Ambient Diffusion: Learning Clean Distributions from Corrupted Data

Giannis Daras
UT Austin
giannisdaras@utexas.edu

Kulin Shah
UT Austin
kulinshah@utexas.edu

Yuval Dagan
UC Berkeley
yuvald@berkeley.edu

Aravind Gollakota
Apple
aravindg@cs.utexas.edu

Alexandros G. Dimakis
UT Austin
dimakis@austin.utexas.edu

Adam Klivans
UT Austin
klivans@utexas.edu

Abstract

We present the first diffusion-based framework that can learn an unknown distribution using only highly-corrupted samples. This problem arises in scientific applications where access to uncorrupted samples is impossible or expensive to acquire. Another benefit of our approach is the ability to train generative models that do not memorize any individual training sample, since they never observe clean training data. Our main idea is to introduce *additional measurement distortion* during the diffusion process and require the model to predict the original corrupted image from the further corrupted image. We prove that our method leads to models that learn the conditional expectation of the full uncorrupted image given this additional measurement corruption. This holds for any corruption process that satisfies some technical conditions (and in particular includes inpainting and compressed sensing). We train models on standard benchmarks (CelebA, CIFAR-10 and AFHQ) and show that we can learn the distribution even when all the training samples have 90% of their pixels missing. We also show that when we finetune foundation models using small corrupted datasets (e.g. MRI scans with block corruptions), we learn the clean distribution without memorizing the training set.

1 Introduction

Diffusion generative models [45, 20, 48] are emerging as versatile and powerful frameworks for learning high-dimensional distributions and solving inverse problems [30, 10, 31, 24]. Numerous recent developments [49, 26] have led to text conditional foundation models like Dalle-2 [37], Latent Diffusion [41] and Imagen [43] with an incredible performance in general image domains. Training these models requires access to high-quality datasets which may be expensive or impossible to obtain. For example, direct images of black holes cannot be observed [11, 16] and high quality MRI images require long scanning times, causing patient discomfort and motion artifacts [24].

Recently, Carlini et al. [7], Somepalli et al. [46], and Jagielski et al. [23] showed that diffusion models can memorize examples from their training set. Further, an adversary can extract dataset samples given only query access to the model, leading to privacy, security and copyright concerns. For many applications, we may want to learn the distribution but not individual training images e.g. we might want to learn the distribution of X-ray scans but not memorize images of specific patient scans from the dataset. Hence, we may want to introduce corruption as a design choice. We show that it is possible to train diffusions that learn a distribution of clean data by only observing highly corrupted samples.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

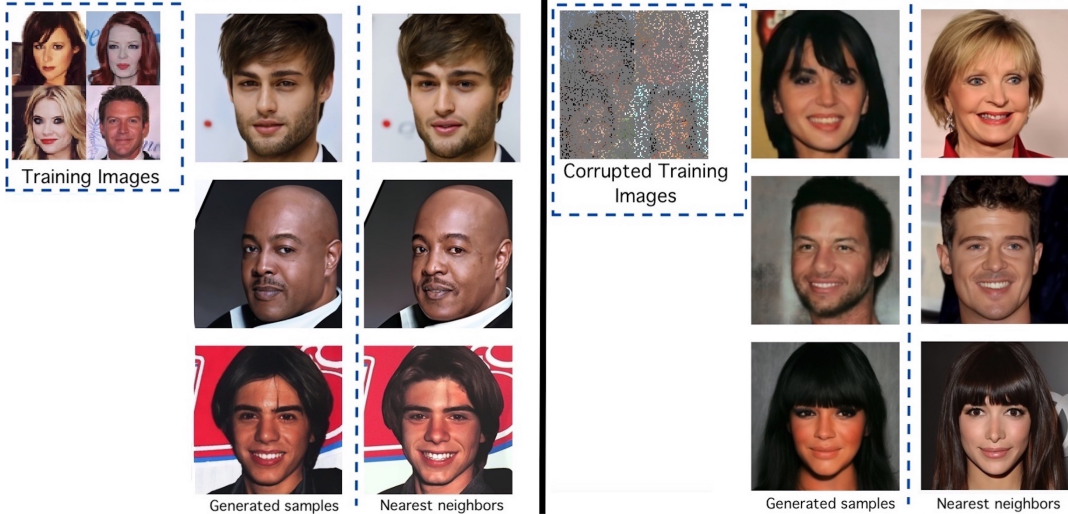



Figure 1: **Left panel:** Baseline method of vanilla finetuning Deepfloyd IF using 3000 images from CelebA-HQ. We show generated sample images and nearest neighbors from the finetuning set. As shown, the generated samples are often near-identical copies from training data. This verifies related work Carlini et al. [7], Somepalli et al. [46], and Jagielski et al. [23] that pointed out that diffusions often generate training samples. **Right panel:** We finetune the same foundation model (Deepfloyd IF) using our method and 3000 highly corrupted training images. The corruption adds noise and removes 80 percent random pixels. We show generated samples and nearest neighbors from the training set. Our method still learns the clean distribution of faces (with some quality deterioration, as shown) but does not memorize training data. We emphasize that our method is performed without ever accessing clean training data.

Prior work in supervised learning from corrupted data. The traditional approach to solving such problems involves training a restoration model using supervised learning to predict the clean image based on the measurements [39, 40, 54, 35]. The seminal Noise2Noise [34] work introduced a practical algorithm for learning how to denoise in the absence of any non-noisy images. This framework and its generalizations [4, 33, 50] have found applications in electron microscopy [13], tomographic image reconstruction [53], fluorescence image reconstruction [56], blind inverse problems [17, 4], monocular depth estimation and proteomics [5]. Another related line of work uses Stein's Unbiased Risk Estimate (SURE) to optimize an unbiased estimator of the denoising objective without access to non-noisy data [15]. The supervised learning framework is effective when the corruption level is sufficiently low so that a reconstruction algorithm is effective. However, it might be either not possible or not desirable to reconstruct individual samples. Instead, the desired goal may be to learn the distribution of the uncorrupted data but without being able to reconstruct individual training samples.

For certain corruption processes, it is theoretically possible to perfectly learn a distribution from highly corrupted samples, even from one-dimensional projections. Specifically, AmbientGAN [6] showed that general d dimensional distributions can be learned from *scalar* observations, by observing only projections on one-dimensional random Gaussian vectors, in the infinite training data limit. The theory requires an infinitely powerful discriminator and hence does not apply to diffusion models.

Our contributions. In this paper, we present the first diffusion-based framework to learn an unknown distribution $\mathcal{D}$ when the training set only contains highly-corrupted examples drawn from $\mathcal{D}$. Our main idea is to introduce *additional measurement distortion* during the diffusion process and require the model to predict the original corrupted image from the further corrupted image.

- Theoretical contributions:
  - We consider the problem of learning to sample from the target distribution $p_0(\boldsymbol{x}_0)$ given corrupted samples $A\boldsymbol{x}_0$ where $A \sim p(A)$ is a random corruption matrix (with known realizations and prior distribution) and $\boldsymbol{x}_0 \sim p_0(\boldsymbol{x}_0)$.

2

| Random in- | Further corrup- | Block inpaint- | Further corrup- |
| painting | tion | ing | tion |

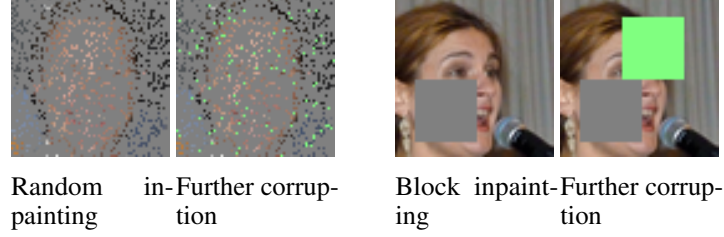



Figure 2: Illustration of our method: Given training data with deleted pixels, we corrupt further by erasing more (illustrated with green color). We feed the learner the further corrupted images and we evaluate it on the originally observed pixels. We can do this during training since the green pixel values are known to us. The score network learner has no way of knowing whether a pixel was missing from the beginning or whether it was corrupted by us. Hence, the score network learns to predict the clean image everywhere. Our method is analogous to grading a random subset of the questions in a test, but the students not knowing which questions will be graded.

- We provide an algorithm that provably learns $\mathbb{E}[\boldsymbol{x}_0 | \tilde{A}(\boldsymbol{x}_0 + \sigma_t \boldsymbol{\eta}), \tilde{A}]$, where $\tilde{A} \sim p(\tilde{A} \mid A)$ is a further corrupted version of A. The result holds for a general family of corruption processes $A \sim p(A)$. For various corruption processes, we show that the further degradation introduced by $\tilde{A}$ can be very small.

- Experimental contributions:
  - We use our algorithm to train diffusion models on standard benchmarks (CelebA, CIFAR-10 and AFHQ) with training data at different levels of corruption.
  - Given the learned conditional expectations we provide an approximate sampler for the target distribution $p_0(\boldsymbol{x}_0)$.
  - We show that for up to 90% missing pixels, we can learn reasonably well the distribution of uncorrupted images. We measure the tradeoff between the amount of corruption, the amount of training data and the quality of the learned generator. We outperform the previous state-of-the-art AmbientGAN [6] and natural baselines.
  - We show that our models perform on par or even outperform state-of-the-art diffusion models for solving certain inverse problems even without ever seeing a clean image during training. Our models do so with a single prediction step while our baselines require hundreds of diffusion steps.
  - We use our algorithm to finetune foundational pretrained diffusion models. Our fine-tuning can be done in a few hours on a consumer GPU and we can use it to learn distributions with a few corrupted samples.
  - We show that models trained on sufficiently corrupted data do not memorize their training set. For reasonably sized training datasets they can perform almost as well as diffusions trained on clean data.

2 Background

Training a diffusion model involves two steps. First, we design a corruption process that transforms the data distribution gradually into a distribution that we can sample from. Typically, this corruption process is described by an Ito SDE of the form: $\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}$, where $\boldsymbol{w}$ is the standard Wiener process. Such corruption processes are *reversible* and the reverse process is also described by an Ito SDE: $\mathrm{d}\boldsymbol{x} = \left(\boldsymbol{f}(\boldsymbol{x}, t) - g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}$. The designer of the diffusion model is usually free to choose the drift function $\boldsymbol{f}(\cdot, \cdot)$ and the diffusion function $g(\cdot)$. Typical choices are setting $\boldsymbol{f}(\boldsymbol{x}, t) = \boldsymbol{0}, g(t) = \sqrt{\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}}$ (Variance Exploding SDE) or setting $\boldsymbol{f}(x, t) = -\beta(t)\boldsymbol{x}, g(t) = \sqrt{\beta(t)}$ (Variance Preserving SDE). Both of these choices lead to a Gaussian terminal distribution and are equivalent up to a linear transformation in the input. The goal of diffusion model training is to learn the function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$, which is known as the score function. To simplify the presentation of the paper, we will focus on the Variance Exploding SDE that leads to conditional distributions $\boldsymbol{x}_t = \boldsymbol{x}_0 + \sigma_t \boldsymbol{\eta}$.

3

Vincent [52] showed that we can learn the score function at level t by optimizing for the score-matching objective:

$$J(\theta) = \frac{1}{2} \mathbb{E}_{(\boldsymbol{x}_0, \boldsymbol{x}_t)} \left|\left| \boldsymbol{h}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right|\right|^2 . \tag{2.1}$$

Specifically, the score function can be written in terms of the minimizer of this objective as:

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) = \frac{h_{\theta^*}(\boldsymbol{x}_t, t) - \boldsymbol{x}_t}{\sigma_t}. \tag{2.2}$$

This result reveals a fundamental connection between the score-function and the best restoration model of $\boldsymbol{x}_0$ given $\boldsymbol{x}_t$, known as Tweedie's Formula [14]. Specifically, it holds that $\boldsymbol{h}_{\theta^*}(\boldsymbol{x}_t, t) = \mathbb{E}[\boldsymbol{x}_0 | \boldsymbol{x}_t]$.

Inspired by this restoration interpretation of diffusion models, the Soft/Cold Diffusion works [12, 3] generalized diffusion models to look at non-Markovian corruption processes: $\boldsymbol{x}_t = C_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\eta}$. Specifically, Soft Diffusion proposes the Soft Score Matching objective:

$$J_{\text{soft}}(\theta) = \frac{1}{2} \mathbb{E}_{(\boldsymbol{x}_0, \boldsymbol{x}_t)} \left|\left| C_t \left(\boldsymbol{h}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right) \right|\right|^2 , \tag{2.3}$$

and shows that it is sufficient to recover the score function via a generalized Tweedie's Formula:

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) = \frac{C_t \overbrace{\mathbb{E}[\boldsymbol{x}_0 | \boldsymbol{x}_t]}^{\text{best restoration}} - \boldsymbol{x}_t}{\sigma_t}. \tag{2.4}$$

For these generalized models, the matrix C_t is a design choice (similar to how we could choose the functions $\boldsymbol{f}, g$). Most importantly, for $t = 0$, the matrix C_t becomes the identity matrix and the noise σ_t gets to zero, i.e. we observe samples from the true distribution.

3 Method

As explained in the introduction, in many cases we do not observe uncorrupted images $\boldsymbol{x}_0$, either by design (to avoid memorization and leaking of sensitive data) or because it is impossible to obtain clean data. Here we study the case where a learner only has access to linear measurements of the clean data, i.e. $\boldsymbol{y}_0 = A\boldsymbol{x}_0$, and the corruption matrices $A : \mathbb{R}^{m \times n}$. We note that we are interested in non-invertible corruption matrices. We ask two questions:

1. Is it possible to learn the best restoration model, i.e. $\mathbb{E}[\boldsymbol{x}_0 | A(\boldsymbol{x}_0 + \sigma_t \boldsymbol{\eta}), A]$, given only access to corrupted samples $(\boldsymbol{y}_0 = A\boldsymbol{x}_0, A)$?

2. If so, is it possible to use this restoration model $\mathbb{E}[\boldsymbol{x}_0 | A(\boldsymbol{x}_0 + \sigma_t \boldsymbol{\eta}), A]$ to recover $\mathbb{E}[\boldsymbol{x}_0 | \boldsymbol{x}_t]$ and thus sample from the true distribution through Tweedie's formula?

We investigate these questions in the rest of the paper. For the first, the answer is affirmative but only after introducing additional corruptions as we explain below. For the second, we approximate $\mathbb{E}[\boldsymbol{x}_0 | \boldsymbol{x}_t]$ given the predictions of $\mathbb{E}[\boldsymbol{x}_0 | A\boldsymbol{x}_t, A]$.

3.1 Training

For the sake of clarity, we first consider the case of random inpainting. If the image $\boldsymbol{x}_0$ is viewed as a vector, we can think of the matrix A as a diagonal matrix with ones in the entries that correspond to the preserved pixels and zeros in the erased pixels. We assume that $p(A)$ samples a matrix where each entry in the diagonal is sampled i.i.d. with a probability $1 - p$ to be 1 and p to be zero.

We would like to train a function h_θ which receives a corruption matrix A and a noisy version of a corrupted image, $A(\boldsymbol{x}_0 + \sigma_t \boldsymbol{\eta})$ where $\boldsymbol{\eta} \sim \mathcal{N}(0, I)$, and produces an estimate for the conditional expectation. The simplest idea would be to simply ignore the missing pixels and optimize for:

$$J_{\text{naive}}^{\text{corr}}(\theta) = \frac{1}{2} \mathbb{E}_{(\boldsymbol{x}_0, \boldsymbol{x}_t, A)} \left|\left| A \left(\boldsymbol{h}_\theta(A, A\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right) \right|\right|^2 , \tag{3.1}$$

Despite the similarities with Soft Score Matching (Eq 2.3), this objective will not learn the conditional expectation. The reason is that the learner is never penalized for performing arbitrarily poorly in

4

the missing pixels. Formally, any function h' satisfying $Ah'(A, \boldsymbol{y}_t, t) = A\mathbb{E}[\boldsymbol{x}_0|A\boldsymbol{x}_t = \boldsymbol{y}_t]$ is a minimizer.

Instead, we propose to *further corrupt* the samples before feeding them to the model, and ask the model to predict the original corrupted sample from the further corruptied image.

Concretely, we randomly corrupt A to obtain $\tilde{A} = BA$ for some matrix B that is selected randomly given A. In our example of missing pixels, $\tilde{A}$ is obtained from A by randomly erasing an additional fraction δ of the pixels that survive after the corruption A. Here, B will be diagonal where each element is 1 with probability $1 - \delta$ and 0 w.p. δ. We will penalize the model on recovering all the pixels that are visible in the sample Ax_0: this includes both the pixels that survive in $\tilde{A}x_0$ and those that are erased by $\tilde{A}$. The formal training objective is given by minimizing the following loss:

$$J^{\text{corr}}(\theta) = \frac{1}{2}\mathbb{E}_{(\boldsymbol{x}_0, \boldsymbol{x}_t, A, \tilde{A})} \left\|A\left(\boldsymbol{h}_\theta(\tilde{A}, \tilde{A}\boldsymbol{x}_t, t) - \boldsymbol{x}_0\right)\right\|^2, \qquad (3.2)$$

The key idea behind our algorithm is as follows: the learner does not know if a missing pixel is missing because we never had it (and hence do not know the ground truth) or because it was deliberately erased as part of the further corruption (in which case we do know the ground truth). Thus, the learner cannot produce bad results in the unobserved pixels because with non-zero probability it might be evaluated on some of them. Notice that the trained model behaves as a denoiser in the observed pixels and as an inpainter in the missing pixels. We also want to emphasize that the probability δ of further corruption can be arbitrarily small as long as it stays positive.

The idea of further corruption can be generalized from the case of random inpainting to a much broader family of corruption processes. For example, if A is a random Gaussian matrix with m rows, we can form $\tilde{A}$ by deleting one row from A at random. If A is a block inpainting matrix (i.e. a random block of fixed size is missing from all of the training images), we can create $\tilde{A}$ by corrupting further with one more non-overlapping missing block. Examples of our further corruption are shown in Figure 2. In our Theory Section, we prove conditions under which it is possible to recover $\mathbb{E}[\boldsymbol{x}_0|\tilde{A}\boldsymbol{x}_t, \tilde{A}]$ using our algorithm and samples $(\boldsymbol{y}_0 = A\boldsymbol{x}_0, A)$. Our goal is to satisfy this condition while adding minimal further corruption, i.e. while keeping $\tilde{A}$ close to A.

3.2 Sampling

Fixed mask sampling. To sample from $p_0(\boldsymbol{x}_0)$ using the standard diffusion formulation, we need access to $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$, which is equivalent to having access to $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ (see Eq. 2.2). Instead, our model is trained to predict $\mathbb{E}[\boldsymbol{x}_0|\tilde{A}\boldsymbol{x}_t, \tilde{A}]$ for all matrices A in the support of $p(A)$.

We note that for random inpainting, the identity matrix is technically in the support of $p(A)$. However, if the corruption probability p is at least a constant, the probability of seeing the identity matrix is exponentially small in the dimension of x. Hence, we should not expect our model to give good estimates of $\mathbb{E}[\boldsymbol{x}_0|\tilde{A}\boldsymbol{x}_t, \tilde{A}]$ for corruption matrices A that belong to the tails of the distribution $p(A)$.

The simplest idea is to sample a mask $\tilde{A} \sim p(\tilde{A})$ and approximate $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ with $\mathbb{E}[\boldsymbol{x}_0|\tilde{A}\boldsymbol{x}_t, \tilde{A}]$. Under this approximation, the discretized sampling rule becomes:

$$\boldsymbol{x}_{t-\Delta t} = \underbrace{\frac{\sigma_{t-\Delta t}}{\sigma_t}}_{\gamma_t} \boldsymbol{x}_t + \underbrace{\frac{\sigma_t - \sigma_{t-\Delta t}}{\sigma_t}}_{1-\gamma_t} \underbrace{\mathbb{E}[\boldsymbol{x}_0|\tilde{A}\boldsymbol{x}_t, \tilde{A}]}_{\hat{x}_0}. \qquad (3.3)$$

This idea works surprisingly well. Unless mentioned otherwise, we use it for all the experiments in the main paper and we show that we can generate samples that are reasonably close to the true distribution (as shown by metrics such as FID and Inception) even with 90% of the pixels missing.

Sampling with Reconstruction Guidance. In the Fixed Mask Sampler, at any point the prediction is a convex combination of the current value and the predicted denoised image. As $t \to 0, \gamma_t \to 0$. Hence, for the masked pixels, the fixed mask sampler outputs the conditional expectation of their value given the observed pixels. This leads to averaging effects as the corruption gets higher. To correct for this problem, we add one more term in the update: the Reconstruction Guidance term. The

5

issue with the previous sampler is that the model never sees certain pixels. We would like to evaluate the model using different masks. However, the model outputs for the denoised image might be very different when evaluated with different masks. To account for this problem, we add an additional term that enforces updates that lead to consistency on the reconstructed image. The update of the sampler with Reconstruction Guidance becomes:

$$\boldsymbol{x}_{t-\Delta t} = \underbrace{\frac{\sigma_{t-\Delta t}}{\sigma_t}}_{\gamma_t} x_t + \underbrace{\frac{\sigma_t - \sigma_{t-\Delta t}}{\sigma_t}}_{1-\gamma_t} \mathbb{E}[\boldsymbol{x}_0|\tilde{A}\boldsymbol{x}_t, \tilde{A}] - \frac{w_t}{2} \nabla_{\boldsymbol{x}_t} \mathbb{E}_{A' \sim p(\tilde{A}')} \left[||\mathbb{E}[\boldsymbol{x}_0|\tilde{A}\boldsymbol{x}_t, \tilde{A}] - \mathbb{E}[\boldsymbol{x}_0|\tilde{A}'\boldsymbol{x}_t, \tilde{A}']||^2 \right].$$

(3.4)

This sampler is inspired by the Reconstruction Guidance term used in Imagen [44]. We see modest improvements over the Fixed Mask Sampler for certain corruption ranges. We ablate this sampler in the Appendix.

We also prove that in theory, whenever it is possible to reconstruct $p_0(\boldsymbol{x}_0)$ from corrupted samples, it is also possible to reconstruct it using access to $\mathbb{E}[\boldsymbol{x}_0|A\boldsymbol{x}_t, A]$. However, as stated in the Limitations section, we were not able to find any practical algorithm to do so.

4 Theory

As elaborated in Section 3, one of our key goals is to learn the best restoration model, i.e., the function $h(\boldsymbol{y}_t, A, t) = \mathbb{E}[\boldsymbol{x}_0|A\boldsymbol{x}_t = \boldsymbol{y}_t, A]$. We now show that under a certain assumption on the distribution of A and $\tilde{A}$, the true population minimizer of Eq. 3.2 is indeed essentially of the form above. This assumption formalizes the notion that even conditional on $\tilde{A}$, A has considerable variability, and the latter ensures that the best way to predict $A\boldsymbol{x}_0$ as a function of $\tilde{A}\boldsymbol{x}_t$ and $\tilde{A}$ is to optimally predict $\boldsymbol{x}_0$ itself. All proofs are deferred to the Appendix.

Theorem 4.1. *Assume a joint distribution of corruption matrices A and further corruption $\tilde{A}$. If for all $\tilde{A}$ in the support it holds that $\mathbb{E}_{A|\tilde{A}}[A^T A]$ is full-rank, then the unique minimizer of the objective in equation 3.2 is given by*

$$h_{\theta^*}(\boldsymbol{y}_t, \tilde{A}, t) = \mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t = \boldsymbol{y}_t, \tilde{A}] \tag{4.1}$$

Two simple examples that fit into this framework (see Corollaries 6.1 and 6.2 in the Appendix) are:

- Inpainting: $A \in \mathbb{R}^{n \times n}$ is a diagonal matrix where each entry $A_{ii} \sim \text{Ber}(1 - p)$ for some $p > 0$ (independently for each i), and the additional noise is generated by drawing $\tilde{A}|A$ such that $\tilde{A}_{ii} = A_{ii} \cdot \text{Ber}(1 - \delta)$ for some small $\delta > 0$ (again independently for each i).[1]
- Gaussian measurements: $A \in \mathbb{R}^{m \times n}$ consists of m rows drawn independently from $\mathcal{N}(0, I_n)$, and $\tilde{A} \in \mathbb{R}^{m \times n}$ is constructed conditional on A by zeroing out its last row.

Notice that the minimizer in Eq 4.1 is not entirely of the form we originally desired, which was $h(\boldsymbol{y}_t, A, t) = \mathbb{E}[\boldsymbol{x}_0 \mid A\boldsymbol{x}_t = \boldsymbol{y}_t, A]$. In place of A, we now have $\tilde{A}$, which is a further degraded matrix. Indeed, one trivial way to satisfy the condition in Theorem 4.1 is by forming $\tilde{A}$ completely independently of A, e.g. by always setting $\tilde{A} = 0$. However, in this case the function we learn is not very useful. For this reason we would like to add as little further noise as possible, and ensure that $\tilde{A}$ is close to A. In natural noise models such as the inpainting noise model, by letting the additional corruption probability δ approach 0, we can indeed ensure that $\tilde{A}$ follows a distribution very close to that of A.

5 Experimental Evaluation

Training from scratch on corrupted data. Our first experiment is to train diffusion models from scratch using corrupted training data at different levels of corruptions. The corruption model we use for these experiments is random inpainting: we form our dataset by deleting each pixel with

---

[1] $\text{Ber}(q)$ indicates a Bernoulli random variable with a probability of q to equal 1 and $1 - q$ for 0.

6

probability p. To create the matrix $\tilde{A}$, we further delete each row of A with probability δ – this removes an additional δ-fraction of the surviving pixels. Unless mentioned otherwise, we use $\delta = 0.1$. We ablate the choice of δ in the Appendix. We train models on CIFAR-10, AFHQ, and CelebA-HQ. All our models are trained with corruption level $p \in \{0.0, 0.2, 0.4, 0.6, 0.8, 0.9\}$. We use the EDM [26] codebase to train our models. The only thing that we change is that we replace convolutions with Gated Convolutions [55] which are known to perform better for inpainting-type problems. The full training details can be found in the Appendix.

We first evaluate the restoration performance of our model for the task it was trained on (random inpainting and noise). We compare with state-of-the-art diffusion models that were trained on clean data. Specifically, for AFHQ we compare with the state-of-the-art EDM model [26] and for CelebA we compare with DDIM [47]. These models were not trained to denoise, but we can use the prior learned in the denoiser as in [51, 25] to solve any inverse problem. We experiment with the state-of-the-art reconstruction algorithms: DDRM [30] and DPS [10].

Our model performs similarly to other diffusion models, even though it has never been trained on clean data. Further, it does so by requiring only one step, while all the baseline diffusion models require hundreds of steps to solve the same task with inferior or comparable performance. We summarize the results in Table 1. We include results with noisy inpainted measurements and comparisons with supervised methods in the Appendix. We want to emphasize that all the baselines we compare against have an advantage: they are trained on *uncorrupted* data. Instead, our models were only trained on corrupted data. This experiment indicates that: i) our training algorithm worked and ii) that the choice of corruption that diffusion models are trained to reverse matters for solving inverse problems.

Next, we evaluate the performance of our diffusion models as generative models. To the best of our knowledge, the only generative baseline with quantitative results for training on corrupted data is AmbientGAN [6] which is trained on CIFAR-10. We further compare with a diffusion model trained without our further corruption algorithm. We plot the results in Figure 5. The diffusion model trained without our further corruption algorithm performs well for low corruption levels, but collapses entirely for high corruption. Instead, our model trained with the further corruption maintains reasonable corruption scores even for high corruption levels, outperforming the previous state-of-the-art AmbientGAN for all ranges of corruption levels.

For CelebA-HQ and AFHQ we could not find any generative baselines trained on corrupted data to compare against. Nevertheless, we report FID and Inception Scores and summarize our results in Table 6 to encourage further research in this area. As shown in the Table, for CelebA-HQ and AFHQ, we manage to maintain a decent FID score even with 90% of the pixels deleted. For CIFAR-10, the performance degrades faster, potentially because of the lower resolution of the training images.

Finetuning foundation models on corrupted data. We can apply our technique to finetune a

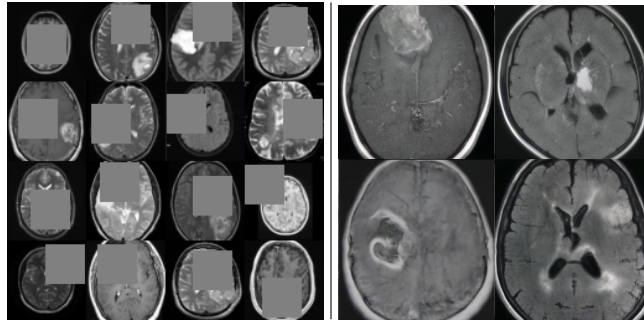



Figure 3: **Left panel:** We finetune the Deepfloyd's IF diffusion model to make it a generative model for MRI images of brains with tumors. We use a small dataset [22] of only 155 images that was corrupted by removing large blocks as shown. **Right panel:** Generated samples from our finetuned model. As shown, the model learns the statistics of full brain tumor MRI images. The training set was resized to 64×64 but the generated images are at 256×256. The higher resolution is obtained by simply leveraging the power of the cascaded IF model.

foundational diffusion model. For all our experiments, we use Deepfloyd's IF model [2], which is one of the most powerful open-source diffusion generative models available. We choose this model over Stable Diffusion [42] because it works in the pixel-space (and hence our algorithm directly

7

Dataset	Corruption Probability	Method	LPIPS	PSNR	NFE
CelebA-HQ		Ours	**0.0373**	**31.51**	1
		DPS	0.0533	28.21	100
	0.6		0.1394	25.76	35
		DDRM	0.0887	27.38	99
			0.0696	28.16	199
		Ours	**0.0849**	**26.80**	1
		DPS	0.1078	24.16	100
	0.8		0.3168	20.37	35
		DDRM	0.1883	22.96	99
			0.1537	23.82	199
		Ours	**0.1528**	**23.34**	1
		DPS	0.1684	20.89	100
	0.9		0.4618	15.87	35
		DDRM	0.3329	18.74	99
			0.2428	20.14	199
AFHQ		Ours	0.0304	33.27	1
		DPS	**0.0203**	**34.06**	100
	0.4		0.1226	25.18	35
		DDRM	0.0912	26.42	99
			0.0889	26.52	199
		Ours	0.0628	29.46	1
		DPS	**0.0518**	**30.03**	100
	0.6		0.2464	20.76	35
		DDRM	0.1665	22.79	99
			0.1600	22.93	199
		Ours	0.1245	**25.37**	1
		DPS	**0.1078**	25.30	100
	0.8		0.5250	14.56	35
		DDRM	0.2957	18.08	99
			0.2583	18.86	199

Table 1: Comparison of our model (trained on corrupted data) with state-of-the-art diffusion models on CelebA (DDIM [47] model) and AFHQ (EDM [26] model) for solving the random inpainting inverse problem. Our model performs on par to state-of-the art diffusion inverse problem solvers, even though it has never seen uncorrupted training data. Further, this is achieved with a single score function evaluation. To solve this problem with a standard pre-trained diffusion model we need to use a reconstruction algorithm (such as DPS [10] or DDRM [30]) that typically requires hundreds of steps. Please also note that the performance of DDRM improves with more function evaluations at the cost of more computation.

applies). **Memorization.** We show that we can finetune a foundational model on a limited dataset without memorizing the training examples. This experiment is motivated by the recent works of Carlini et al. [7], Somepalli et al. [46], and Jagielski et al. [23] that show that diffusion generative models memorize training samples and they do it significantly more than previous generative models, such as GANs, especially when the training dataset is small. Specifically, Somepalli et al. [46] train diffusion models on subsets of size $\{300, 3000, 30000\}$ of CelebA and they show that models trained on 300 or 3000 memorize and blatantly copy images from their training set.

We replicate this training experiment by finetuning the IF model on CelebA on same-size subsets of CelebA. Results are shown in Figure 1. Standard finetuning of Deepfloyd's IF on 3000 images memorizes samples and produces almost exact copies of the training set. Instead, if we corrupt the images by deleting 80% of the pixels prior to training and finetune, the memorization decreases sharply and there are distinct differences between the generated images and their nearest neighbors from the dataset. This is in spite of finetuning until convergence.

Following Somepalli et al. [46], we generate 10000 images from each model and we use DINO [8]-v2 [38] to compute top-1 similarity to the training images. Results are shown in Figure 4. Similarity values above 0.95 roughly correspond to the same person while similarities below 0.75 typically correspond to random faces. The standard finetuning (Red) often generates images that are near-

8

Distribution of Similarity Values to the nearest training sample

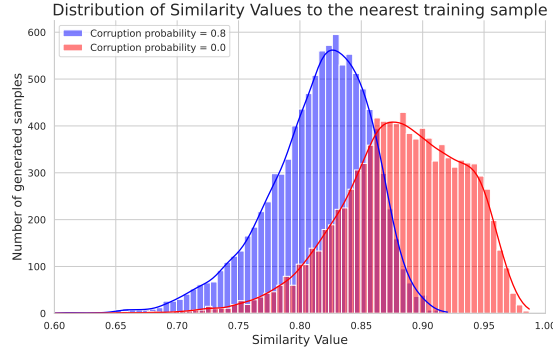



Figure 4: Distribution of similarity values to the nearest neighbor in the dataset for a finetuned IF model on a 3000 samples CelebA subset. Please note that similarity values above 0.95 roughly correspond to the same person. Therefore the baseline finetuning process (red) often generates images that are near-copies of the training set. On the contrary, our fine-tuning with corrupted samples (blue) shows a clear shift to the left. Visually we never observed a near-identical image generated from our process, see also Figure 1 for qualitative results.

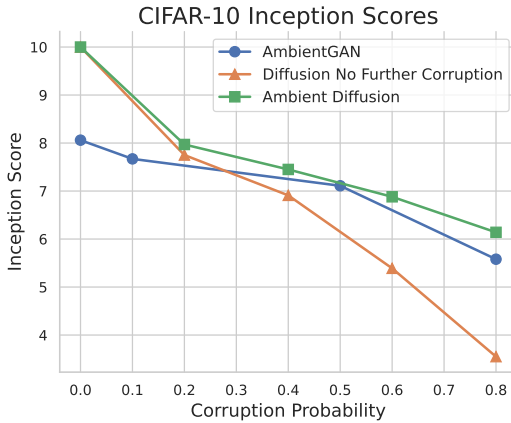



Figure 5: Performance on CIFAR-10 as a function of the corruption level. We compare our method with a diffusion model trained without our further corruption trick and AmbientGAN [6]. Ambient Diffusion outperforms both baselines for all ranges of corruption levels.

Dataset	Corruption Probability	FID	Inception Score
CelebA-HQ	0.0	3.26	
	0.2	4.18	
	0.6	6.08	N/A
	0.8	11.19	
	0.9	25.53	
AFHQ	0.0	2.41	
	0.2	4.47	
	0.4	6.96	
	0.6	10.11	N/A
	0.8	16.78	
	0.9	41.00	
CIFAR-10	0.0	1.85	9.94
	0.2	11.70	7.97
	0.4	18.85	7.45
	0.6	28.88	6.88
	0.8	46.27	6.14

Figure 6: Inception/FID results on random inpainting for models trained with our algorithm on CelebA-HQ, AFHQ and CIFAR-10.

identical with the training set. Instead, fine-tuning with corrupted samples (blue) shows a clear shift to the left. Visually we never observed a near-copy generated from our process – see also Figure 1.

New domains and different corruption. We show that we can finetune a pre-trained foundation model on a limited sized dataset on a new domain in a few hours in a consumer's GPU. Figure 3 shows generated samples from our finetuned model. As shown, the model learns the statistics of full brain tumor MRI images while only trained on brain-tumor images that have a random box obfuscating 25% of the image. The training set was resized to 64×64 but the generated images are at 256×256 by simply leveraging the power of the cascaded Deepfloyd IF.

Limitations. Our work has several limitations. First, there is a tradeoff between generator quality and corruption levels. For higher corruption, it is less likely that our generator memorizes parts of training examples, but at a cost of degrading quality (see related Ablation Study in the Appendix). Precisely characterizing this trade-off is an open-research problem. Further, in this work we only experimented with very simple approximation algorithms to estimate $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ using our trained models. Additionally, we cannot make any strict privacy claim about the protection of any training sample without making assumptions about the data distribution. We show in the Appendix that it is possible to recover $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ exactly using our restoration oracle, but we do not have an algorithm

9

to do so. Finally, our method cannot handle measurements that also have noise. Future work could potentially address this limitation by exploiting SURE regularization as in [1].

Acknowledgments and Disclosure of Funding

The authors would like to thank Tom Goldstein for insightful discussions that benefited this work. This research has been supported by NSF Grants CCF 1763702, AF 1901292, CNS 2148141, Tripods CCF 1934932, NSF AI Institute for Foundations of Machine Learning (IFML) 2019844, the Texas Advanced Computing Center (TACC) and research gifts by Western Digital, WNCG IAP, UT Austin Machine Learning Lab (MLL), Cisco and the Archie Straiton Endowed Faculty Fellowship. Giannis Daras has been supported by the Onassis Fellowship (Scholarship ID: F ZS 012-1/2022-2023), the Bodossaki Fellowship and the Leventis Fellowship. Aravind Gollakota was at UT Austin while this work was done.

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) (page 10).

[2] Stability AI. *Deepfloyd IF*. https://github.com/deep-floyd/IF. 2013 (pages 7, 17).

[3] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. "Cold diffusion: Inverting arbitrary image transforms without noise". In: *arXiv preprint arXiv:2208.09392* (2022) (page 4).

[4] Joshua Batson and Loic Royer. "Noise2self: Blind denoising by self-supervision". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 524–533 (page 2).

[5] Felix JB Bäuerlein and Wolfgang Baumeister. "Towards visual proteomics at high resolution". In: *Journal of Molecular Biology* 433.20 (2021), p. 167187 (page 2).

[6] Ashish Bora, Eric Price, and Alexandros G Dimakis. "AmbientGAN: Generative models from lossy measurements". In: *International conference on learning representations*. 2018 (pages 2, 3, 7, 9).

[7] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. "Extracting Training Data from Large Language Models". In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021, pp. 2633–2650. ISBN: 978-1-939133-24-3. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting (visited on 11/10/2022) (pages 1, 2, 8).

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660 (page 8).

[9] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. "Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions". In: *arXiv preprint arXiv:2209.11215* (2022) (pages 15, 16).

[10] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. "Diffusion Posterior Sampling for General Noisy Inverse Problems". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=OnD9zGAGT0k (pages 1, 7, 8, 20).

[11] The Event Horizon Telescope Collaboration et al. "First M87 Event Horizon Telescope Results. IV. Imaging the Central Supermassive Black Hole". In: *The Astrophysical Journal Letters* 875.1 (Apr. 2019), p. L4. DOI: 10.3847/2041-8213/ab0e85. URL: https://dx.doi.org/10.3847/2041-8213/ab0e85 (page 1).

[12] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alex Dimakis, and Peyman Milanfar. "Soft Diffusion: Score Matching with General Corruptions". In: *Transactions on Machine Learning Research* (2023). ISSN: 2835-8856. URL: https://openreview.net/forum?id=W98rebBxlQ (page 4).

[13] Jeffrey M Ede. "Deep learning in electron microscopy". In: *Machine Learning: Science and Technology* 2.1 (2021), p. 011004 (page 2).

10

[14]   Bradley Efron. "Tweedie's formula and selection bias". In: *Journal of the American Statistical Association* 106.496 (2011), pp. 1602–1614 (page 4).

[15]   Yonina C. Eldar. "Generalized SURE for Exponential Families: Applications to Regularization". In: *IEEE Transactions on Signal Processing* 57.2 (2009), pp. 471–481. DOI: 10.1109/TSP.2008.2008212 (page 2).

[16]   Angela F Gao, Oscar Leong, He Sun, and Katherine L Bouman. "Image Reconstruction without Explicit Priors". In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5 (page 1).

[17]   Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. "Toward convolutional blind denoising of real photographs". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 1712–1722 (page 2).

[18]   Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16000–16009 (pages 18, 19).

[19]   Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in neural information processing systems* 30 (2017) (page 17).

[20]   Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851 (pages 1, 17).

[21]   Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. "Cascaded Diffusion Models for High Fidelity Image Generation." In: *J. Mach. Learn. Res.* 23.47 (2022), pp. 1–33 (page 17).

[22]   "Huggingface Brain Tumor MRI Dataset". In: (2020). URL: https://huggingface.co/datasets/miladfa7/Brain-MRI-Images-for-Brain-Tumor-Detection/ (page 7).

[23]   Matthew Jagielski, Om Thakkar, Florian Tramèr, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Chiyuan Zhang. "Measuring Forgetting of Memorized Training Examples". In: *arxiv:2207.00099[cs]* (June 2022). DOI: 10.48550/arXiv.2207.00099. arXiv: 2207.00099 [cs]. URL: http://arxiv.org/abs/2207.00099 (visited on 11/10/2022) (pages 1, 2, 8).

[24]   Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. "Robust compressed sensing mri with deep generative priors". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 14938–14954 (page 1).

[25]   Zahra Kadkhodaie and Eero P Simoncelli. "Solving linear inverse problems using the prior implicit in a denoiser". In: *arXiv preprint arXiv:2007.13640* (2020) (page 7).

[26]   Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. "Elucidating the design space of diffusion-based generative models". In: *arXiv preprint arXiv:2206.00364* (2022) (pages 1, 7, 8, 16–18).

[27]   Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Alias-free generative adversarial networks". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 852–863 (page 16).

[28]   Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410 (page 16).

[29]   Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8110–8119 (page 16).

[30]   Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. "Denoising Diffusion Restoration Models". In: *Advances in Neural Information Processing Systems* (pages 1, 7, 8, 20).

[31]   Bahjat Kawar, Gregory Vaksman, and Michael Elad. "SNIPS: Solving noisy inverse problems stochastically". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 21757–21769 (page 1).

[32]   Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. "Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 11201–11228 (page 16).

11

[33] 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 2).

[34] 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 2).

[35] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. "Coherent Semantic Attention for Image Inpainting". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2019). DOI: 10.1109/iccv.2019.00427. URL: http://dx.doi.org/10.1109/ICCV.2019.00427 (page 2).

[36] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. "Pulse: Self-supervised photo upsampling via latent space exploration of generative models". In: *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2020, pp. 2437–2445 (page 16).

[37] Alex Nichol, Aditya Ramesh, Pamela Mishkin, Prafulla Dariwal, Joanne Jang, and Mark Chen. *DALL·E 2 Pre-Training Mitigations*. June 2022. URL: https://openai.com/blog/dall-e-2-pre-training-mitigations/ (visited on 11/10/2022) (page 1).

[38] 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 8).

[39] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. "Context encoders: Feature learning by inpainting". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2536–2544 (page 2).

[40] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. "Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation". In: *arXiv preprint arXiv:2008.00951* (2020) (page 2).

[41] 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).

[42] 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 7).

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. "Photorealistic text-to-image diffusion models with deep language understanding". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36479–36494 (page 1).

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. "Photorealistic text-to-image diffusion models with deep language understanding". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36479–36494 (page 6).

[45] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265 (page 1).

[46] 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, 2, 8).

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models". In: *arXiv preprint arXiv:2010.02502* (2020) (pages 7, 8, 18).

[48] 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).

[49] 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, 17).

12

[50] Julián Tachella, Dongdong Chen, and Mike Davies. "Unsupervised Learning From Incomplete Measurements for Inverse Problems". In: *arXiv preprint arXiv:2201.12151* (2022) (page 2).

[51] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. "Plug-and-play priors for model based reconstruction". In: *2013 IEEE Global Conference on Signal and Information Processing*. IEEE. 2013, pp. 945–948 (page 7).

[52] Pascal Vincent. "A connection between score matching and denoising autoencoders". In: *Neural computation* 23.7 (2011), pp. 1661–1674 (page 4).

[53] Ge Wang, Jong Chul Ye, and Bruno De Man. "Deep learning for tomographic image reconstruction". In: *Nature Machine Intelligence* 2.12 (2020), pp. 737–748 (page 2).

[54] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. "Free-Form Image Inpainting With Gated Convolution". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2019). DOI: 10.1109/iccv.2019.00457. URL: http://dx.doi.org/10.1109/ICCV.2019.00457 (page 2).

[55] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. "Free-form image inpainting with gated convolution". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 4471–4480 (pages 7, 16).

[56] Yide Zhang, Yinhao Zhu, Evan Nichols, Qingfei Wang, Siyuan Zhang, Cody Smith, and Scott Howard. "A poisson-gaussian denoising dataset with real fluorescence microscopy images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11710–11718 (page 2).

13

A Proofs

Proof of Theorem 4.1. Let $\boldsymbol{h}_{\theta^*}$ be a minimizer of equation 3.2, and for brevity let

$$\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A}) = \boldsymbol{h}_{\theta^*}(\tilde{A}\boldsymbol{x}_t, \tilde{A}) - \mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}]$$

be the difference between $\boldsymbol{h}_{\theta^*}$ and the claimed optimal solution. We will now argue that $\boldsymbol{f}$ must be identically zero.

First, by adding and subtracting $A\mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}]$, we can expand the objective value achieved at $\theta = \theta^*$ in equation 3.2 as follows:

$$J^{\text{corr}}(\theta^*) = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, A, \tilde{A}}\left[\left\|(A\boldsymbol{x}_0 - A\mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}]) - A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A}))\right\|^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, A, \tilde{A}}\left[\left\|A\boldsymbol{x}_0 - A\mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}]\right\|^2\right] + \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, A, \tilde{A}}\left[\left\|A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right\|^2\right]$$

$$- 2\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, A, \tilde{A}}\left[(A\boldsymbol{x}_0 - A\mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}])^T A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right].$$

Here the first term is the irreducible error, while the third term vanishes by the tower law of expectations:

$$\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, A, \tilde{A}}\left[(A\boldsymbol{x}_0 - A\mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}])^T A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right]$$

$$= \mathbb{E}_{\boldsymbol{x}_0, A, \tilde{A}, \tilde{A}\boldsymbol{x}_t}\left[(\boldsymbol{x}_0 - \mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}])^T A^T A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right]$$

$$= \mathbb{E}_{A, \tilde{A}, \tilde{A}\boldsymbol{x}_t}\left[\mathbb{E}_{\boldsymbol{x}_0 \mid A, \tilde{A}, \tilde{A}\boldsymbol{x}_t}\left[\boldsymbol{x}_0 - \mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}]\right]^T A^T A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right]$$

$$= \mathbb{E}_{A, \tilde{A}, \tilde{A}\boldsymbol{x}_t}\left[(\mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}] - \mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}])^T A^T A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right]$$

$$= 0.$$

Thus the only part of $J^{\text{corr}}(\theta^*)$ that actually depends on the parameter value θ^* is the second term. We now show that the second term can be made to vanish, and that this occurs precisely when $\boldsymbol{f}$ is identically 0:

$$\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, A, \tilde{A}}\left[\left\|A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right\|^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, A, \tilde{A}}\left[\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})^T A^T A\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right]$$

$$= \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, \tilde{A}}\left[\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})^T \mathbb{E}_{A \mid \boldsymbol{x}_0, \boldsymbol{x}_t, \tilde{A}}\left[A^T A\right] f(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right]$$

$$= \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t, \tilde{A}}\left[f(\tilde{A}\boldsymbol{x}_t, \tilde{A})^T \mathbb{E}_{A \mid \tilde{A}}\left[A^T A\right] \boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A})\right].$$

For every $\tilde{A}$ and $\boldsymbol{x}_0, \boldsymbol{x}_t$, by assumption we have that $\mathbb{E}_{A \mid \tilde{A}}\left[A^T A\right]$ is full-rank, and so the inner quadratic form is minimized when $\boldsymbol{f}(\tilde{A}\boldsymbol{x}_t, \tilde{A}) = 0$. Further, the term as a whole vanishes exactly when this holds for every $\tilde{A}$ and $\boldsymbol{x}_0, \boldsymbol{x}_t$ in the support, which means $\boldsymbol{f}$ must be identically zero. $\square$

Corollary A.1 (Inpainting noise model). *Consider the following inpainting noise model: $A \in \mathbb{R}^{n \times n}$ is a diagonal matrix where each entry $A_{ii} \sim \text{Ber}(1 - p)$ for some $p > 0$ (independently for each i), and the additional noise is generated by drawing $\tilde{A} \mid A$ such that $\tilde{A}_{ii} = A_{ii}\text{Ber}(1 - \delta)$ for some small $\delta > 0$ (again independently for each i). Then the unique minimizer of the objective in equation 3.2 is*

$$\boldsymbol{h}_{\theta^*}(\tilde{A}\boldsymbol{x}_t, \tilde{A}) = \mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}].$$

Proof. By Theorem 4.1, what we must show is that for any $\tilde{A}$ in the support, $E_{A \mid \tilde{A}}[A^T A]$ is full-rank. Fix any particular realization of $\tilde{A}$, which will be a diagonal matrix with only 0s and 1s. For indices i where $\tilde{A}_{ii} = 1$, we know that for any A drawn conditional on $\tilde{A}$, $A_{ii} = 1$ as well, i.e.

$$\Pr(A_{ii} = 1 \mid \tilde{A}_{ii} = 1) = 1.$$

14

And for indices i where $\tilde{A}_{ii} = 0$, by Bayes' rule we have

$$\Pr(A_{ii} = 1 \mid \tilde{A}_{ii} = 0) = \frac{\Pr(A_{ii} = 1, \tilde{A}_{ii} = 0)}{\Pr(A_{ii} = 0, \tilde{A}_{ii} = 0) + \Pr(A_{ii} = 1, \tilde{A}_{ii} = 0)} = \frac{(1-p)\delta}{(1-p)\delta + p} =: q.$$

Thus we see that $E_{A|\tilde{A}}[A^T A] = E_{A|\tilde{A}}[A]$ is a diagonal matrix whose entries are 1 wherever $\tilde{A}_{ii} = 1$ and q wherever $\tilde{A}_{ii} = 0$. This is clearly of full rank since $q > 0$. $\qquad\square$

Corollary A.2 (Gaussian measurements noise model). *Consider the following noise model where we only observe m independent Gaussian measurements of the ground truth, and then one of the measurements is further omitted: $A \in \mathbb{R}^{m \times n}$ consists of m rows drawn independently from $\mathcal{N}(0, I_n)$, and $\tilde{A} \in \mathbb{R}^{m \times n}$ is constructed condtional on A by zeroing out its last row. Then the unique minimizer of the objective in equation 3.2 is*

$$\boldsymbol{h}_{\theta^*}(\tilde{A}\boldsymbol{x}_t, \tilde{A}) = \mathbb{E}[\boldsymbol{x}_0 \mid \tilde{A}\boldsymbol{x}_t, \tilde{A}].$$

Proof. Again by Theorem 4.1, we must show that for any $\tilde{A}$ in the support, $E_{A|\tilde{A}}[A^T A]$ is full-rank. Fix any realization of $\tilde{A}$, which will have the following form:

$$\tilde{A} = \begin{bmatrix} \underline{\quad} & \boldsymbol{a}_1^T & \underline{\quad} \\ & \vdots & \\ \underline{\quad} & \boldsymbol{a}_{m-1}^T & \underline{\quad} \\ \underline{\quad} & 0^T & \underline{\quad} \end{bmatrix}.$$

Then it is clear that conditional on $\tilde{A}$, A has the following distribution:

$$A \mid \tilde{A} = \begin{bmatrix} \underline{\quad} & \boldsymbol{a}_1^T & \underline{\quad} \\ & \vdots & \\ \underline{\quad} & \boldsymbol{a}_{m-1}^T & \underline{\quad} \\ \underline{\quad} & \boldsymbol{b}_m^T & \underline{\quad} \end{bmatrix} \quad \text{where} \quad \boldsymbol{b}_m \sim \mathcal{N}(0, I_n).$$

Here $\boldsymbol{b}_m$ is drawn entirely independently from $\mathcal{N}(0, I_n)$. Elementary manipulations now reveal that $\mathbb{E}_{A|\tilde{A}}[A^T A] = \mathbb{E}_{\boldsymbol{b}_m \sim \mathcal{N}(0, I_n)}[\tilde{A}^T \tilde{A} + \boldsymbol{b}_m \boldsymbol{b}_m^T] = \tilde{A}^T \tilde{A} + I_n$, which is clearly full rank (indeed, it is PSD with strictly positive eigenvalues). $\qquad\square$

A.1 Reduction

In this section we argue that if there is an algorithm that recovers the target distribution $p_0(\boldsymbol{x}_0)$ from i.i.d. samples $(A\boldsymbol{x}_0, A)$ where $A \sim p(A)$ and $\boldsymbol{x}_0 \sim p_0(\boldsymbol{x}_0)$, then, there is an algorithm that recovers $p_0(\boldsymbol{x}_0)$ without a sample access, but instead, using an access to an oracle that given t, x and A in the support of $p(A)$, returns $\mathbb{E}[\boldsymbol{x}_0 \mid A\boldsymbol{x}_t = x]$. Indeed, for any A, Chen et al. [9] show that it is possible to recover the distribution of $A\boldsymbol{x}_0$ given access to $\mathbb{E}[A\boldsymbol{x}_0 \mid A\boldsymbol{x}_t = x]$ for any t and x (this requires some minimal assumptions on the data distribution $p_0(\boldsymbol{x}_0)$ [9, Assumptions 1-3]). Using our oracle and using this theorem, we can recover the distribution of $A\boldsymbol{x}_0$ for all A in the support. By sampling from these distributions, one can as well obtain samples of $A\boldsymbol{x}_0$ for $A \sim p(A)$ and $\boldsymbol{x}_0 \sim p_0(\boldsymbol{x}_0)$. Hence, if these samples are sufficient to recover $p_0(\boldsymbol{x}_0)$, then, having an oracle to these conditional expectations is sufficient as well.

This intuition can be formalized as follows. Fix a distribution $p_A(A)$ over corruption matrices. For a distribution $p_0(\boldsymbol{x}_0)$, denote by $\mathrm{corrupt}(p_A, p_0)$ the distribution over pairs $(A, A\boldsymbol{x}_0)$ where $A \sim p_A(A)$ and $\boldsymbol{x}_0 \sim p_0(\boldsymbol{x}_0)$. We say that *it is possible to reconstruct $p_0(\boldsymbol{x}_0)$ from random corruptions $A \sim p_A(A)$* if the following holds: for any two distributions, $p_0(\boldsymbol{x}_0)$ and $p_0'(\boldsymbol{x}_0')$ that satisfy Assumptions 1-3 of Chen et al. [9], if $\mathrm{corrupt}(p_A, p_0) = \mathrm{corrupt}(p_A, p_0')$, then $p_0 = p_0'$. Similarly, we say that *it is possible to reconstruct $p_0(x_0)$ from conditional expectations given $A \sim p(A)$* if the following holds: for any distribution $p_0(\boldsymbol{x}_0)$ and $p_0'(\boldsymbol{x}_0')$ that satisfy Assumptions 1-3 of Chen et al. [9], if for all x, t and A in the support of p_A,

$$\mathbb{E}_{(\boldsymbol{x}_0, \boldsymbol{x}_t) \sim p_{0,t}(\boldsymbol{x}_0, \boldsymbol{x}_t)}[\boldsymbol{x}_0 \mid A\boldsymbol{x}_t = x] = \mathbb{E}_{(\boldsymbol{x}_0', \boldsymbol{x}_t') \sim p_{0,t}'(\boldsymbol{x}_0', \boldsymbol{x}_t')}[\boldsymbol{x}_0' \mid A\boldsymbol{x}_t' = x] \qquad \text{(A.1)}$$

then $p_0 = p_0'$. Here, $p_{0,t}(\boldsymbol{x}_0, \boldsymbol{x}_t)$ is obtained by sampling $\boldsymbol{x}_0 \sim p_0$ and $\boldsymbol{x}_t = \boldsymbol{x}_0 + \sigma_t \boldsymbol{\eta}$ where $\boldsymbol{\eta} \sim \mathcal{N}(0, I)$. Similarly, $p_{0,t}'(\boldsymbol{x}_0', \boldsymbol{x}_t')$ is obtained by the same process where $\boldsymbol{x}_0'$ is instead sampled from p_0'. We state the following lemma:

15

Lemma A.3. *Fix a distribution $p_A(A)$. If it is possible to reconstruct $p_0(\boldsymbol{x}_0)$ from random corruptions $\boldsymbol{y}_0 = A\boldsymbol{x}_0$, $A \sim p_A(A)$, then it is possible to reconstruct $p_0(\boldsymbol{x}_0)$ given access to an oracle that computes the conditional expectations $\mathbb{E}[\boldsymbol{x}_0 | A\boldsymbol{x}_t, A]$, for $A \sim p_A(A)$ and $\boldsymbol{x}_t = \boldsymbol{x}_0 + \sigma_t \boldsymbol{\eta}$.*

Proof. Fix a distribution $p_A(A)$ over corruption matrices. Assume that it is possible to reconstruct $p_0(\boldsymbol{x}_0)$ from random corruptions $A \sim p_A(A)$ and we will prove that it is possible to reconstruct $p_0(\boldsymbol{x}_0)$ from conditional expectations given $A \sim p_A(A)$. To do so, let $p_0(\boldsymbol{x}_0)$ and $p'_0(\boldsymbol{x}'_0)$ be two distributions that satisfy Assumptions 1-3 of Chen et al. [9]. Assume that equation A.1 holds. We will prove that $p_0 = p'_0$. Fix some A in the support of p_A. By Chen et al. [9], there is an algorithm that samples from the distribution of $A\boldsymbol{x}_0$, $\boldsymbol{x}_0 \sim p_0(\boldsymbol{x}_0)$, that only has access to $\mathbb{E}[A\boldsymbol{x}_0 \mid A\boldsymbol{x}_t, A]$ and similarly, there is an algorithm that samples from the distribution of $A\boldsymbol{x}'_0$ that only has access to $\mathbb{E}[A\boldsymbol{x}'_0 \mid A\boldsymbol{x}'_t, A]$. Since these two conditional expectations are assumed to be the same, then the distribution of $A\boldsymbol{x}_0$ equals the distribution of $A\boldsymbol{x}'_0$. Consequently, $\text{corrupt}(p_A, p_0) = \text{corrupt}(p_A, p'_0)$. By the assumption that it is possible to reconstruct $p_0(\boldsymbol{x}_0)$ from random corruptions $A \sim p_A(A)$, this implies that $p_0 = p'_0$. This completes the proof. $\square$

B   Broader Impact and Risks

Generative models in general hold the potential to have far-reaching impacts on society in a variety of forms, coupled with several associated risks [36, 28, 29, 27]. Among other potential applications, they can be utilized to create deceptive images and perpetuate societal biases. To the best of our knowledge, our paper does not amplify any of these existing risks. Regarding the included MRI results, we want to clarify that we make no claim that such results are diagnostically useful. This experiment serves only as a toy demonstration that it can be potentially feasible to learn the distribution of MRI scans with corrupted samples. Significant further research must be done in collaboration with radiologists before our algorithm gets tested in clinical trials. Finally, we want to underline again that even though our approach seems to mitigate the memorization issue in generative models, we cannot guarantee the privacy of any training sample unless we make assumptions about the data distribution. Hence, we strongly discourage using this algorithm in applications where privacy is important before this research topic is investigated further.

C   Training Details

We plan to open-source our code to facilitate further research in this area. We include an anonymized version of our code and our pre-trained models.

Note on maintaining anonymity. Our code is fork of the **public** repository https://github.com/NVlabs/edm which contains the official implementation of the published EDM [26] paper.

Any non-anonymous reference to this repository, the authors or the license of this repository **does not** imply any affiliation with the authors of EDM. We provide our checkpoints through anonymous (Zenodo) URLs.

Further, our repository has a `diffusers` folder. We use the Huggingface diffusers library for our finetuning experiments. Again, any non-anonymous reference to this repository, the authors or the license of this repository **does not** imply any affiliation with the authors of the library.

Models trained from scratch.   We trained models from scratch at different corruption levels on CelebA-HQ, AFHQ and CIFAR-10. The resolution of the first two datasets was set to 64×64 and for CIFAR-10 we trained on 32×32.

We started from EDM's [26] official implementation and made some necessary changes. Architecturally, the only change we made was to replace the convolutional layers with Gated Convolutions [55] that are known to perform well for inpainting problems. We observed that this change stabilized the training significantly, especially in the high-corruptions regime. As in EDM, we use the architecture from the DDPM++ [32] paper.

To avoid additional design complexity, we tried to keep our hyperparameters as close as possible to the EDM paper. We observed that for high corruption levels, it was useful to add gradient clipping,

16

otherwise, the training would often diverge. For all our experiments, we use gradient clipping with max-norm set to 1.0. We underline that unfortunately, even with gradient clipping, the training at high corruption levels ($p \geq 0.8$), still diverges sometimes. Whenever this happened, we restarted the training from an earlier checkpoint. We list the rest of the hyperparameters we used in Table 2.

Table 2: Training Hyperparameters

Dataset	Iters	Batch	LR	SDE	p	δ	Aug. Prob.	Ch. Multipliers	Dropout
CIFAR-10		512	1e-3		$\{0.2, 0.4, 0.6, 0.8\}$		0.12	$(1, 1, 1, 1)$	0.13
AFHQ	200000			VP	$\{0.2, 0.4, 0.6, 0.8, 0.9\}$	0.1			0.25
CelebA-HQ		256	2e-4		$\{0.2, 0.6, 0.8\}$ 0.9	0.3	0.15	$(1, 2, 2, 2)$	0.1

Training diffusion models from scratch is quite computationally intensive. We trained all our models for 200000 iterations. Our CIFAR-10 models required ≈ 2 days of training each on 6 A100 GPUs. Our AFHQ and CelebA-HQ models required ≈ 6 days of training each on 6 A100 GPUs. These numbers roughly match the performance reported in the EDM paper, indicating that the extra corruption we need to do on matrix A does not increase training time.

Due to the increased computational complexity of training these models, we could not extensively optimize the hyperparameters, e.g. the δ probability in our extra corruption. For higher corruption, e.g. for $p = 0.9$, we noticed that we had to increase δ in order for the model to learn to perform well on the unobserved pixels.

Finetuning Deepfloyd's IF. We access Deepfloyd's IF [2] model through the `diffusers` library. The model is a Cascaded Diffusion Model [21]. The first part of the pipeline is a text-conditional diffusion model that outputs images at resolution 64×64. Next in the pipeline, there are two diffusion models that are conditioned both in the input text and the low-resolution output of the previous stage. The first upscaling module increases the resolution from 64×64 to 256×256 and the final one from 256×256 to 1024×1024.

To reduce the computational requirements of the finetuning, we only work with the first text-conditional diffusion model that works with 64×64 resolution images. Once the finetuning is completed, we use again the whole model to generate high-resolution images.

For the finetuning, we set the training batch size to 32 and the learning rate to $3\mathrm{e} - 6$. We train for a maximum of 15000 steps and we keep the checkpoint that gives the lowest error on the pixels that we further corrupted. To further reduce the computational requirements, we use an 8-bit Adam Optimizer and we train with half-precision.

For our CelebA finetuning experiments, we set $\delta = 0.1$ and $p = 0.8$. We experiment with the full training set, a subset of size 3000 (see Figure 1) and a subset of 300. For the model trained with only 300 heavily corrupted samples, we did not observe memorization but the samples were of very low quality. Intuitively, our algorithm provides a way to control the trade-off between memorization and fidelity. Fully exploring this trade-off is a very promising direction for future work. For our MRI experiments, we use two non-overlapping blocks that each obfuscates 25% of the image and we evaluate the model in one of them.

All of our fine-tuning experiments can be completed in a few hours. Training for 15000 iterations takes ≈ 10 hours on an A100 GPU, but we usually get the best checkpoints earlier in the training.

D   Evaluation Details

FID evaluation. Our FID [19] score is computed with respect to the training set, as is standard practice, e.g. see [20, 26, 49]. For each of our models trained from scratch, we generate 50000 images using the seeds $0 - 49999$. Once we generate our images, we use the code provided in the official implementation of the EDM [26] paper for the FID computation.

17

E Additional Experiments

E.1 Restoration performance with noisy measurements

In Table 1 of the main paper, we compare the restoration performance of our models and vanilla diffusion models (trained with uncorrupted images). We compare the restoration performance in the task of random inpainting because it is straightforward to use our models to solve this inverse problem. It is potentially feasible to use our trained generative models to solve any (linear or non-linear) inverse problem but we leave this direction for future work.

In this section, we further compare our models (trained with randomly inpainted images) in the task of inpainting with noisy measurements. Concretely, we want to predict x_0, given measurements $y_t = A(x_0 + \sigma_{y_t} \eta)$, $\eta \sim \mathcal{N}(0, I)$. As in the rest of the paper, we assume that the mask matrix A is known. We can solve this problem with one model prediction using our trained models since according to Eq. 4.1, we are learning: $h_\theta(y_t, A, t) = \mathbb{E}[x_0 | A x_t = y_t, A]$.

We use the EDM [26] state-of-the-art model trained on AFHQ as our baseline (as we did in the main paper). To use this pre-trained model to solve the noisy random inpainting inverse problem, we need a reconstruction algorithm. We experiment again with DPS and DDRM which can both handle inverse problems with noise in the measurements. We present our results in Table 3. As shown, our models significantly outperform the EDM models that use the DDRM reconstruction algorithm and perform on-par with the EDM models that use the DPS reconstruction algorithm.

Dataset	Corruption Probability	Measurement Noise (σ_{y_0})	Method	LPIPS	PSNR	NFE
AFHQ			Ours	0.0861	29.46	1
			DPS	**0.0846**	**29.83**	100
	0.4	0.05		0.2061	24.47	35
			DDRM	0.1739	25.38	99
				0.1677	25.58	199
			Ours	0.1031	27.40	1
			DPS	**0.0949**	**27.92**	100
	0.6	0.05		0.4066	18.73	35
			DDRM	0.3626	19.49	99
				0.3506	19.70	199
			Ours	0.1792	**23.21**	1
			DPS	**0.1778**	23.01	100
	0.8	0.05		0.5879	13.65	35
			DDRM	0.5802	13.99	99
				0.5753	14.09	199

Table 3: Comparison of our model (trained on corrupted data) with state-of-the-art diffusion models on CelebA (DDIM [47] model) and AFHQ (EDM [26] model) for solving the random inpainting inverse problem with **noise**.

E.2 Comparison with Supervised Methods

For completeness, we include a comparison with Masked AutoEncoders [18], a state-of-the-art supervised method for solving the random inpainting problem. The official repository of this paper does not include models trained on AFHQ. We compare with the available models that are trained on the iNaturalist dataset which is the most semantically close dataset we could find. We emphasize that this model was trained with access to uncorrupted images. Results are shown in 4. As shown, our method and DPS outperform this supervised baseline. We underline that this experiment is included for completeness and does not exclude the possibility that there are more performant supervised alternatives for random inpainting.

E.3 Sampler Ablation

By Lemma A.3, if it is possible to learn $p_0(x_0)$ using corrupted samples then it is also possible to use our learned model to sample from $p_0(x_0)$. Even though such an algorithm exists, we do not know which one it is.

18

Corruption Probability	Method	LPIPS	PSNR	NFE
	Ours	0.0304	33.27	1
0.4	DPS	**0.0203**	**34.06**	100
	MAE	0.0752	28.88	1
	Ours	0.0628	29.46	1
0.6	DPS	**0.0518**	**30.03**	100
	MAE	0.0995	25.89	1
	Ours	0.1245	**25.37**	1
0.8	DPS	**0.1078**	25.30	100
	MAE	0.1794	22.01	1

Table 4: Comparison with the MAE [18], a state-of-the-art supervised method for solving the restoration task of random inpainting.

In the paper, we proposed to simple ideas for sampling, the Fixed Mask Sampler and the Reconstruction Guidance Sampler. The Fixed Mask Sampler fixes a mask throughout the sampling process. Hence, sampling with this algorithm is equivalent to first sampling some pixels from the marginals of $p_0(\boldsymbol{x}_0)$ and then completing the rest of the pixels with the best reconstruction (the conditional expectation) in the last step. This simple sampler performs remarkably well and we use it throughout the main paper.

To demonstrate that we can further improve the sampling performance, we proposed the Reconstruction Guidance sampler that takes into account all the pixels by forcing predictions of the model with different masks to be consistent. In Table 5 we ablate the performance of this alternative sampler. For the reconstruction guidance sampler, we select each time four masks at random and we add an extra update term to the Fixed Mask Sampler that ensures that the predictions of the Fixed Mask Sampler are not very different compared to the predictions with the other four masks (that have different context regarding the current iterate $\boldsymbol{x}_t$). We set the guidance parameter w_t to the value $5e-4$. As shown, this sampler improves the performance, especially for the low corruption probabilities where the extra masks give significant information about the current state to the predictions given only one fixed mask. However, the benefits of this sampler are vanishing for higher corruption. Given that the two samples perform on par and that the Reconstruction Guidance Sampler is much more computationally intensive (we need one extra prediction per step for each extra mask), we choose to use the Fixed Mask Sampler for all the experiments in the paper.

Sampler Type	Corruption Probability	FID	Inception Score
	0.2	11.70	7.97
Fixed Mask	0.4	18.85	7.45
	0.6	**28.88**	6.88
	0.8	**46.27**	**6.14**
	0.2	**11.59**	**8.01**
Reconstruction Guidance	0.4	**18.52**	**7.51**
	0.6	28.90	**6.91**
	0.8	46.31	6.13

Table 5: Comparison between the Fixed Mask sampler and the Reconstruction Guidance Sampler.

E.4   Additional Figures

Figure 7 shows reconstructions of AFHQ corrupted images with the EDM AFHQ model trained on clean data (columns 3, 4) and our model trained on corrupted data (column 5). The restoration task is random inpainting at probability $p = 0.8$. The last two rows also have measurement noise with $\sigma_{\boldsymbol{y}_0} = 0.05$.

19

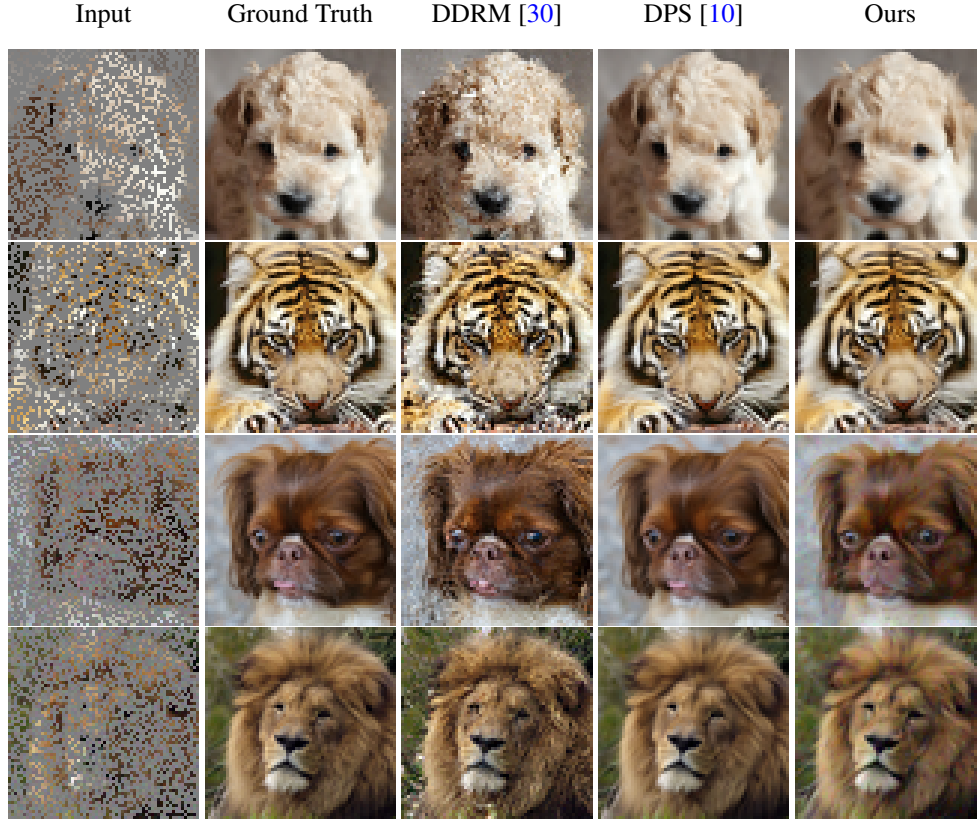



Figure 7: Reconstructions of AFHQ corrupted images with the EDM AFHQ model trained on clean data (columns 3, 4) and our model trained on corrupted data (column 5). The restoration task is random inpainting at probability $p = 0.6$. The last two rows also have measurement noise with $\sigma_{\boldsymbol{y}_0} = 0.05$.

In the remaining pages, we include uncurated unconditional generations of our models trained at different corruption levels p. Results are shown in Figures 8,9,10.

20

CelebA dataset, $p = 0.6, \delta = 0.1$ Uncurated samples from our model

CelebA dataset, $p = 0.8, \delta = 0.1$ Uncurated samples from our model

CelebA dataset, $p = 0.9, \delta = 0.3$ Uncurated samples from our model

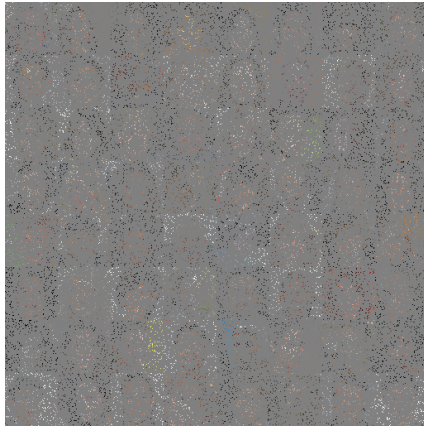
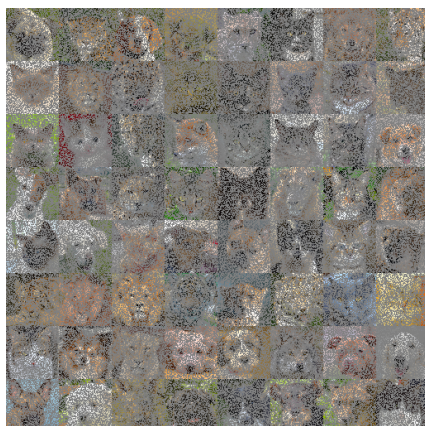



Figure 8: Left column: CelebA-HQ training dataset with random inpainting at different levels of corruption p (the survival probability is $(1 - p) \cdot (1 - \delta)$). Right column: Unconditional generations from our models trained with the corresponding parameters. As shown, the generations become slightly worse as we increase the level of corruption, but we can reasonably well learn the distribution even with 93% pixels missing (on average) from each training image.

21

AFHQ dataset, $p = 0.4, \delta = 0.1$ Uncurated samples from our model

AFHQ dataset, $p = 0.6, \delta = 0.1$ Uncurated samples from our model

AFHQ dataset, $p = 0.8, \delta = 0.1$ Uncurated samples from our model

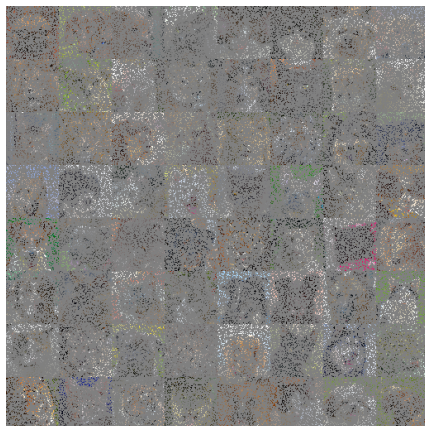
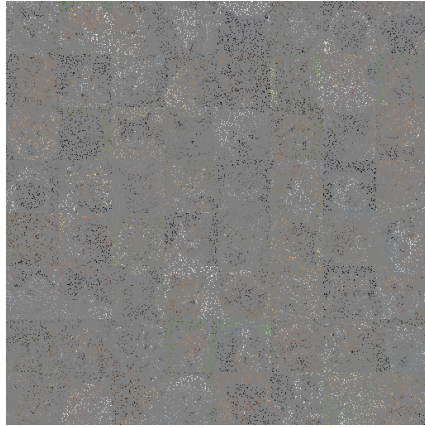
22

AFHQ dataset, $p = 0.9, \delta = 0.1$ Uncurated samples from our model

Figure 9: Left column: AFHQ training dataset with random inpainting at different levels of corruption p (the survival probability is $(1 - p) \cdot (1 - \delta)$). Right column: Unconditional generations from our models trained with the corresponding parameters. As shown, the generations become slightly worse as we increase the level of corruption, but we can reasonably well learn the distribution even with 91% pixels missing (on average) from each training image.

23

CIFAR-10 training dataset,
$p = 0.2, \delta = 0.1$

Uncurated samples from our
model

CIFAR-10 training dataset,
$p = 0.4, \delta = 0.1$

Uncurated samples from our
model

CIFAR-10 training dataset,
$p = 0.6, \delta = 0.1$

Uncurated samples from our
model

CIFAR-10 training dataset,
$p = 0.8, \delta = 0.1$

Uncurated samples from our
model

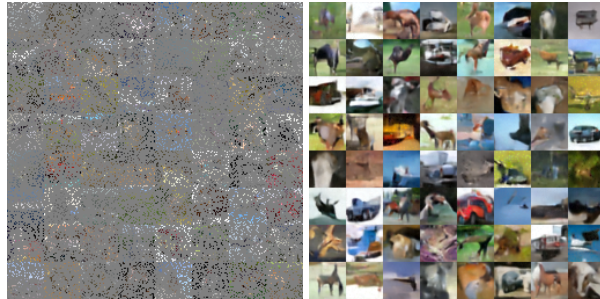



Figure 10: Left column: CIFAR-10 training dataset with random inpainting at different levels of corruption p (the survival probability is $(1 - p) \cdot (1 - \delta)$). Right column: Unconditional generations from our models trained with the corresponding parameters. As shown, the generations become slightly worse as we increase the level of corruption, but we can reasonably well learn the distribution even with high corruptions.

24