

Consistent Diffusion Meets Tweedie: Training Exact Ambient Diffusion Models with Noisy Data

Giannis Daras	Alexandros G. Dimakis	Constantinos Daskalakis
UT Austin	UT Austin	CSAIL MIT
Computer Science Department	ECE Department	Archimedes AI
giannisdaras@utexas.edu	dimakis@austin.utexas.edu	costis@csail.mit.edu

Abstract

Ambient diffusion is a recently proposed framework for training diffusion models using corrupted data. Both Ambient Diffusion and alternative SURE-based approaches for learning diffusion models from corrupted data resort to approximations which deteriorate performance. We present the first framework for training diffusion models that provably sample from the uncorrupted distribution given only noisy training data, solving an open problem in this space. Our key technical contribution is a method that uses a double application of Tweedie’s formula and a consistency loss function that allows us to extend sampling at noise levels below the observed data noise. We also provide further evidence that diffusion models memorize from their training sets by identifying extremely corrupted images that are almost perfectly reconstructed, raising copyright and privacy concerns. Our method for training using corrupted samples can be used to mitigate this problem. We demonstrate this by fine-tuning Stable Diffusion XL to generate samples from a distribution using only noisy samples. Our framework reduces the amount of memorization of the fine-tuning dataset, while maintaining competitive performance.

1 Introduction

In recent years, we have witnessed remarkable progress in image generation as exemplified by state-of-the-art models such as Stable Diffusion (-XL) [22, 20] and DALL-E (2, 3) [5]. This progress has been driven by two major enablers: i) the diffusion modeling framework [11, 28, 29]; and ii) the existence of massive datasets of image-text pairs [23, 9].

The need for high-quality, web-scale data and the intricacies involved in curating datasets at that scale often result in the inclusion of copyrighted content. Making things worse, diffusion models memorize training examples more [25, 6] than previous generative modeling approaches, such as Generative Adversarial Networks [10], often replicating parts or whole images from their training set [6, 25].

A recently proposed strategy for mitigating the memorization issue is to train (or fine-tune) diffusion models using corrupted data [8, 26]. Indeed, developing a capability for training diffusion models using corrupted data can also find applications in domains where access to uncorrupted data is expensive or impossible, e.g. in MRI [1] or black-hole imaging [15, 31]. Unfortunately, existing methods for learning diffusion models from corrupted data [8, 1, 12, 32] resort to approximations (during training or sampling) that significantly hurt performance. Our contributions are as follows:

1. We propose the first *exact* framework for learning diffusion models using only corrupted samples. Our key technical contributions are: i) a computationally efficient method for



Figure 1: Top row: images from LAION [23], middle row: masked images, bottom row: reconstructed images with the SDXL [20] inpainting model. The accuracy of the reconstructions presents strong evidence that the images on the top-row were in the training set of SDXL (or SDXL Inpainting) and have been memorized. To the best of our knowledge, SDXL does not disclose its training set.

learning optimal denoisers for all levels of noise $\sigma \geq \sigma_n$, where σ_n is the standard deviation of the noise in the training data, obtained by applying Tweedie’s formula twice; and ii) a consistency loss function [7] for learning the optimal denoisers for noise levels $\sigma \leq \sigma_n$. Note that given samples at level of noise σ_n it is possible to obtain samples at levels of noise $\sigma > \sigma_n$ (by adding further noise) but prior to our work it was not known how to train diffusion models to obtain samples at levels of noise $\sigma < \sigma_n$.

2. We provide further evidence that foundation diffusion models memorize from their training sets by showing that extremely corrupted training images can be almost perfectly reconstructed. Moreover, we show that memorization occurs at a higher rate than previously anticipated.
3. We use our framework to fine-tune diffusion foundation models using corrupted data and show that the performance of our trained model declines (as the corruption in the training data increases) at a much slower rate compared to previously proposed approaches.
4. We evaluate our trained models against our as well as a baseline method for testing data replication and we show that models trained under data corruption memorize significantly less.
5. We open-source our code to facilitate further research in this area: <https://github.com/giannisdaras/ambient-tweedie>.

2 Background and Related Work

Consider a distribution of interest admitting a density function p_0 . Our goal is to train a diffusion model that generates samples from p_0 . However, we only have access to noisy samples from p_0 . In particular, we have samples of the form $\mathbf{X}_{t_n} = \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}$, where $\mathbf{X}_0 \sim p_0$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I_d)$. We denote by p_{t_n} the distribution density of these samples. Throughout the paper we fix an increasing non-negative function $\sigma(t)$, where $t \in [0, T]$, $T > 0$, and $\sigma(0) = 0$, and denote $\sigma(t)$ by σ_t . We take $t_n \in (0, T)$. The subscript ‘ n ’ in t_n refers to “nature” and, as stated above, we assume that nature is giving us access to samples at noise level σ_{t_n} . We denote by p_t the distribution of random variable $\mathbf{X}_t = \mathbf{X}_0 + \sigma_t \mathbf{Z}$, where $\mathbf{X}_0 \sim p_0$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I_d)$.



Figure 2: SDXL [20] posterior samples (Row e) given extremely noisy encodings (Row c) of LAION images (Row a). The level of fidelity of the reconstructions to the original images, despite the severe corruption (c) and the blurriness of the MMSE solution (d), indicates that the images were potentially in the training set and have been memorized.

2.1 Background on denoising diffusion models

Denoising diffusion models can equivalently be viewed as denoisers at many different noise levels σ_t , $t \in [0, T]$. They are typically trained with the Denoising Score Matching loss:

$$J_{\text{DSM}}(\theta) = \mathbb{E}_{\mathbf{x}_0 \sim p_0(\mathbf{x}_0)} \mathbb{E}_{t \sim \mathcal{U}[0, T]} \mathbb{E}_{\mathbf{x}_t \sim p_t(\mathbf{x}_t | \mathbf{x}_0)} \left[\|\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|^2 \right].$$

If the function class $\{\mathbf{h}_\theta\}$ is sufficiently rich, the minimizer of this loss satisfies $\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t]$ for all t, \mathbf{x}_t . Tweedie’s formula connects the conditional expectation, i.e. the best denoiser in the ℓ_2^2 sense, with the score function $\nabla \log p_t(\mathbf{x}_t)$,

$$\nabla \log p_t(\mathbf{x}_t) = \frac{\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t] - \mathbf{x}_t}{\sigma_t^2}. \quad (2.1)$$

The score can be then used to sample from $p_0(\mathbf{x}_0)$ [3], by sampling a trajectory from the following Stochastic Differential Equation (which is the reverse diffusion process of the process that adds Gaussian noise to a sample from p_0 according to the noise schedule σ_t):

$$d\mathbf{x}_t = -2\sigma_t d\sigma_t \nabla \log p_t(\mathbf{x}_t) + \sqrt{\frac{d\sigma_t^2}{dt}} d\bar{\mathbf{w}}, \quad (2.2)$$

initialized at $\mathbf{x}_T \sim p_T(\mathbf{x}_T)$, where $\bar{\mathbf{w}}$ is a standard Wiener process when the time flows backwards (from T to 0). In practice, the score function is replaced with its estimate, i.e. we run the process:

$$d\mathbf{x}_t = -2d\sigma_t \frac{\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t} + \sqrt{\frac{d\sigma_t^2}{dt}} d\bar{\mathbf{w}}. \quad (2.3)$$

As we have described, in our setting of interest, we do not have access to samples from p_0 , thus we are not in a position to train a diffusion model using Equation 2.1. Yet, we still want to learn $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t]$ (and thus, the score) for *all noise levels* t . A natural first question is whether we can at least learn denoisers for noise levels that are equal to that of the available data or larger, i.e. for $t : \sigma_t \geq \sigma_{t_n}$.

2.2 Prior Work on learning denoisers from noisy data

Prior work has given an affirmative answer to the question at the end of the previous section. One of the most established methods is Stein’s Unbiased Risk Estimate (SURE) [30]. SURE learns the conditional expectation of \mathbf{X}_0 given a sample \mathbf{X}_{t_n} , by minimizing the following objective *that only uses the noisy realizations*:

$$J_{\text{SURE}}(\theta) = \mathbb{E}_{\mathbf{x}_{t_n} \sim p_{t_n}} \left[\|\mathbf{h}_\theta(\mathbf{x}_{t_n}) - \mathbf{x}_{t_n}\|^2 + 2\sigma_{t_n}^2 (\nabla_{\mathbf{x}} \cdot \mathbf{h}_\theta(\mathbf{x}_{t_n}))^2 \right],$$

The divergence term is expensive to evaluate and is typically replaced with the Monte Carlo approximation:

$$(\nabla_{\mathbf{x}} \cdot \mathbf{h}_\theta(\mathbf{x}_{t_n}))^2 \approx \mathbf{z}^T \left(\frac{\mathbf{h}_\theta(\mathbf{x}_{t_n} + \epsilon \mathbf{z}) - \mathbf{h}_\theta(\mathbf{x}_{t_n})}{\epsilon} \right),$$

for some small, positive parameter ϵ and $\mathbf{z} \sim N(\mathbf{0}, I_d)$ [2, 24, 16]. An alternative approximation is to compute the Jacobian Vector Product $\mathbf{z}^T \nabla \mathbf{h}_\theta(\mathbf{x}_{t_n}) \mathbf{z}$, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I_d)$, with automatic differentiation tools [12]. Both methods give unbiased estimators for the divergence term and the variance can be decreased by averaging over many \mathbf{z} (at the cost of increased computation).

Another line of work is the Noise2Noise [14] framework and its generalizations [4, 13, 19, 33, 17]. Most relevant to our work are the Noisier2Noise [17] and Noisy-As-Clean [33] approaches wherein the training goal is to predict the noisy signal from a further corrupted version of it. Noisier2Noise comes with the theoretical guarantee of learning $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t]$ for a single $t : \sigma_t \geq \sigma_{t_n}$, where by “a single t ” we mean that, for each t of interest, a new problem needs to be solved.

2.3 Prior work on learning diffusion models from corrupted data

In contrast to the previous section, there is no known approach for learning $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t]$ for noise levels $t : \sigma_t \leq \sigma_{t_n}$. Thus, it is not known how to train an exact diffusion model using noisy samples, so various approximations have been considered, as described below.

Aali et al. [1] uses SURE to learn the optimal denoiser at the noise level of the available data, i.e. $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_{t_n} = \mathbf{x}_{t_n}]$, and then creates noisy iterates $\tilde{\mathbf{X}}_t = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_{t_n}] + \sigma_t \mathbf{Z}$ at all noise levels t to train with Denoising Score Matching. However, the underlying noisy distributions, $\tilde{p}_t(\mathbf{x}_t)$, are different than $p_t(\mathbf{x}_t)$, since the R.V. \mathbf{X}_0 has been replaced with $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_{t_n}]$. An alternative approach is to use SURE to learn $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t]$ for all levels $t : \sigma_t \geq \sigma_{t_n}$ and early stop diffusion sampling at time $t : \sigma_t = \sigma_{t_n}$. This type of approach is adopted by Kawar et al. [12] and Xiang et al. [32] and it only guarantees samples from the distribution $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_{t_n}]$.

Notably, similar problems arise in the setting of training diffusion models from linearly corrupted data, i.e. when the available samples are $\mathbf{Y}_0 = A\mathbf{X}_0$, for a known matrix A , as considered in the Ambient Diffusion paper [8]. In this setting, the authors manage to learn $\mathbb{E}[\mathbf{X}_0 | A\mathbf{X}_t]$ for all t , but not $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t]$, where $\mathbf{X}_t = \mathbf{X}_0 + \sigma_t \mathbf{Z}$, as always in this paper. Similar challenges are encountered in the G-SURE paper [12].

In sum, training exact diffusion models, i.e. diffusion models sampling the target distribution p_0 , given corrupted data remains unsolved. In this paper, we resolve this open-problem with two key technical contributions: i) an efficiently computable objective for learning the optimal denoisers for all levels of noise $t : \sigma_t \geq \sigma_{t_n}$, obtained by applying Tweedie’s formula twice; and ii) a consistency loss for learning the optimal denoisers for levels of noise $t : \sigma_t \leq \sigma_{t_n}$. We describe these contributions in the next section.

3 Method

3.1 Learning the Optimal Denoiser for $\sigma_t > \sigma_{t_n}$

We first present an efficiently computable objective that resembles Denoising Score Matching and enables learning the optimal denoisers for all noise levels $t : \sigma_t > \sigma_{t_n}$.

Theorem 3.1 (Ambient Denoising Score Matching). *Define \mathbf{X}_t as in the beginning of Section 2. Suppose we are given samples $\mathbf{X}_{t_n} = \mathbf{X}_0 + \sigma_{t_n}\mathbf{Z}$, where $\mathbf{X}_0 \sim p_0$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I)$. Consider the following objective:*

$$J(\theta) = \mathbb{E}_{\mathbf{x}_{t_n}} \mathbb{E}_{t \sim \mathcal{U}(t_n, T)} \mathbb{E}_{\mathbf{x}_t = \mathbf{x}_{t_n} + \sqrt{\sigma_t^2 - \sigma_{t_n}^2} \boldsymbol{\eta}} \left[\left\| \frac{(\sigma_t^2 - \sigma_{t_n}^2)}{\sigma_t^2} (\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \mathbf{x}_t - \mathbf{x}_{t_n} \right\|^2 \right],$$

where $\boldsymbol{\eta}$ in the above is a standard Gaussian vector. Suppose that the family of functions $\{\mathbf{h}_\theta\}$ is rich enough to contain the minimizer of the above objective overall functions $\mathbf{h}(\mathbf{x}, t)$. Then the minimizer θ^* of J satisfies:

$$\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t], \quad \forall \mathbf{x}_t, t > t_n. \quad (3.1)$$

The theorem above states that we can estimate the best l_2^2 denoisers for all noise levels $t : \sigma_t > \sigma_{t_n}$ without ever seeing clean data from p_0 and using an efficiently computable objective that contains no divergence term.

Proof Overview. The central idea for this proof is to apply Tweedie’s Formula twice, on appropriate random variables. We start by stating (a generalized version of) Tweedie’s formula, the proof of which is given in the Appendix.

Lemma 3.2 (Generalized Tweedie’s Formula). *Let:*

$$\mathbf{X}_t = \alpha_t \mathbf{X}_0 + \sigma_t \mathbf{Z}, \quad (3.2)$$

for $\mathbf{X}_0 \sim p_0$, $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I)$, and some positive function α_t of t . Then,

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) = \frac{\alpha_t \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t] - \mathbf{x}_t}{\sigma_t^2}. \quad (3.3)$$

For $t : \sigma_t > \sigma_{t_n}$, the R.V. \mathbf{X}_t can be written in the following two equivalent ways:

$$\begin{cases} \mathbf{X}_t = \mathbf{X}_0 + \sigma_t \mathbf{Z} \\ \mathbf{X}_t = \mathbf{X}_{t_n} + \sqrt{\sigma_t^2 - \sigma_{t_n}^2} \mathbf{Z} \end{cases} \quad (3.4)$$

By applying Tweedie’s formula twice, we get two alternative expressions for the same score-function since the distribution remains the same, irrespectively of how we choose to express \mathbf{X}_t . By equating the two expressions for the score, we arrive at the following result:

$$\mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t] = \frac{\sigma_t^2 - \sigma_{t_n}^2}{\sigma_t^2} (\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t] - \mathbf{x}_t) + \mathbf{x}_t.$$

We can train a network with denoising score matching to estimate $\mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t]$ and hence we can use the above equation to obtain $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t]$, as desired.

The method we propose is conceptually similar to Noisier2Noise [17] but instead of adding noise with a fixed magnitude to create further corrupted iterates, we consider a continuum of noise scales and we train the model jointly in a Denoising Score Matching fashion.

Finally, we underline that our method can be easily extended to the Variance Preserving (VP) [29] case, i.e. when the available data are $\mathbf{X}_{t_n} = \sqrt{1 - \sigma_{t_n}^2} \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}$. This is the setting for our Stable Diffusion finetuning experiments (see Section 4). For the sake of simplicity, we avoid these calculations in the main paper and we point the interested reader to the Appendix (see Theorem A.5).

3.2 Learning the Optimal Denoiser for $\sigma_t \leq \sigma_{t_n}$

Theorem 3.1 allows us to learn the optimal denoisers for $t : \sigma_t > \sigma_{t_n}$. However, to perform exact sampling we need to also learn $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t]$ for $t : \sigma_t \leq \sigma_{t_n}$. We achieve this by *training the network to be consistent*.

Definition 3.3 (Consistent Denoiser [7]). Let $p_\theta(\mathbf{x}_{t'}, t' | \mathbf{x}_t, t)$ be the density of the sample $\mathbf{X}_{t'}$ of the stochastic diffusion process of Equation 2.3 at time t' when initialized with \mathbf{x}_t at time $t > t'$. The network $\mathbf{h}_\theta(\cdot, t)$ that drives the process is a *consistent denoiser* if:

$$\mathbf{h}_\theta(\mathbf{x}_t, t) = \mathbb{E}_{\mathbf{X}_{t'} \sim p_\theta(\mathbf{x}_{t'}, t' | \mathbf{x}_t, t)} [\mathbf{h}_\theta(\mathbf{X}_{t'}, t')]. \quad (3.5)$$

The concept of consistency was introduced by Daras et al. [7] as a way to reduce error propagation in diffusion sampling and improve performance. Here, we find a completely different use case: we use consistency to learn the optimal denoisers for levels below the noise level of the available data. We are now ready to state our main theorem.

Theorem 3.4 (Main Theorem (informal)). *Define \mathbf{X}_t as in the beginning of Section 2. Suppose we are given samples $\mathbf{X}_{t_n} = \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}$, where $\mathbf{X}_0 \sim p_0$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I)$. Consider the following objective:*

$$J(\theta) = \underbrace{\mathbb{E}_{\mathbf{x}_{t_n}} \mathbb{E}_{t \sim \mathcal{U}(t_n, T)} \mathbb{E}_{\mathbf{x}_t = \mathbf{x}_{t_n} + \sqrt{\sigma_t^2 - \sigma_{t_n}^2} \boldsymbol{\eta}} \left[\left\| \frac{(\sigma_t^2 - \sigma_{t_n}^2)}{\sigma_t^2} (\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \mathbf{x}_t - \mathbf{x}_{t_n} \right\|^2 \right]}_{\text{Ambient Score Matching}} + \underbrace{\mathbb{E}_{t \sim \mathcal{U}(t_n, T), t' \sim \mathcal{U}(\epsilon, t), t'' \sim \mathcal{U}(t' - \epsilon, t')} \mathbb{E}_{\mathbf{x}_t} \mathbb{E}_{\mathbf{x}_{t'} | \mathbf{x}_t} \left[\left\| \mathbf{h}_\theta(\mathbf{x}_{t'}, t') - \mathbb{E}_{\mathbf{x}_{t''} \sim p_\theta(\mathbf{x}_{t''), t'' | \mathbf{x}_{t'}, t'} [\mathbf{h}_\theta(\mathbf{x}_{t''), t'']} \right\|^2 \right]}_{\text{Consistency Loss}},$$

where $\boldsymbol{\eta}$ in the above is a standard Gaussian vector. Suppose that the family of functions $\{\mathbf{h}_\theta\}$ is rich enough to contain the minimizer of the above objective overall functions $\mathbf{h}(\mathbf{x}, t)$. Then the minimizer θ^* of J satisfies:

$$\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t], \quad \forall \mathbf{x}_t, t. \quad (3.6)$$

The formal statement and the proof of this Theorem is given in the Appendix (see Theorem A.6).

Intuition and Proof Overview. It is useful to build some intuition about how this objective works. There are two terms in the loss: i) the Ambient Score Matching term and ii) the Consistency Loss. The Ambient Score Matching term regards only noise levels $t : \sigma_t > \sigma_{t_n}$. Per Theorem 3.1, this term has a unique minimizer that is the optimal denoiser for all levels $t : \sigma_t > \sigma_{t_n}$. The consistency term in the loss, penalizes for violations of the Consistency Property (see Definition 3.3) for all pairs of times t, t' . The desired solution, $\mathbf{h}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t]$, $\forall t, \mathbf{x}_t$, minimizes the first term and makes the second term 0, since it corresponds to a consistent denoiser. Hence, the desired solution is an optimal solution for the objective we wrote and the question becomes whether this solution is unique. The uniqueness of the solution arises from the Fokker-Planck PDE that describes the evolution of density: there is unique extension to a function that is $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t]$, $t : \sigma_t > \sigma_{t_n}$ and is consistent for all t . The latter result comes from Theorem 3.2 in Consistent Diffusion Models [7].

Implementation Trade-offs and Design Choices. When it comes to implementing the Consistency Loss there are trade-offs that need to be considered. First, we need to run partially the sampling chain. Doing so at every training step can lead to important slow-downs, as explained in Daras et al. [7]. To mitigate this, we choose the times t', t'' to be very close to one another, as in Consistent Diffusion, using a uniform distribution with support of width ϵ . This helps us run only 1 step of the sampling chain (without introducing big discretization errors) and it works because local consistency implies global consistency. Second, for the inner-term in the consistency loss we need to compute an expectation over samples of p_θ . To avoid running the sampling chain many times during training, we

opt for an unbiased estimator of this term that uses only two samples, following the implementation of Daras et al. [7]. Specifically, we use the approximation:

$$\begin{aligned} & \left| \left| \mathbf{h}_\theta(\mathbf{x}_{t'}, t') - \mathbb{E}_{\mathbf{x}_{t''} \sim p_\theta(\mathbf{x}_{t''}, t'' | \mathbf{x}_{t'}, t')} [\mathbf{h}_\theta(\mathbf{x}_{t''}, t'')] \right| \right|^2 \\ & \approx (\mathbf{h}_\theta(\mathbf{x}_{t''}^1, t'') - \mathbf{h}_\theta(\mathbf{x}_{t'}, t'))^T (\mathbf{h}_\theta(\mathbf{x}_{t''}^2, t'') - \mathbf{h}_\theta(\mathbf{x}_{t'}, t'')), \end{aligned}$$

where $\mathbf{x}_{t''}^1, \mathbf{x}_{t''}^2$ are samples from $p_\theta(\cdot | \mathbf{x}_{t'}, t')$.

3.3 Testing Training Data Replication

Learning from corrupted data is a potential mitigation strategy for the problem of training data replication. Thus, we need effective ways to evaluate the degree to which our models (and baselines trained on clean data) memorize.

A standard approach is to generate a few thousand samples with the trained models (potentially using the dataset prompts) and then measure the similarities of the generated samples with their nearest neighbors in the dataset [25, 8]. This approach is known to “systematically underestimate the amount of replication in Stable Diffusion and other models”, as noted by Somepalli et al. [25].

We propose a novel attack that shows that diffusion models memorize their training sets at a higher rate than previously known. We use the trained diffusion priors to solve inverse problems at extremely high corruption levels and we show that the reconstructions are often almost perfect.

4 Experimental Evaluation

4.1 Experiments with pre-trained models

In this section, we measure how much pre-trained foundation diffusion models memorize data from their training set. We perform our experiments with Stable Diffusion XL [20] (SDXL), as it is the state-of-the-art open-source image generation diffusion model.

We take a random 10,000 image subset of LAION and we corrupt it severely. We consider two models of corruption. In the first model, we take the LAION images and mask significant portions of them, as shown in Figure 1. The masked regions are selected automatically, using a YOLO object detection network [21], to contain whole faces or large objects that are impossible to perfectly predict by only observing the non-masked content of the image. Yet, as seen in the last row of Figure 1, some posterior samples are almost pixel-perfect matches of the original images. This strongly indicates that the images in the top row of Figure 1 were in the training set of SDXL and have been memorized. It is important to note that the captions (from the LAION dataset) are entered as input in the inpainting model and this attack did not seem to work with null captions.

In the second corruption model, we encode LAION images with the SDXL encoder and we add a significant amount of noise to them. In Figure 2, we show images from LAION dataset, their encodings (visualizing them as 3-channel RGB images), the noisy latents, the MMSE reconstruction (using the model’s one-step prediction at the noise level of the corruption) and posterior samples from the model. Again, even if the corruption is severe and the MMSE denoised images are very blurry, the posterior samples from the model are very close to the original images from the dataset, indicating potential memorization. In this corruption model the near-duplicate reconstructed images were obtained with null captions, so no text guidance was needed.

To quantify the degree of memorization and detect replication automatically, we adapt the methodology of Somepalli et al. [25]. In this work, the authors embed both the generated images and the dataset images to the DINOv2 [18] latent space, and for each generated image compute its maximum inner product (similarity score) with its nearest neighbor in the dataset. We repeat this experiment, previously done for Stable Diffusion v1.4, for the latest SDXL model. We empirically find that similarities above 0.95 correspond to almost identical samples to the ones in the training set and similarities above 0.9 correspond to close matches. We compare the distribution obtained using the Somepalli et al. [25] method with the distribution obtained using our noising approach (for two different noise levels) in Figure 3. As shown, our approach finds significantly more examples that have similarity values close to 1. Also as mentioned, our attack did not need the prompts in this case. This is not necessarily surprising since our approach uses more information (the noisy latents)

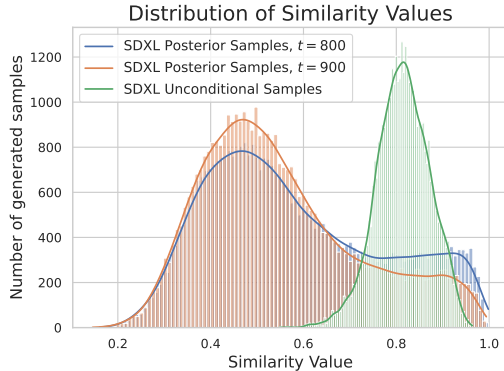


Figure 3: Distribution of image similarities of generated images with their nearest neighbors in the dataset for: i) the Somepalli et al. [25] method, and ii) for our noising method for two different noise levels. As shown, the fraction of images with similarities above 0.95 (near-identical to training set) is much higher for our method compared to the Somepalli et al. [25] baseline.

compared to the previously proposed method that only uses the prompts. Still, our results present evidence that diffusion models memorize significantly more training data compared to what was previously known. For the inpainting case, we only compute embeddings for the infilled regions and hence the similarity numbers are not directly comparable. We present these results in Figure 9 in the Appendix.

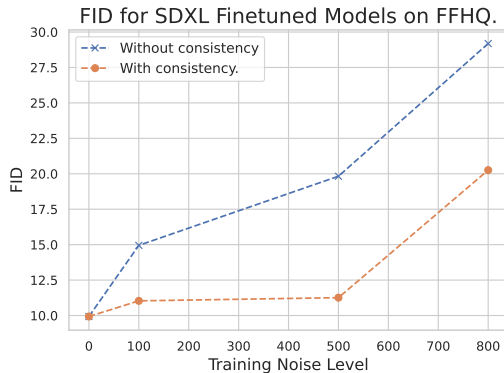


Figure 4: FID results for SDXL finetuned models, with and without consistency, on FFHQ, as we change the corruption level. The performance of models trained without consistency deteriorates significantly as we increase the corruption. Models trained with consistency maintain comparable performance to the baseline model (trained on clean data) for noise levels up to $t_n = 500$.

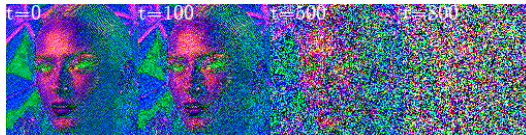
4.2 Finetuning Stable Diffusion XL

The next step is to use our framework, detailed in Sections 3.1, 3.2, to finetune SDXL on corrupted data. We finetune our models on FFHQ, at 1024×1024 resolution, since it is a standard benchmark for image generation. Given that SDXL is a latent model, we first encode the clean images using the SDXL encoder and then we add noise to the latents. We consider four noise levels which we will be referring to as: i) noiseless, $t_n = 0$, $\sigma_{t_n} = 0$, ii) low-noise, $t_n = 100$, $\sigma_{t_n} = 0.325$, iii) medium-noise, $t_n = 500$, $\sigma_{t_n} = 0.850$, and, iv) high-noise, $t_n = 800$, $\sigma_{t_n} = 0.981$. For reference, we fix an image from the training set and we visualize posterior samples for each one of the noise levels in Figure 5. We train models with our Ambient Denoising Score Matching loss, with and without consistency. We provide the training details in the Appendix, Section B.

We first evaluate the denoising performance of our models. To do so, we take 32 evaluation samples from FFHQ, we add noise to levels $t_{\text{eval}} \in \{900, 800, 500, 100\}$, we use our trained models to

Noise Level Eval	Noise Level Training	Latent MSE	Pixel MSE
900	100	0.3369 (0.0521)	0.0802 (0.0257)
	500	0.3377 (0.0514)	0.0804 (0.0253)
	800	0.3375 (0.0514)	0.0796 (0.0254)
800	100	0.2974 (0.0463)	0.0566 (0.0219)
	500	0.2978 (0.0464)	0.0566 (0.0222)
	800	0.3001 (0.0466)	0.0570 (0.0220)
500	100	0.2153 (0.0283)	0.0219 (0.0092)
	500	0.2159 (0.0283)	0.0221 (0.0092)
	800	0.2182 (0.0284)	0.0226 (0.0094)
100	100	0.0405 (0.0029)	0.0068 (0.0027)
	500	0.0409 (0.0029)	0.0069 (0.0028)
	800	0.0411 (0.0029)	0.0070 (0.0028)

Table 1: Restoration performance of models trained with noisy data at different noise levels. All the models have comparable performance, irrespective of the noise level of the dataset they were trained with.



(a) Noisy Latents for $t_n \in \{0, 100, 500, 800\}$.



(b) Posterior samples for $t_n \in \{0, 100, 500, 800\}$.

Figure 5: Visualization of the noise levels considered in the paper. The top row shows noisy latents, visualized as RGB images. The bottom row shows posterior samples obtained by the SDXL [20] model given these noise latents.

denoise and we measure the reconstruction error. Since SDXL is a latent diffusion model, the noise (and the denoising) happens in the latent space. Hence, the MSE reconstruction error can be measured directly in the latent space or pixel space (by decoding the reconstructed latents). We present our results in Table 1. As shown, all the models have comparable performance across all noise levels, irrespective of the noise level of the data they saw during training. This is in line with our theory: all the models are trained to estimate $\mathbb{E}[x_0|x_t]$ for all levels t .

To understand better the role of consistency, we visualize unconditional samples from our models trained with and without consistency in Figure 6. As shown in the left column of Figure 6, models trained without consistency lead to increasingly blurry generations as the level of noise encountered during training increases. This is not surprising: as explained in Subsection 2.3, models trained without consistency sample from the distribution of MMSE denoised images, $\mathbb{E}[X_0|X_{t_n}]$. As the noise level t_n increases, these images become averaged and high-frequency detail is lost. As shown in the right column on Figure 6, training with consistency recovers high-frequency details and leads to significantly improved images, especially for models trained with highly noisy data ($t_n \in \{500, 800\}$).

We proceed to evaluate unconditional generation performance. For each of our models, we generate 50,000 images and we compute the FID score. We visualize the performance of our models trained with and without consistency in Figure 4. As shown, the performance of models trained without consistency deteriorates significantly as we increase the corruption. This is compatible with Figure 6 (left) that shows that the generations become blurrier. Models trained on noisy data with consistency maintain comparable performance to the baseline model (trained on clean data) for noise levels up to $t_n = 500$ and are better everywhere compared to their counterparts trained without consistency.

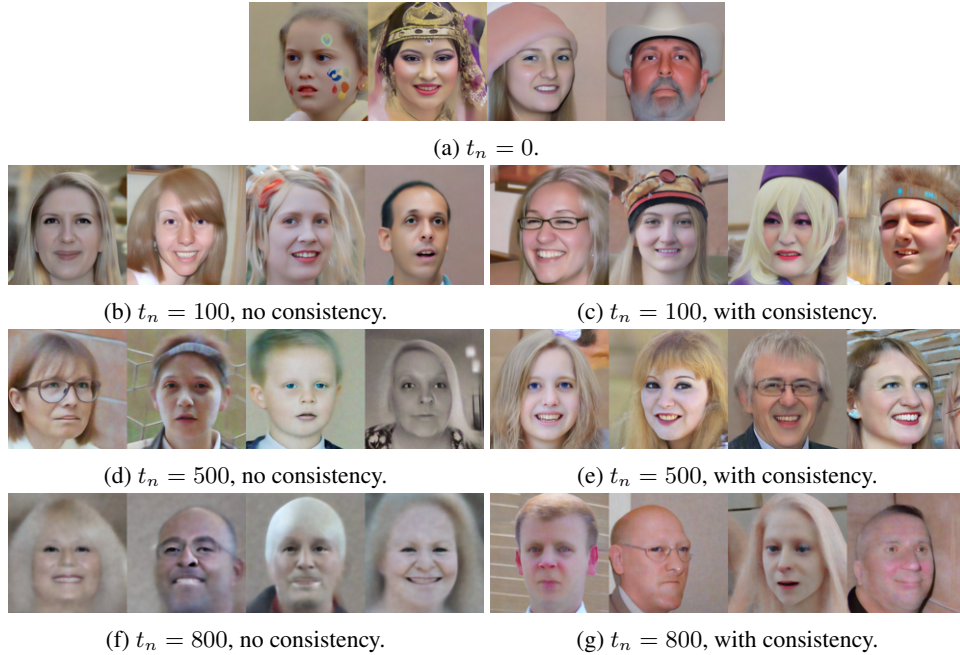


Figure 6: Unconditional generations for models trained with and without consistency at various noise levels t_n . Models trained without consistency lead to increasingly blurry generations as the noise level of the training data increases. Training with consistency recovers high-frequency details and leads to significantly improved images, especially for models trained on highly noisy data.

4.3 Additional Finetuning Experiments

We perform additional experiments to show that our framework can be used to fine-tune SDXL on datasets beyond FFHQ. We finetune SDXL on a 10k subset of LAION at different levels of noise in the training data and we show generations for different textual prompts at Figure 7. As shown, even for high levels of training corruption, the model is still capable of generating plausible images for arbitrary user prompts.

To further show that our method can be used for data that follow a distribution significantly different to the training distribution, we finetune SDXL on medical images. Specifically, we train the model on a dataset of MRIs of brain tumors and a dataset of chest x-rays. In Figure 11 of the Appendix, we provide some samples of the training dataset (Row 1), generated samples without fine-tuning (Row 2), noisy samples that were used to fine-tune the model (Row 3), generated samples after fine-tuning without consistency (Row 4) and finally generated samples after fine-tuning with consistency. For all our generations, we use prompts from the dataset of interest. The generations of the model without fine-tuning are very different compared to the dataset samples, hinting that the model initially models a different distribution conditioned on the given prompt. After fine-tuning with noisy data, the generated samples are more closely related to the samples from the dataset. As we also observed in the rest of the experiments in this paper, consistency decreases the blurriness of the generated samples.

4.4 Measuring Memorization of Finetuned Models

The final step in our experimental evaluation is to investigate to what extent training with noisy data reduced the rate of training data replication. To do so, we use the method we proposed in Section 4.1. Specifically, we get the FFHQ training images, we encode them to the latent space of the SDXL Encoder and we add noise to them that corresponds to $t_n = 900$. We then use the model trained with clean images and the model trained with data at $t_n = 500$ noise level to perform posterior sampling, given the noisy latents. For each generated sample, we measure its DINO similarity to the nearest neighbor in the dataset. We plot the resulting distributions for the model trained with clean data and the $t_n = 500$ model in Figure 8. As shown, the model trained with clean data (blue



(a) Finetuned model, no noise. (b) Finetuned model, $t_n = 100$. (c) Finetuned model, $t_n = 500$.

Figure 7: Generations of finetuned SDXL models on a 10k subset of LAION at different levels of noise in the training data. The following prompts were used: Row 1) “A propaganda poster depicting a cat dressed as french emperor napoleon.”, Row 2) “A high-quality image of a dog.”, Row 3) “Panda mad scientist mixing sparkling chemicals, artstation.”, Row 4) “A robot painted as graffiti on a brick wall.”.

curve), has a distribution of similarity values that is more shifted to the right, indicating higher dataset memorization compared to the model trained with corrupted data (orange curve). Finally, we once again compare with the method of Somepalli et al. [25] for identifying training data replications. We use the model trained with clean data, we take the 50,000 images that we used for FID generation and we compute their similarity to their nearest neighbor in the dataset. We compare with our approach in Figure 10 in the Appendix.

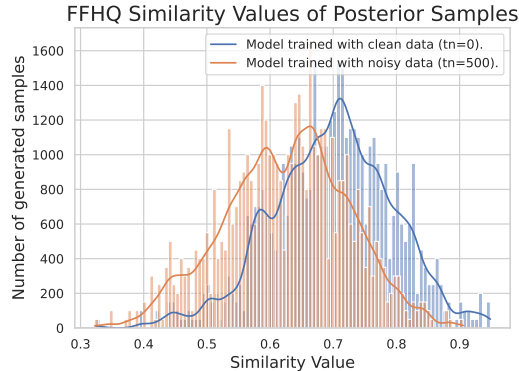


Figure 8: Distribution of similarities of posterior samples to their nearest neighbor in the dataset, given noisy latents (at $t = 900$) for two models. The model trained with clean data (blue curve), has a distribution of similarity values that is more shifted to the right, indicating higher dataset memorization compared to the model trained with corrupted data (orange curve).

5 Conclusions, Limitations and Future Work

We presented the first exact framework for training diffusion models to sample from an uncorrupted distribution using access to noisy data. We used our framework to finetune SDXL and we showed that training with corrupted data reduces memorization of the training set, while maintaining competitive performance. Our method has several limitations. First, it does not solve the problem of training diffusion models with *linearly* corrupted data that provably sample from the uncorrupted distribution. Second, training with consistency increases the training time [7]. Finally, in some preliminary experiments on very limited datasets (< 100 samples), the proposed Ambient Denoising Score Matching objective did not work. We plan to explore all these exciting open directions in future work.

6 Acknowledgments

This research has been supported by NSF Grants AF 1901292, CNS 2148141, Tripods CCF 1934932, IFML CCF 2019844 and research gifts by Western Digital, Amazon, WNCG IAP, UT Austin Machine Learning Lab (MLL), Cisco and the Stanly P. Finch Centennial Professorship in Engineering. Constantinos Daskalakis has been supported by NSF Awards CCF-1901292, DMS-2022448 and DMS-2134108, a Simons Investigator Award, and the Simons Collaboration on the Theory of Algorithmic Fairness. Giannis Daras has been supported by the Onassis Fellowship (Scholarship ID: F ZS 012-1/2022-2023), the Bodossaki Fellowship and the Leventis Fellowship.

References

- [1] Asad Aali, Marius Arvinte, Sidharth Kumar, and Jonathan I Tamir. “Solving Inverse Problems with Score-Based Generative Priors learned from Noisy Data”. In: *arXiv preprint arXiv:2305.01166* (2023) (pages 1, 4).
- [2] Hemant Kumar Aggarwal, Aniket Pramanik, Maneesh John, and Mathews Jacob. “ENSURE: A general approach for unsupervised training of deep image reconstruction algorithms”. In: *IEEE Transactions on Medical Imaging* 42.4 (2022), pp. 1133–1144 (page 4).
- [3] Brian D.O. Anderson. “Reverse-time diffusion equation models”. In: *Stochastic Processes and their Applications* 12.3 (1982), pp. 313–326 (page 3).
- [4] Joshua Batson and Loic Royer. “Noise2self: Blind denoising by self-supervision”. In: *International Conference on Machine Learning*. PMLR, 2019, pp. 524–533 (page 4).
- [5] 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”. In: *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf> 2 (2023), p. 3 (page 1).
- [6] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. “Extracting training data from diffusion models”. In: *32nd USENIX Security Symposium (USENIX Security 23)*. 2023, pp. 5253–5270 (page 1).
- [7] Giannis Daras, Yuval Dagan, Alexandros G Dimakis, and Constantinos Daskalakis. “Consistent diffusion models: Mitigating sampling drift by learning to be consistent”. In: *arXiv preprint arXiv:2302.09057* (2023) (pages 2, 6, 7, 11, 17).
- [8] 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*. 2023. URL: <https://openreview.net/forum?id=wBJBLy9kBY> (pages 1, 4, 7).
- [9] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, et al. “DataComp: In search of the next generation of multimodal datasets”. In: *arXiv preprint arXiv:2304.14108* (2023) (page 1).
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144 (page 1).
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851 (page 1).
- [12] Bahjat Kawar, Noam Elata, Tomer Michaeli, and Michael Elad. “GSURE-Based Diffusion Model Training with Corrupted Data”. In: *arXiv preprint arXiv:2305.13128* (2023) (pages 1, 4).
- [13] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. “Noise2void-learning denoising from single noisy images”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2129–2137 (page 4).
- [14] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. “Noise2Noise: Learning image restoration without clean data”. In: *arXiv preprint arXiv:1803.04189* (2018) (page 4).
- [15] Yvette Y Lin, Angela F Gao, and Katherine L Bouman. “IMAGING AN EVOLVING BLACK HOLE BY LEVERAGING SHARED STRUCTURE”. In: () (page 1).

- [16] Christopher A Metzler, Ali Mousavi, Reinhard Heckel, and Richard G Baraniuk. “Unsupervised learning with Stein’s unbiased risk estimator”. In: *arXiv preprint arXiv:1805.10531* (2018) (page 4).
- [17] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. “Noisier2noise: Learning to denoise from unpaired noisy data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12064–12072 (pages 4, 5).
- [18] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. “DINOv2: Learning Robust Visual Features without Supervision”. In: *arXiv preprint arXiv:2304.07193* (2023) (page 7).
- [19] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. “Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 2043–2052 (page 4).
- [20] 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: *arXiv preprint arXiv:2307.01952* (2023) (pages 1–3, 7, 9).
- [21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788 (page 7).
- [22] 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*. 2022, pp. 10684–10695 (page 1).
- [23] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. “Laion-5b: An open large-scale dataset for training next generation image-text models”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 25278–25294 (pages 1, 2).
- [24] Shakarim Soltanayev and Se Young Chun. “Training deep learning based denoisers without ground truth data”. In: *Advances in neural information processing systems* 31 (2018) (page 4).
- [25] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. “Diffusion Art or Digital Forgery? Investigating Data Replication in Diffusion Models”. In: *arXiv preprint arXiv:2212.03860* (2022) (pages 1, 7, 8, 11, 18, 19).
- [26] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. “Understanding and Mitigating Copying in Diffusion Models”. In: *arXiv preprint arXiv:2305.20086* (2023) (page 1).
- [27] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising diffusion implicit models”. In: *arXiv preprint arXiv:2010.02502* (2020) (page 18).
- [28] Yang Song and Stefano Ermon. “Generative modeling by estimating gradients of the data distribution”. In: *Advances in Neural Information Processing Systems* 32 (2019) (page 1).
- [29] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. “Score-based generative modeling through stochastic differential equations”. In: *arXiv preprint arXiv:2011.13456* (2020) (pages 1, 5, 15).
- [30] Charles M Stein. “Estimation of the mean of a multivariate normal distribution”. In: *The annals of Statistics* (1981), pp. 1135–1151 (page 4).
- [31] The Event Horizon Telescope Collaboration. “First M87 Event Horizon Telescope Results. IV. Imaging the Central Supermassive Black Hole”. In: *The Astrophysical Journal Letters* 875.1 (2019), p. L4. DOI: [10.3847/2041-8213/ab0e85](https://doi.org/10.3847/2041-8213/ab0e85). URL: <https://dx.doi.org/10.3847/2041-8213/ab0e85> (page 1).
- [32] Tiange Xiang, Mahmut Yurt, Ali B Syed, Kawin Setsompop, and Akshay Chaudhari. “DDM²: Self-Supervised Diffusion MRI Denoising with Generative Diffusion Models”. In: *arXiv preprint arXiv:2302.03018* (2023) (pages 1, 4).
- [33] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. “Noisy-as-clean: Learning self-supervised denoising from corrupted image”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9316–9329 (page 4).

A Theoretical Results

In this section, we provide the proofs for the theoretical results of the main paper.

A.1 Preliminaries

We start by stating and proving a generalized version of Tweedie's formula that will be useful for learning the optimal denoisers, given only noisy data at σ_{t_n} , for noise levels higher than the level of the noise in the data, i.e. for $\sigma_t \geq \sigma_{t_n}$.

Lemma A.1 (Generalized Tweedie's Formula). *Let:*

$$\mathbf{X}_t = \alpha_t \mathbf{X}_0 + \sigma_t \mathbf{Z}, \quad (\text{A.1})$$

for $\mathbf{X}_0 \sim p_0$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I)$. Then,

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \frac{\alpha_t \mathbb{E}[\mathbf{X}_0 | \mathbf{x}_t] - \mathbf{x}_t}{\sigma_t^2}. \quad (\text{A.2})$$

Proof.

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \frac{1}{p_t(\mathbf{x}_t)} \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t) = \frac{1}{p_t(\mathbf{x}_t)} \nabla_{\mathbf{x}_t} \int p_t(\mathbf{x}_t, \mathbf{x}_0) d\mathbf{x}_0 \quad (\text{A.3})$$

$$= \frac{1}{p_t(\mathbf{x}_t)} \nabla_{\mathbf{x}_t} \int p_t(\mathbf{x}_t | \mathbf{x}_0) p_0(\mathbf{x}_0) d\mathbf{x}_0 \quad (\text{A.4})$$

$$= \frac{1}{p_t(\mathbf{x}_t)} \int \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t | \mathbf{x}_0) p_0(\mathbf{x}_0) d\mathbf{x}_0 \quad (\text{A.5})$$

$$= \frac{1}{p_t(\mathbf{x}_t)} \int p_t(\mathbf{x}_t | \mathbf{x}_0) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}_0) p_0(\mathbf{x}_0) d\mathbf{x}_0 \quad (\text{A.6})$$

$$= \int p_0(\mathbf{x}_0 | \mathbf{x}_t) \frac{\alpha_t \mathbf{x}_0 - \mathbf{x}_t}{\sigma_t^2} d\mathbf{x}_0 \quad (\text{A.7})$$

$$= \frac{\alpha_t \mathbb{E}[\mathbf{X}_0 | \mathbf{x}_t] - \mathbf{x}_t}{\sigma_t^2}. \quad (\text{A.8})$$

□

A.2 Learning the Optimal Denoisers for $\sigma_t \geq \sigma_{t_n}$

We will now use Lemma 3.2 to connect the conditional expectation of \mathbf{X}_0 given \mathbf{X}_t , $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t]$, to the conditional expectation of \mathbf{X}_{t_n} given \mathbf{X}_t , $\mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t]$, for $t : \sigma_t \geq \sigma_{t_n}$. The latter can be learned with supervised learning and hence this connection will give us a way to learn how to find the best denoised image at level $t = 0$ given only access to data at t_n .

Lemma A.2 (Connecting Conditional Expectations – Variance Exploding). *Let $\mathbf{X}_{t_n} = \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}_1$ and $\mathbf{X}_t = \mathbf{X}_0 + \sigma_t \mathbf{Z}_2$, $\mathbf{Z}_1, \mathbf{Z}_2 \sim \mathcal{N}(\mathbf{0}, I)$ i.i.d. Then, for any $\sigma_t > \sigma_{t_n}$, we have that:*

$$\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t] = \frac{\sigma_t^2}{\sigma_t^2 - \sigma_{t_n}^2} \mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t] - \frac{\sigma_{t_n}^2}{\sigma_t^2 - \sigma_{t_n}^2} \mathbf{x}_t + \mathbf{x}_t. \quad (\text{A.9})$$

Proof. Applying Tweedie's formula (Lemma 3.2) for the pair $\mathbf{X}_t, \mathbf{X}_0$ we have:

$$\nabla \log p_t(\mathbf{x}_t) = \frac{\mathbb{E}[\mathbf{X}_0 | \mathbf{x}_t] - \mathbf{x}_t}{\sigma_t^2}. \quad (\text{A.10})$$

But also, $\mathbf{X}_t = \mathbf{X}_{t_n} + \sqrt{\sigma_t^2 - \sigma_{t_n}^2} \mathbf{Z}$. Applying again Tweedie's formula for $\mathbf{X}_t, \mathbf{X}_{t_n}$ we get:

$$\nabla \log p_t(\mathbf{x}_t) = \frac{\mathbb{E}[\mathbf{X}_{t_n} | \mathbf{x}_t] - \mathbf{x}_t}{\sigma_t^2 - \sigma_{t_n}^2}. \quad (\text{A.11})$$

From A.10, A.11, we get:

$$\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t] = \frac{\sigma_t^2}{\sigma_t^2 - \sigma_{t_n}^2} \mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t] - \frac{\sigma_{t_n}^2}{\sigma_t^2 - \sigma_{t_n}^2} \mathbf{x}_t + \mathbf{x}_t. \quad (\text{A.12})$$

□

We are now ready to present the proof of Theorem 3.1. For completeness, we restate the theorem here.

Theorem A.3 (Ambient Denoising Score Matching; restated Theorem 3.1). *Define \mathbf{X}_t as in the beginning of Section 2. Suppose we are given samples $\mathbf{X}_{t_n} = \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}$, where $\mathbf{X}_0 \sim p_0$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I)$. Consider the following objective:*

$$J(\theta) = \mathbb{E}_{\mathbf{x}_{t_n}} \mathbb{E}_{t \sim \mathcal{U}(t_n, T)} \mathbb{E}_{\mathbf{x}_t = \mathbf{x}_{t_n} + \sqrt{\sigma_t^2 - \sigma_{t_n}^2} \boldsymbol{\eta}} \left[\left\| \frac{(\sigma_t^2 - \sigma_{t_n}^2)}{\sigma_t^2} (\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \mathbf{x}_t - \mathbf{x}_{t_n} \right\|^2 \right],$$

where $\boldsymbol{\eta}$ in the above is a standard Gaussian vector. Suppose that the family of functions $\{\mathbf{h}_\theta\}$ is rich enough to contain the minimizer of the above objective overall functions $\mathbf{h}(\mathbf{x}, t)$. Then the minimizer θ^* of J satisfies:

$$\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t], \quad \forall \mathbf{x}_t, t > t_n. \quad (\text{A.13})$$

Proof. We start by using the definition of conditional expectation as the unique minimizer of the mean squared error objective. Specifically, we know that the solution to the optimization problem:

$$\tilde{J}(\theta) = \mathbb{E}_{\mathbf{x}_{t_n}} \mathbb{E}_{t \sim \mathcal{U}(t_n, T)} \mathbb{E}_{\mathbf{x}_t = \mathbf{x}_{t_n} + \sqrt{\sigma_t^2 - \sigma_{t_n}^2} \boldsymbol{\eta}} \left[\|\mathbf{g}_\theta(\mathbf{x}_t, t) - \mathbf{x}_{t_n}\|^2 \right], \quad (\text{A.14})$$

for a rich enough family of functions \mathbf{g}_θ is $\mathbf{g}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t]$. We can parametrize \mathbf{g}_θ as $\mathbf{g}_\theta(\mathbf{x}_t, t) = \frac{\sigma_t^2 - \sigma_{t_n}^2}{\sigma_t^2} (\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \mathbf{x}_t$ and solve the following optimization problem:

$$\min_{\theta: \mathbf{g}_\theta(\mathbf{x}_t, t) = \frac{\sigma_t^2 - \sigma_{t_n}^2}{\sigma_t^2} (\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \mathbf{x}_t} \mathbb{E}_{\mathbf{x}_{t_n}} \mathbb{E}_{t \sim \mathcal{U}(t_n, T)} \mathbb{E}_{\mathbf{x}_t = \mathbf{x}_{t_n} + \sqrt{\sigma_t^2 - \sigma_{t_n}^2} \boldsymbol{\eta}} \left[\|\mathbf{g}_\theta(\mathbf{x}_t, t) - \mathbf{x}_{t_n}\|^2 \right], \quad (\text{A.15})$$

which will have the same minimizer since any solution of $\tilde{J}(\theta)$ remains feasible. Hence,

$$\mathbf{g}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t] \iff \quad (\text{A.16})$$

$$\frac{\sigma_t^2 - \sigma_{t_n}^2}{\sigma_t^2} (\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) - \mathbf{x}_t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \mathbf{x}_t = \mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t]. \quad (\text{A.17})$$

Using Lemma A.2, the latter implies that $\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0 | \mathbf{X}_t = \mathbf{x}_t]$, as needed. □

A.3 Extensions to Variance Preserving Diffusion

The results that we presented can be easily extended to the Variance Preserving [29] case, where the observed data are:

$$\mathbf{X}_{t_n} = \sqrt{1 - \sigma_{t_n}^2} \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}, \quad 0 < \sigma_{t_n} < 1. \quad (\text{A.18})$$

We first extend Lemma A.2.

Lemma A.4 (Connecting Conditional Expectations – Variance Preserving). *Let $\mathbf{X}_{t_n} = \sqrt{1 - \sigma_{t_n}^2} \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}_1$, $0 < \sigma_{t_n} < 1$, and $\mathbf{X}_t = \sqrt{1 - \sigma_t^2} \mathbf{X}_0 + \sigma_t \mathbf{Z}_2$, $\mathbf{Z}_1, \mathbf{Z}_2 \sim \mathcal{N}(\mathbf{0}, I)$ i.i.d. Then, for any $1 > \sigma_t > \sigma_{t_n}$, we have that:*

$$\mathbb{E}[\mathbf{x}_0 | \mathbf{X}_t = \mathbf{x}_t] = \frac{\sigma_t^2}{\sigma_t^2 - \sigma_{t_n}^2} \sqrt{1 - \sigma_{t_n}^2} \mathbb{E}[\mathbf{x}_{t_n} | \mathbf{X}_t = \mathbf{x}_t] - \sigma_{t_n}^2 \frac{\sqrt{1 - \sigma_t^2}}{\sigma_t^2 - \sigma_{t_n}^2} \mathbf{x}_t. \quad (\text{A.19})$$

Proof. Applying Tweedie's formula (Lemma 3.2) for the pair $\mathbf{X}_t, \mathbf{X}_0$ we have:

$$\nabla \log p_t(\mathbf{x}_t) = \frac{\sqrt{1 - \sigma_t^2} \mathbb{E}[\mathbf{X}_0 | \mathbf{x}_t] - \mathbf{x}_t}{\sigma_t^2}. \quad (\text{A.20})$$

The next step is to express \mathbf{X}_t as a function of \mathbf{X}_{t_n} . We want to find co-efficients α, β such that:

$$\mathbf{X}_t = \alpha \mathbf{X}_{t_n} + \beta \mathbf{Z}_2 \iff (\text{A.21})$$

$$\mathbf{X}_t = \alpha \sqrt{1 - \sigma_{t_n}^2} \mathbf{X}_0 + \alpha \sigma_{t_n} \mathbf{Z}_1 + \beta \mathbf{Z}_2 \iff (\text{A.22})$$

$$\mathbf{X}_t = \alpha \sqrt{1 - \sigma_{t_n}^2} \mathbf{X}_0 + \sqrt{\alpha^2 \sigma_{t_n}^2 + \beta^2} \mathbf{Z}. \quad (\text{A.23})$$

Since $\mathbf{X}_t = \sqrt{1 - \sigma_t^2} \mathbf{X}_0 + \sigma_t \mathbf{Z}$, this implies that the values of the desired co-efficients are:

$$\alpha = \sqrt{\frac{1 - \sigma_t^2}{1 - \sigma_{t_n}^2}}, \quad \beta = \sqrt{\frac{\sigma_t^2 - \sigma_{t_n}^2}{1 - \sigma_{t_n}^2}}, \quad (\text{A.24})$$

i.e. \mathbf{X}_t can be expressed as:

$$\mathbf{X}_t = \sqrt{\frac{1 - \sigma_t^2}{1 - \sigma_{t_n}^2}} \mathbf{X}_{t_n} + \sqrt{\frac{\sigma_t^2 - \sigma_{t_n}^2}{1 - \sigma_{t_n}^2}} \mathbf{Z}_2. \quad (\text{A.25})$$

By applying Tweedie's formula again for the $\mathbf{X}_t, \mathbf{X}_{t_n}$ pair, we get that:

$$\nabla \log p_t(\mathbf{x}_t) = \frac{\sqrt{\frac{1 - \sigma_t^2}{1 - \sigma_{t_n}^2}} \mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t] - \mathbf{x}_t}{\frac{\sigma_t^2 - \sigma_{t_n}^2}{1 - \sigma_{t_n}^2}} \iff (\text{A.26})$$

$$\nabla \log p_t(\mathbf{x}_t) = \frac{\sqrt{(1 - \sigma_t^2)(1 - \sigma_{t_n}^2)} \mathbb{E}[\mathbf{X}_{t_n} | \mathbf{X}_t = \mathbf{x}_t] - (1 - \sigma_{t_n}^2) \mathbf{x}_t}{\sigma_t^2 - \sigma_{t_n}^2}. \quad (\text{A.27})$$

Finally, by equating A.20, A.27, we find that:

$$\mathbb{E}[\mathbf{x}_0 | \mathbf{X}_t = \mathbf{x}_t] = \frac{\sigma_t^2}{\sigma_t^2 - \sigma_{t_n}^2} \sqrt{1 - \sigma_{t_n}^2} \mathbb{E}[\mathbf{x}_{t_n} | \mathbf{X}_t = \mathbf{x}_t] - \sigma_{t_n}^2 \frac{\sqrt{1 - \sigma_t^2}}{\sigma_t^2 - \sigma_{t_n}^2} \mathbf{x}_t. \quad (\text{A.28})$$

□

We can now use this result to write an objective for the VP case that learns $\mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t], \forall t : \sigma_t > \sigma_{t_n}$, as in Theorem 3.1.

Theorem A.5 (Ambient Denoising Score Matching – VP case). *Let $\mathbf{X}_{t_n} = \sqrt{1 - \sigma_{t_n}^2} \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}$, $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I)$ and $\mathbf{X}_t = \sqrt{1 - \sigma_t^2} \mathbf{X}_0 + \sigma_t \mathbf{Z}$, $t : 1 > \sigma_t > \sigma_{t_n} > 0$. Then, the unique minimizer of the objective:*

$$J(\theta) = \mathbb{E}_{\mathbf{x}_{t_n}} \mathbb{E}_{t \sim \mathcal{U}(t_n, T)} \mathbb{E}_{\mathbf{x}_t | \mathbf{x}_{t_n}} \left[\left\| \frac{\sigma_t^2 - \sigma_{t_n}^2}{\sigma_t^2 \sqrt{1 - \sigma_{t_n}^2}} \mathbf{h}_\theta(\mathbf{x}_t, t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \sqrt{\frac{1 - \sigma_t^2}{1 - \sigma_{t_n}^2}} \mathbf{x}_t - \mathbf{x}_{t_n} \right\|^2 \right] \quad (\text{A.29})$$

is:

$$\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0 | \mathbf{X}_t = \mathbf{x}_t], \quad \forall t : 1 > \sigma_t > \sigma_{t_n}. \quad (\text{A.30})$$

The proof of this Theorem is skipped for brevity since it follows the same steps as the proof of Theorem 3.1, with the only difference being that it invokes Lemma A.4 instead of Lemma A.2.

A.4 Learning the Optimal Denoisers for $\sigma_t \leq \sigma_{t_n}$

We are now ready to present the theory for learning the optimal denoisers for $\sigma_t \leq \sigma_{t_n}$. The formal version of our main Theorem (3.4) is given below.

Theorem A.6 (Main Theorem). *Define \mathbf{X}_t as in the beginning of Section 2. Suppose we are given samples $\mathbf{X}_{t_n} = \mathbf{X}_0 + \sigma_{t_n} \mathbf{Z}$, where $\mathbf{X}_0 \sim p_0$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, I)$. Let also $p_\theta(\mathbf{x}_{t'}, t' | \mathbf{x}_t, t)$ be the density of the sample $\mathbf{X}_{t'}$ sampled by the stochastic diffusion process of Equation 2.3, at time t' when initialized with \mathbf{x}_t at time $t > t'$. Consider the following objective:*

$$\begin{aligned}
 J(\theta) = & \overbrace{\mathbb{E}_{\mathbf{x}_{t_n}} \mathbb{E}_{t \sim \mathcal{U}(t_n, T)} \mathbb{E}_{\mathbf{x}_t = \mathbf{x}_{t_n} + \sqrt{\sigma_t^2 - \sigma_{t_n}^2} \boldsymbol{\eta}} \left[\left\| \frac{(\sigma_t^2 - \sigma_{t_n}^2)}{\sigma_t^2} (\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t) + \frac{\sigma_{t_n}^2}{\sigma_t^2} \mathbf{x}_t - \mathbf{x}_{t_n} \right\|^2 \right]}^{\text{Ambient Score Matching}} \\
 & + \underbrace{\mathbb{E}_{t \sim \mathcal{U}(t_n, T), t' \sim \mathcal{U}(0, t), t'' \sim \mathcal{U}[t' - \epsilon, t']} \mathbb{E}_{\mathbf{x}_t} \mathbb{E}_{\mathbf{x}_{t'} \sim p_\theta(\mathbf{x}_{t'}, t' | \mathbf{x}_t, t)} \left[\left\| \mathbf{h}_\theta(\mathbf{x}_{t'}, t') - \mathbb{E}_{\mathbf{x}_{t''} \sim p_\theta(\mathbf{x}_{t''), t'' | \mathbf{x}_{t'}, t'} [\mathbf{h}_\theta(\mathbf{x}_{t''), t'']} \right\|^2 \right]}_{\text{Consistency Loss}},
 \end{aligned} \tag{A.31}$$

where $\boldsymbol{\eta}$ in the above is a standard Gaussian vector, and subject to:

$$\begin{cases}
 \text{(A1): } \mathbf{h}_\theta(\mathbf{x}_0, 0) = \mathbf{x}_0, \forall \mathbf{x}_0; \\
 \text{(A2): } \frac{\mathbf{h}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t}{\sigma_t^2} = \nabla \Phi(\mathbf{x}_t), \text{ for some scalar-valued function } \Phi, \forall t, \mathbf{x}_t.
 \end{cases} \tag{A.32}$$

Suppose that the family of functions $\{\mathbf{h}_\theta\}$ is rich enough to contain the minimizer of the above objective overall functions $\mathbf{h}(\mathbf{x}, t)$. Then the minimizer θ^* of J satisfies:

$$\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t], \quad \forall \mathbf{x}_t, t. \tag{A.33}$$

Proof. The first term of the loss involves predictions of the network only for $t : \sigma_t > \sigma_{t_n}$. By Theorem 3.1, for these times t , there is a unique minimizer and the solution should satisfy:

$$\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t], \quad \forall t : \sigma_t > \sigma_{t_n}. \tag{A.34}$$

The solution:

$$\mathbf{h}_{\theta^*}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{X}_0 | \mathbf{X}_t = \mathbf{x}_t], \quad \forall t, \tag{A.35}$$

is one minimizer of the loss since: i) it is a feasible solution (satisfies (A1), (A2)), ii) it satisfies the condition of Equation A.34 that corresponds to the minimization of the first term, and iii) it makes the second term of the loss 0 (by the tower law of expectation). Hence, the only thing left to show is that the solution is unique for times $t : \sigma_t \leq \sigma_{t_n}$.

Let $\mathbf{h}_{\bar{\theta}}$ be another optimal solution. It has to satisfy the following properties:

1. $\mathbf{h}_{\bar{\theta}}$ needs to make the second term in the loss 0, i.e. $\mathbf{h}_{\bar{\theta}}$ is a consistent denoiser (see Definition 3.3) for all t . This is because we found another minimizer that minimizes the first term of the loss and makes the second term 0.
2. $\mathbf{h}_{\bar{\theta}}$ satisfies (A1), (A2) since the optimal solution should be a feasible solution.
3. $\mathbf{h}_{\bar{\theta}}$ needs to satisfy Eq. A.34, since the first term in the loss has a unique minimizer.

By Theorem 3.2 (part ii) of Consistent Diffusion Models [7], the only function that satisfies properties 1., 2., 3. is the function \mathbf{h}_{θ^*} and hence the solution is unique. \square

B Experimental Details

In this section, we provide further regarding the SDXL finetuning experiments. We train all our models with a batch size of 16 using a constant learning rate $1e-5$. For all our experiments, we use the Adam optimizer with the following hyperparameters: $\beta_1 = 0.9, \beta_2 = 0.999$, weight decay = 0.01. We train all of our models for at least 200,000 steps or roughly 45 epochs on FFHQ. The models trained with consistency were finetuned for 50,000 steps, initialized from the models trained without consistency after 150,000 steps. We did this to save computation time since training with consistency loss takes $\approx 3\times$ more time compared to vanilla training. During finetuning, we used a weight of 0.01 for the consistency loss for the $t_n \in \{100, 500\}$ models and a weight of $1e-4$ for our $t_n = 800$ model. We noticed that further increasing the weight for the latter led to training collapse.

For our finetuning, we use LoRA with rank 4, following the implementation of SDXL finetuning from the `diffusers` Github repository. We train all of our models on 16-bit precision to reduce the memory requirements and accelerate training speed. For all the experiments in the paper (including FID evaluation) the images were generated using 25 inference steps and the DDIM [27] sampling algorithm. We underline that better performance could have been achieved by increasing the number of steps, the training and sampling precision and by carefully tuning the batch size. However, in this paper, we did not optimize for state-of-the-art unconditional generation performance but rather we focused on building a complete and exact framework for learning diffusion models from noisy data.

C Additional Results

In this section, we provide additional results that were not included in the main paper. Figure 9 shows the memorization curves for the SDXL inpainting experiment (see also Figure 1). Figure 10 compares the Somepalli et al. [25] method for detecting training data replication with our proposed method that works by denoising extremely corrupted encodings of dataset images. We once again underline that it is not surprising that our method indicates higher memorization since it has access to more information (the noisy latents).

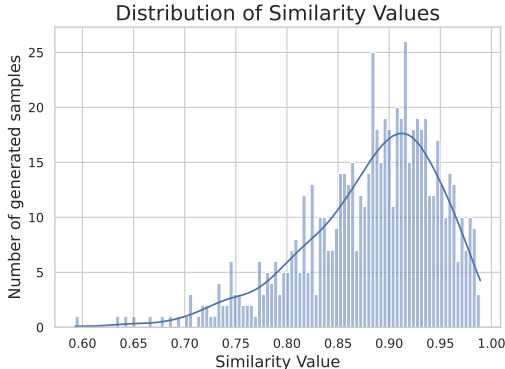


Figure 9: Distribution of image similarities of generated images with their nearest neighbors in the dataset for inpainting attack.

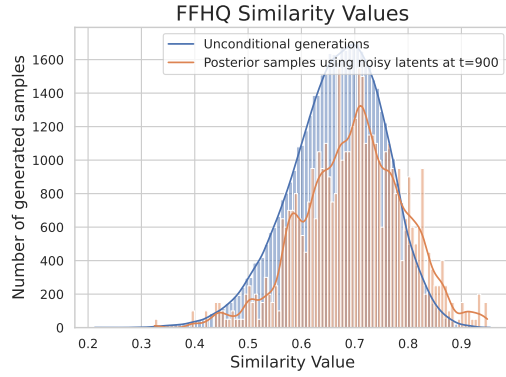


Figure 10: Comparison of [25] method for detecting training data replication with our proposed method that works by denoising extremely noisy dataset latents. The comparison is given for an SDXL model finetuned on clean FFHQ images. Our method (orange curve) gives a distribution that is more shifted to the right, indicating higher dataset memorization. This is not surprising since we have access to more information (noisy latents) compared to the baseline method.

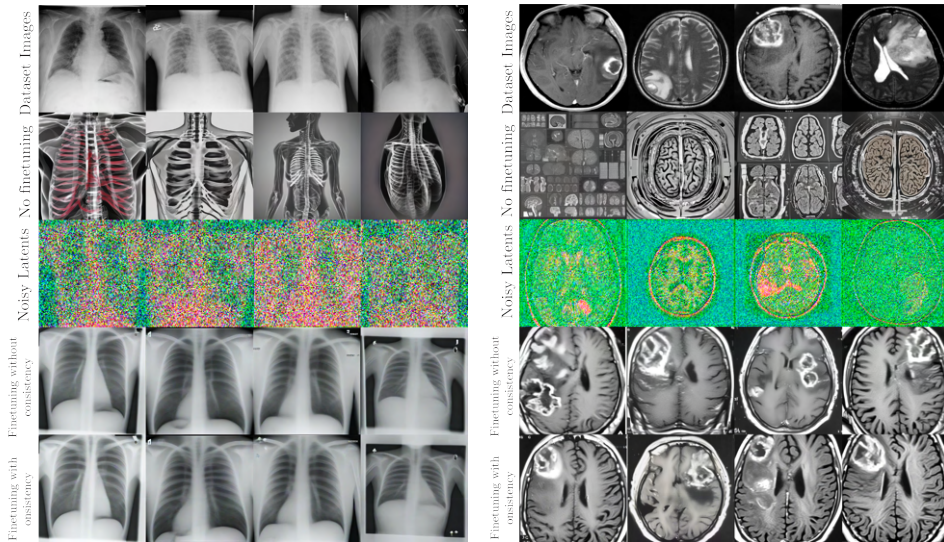


Figure 11: SDXL Finetuning on medical datasets. Row 1: samples of the training dataset, Row 2: generated samples without fine-tuning, Row 3: noisy samples that were used to fine-tune the model, Row 4: generated samples after fine-tuning without consistency, and, Row 5: generated samples after fine-tuning with consistency.