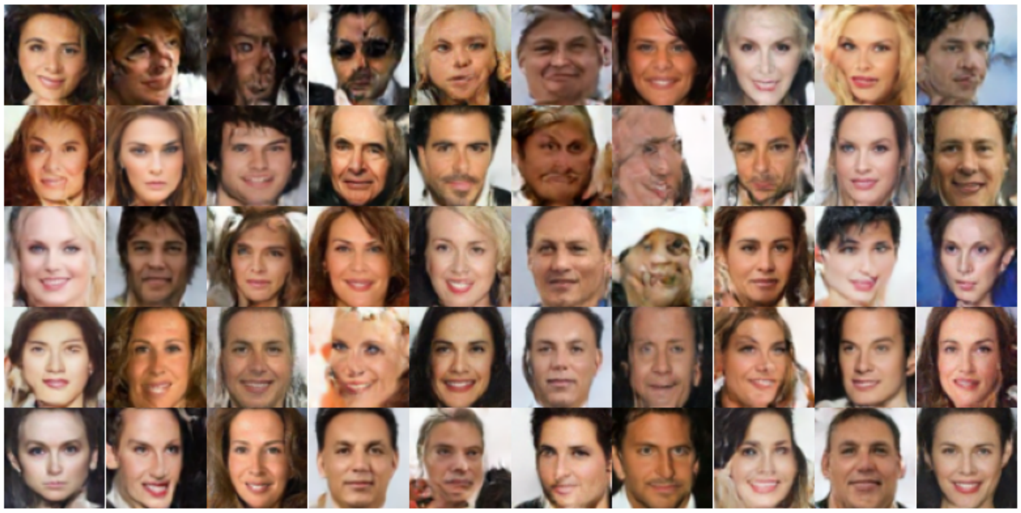


Figure 19: cfg B: $\sigma_\zeta = 1.38$; 50 clean images for pretraining – FID: 23.63

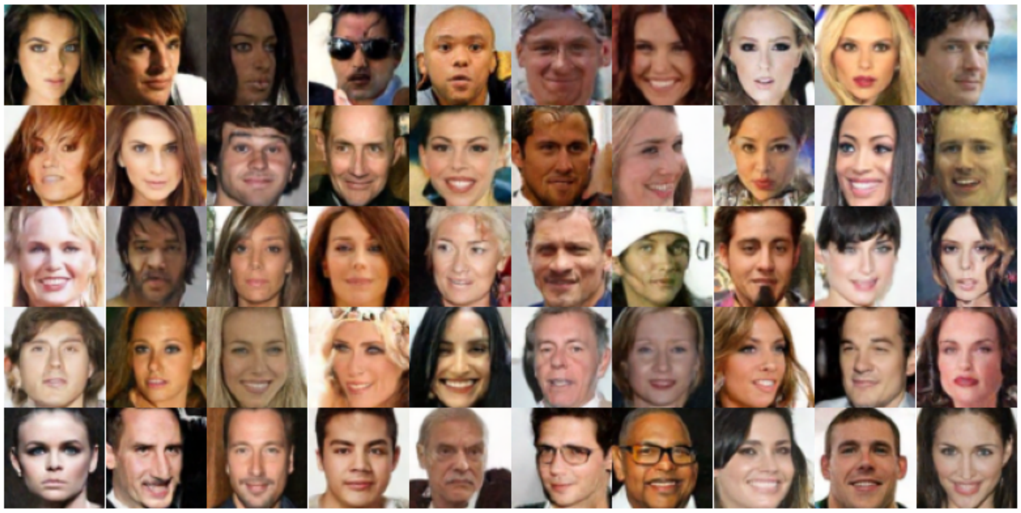


Figure 20: cfg C: $\sigma_\zeta = 0.20$; 50 clean images for pretraining – FID: 6.48

E. Experiment Configurations

E.1. Model Architectures

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

Table 3: 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 3(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.