DiffEM: Learning from Corrupted Data with Diffusion Models via Expectation Maximization

Anonymous authors
Paper under double-blind review

Abstract

Diffusion models have emerged as powerful generative priors for high-dimensional inverse problems, yet learning them when observations are only available through a *corruption channel* remains challenging. In this work, we propose DiffEM, a new method for training diffusion models with Expectation-Maximization (EM) from corrupted data that does not rely on any approximations or heuristics. DiffEM utilizes conditional diffusion models to reconstruct clean data from observations in the E-step, and then uses the reconstructed data to refine the conditional diffusion model in the M-step. Theoretically, we provide monotonic convergence guarantees for the DiffEM iteration, assuming appropriate statistical conditions. We demonstrate the effectiveness of our approach through experiments on various image reconstruction tasks.

1 Introduction

Diffusion models (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2020) have emerged as powerful tools for learning high-dimensional distributions, achieving remarkable success across a broad range of generative tasks. Their effectiveness as learned priors has led to significant advances in solving inverse problems (Kawar et al., 2021; Choi et al., 2021; Saharia et al., 2022), including image inpainting, denoising, and super-resolution. However, in many real-world scenarios, acquiring clean training data remains difficult, costly, or proprietary, and can raise significant concerns, as training on clean data might lead to memorization (Somepalli et al., 2023a; Carlini et al., 2023; Somepalli et al., 2023b; Shah et al., 2025), posing privacy and copyright risks. While data with mild or moderate corruption is often more readily available and cheaper to acquire, particularly in domains like medical imaging (Wang et al., 2016; Zbontar et al., 2018) and compressive sensing, training diffusion models effectively using only corrupted or noisy observations presents substantial technical challenges.

The fundamental difficulty lies in the fact that standard techniques for training diffusion models are designed for settings with access to clean data from the data distribution. When only corrupted or noisy observations are available, these techniques become inapplicable, and training diffusion models effectively reduces to learning a latent variable model from corrupted observations, a problem well-known for its theoretical and practical challenges.

Recent work (Rozet et al., 2024; Bai et al., 2024) has proposed addressing this challenge by applying the Expectation-Maximization (EM) method with diffusion models as priors. However, this approach faces a critical difficulty: in each E-step, the algorithm must sample from the posterior distribution given the corrupted observations, whereas it only has access to the score function of the diffusion prior. To overcome this, these works adopt ad hoc posterior sampling schemes that rely on various approximations of the posterior score function that explicitly incorporate the corruption channel. Such approximation schemes, however, are based on implicit structural assumptions about the true data distribution and the corruption channel, making their approximation errors difficult to quantify.

In this work, we propose a new method that combines diffusion models with the EM framework. Our key insight is that instead of learning a diffusion prior and then performing approximate sampling, we can directly model the posterior distribution using a conditional diffusion model (Saharia et al., 2022; Daras

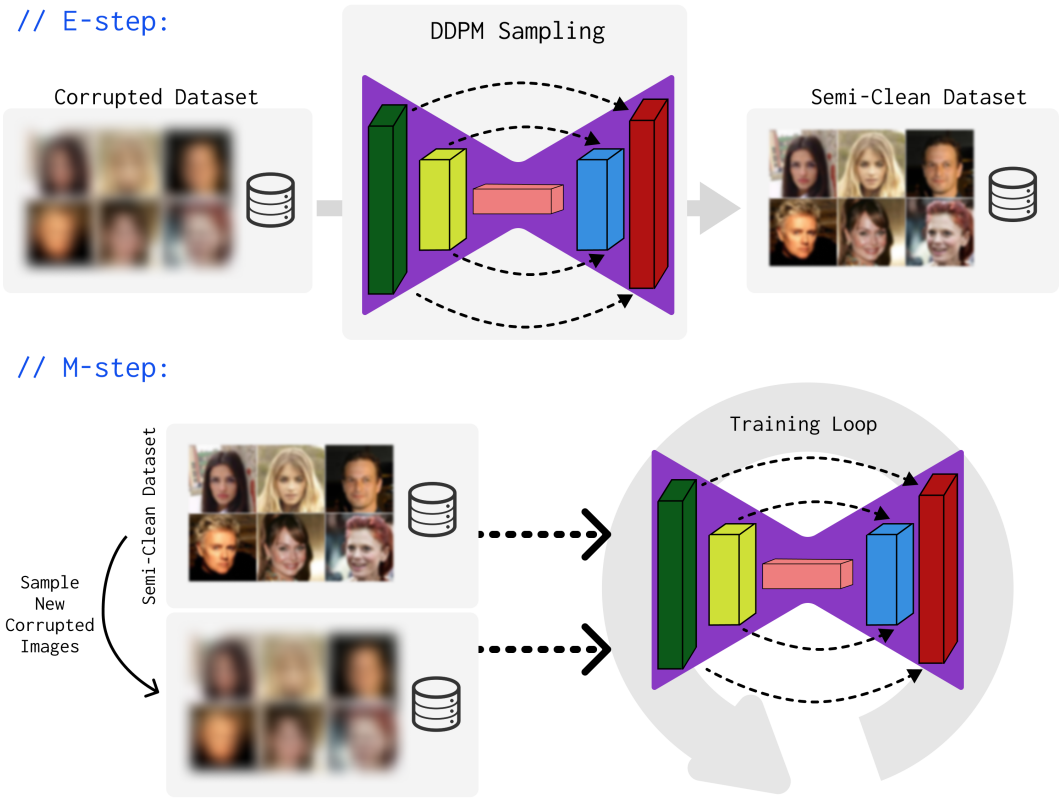

Figure 1: Illustration of one DiffEM iteration. **E-step** (top): the conditional diffusion model samples reconstructions of the clean data given the corrupted observations. **M-step** (bottom): the conditional diffusion model is retrained on the reconstructed data.

et al., 2024a). The primary advantage of our approach lies in its independence from specific approximate posterior sampling schemes. Notably, it can handle any corruption channel, as it makes no assumptions about the data distribution and corruption channel beyond requiring that the posterior score function can be expressed by the denoiser network. Furthermore, we provide theoretical analysis of the proposed EM iteration, demonstrating its convergence under appropriate conditions on the approximation error of the denoiser network along with the rate of convergence. We demonstrate the effectiveness of DiffEM on a synthetic manifold learning task and real-world image datasets (CIFAR-10 and CelebA) across a variety of corruption channels, including random masking, Gaussian blur, and JPEG compression. We further experiment with corruption mismatch, where the method operates under a slightly misspecified corruption channel, demonstrating the robustness of our approach.

1.1 Related work

Learning diffusion models with corrupted datasets Several approaches have been proposed to train diffusion models from corrupted observations. For linear corruption $Y \sim \mathsf{N}\big(AX, \sigma_Y^2 \mathbf{I}\big)$, where Y is the observation and X is the underlying clean sample (described in eq. (2)), methods such as SURE-score (Aali et al., 2023), GSURE (Kawar et al., 2023), and Ambient-Diffusion (Daras et al., 2023b; Aali et al., 2025; Daras et al., 2025a) train the denoiser network using a surrogate loss function. Specializing to Gaussian corruption $Y \sim \mathsf{N}\big(X, \sigma_Y^2 \mathbf{I}\big)$, Daras et al. (2023a; 2024b) propose enforcing *consistency* of the diffusion model to enable generalization to unseen noise levels, while Lu et al. (2025) develop an iterative scheme to refine the diffusion prior. Recent work (Rozet et al., 2024; Bai et al., 2024) identifies the Expectation-Maximization

(EM) method as a promising framework for training diffusion priors with linearly corrupted observations. However, as these EM approaches employ diffusion models as *priors*, they rely heavily on approximation schemes for posterior sampling and, moreover, assume a linear corruption channel (detailed discussion in section 2.1). In contrast, DiffEM uses a conditional diffusion model to directly represent the posterior, requiring no such approximations and supporting arbitrary corruption channels.

Solving inverse problems with diffusion models Diffusion models have also been shown to be powerful priors for a wide range of inverse problems in computer vision and medical imaging. A line of work—including SNIPS (Kawar et al., 2021), ILVR (Choi et al., 2021), DDRM (Kawar et al., 2022), Palette (Saharia et al., 2022), and DPS (Chung et al., 2022), among others—has demonstrated the effectiveness of both unconditional and conditional diffusion models in addressing various tasks, such as super-resolution, inpainting, deblurring, and compressed sensing. As surveyed by Daras et al. (2024a), many of these approaches leverage learned diffusion priors and perform posterior sampling through approximations of the posterior score function.

1.2 Preliminaries

Problem setup The *data distribution* $P_X^\star$ is a distribution over the space $\mathcal{X}$ of latent variables, and the *corruption channel* $\mathbf{Q}(\cdot|X)$ represents the conditional distribution of observations given the latent variables, mapping each point $X \in \mathcal{X}$ to a distribution over the observation space $\mathcal{Y}$. The observation is generated as

$$Y \sim \mathbf{Q}(\cdot|X), \quad \text{where } X \sim P_X^\star, \tag{1}$$

We denote by $P^\star$ the joint distribution of (X, Y), and by $P_Y^\star$ the marginal distribution of Y. This formulation encompasses classical inverse problems by specifying $\mathbf{Q}(\cdot|X) = \mathsf{N}\big(\mathcal{A}(X), \sigma_Y^2\mathbf{I}\big)$, where $\mathcal{A} : \mathcal{X} \to \mathbb{R}^d$ is a known forward operator.

In our setting, the learner only has access to a dataset $\{Y^{[1]}, \cdots, Y^{[N]}\}$ consisting of i.i.d. observations from $P_Y^\star$, and $\mathbf{Q}$ is assumed to be known. The goal is two-fold:

- **Unconditional generation**: to generate new samples from the ground-truth data distribution $P_X^\star$.
- **Posterior sampling**: to sample $X \sim P^\star(\cdot|Y)$ given an observation Y.

Our setup strictly generalizes the settings studied in recent work (Daras et al., 2023b;a; Rozet et al., 2024; Bai et al., 2024; Daras et al., 2024b). In those prior works, the latent space is $\mathcal{X} = \mathbb{R}^{d_x}$ (consisting of "clean images"), and there is a known distribution P_A of corruption matrices $A \in \mathbb{R}^{d \times d_x}$. The observation is drawn as

$$\begin{aligned} &Y = (AX + \epsilon, A), \\ &\text{where } X \sim P_X^\star, A \sim P_A, \epsilon \sim \mathsf{N}\big(0, \sigma_Y^2\mathbf{I}\big), \end{aligned} \tag{2}$$

i.e., the observation $Y \in \mathbb{R}^d \times \mathbb{R}^{d \times d_x}$ is a (corrupted image, corruption matrix) pair, with the image corrupted by the matrix $A \sim P_A$ and the additive Gaussian noise ϵ. By choosing different distributions P_A for the corruption matrix, (2) can model problems including random masking (Daras et al., 2023b; Rozet et al., 2024; Bai et al., 2024) and blurring (Bai et al., 2024). Our setting subsumes this family of corruptions while extending to arbitrary corruption mechanisms, including non-linear and discrete ones.

Diffusion models Diffusion models aim to learn how to generate new samples from a distribution p_0 over $\mathbb{R}^d$, given access to samples from p_0. Following Song et al. (2020), we consider the diffusion process $(X_t)_{t\in[0,1]}$ with $X_0 \sim p_0$, and $X_t|X_0 \sim \mathsf{N}\big(X_0, \sigma_t^2\mathbf{I}\big)$. The diffusion process is described by the following stochastic differential equation (SDE):

$$\mathrm{d}X_t = g(t)\mathrm{d}\mathbf{B}_t, \qquad X_0 \sim p_0, \tag{3}$$

where $g(t)^2 = \frac{d\sigma_t^2}{dt}$, and $(\mathbf{B}_t)_{t\in[0,1]}$ is the standard Brownian motion. Let $p_t(x)$ be the density function of $X_t \in \mathbb{R}^d$. It is well-known that the reverse of process (3) is described by the following reverse-time SDE:

$$\mathrm{d}X_t = -g(t)^2\nabla_x \log p_t(X_t)\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{B}}_t, \quad X_1 \sim p_1, \tag{4}$$

where $(\bar{\mathbf{B}}_t)_{t\in[0,1]}$ denotes a reverse-time Brownian motion. For sufficiently large σ_1, we have $p_1 \approx \mathsf{N}\big(0, \sigma_1^2\mathbf{I}\big)$. The score function $(x,t) \mapsto \nabla_x \log p_t(x)$ is typically parameterized by a neural network $\mathbf{s}_\theta(x,t)$. By Tweedie's formula, $\nabla_x \log p_t(x) = \frac{\mathbb{E}[X_0|X_t=x]-x}{\sigma_t^2}$, where the expectation is taken under the forward process (3). Hence, $\mathbf{s}_\theta(x,t)$ is trained via denoising score matching (Vincent, 2011).

2 Expectation-Maximization Approach

When applied to our setup, the Expectation-Maximization (EM) method optimizes over a class of parameterized latent variable models $\{q_\theta(x,y)\}_\theta$ that aims to represent the joint ground-truth distribution of (X,Y). Here, $q_\theta(x,y) : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}_{\geq 0}$ is the probability density function associated with the model parameterized by parameter θ, and we denote by $q_\theta(y) : \mathcal{Y} \to \mathbb{R}_{\geq 0}$ the probability density function of the marginal distribution of the observable Y. EM seeks a parameter θ that maximizes the population log-likelihood of the observable variable:

$$\max_\theta \mathcal{L}(\theta), \quad \text{where } \mathcal{L}(\theta) := \mathbb{E}_{Y\sim P_Y^\star} \log q_\theta(Y).$$

This optimization problem is equivalent to minimizing the KL divergence between $P_Y^\star$ and $q_\theta(y)$. However, direct optimization is computationally intractable in general. To overcome this challenge, each step of the EM method optimizes the following evidence lower bound (ELBO) with a parameter $\widehat{\theta}$:

$$\mathcal{L}(\theta) \geq \mathbb{E}_{Y\sim P_Y^\star} \mathbb{E}_{X\sim q_{\hat{\theta}}(X|Y)} \log \tfrac{q_\theta(X,Y)}{q_{\hat{\theta}}(X|Y)}.$$

In particular, the EM algorithm can be succinctly written as: Starting from an initial point $\theta^{(0)}$, iterate

$$\theta^{(k+1)} = \arg\max_\theta \mathbb{E}_{Y\sim P_Y^\star} \mathbb{E}_{X\sim q_{\theta^{(k)}}(X|Y)} \log q_\theta(X,Y).$$

In our setting, since the corruption channel $\mathbf{Q}$ is known, the parameterized model satisfies $q_\theta(x,y) = \mathbf{Q}(Y=y|X=x)q_\theta(x)$. In this case, the EM iterations reduce to

$$\theta^{(k+1)} = \arg\max_\theta \mathbb{E}_{Y\sim P_Y^\star} \mathbb{E}_{X\sim q_{\theta^{(k)}}(X|Y)} \log q_\theta(X). \tag{5}$$

This specialization of EM has been studied in (Aubin-Frankowski et al., 2022; Rozet et al., 2024; Bai et al., 2024), and it is also the basis of our framework. To simplify the notation, we consider the *mixture posterior distribution* $\pi^{(k)}$ with density $\pi^{(k)}(x) = \mathbb{E}_{Y\sim P_Y^\star}[q_{\theta^{(k)}}(x|Y)]$, which is a mixture with respect to the observation distribution $P_Y^\star$ of the posteriors $q_{\theta^{(k)}}(X|Y)$ (Rozet et al., 2024). Then, the EM update (5) can be rewritten as

$$\theta^{(k+1)} = \arg\min_\theta D_{\mathrm{KL}}(\pi^{(k)}(x) \parallel q_\theta(x)), \tag{6}$$

i.e., the model $q_{\theta^{(k+1)}}$ minimizes the distance to the mixture posterior distribution $\pi^{(k)}$. Crucially, to implement this update, we must sample from the conditional distribution $q_{\theta^{(k)}}(X|Y)$.

2.1 Prior approach: EM with diffusion priors

Prior work (Rozet et al., 2024; Bai et al., 2024) performs posterior sampling with diffusion models as priors. Their methods are restricted to the *linear corruption model* described in eq. (2), where the observation is $Y = (AX+\epsilon, A)$, with $\epsilon \sim \mathsf{N}\big(0, \sigma_Y^2\mathbf{I}\big)$ the noise and $A \sim P_A$ a random corruption matrix. For simplicity, to describe these results, we focus on the case where A is fixed, i.e. $\mathbf{Q}(\cdot|X) = \mathbf{Q}_A(\cdot|X) = \mathsf{N}\big(AX, \sigma_Y^2\mathbf{I}\big)$.

In the EM approach of Rozet et al. (2024); Bai et al. (2024), the latent variable models are described by diffusion models. More precisely, each θ parameterizes a score function $\mathbf{s}_\theta(x,t)$, and $q_\theta(x)$ corresponds to the distribution of X_0 obtained by running the backward diffusion process with the score function $\mathbf{s}_\theta$. However, to sample from $q_\theta(X|Y)$, one needs to approximate the conditional score function $\nabla_x \log q_\theta(X_t = x|Y=y)$. Following previous work on posterior sampling with diffusion priors (Chung et al., 2022), the conditional score is decomposed according to Bayes' rule:

$$\nabla_x \log q_\theta(X_t = x|Y) = \nabla_x \log q_\theta(Y|X_t = x) + \nabla_x \log q_\theta(X_t = x).$$

The second term is given by the score function $\mathbf{s}_\theta(x,t)$. To approximate the first term, Bai et al. (2024) applies a simple approximation $q_\theta(Y=\cdot|X_t=x)\approx \mathsf{N}\big(A\mathbb{E}_\theta[X|X_t=x],\sigma_Y^2\mathbf{I}\big)$. Alternatively, Rozet et al. (2024) takes a more sophisticated approach by first applying a Gaussian approximation $q_\theta(X=\cdot|X_t=x)\approx\mathsf{N}(\mu_\theta,\Sigma_\theta)$, where $\mu_\theta=\mathbb{E}_\theta[X|X_t=x]$ and $\Sigma_\theta=\mathbb{V}_\theta[X|X_t=x]$, from which the conditional distribution of Y is approximated as

$$q_\theta(Y=\cdot|X_t=x)\approx\mathsf{N}\big(A\mu_\theta,\sigma_Y^2\mathbf{I}+A\Sigma_\theta A^\top\big).$$

Then, to calculate $\nabla_x\log q_\theta(Y|X_t=x)$, Rozet et al. (2024) introduces moment matching techniques to approximate the variance function $\mathbb{V}_\theta[X|X_t=x]$.

However, these approximations all rely on the assumption that $q_\theta(X_0=\cdot|X_t=x)$ is close to a Gaussian distribution. This assumption may not hold for general diffusion priors, which are highly multi-modal. Therefore, errors in these approximation schemes can be difficult to control and can compound across EM iterations. Furthermore, even when the learned diffusion prior q_θ is close to the ground truth, the posterior distribution of $X|Y$ (obtained by approximating the score $\nabla_x\log q_\theta(X_t=x|Y)$) might not accurately represent the true conditional distribution $q_\theta(X|Y)$ under the diffusion prior $q_\theta(X)$.

Additionally, the moment matching techniques of Rozet et al. (2024) are highly specialized to eq. (2). For a general corruption channel with non-linear transformations, calculating the score $\nabla_x\log q_\theta(Y|X_t=x)$ can be challenging even under the Gaussian approximation assumption.

2.2 Our Approach: EM with conditional diffusion model

Instead of parameterizing the data distribution $q_\theta(x)$ using a diffusion model, we directly model the posterior distribution $q_\theta(x|y)$ through a conditional score function network $\mathbf{s}_\theta(x,t|y)$.

Conditional diffusion process Given a latent variable model q, we consider the diffusion process

$$(X_0,Y)\sim q,\quad \mathrm{d}X_t=g(t)\mathrm{d}\mathbf{B}_t. \tag{7}$$

Let p be the joint distribution of $(\{X_t\}_{t\in[0,1]},Y)$. To sample from $q(X_0|Y)$, we consider the following reverse-time process:

$$\mathrm{d}X_t=-g(t)^2\mathbf{s}_\theta(X_t,t|Y)\mathrm{d}t+g(t)\mathrm{d}\bar{\mathbf{B}}_t,\quad X_1\sim q_1(\cdot|Y), \tag{8}$$

where the network $\mathbf{s}_\theta$ directly approximates the true conditional score function

$$\mathbf{s}_\theta(x,t\mid y)\approx\nabla_x\log p(X_t=x\mid y)=\frac{\mathbb{E}\left[X_0\,|X_t=x,\,Y=y\right]-x}{\sigma_t^2}. \tag{9}$$

where the expectation is taken over the process (7) (see e.g. (Daras et al., 2024a)). For a given parameter θ that parameterizes the conditional denoiser network $\mathbf{s}_\theta$, we let $q_\theta(X=\cdot|Y)$ be the distribution of X_0 generated by eq. (8). In particular, when $\mathbf{s}_\theta(x,t|y)=\nabla_x\log p(X_t=x|y)$, the reverse process (8) exactly generates $X_0\sim q(\cdot|Y)$, i.e., $q_\theta(\cdot|Y)=q(\cdot|Y)$.

EM with conditional diffusion models Based on the conditional diffusion process, we propose the EM procedure described in algorithm 1, using a conditional diffusion model to learn the *posterior* directly.

In the E-step, the algorithm generates the dataset $\mathcal{D}_X^{(k)}=\{X^{[1]},\cdots,X^{[N]}\}$ consisting of the reconstructions $X^{[i]}\sim q_{\theta^{(k)}}(\cdot|Y^{[i]})$.

Then, in the M-step, the algorithm uses $\mathcal{D}_X^{(k)}$ to train the conditional diffusion model $\theta^{(k+1)}$ to sample from $\widehat{P}^{(k)}(X|Y)$, the posterior of the joint distribution $\widehat{P}^{(k)}(X,Y)$ defined by $X\sim\mathcal{D}_X^{(k)}$ and $Y|X\sim\mathbf{Q}(\cdot|X)$. To train this model, we consider the following conditional score matching loss:

$$\begin{aligned}&L_{\mathrm{SM},k}(\theta)=\int_0^1\lambda_t\mathbb{E}_{X\sim\mathcal{D}_X^{(k)},Y\sim\mathbf{Q}(\cdot|X)}\mathbb{E}_{X_t=X+\sigma_tZ}L^\theta_{(X_t,t,Y,Z)}\mathrm{d}t,\\&\text{where } L^\theta_{(X_t,t,Y,Z)}=\left\|\mathbf{s}_\theta(X_t,t|Y)+Z\right\|^2\end{aligned} \tag{10}$$

Algorithm 1 DiffEM: Expectation-Maximization with a conditional diffusion model

Require: Dataset of corrupted observations $\mathcal{D}_Y = \{Y^{[1]}, \cdots, Y^{[N]}\}$, likelihood $\mathbf{Q}(\cdot|X)$, and an initialization for the conditional model $\theta^{(0)}$.

for $k = 0, 1, \cdots, K-1$ **do**

// E-step:

for $i \in [N]$ **do**

Generate the reconstruction $X^{[i]} \sim q_{\theta^{(k)}}(\cdot|Y^{[i]})$ using the current conditional model $\theta^{(k)}$.

end for

// M-step:

Train a new conditional diffusion model using the dataset $\mathcal{D}_X^{(k)} = \{X^{[1]}, \cdots, X^{[N]}\}$ by minimizing the objective provided in eq. (10):

$$\theta^{(k+1)} = \arg\min_{\theta} L_{\mathrm{SM},k}(\theta).$$

end for

Ensure: (1) The conditional diffusion model $\theta^{(K)}$, and

Ensure: (2) An *unconditional* diffusion model $\widehat{\theta}$ trained on the dataset $\mathcal{D}_X^{(K-1)}$.

where $Z \sim \mathsf{N}(0, \mathbf{I})$ is the unit noise, and $\lambda_t \geq 0$ is a weight sequence. It is straightforward to verify that, assuming the network $\mathbf{s}_\theta$ is expressive enough, the minimizer $\theta^\star$ of $L_{\mathrm{SM},k}$ satisfies $\mathbf{s}_{\theta^\star}(x, t|y) = \frac{\mathbb{E}[X_0|X_t=x, Y=y]-x}{\sigma_t^2}$, where the conditional expectation is taken with respect to the distribution defined by sampling $X_0 \sim \mathcal{D}_X^{(k)}$, $Y \sim \mathbf{Q}(\cdot|X_0)$, $X_t \sim \mathsf{N}\big(X_0, \sigma_t^2 \mathbf{I}\big)$. Therefore, provided the M-step loss is minimized, we expect to have $q_{\theta^{(k+1)}}(X|Y) \approx \widehat{P}^{(k)}(X|Y)$ (cf. section 3).

The advantage of conditional diffusion model Unlike approaches that rely on ad hoc approximation schemes for the posterior score function using unconditional diffusion models (Rozet et al., 2024; Bai et al., 2024), our framework directly employs a conditional diffusion model. Both the *data distribution* and the *corruption channel* are implicitly encoded in this model through the minimization of the conditional score matching loss (10). In experiments (section 4), we observe that DiffEM consistently outperforms EM methods with diffusion priors. As predicted by our theoretical analysis (section 3), this improvement is because conditional models avoid the approximation bottleneck inherent in heuristic posterior sampling schemes.

Output: Posterior sampler $\theta^{(K)}$ and diffusion prior $\widehat{\theta}$ Our framework is designed to address two complementary goals: (1) posterior sampling and (2) unconditional generation (cf. section 1.2). The conditional diffusion model trained by DiffEM naturally serves as a posterior sampler. For unconditional generation, we leverage the reconstructed dataset $\mathcal{D}_X^{(K-1)}$ generated during the final EM iteration, and train an unconditional diffusion prior on this dataset. In particular, when the target application requires only a diffusion prior (Daras et al., 2023b; Rozet et al., 2024; Bai et al., 2024), we may directly use $\widehat{\theta}$. In such cases, the conditional model adopted by our approach primarily serves as a means to accelerate EM convergence.

Computational efficiency of DiffEM The computational cost of DiffEM can be decomposed as

$$\text{Total Time} = T_{\mathsf{init}} + K \cdot T_{\mathsf{ft}} + T_{\mathsf{u}}, \tag{11}$$

where K is the number of EM iterations, T_{init} is the time of training a standard conditional diffusion model from scratch, $T_{\mathsf{ft}} \leq T_{\mathsf{init}}$ is the average time of *fine-tuning* the conditional diffusion model for each M-step, and T_{u} is the cost of training an unconditional model for unconditional generation. The cost $T_{\mathsf{init}} \geq T_{\mathsf{ft}}$ of training a diffusion model is intrinsic to diffusion-based learning methods.

In general, the computational cost of EM-based methods (Rozet et al., 2024; Bai et al., 2024) can be decomposed as eq. (11). In our experiments, we compare the compute cost of DiffEM and EM-MMPS at matched performance levels (Figure 3), showing that DiffEM is substantially more compute-efficient. Variable values in eq. (11) are also available in table 6.

3 Monotonic Improvement Property and Convergence

As observed by Aubin-Frankowski et al. (2022), when the iteration (6) is *exact*, i.e., when the sample size is infinite and the conditional model $q_{\theta^{(k+1)}}$ learns the mixture posterior exactly in each M-step, the EM iteration is equivalent to *mirror descent* in the space of measures. Therefore, the convergence of the *exact* EM iteration follows directly from the guarantees of mirror descent.

We study the DiffEM iteration, taking the *score-matching error* introduced by the M-step into account. For simplicity, we analyze the EM iteration with *fresh* corrupted samples. Specifically, we consider the variant of algorithm 1 where, at each iteration $k = 0, 1, \cdots, K-1$, a new dataset of corrupted observations $\mathcal{D}_Y^{(k)} = \left\{Y^{[1]}, \cdots, Y^{[N]}\right\} \sim P_Y^\star$ is drawn in the E-step. We continue to refer to this procedure as DiffEM throughout this section.

Under this variant, for each k, the reconstructed dataset $\mathcal{D}_X^{(k)} = \{X^{[1]}, \cdots, X^{[N]}\}$ consists of i.i.d. samples from the posterior mixture distribution $\pi^{(k)} = \mathbb{E}_{Y\sim P_Y^\star}[q_{\theta^{(k)}}(\cdot|Y)]$. We let $P^{(k)}$ be the joint probability distribution of (X, Y) under $X \sim \pi^{(k)}, Y \sim \mathbf{Q}(\cdot|X)$, and write $P_Y^{(k)}$ for the marginal of Y. The convergence is measured in terms of $D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)})$, the Kullback-Leibler (KL) divergence between the true observation distribution $P_Y^\star$ and the distribution $P_Y^{(k)}$. Intuitively, this measures how plausible the prior $\pi^{(k)}$ is by comparing the induced observation distribution $P_Y^{(k)}$ to $P_Y^\star$. [1]

Score-matching error We define the *score-matching error* of the kth M-step as

$$\varepsilon_{\mathsf{SM}}^{(k)} := \mathbb{E}_{Y\sim P_Y^\star} D_{\mathrm{KL}}(q_{\theta^{(k+1)}}(\cdot|Y) \parallel P^{(k)}(\cdot|Y)),$$

which measures the KL divergence between the conditional distribution $q_{\theta^{(k+1)}}(\cdot|Y)$ learned by the diffusion model in the kth M-step and the true posterior $P^{(k)}(\cdot|Y)$. This error can be decomposed into two components: (1) the error of the learned score function, which is the statistical error of score matching (10) with a finite sample size, and (2) the sampling error, which comes from the discretized backward diffusion process (8) starting from Gaussian noise. When the denoiser network is sufficiently expressive, the score-matching error can be upper bounded through statistical learning theory (Dou et al., 2024; Zhang et al., 2024; Wibisono et al., 2024; Chen et al., 2024; Gatmiry et al., 2024, etc.). The sampling error is addressed by existing work on backward diffusion sampling (see e.g., Chen et al., 2022; Conforti et al., 2023; 2025). Therefore, under appropriate conditions, it can be shown that the score-matching error $\varepsilon_{\mathsf{SM}}^{(k)} \to 0$ as the sample size N increases.

Monotonicity of EM Our first result (shown in section A.1) is the following approximate *monotonicity* property of the EM iteration in terms of the statistical error $\varepsilon_{\mathsf{SM}}^{(k)}$.

Lemma 1 (Monotonic improvement)**.** *For any $k \geq 0$, it holds that*

$$\underbrace{D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k+1)})}_{\text{error of prior } \pi^{(k+1)}} \leq \underbrace{D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)})}_{\text{error of prior } \pi^{(k)}} - \underbrace{D_{\mathrm{KL}}(\pi^{(k+1)} \parallel \pi^{(k)})}_{\text{difference between priors}} + \underbrace{\varepsilon_{\mathsf{SM}}^{(k)}}_{\text{score-matching error of } q_{\theta^{(k+1)}}}.$$

Therefore, when the statistical error $\varepsilon_{\mathsf{SM}}^{(k)} \to 0$, the divergence $D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)})$ is monotonically decreasing. In other words, in the EM iteration, the observation distribution induced by prior $\pi^{(k+1)}$ is always closer to $P_Y^\star$ compared to the observation distribution induced by $\pi^{(k)}$, modulo the score-matching error $\varepsilon_{\mathsf{SM}}^{(k)}$. In section 4.2.1, we corroborate this property in experiments, showing that DiffEM can improve upon the learned prior produced by EM-MMPS (Rozet et al., 2024).

Convergence rate Beyond monotonicity, we show that the EM iteration enjoys a convergence rate guarantee. However, this guarantee requires that the conditional model achieves a small approximation error measured in the latent space. Specifically, for each $k \geq 0$, we define the error

$$\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)} = \mathbb{E}_{(X,Y)\sim P^\star}\left[\log \frac{P^{(k)}(X|Y)}{q_{\theta^{(k+1)}}(X|Y)}\right],$$

[1] Here, the convergence is not measured in terms of the divergence between the data distribution $P_X^\star$ and $\pi^{(k)}$ because in general, the problem (1) might not be *identifiable*, i.e., there may exist a distribution $P_X' \neq P_X^\star$ that induces the same observation distribution $\mathbf{Q}_\# P_X' = P_Y^\star$. Therefore, convergence of the data distribution can only be obtained under the additional assumption of *identifiability* (cf. assumption 1).

which measures the closeness of the posterior likelihoods computed under $P^{(k)}$ and $q_{\theta^{(k+1)}}$ with respect to samples $(X, Y) \sim P^\star$. The error $\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}$ can be larger than $\varepsilon_{\mathsf{SM}}^{(k)}$ since it is measured under the unknown prior distribution $P_X^\star$. Nevertheless, we show that $\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}$ can be related to $\varepsilon_{\mathsf{SM}}^{(k)}$ under appropriate assumptions (detailed in section A.4). Below, we state the convergence guarantee of the EM iteration. The proof is in section A.2.

Proposition 1 (Convergence of EM iteration)**.** *For each $K \geq 0$, we have*

$$\min_{k \leq K} D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)}) \leq \tfrac{1}{K+1} \textstyle\sum_{k=0}^{K} D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)}) \leq \tfrac{D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(0)})}{K+1} + \max_{k \leq K} \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}.$$

Therefore, as the number of EM iterations increases, $P_Y^{(k)}$ converges to $P_Y^\star$ at the rate of $\frac{1}{k}$, up to the statistical error $\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}$. Furthermore, we can also derive the following last-iterate convergence by invoking lemma 1:

$$D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(K)}) \leq \tfrac{D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(0)})}{K+1} + \max_{k \leq K} \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)} + \textstyle\sum_{k=0}^{K} \varepsilon_{\mathsf{SM}}^{(k)}, \quad \forall K \geq 0.$$

Given that each EM update is computationally expensive, the above convergence rate is most relevant in the regime where $D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(0)}) \lesssim 1$, i.e., where the initial diffusion model provides a prior that is not too far from the ground-truth $P_X^\star$. Such a *warm start* model can be trained using existing methods (Daras et al., 2023b) that are computationally cheaper.

Stronger convergence under identifiability Under the assumption that the latent variable problem (1) is *identifiable*, we show that EM achieves *linear* convergence in terms of $D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)})$.

Assumption 1 (Identifiability)**.** *There exist parameters $\kappa \geq 1, R \geq 0$ such that for any distribution $P(x)$ with $D_{\mathrm{KL}}(P_X^\star \parallel P) \leq R$, it holds that*

$$D_{\mathrm{KL}}(P_X^\star \parallel P) \leq \kappa \cdot D_{\mathrm{KL}}(P_Y^\star \parallel \mathbf{Q}_\# P),$$

where $\mathbf{Q}_\# P$ is the distribution of Y under $X \sim P, Y \sim \mathbf{Q}(\cdot|X)$.

In other words, assumption 1 requires that for any prior P whose induced observation distribution $\mathbf{Q}_\# P$ is close to $P_Y^\star$, P itself must be close to the true data distribution $P_X^\star$. Intuitively, assumption 1 quantifies the *identifiability* of the latent variable problem (1). We show the following in section A.3.

Proposition 2 (Linear convergence of EM)**.** *Suppose that assumption 1 holds, $D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(0)}) \leq R$, and $\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)} \leq \frac{R}{\kappa}$ for each $k \geq 0$. Then it holds that*

$$D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(K)}) \leq \exp\Big(-\tfrac{K}{\kappa+1}\Big) \cdot D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(0)}) + (\kappa + 1) \max_k \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}.$$

4 Experiments

We evaluate the proposed method, DiffEM, through a series of experiments. We begin with a synthetic manifold learning task (section 4.1), where we show that the conditional diffusion model yields more accurate posterior samples than existing approximate posterior sampling schemes (Rozet et al., 2024). We then conduct distributional learning and image reconstruction experiments on CIFAR-10 (section 4.2) and CelebA (section 4.3), demonstrating that DiffEM outperforms prior approaches for learning diffusion models from corrupted data.

4.1 Synthetic Manifold

We evaluate our method's performance on a synthetic problem introduced by Rozet et al. (2024). In this setting, the latent space is $\mathcal{X} = \mathbb{R}^5$, with the latent distribution $P_X^\star$ supported on a one-dimensional curve in $\mathbb{R}^5$. The observation $Y = (AX + \epsilon, A)$ is generated through the following steps: (1) sample a latent point $X \sim P_X^\star$, (2) sample a corruption matrix $A \in \mathbb{R}^{2 \times 5} \sim P_A$ with rows drawn uniformly from the unit sphere $\mathbb{S}^4$, and (3) add Gaussian noise $\epsilon \sim \mathsf{N}\big(0, \sigma_Y^2 \mathbf{I}\big)$.

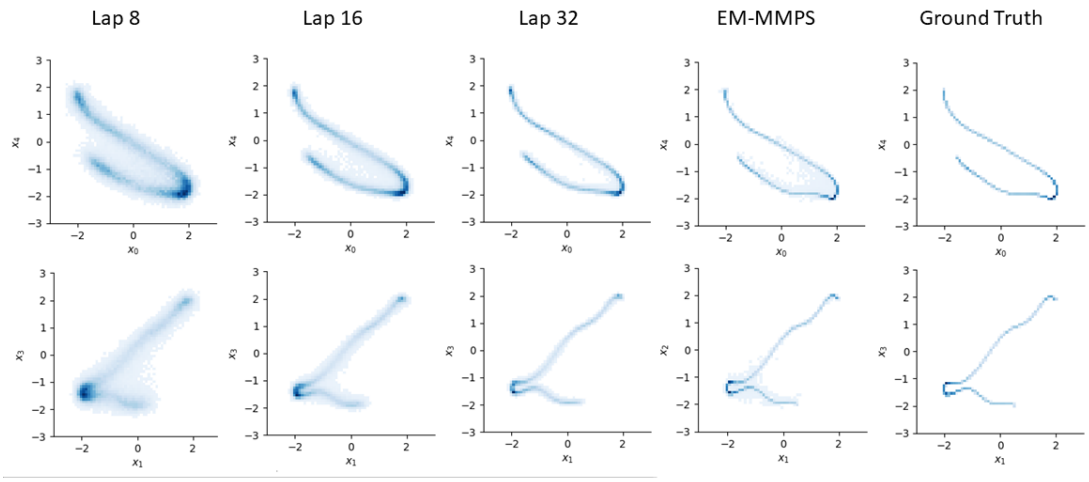

Figure 2: Evolution of the learned latent distribution on the synthetic manifold task. From left to right: reconstructed distributions from our model at DiffEM iterations 8, 16, and 32, followed by the distribution from EM-MMPS ((Rozet et al., 2024), 32nd iteration) and the ground-truth $P_X^\star$. Our method shows progressively better concentration around the ground-truth curve, demonstrating more accurate posterior learning compared to previous work.

Following Rozet et al. (2024), we apply our method to a dataset of $2^{16} = 65,536$ independent observations with noise variance $\sigma_Y = 10^{-2}$. Figure 2 illustrates the two-dimensional marginals of the reconstructed latent distribution compared to those obtained by Rozet et al. (2024). Qualitatively, the results demonstrate that our method achieves better concentration around the ground-truth curve, providing empirical evidence that the conditional diffusion model learns the posterior distribution more accurately than the approximate posterior sampling scheme of Rozet et al. (2024) (cf. section 2.1). Quantitatively, we compute the Sinkhorn distance between the reconstruction and the ground-truth (see Figure 4), confirming DiffEM's superior reconstruction accuracy. Detailed experimental settings and results are presented in section B.1.

4.2 Corrupted CIFAR-10

We next evaluate our method on the CIFAR-10 dataset (Krizhevsky, 2009), treating the 50000 training images as samples from the latent distribution $P_X^\star$.

Masked corruption Following (Daras et al., 2023b; Rozet et al., 2024), we consider randomly masking each pixel with probability ρ, i.e., the matrix $A \sim P_A$ in eq. (2) is diagonal with entries independently drawn from Bernoulli$(1-\rho)$. In this setting, the observation is generated as $Y = (AX + \epsilon, A)$, with $A \sim P_A$, $X \sim P_X^\star$, $\epsilon \sim \mathsf{N}\big(0, \sigma_Y^2 \mathbf{I}\big)$. In other words, each image is corrupted by (1) first randomly deleting every pixel independently with probability ρ, and then (2) adding isotropic Gaussian noise with variance σ_Y^2.

In our experiments, we set $\rho = 0.75$, $\sigma_Y = 10^{-3}$, i.e., each image has 75% of the pixels deleted and is corrupted by negligible Gaussian noise. We also perform experiments for when $\rho = 0.9$ and report the results in table 7. Apart from random masking, we explore other corruption channels such as Gaussian blurring (section B.4), JPEG compression (section B.6), and Gaussian noise with random masking (section B.9).

Experiment setup Our conditional diffusion model $q_\theta(x|y)$ is parameterized by a denoiser network $d_\theta(x_t, \sigma_t, y)$ with U-Net architecture and positional time-step embedding. We train the model for 21 DiffEM iterations, initializing with a Gaussian prior (detailed in section B). For each iteration, we train the denoiser network with conditional score matching in eq. (10) to learn the conditional mean $\mathbb{E}[X_0|X_t, Y]$. We then compare DiffEM to prior methods (Daras et al., 2023b; Rozet et al., 2024) under the following evaluation metrics, which correspond to the *posterior sampling* task and *unconditional generation* task (cf. section 1.2).

Task	Method	IS ↑	FID ↓	$\text{FD}_{\text{DINOv2}}$ ↓	FD_{∞} ↓
Posterior Sampling	Ambient-Diffusion	7.70	30.76	260.23	256.11
	EM-MMPS	<u>9.77</u>	6.49	237.02	231.80
	DiffEM (Ours)	**9.81**	<u>4.68</u>	<u>220.97</u>	<u>216.53</u>
	DiffEM (Warm-started)	9.66	**4.66**	**186.90**	**180.70**
Unconditional Generation	Ambient-Diffusion	6.88	28.88	1068.00	1062.98
	EM-MMPS	8.14	13.18	643.59	640.14
	DiffEM (Ours)	**8.57**	**10.24**	<u>598.18</u>	<u>594.75</u>
	DiffEM (Warm-started)	<u>8.49</u>	<u>10.33</u>	**546.07**	**541.53**

Table 1: Posterior sampling and unconditional generation performance on CIFAR-10 under random masking ($\rho = 0.75$), comparing DiffEM against Ambient-Diffusion (Daras et al., 2023b) and EM-MMPS (Rozet et al., 2024). DiffEM outperforms both baselines across all main metrics. Details of DiffEM with warm-start are in section 4.2.1.

Task	Method	density	recall	precision	coverage
Posterior Sampling	Ambient-Diffusion	0.87616	0.75420	**0.79210**	<u>0.67930</u>
	EM-MMPS	0.68918	0.83780	0.72770	0.67160
	DiffEM (Ours)	0.58080	**0.87110**	0.70150	0.64080
	DiffEM (Warm-started)	0.72216	<u>0.86300</u>	<u>0.76320</u>	**0.72490**
Unconditional Generation	Ambient-Diffusion	1.40812	0.0825	**0.79370**	0.08170
	EM-MMPS	0.80986	0.4895	0.64740	0.24380
	DiffEM (Ours)	0.81284	<u>0.50490</u>	0.64900	<u>0.25640</u>
	DiffEM (Warm-started)	0.75816	**0.52560**	<u>0.65980</u>	**0.29370**

Table 2: Additional metrics (density, recall, precision, and coverage) using the evaluation protocol of Stein et al. (2023), for posterior sampling and unconditional generation on CIFAR-10 under random masking ($\rho = 0.75$). DiffEM and DiffEM (Warm-started) achieve competitive or superior recall and coverage compared to EM-MMPS (Rozet et al., 2024) and Ambient-Diffusion (Daras et al., 2023b).

Eval 1: Posterior sampling performance The final model returned by DiffEM is a *conditional diffusion model*, i.e., given any corrupted observation Y, the model samples a reconstructed image $X \sim q_\theta(\cdot|Y)$. Therefore, to evaluate the performance of posterior sampling, for each observation $Y^{[i]}$ in our dataset, we use the trained model to generate a reconstructed image $X^{[i]} \sim q_\theta(\cdot|Y^{[i]})$ and obtain the reconstructed dataset $\mathcal{D}_{\text{recon}} = \{X^{[1]}, \cdots, X^{[50000]}\}$ (similar to the E-step of algorithm 1). We then evaluate the quality of $\mathcal{D}_{\text{recon}}$ by computing the Inception Score (IS) (Salimans et al., 2016) and the Fréchet distance to the uncorrupted dataset in various representation spaces[2] to obtain the metrics FID (Heusel et al., 2017), $\text{FD}_{\text{DINOv2}}$ (Oquab et al., 2023; Stein et al., 2023), and FD_{∞} (Chong & Forsyth, 2020). The results are reported in table 1. Furthermore, we evaluate Precision, Coverage, Recall, and Density and report them in table 2. We use the publicly available codebase from Stein et al. (2023) to compute these metrics.

Eval 2: Unconditional generation performance We also note that the models trained by existing works (Daras et al., 2023b; Rozet et al., 2024; Bai et al., 2024) are *unconditional* diffusion models, which can be regarded as the reconstruction of the ground-truth data distribution $P_X^\star$. In DiffEM, the reconstructed data distribution is implicitly described by the conditional diffusion model q_θ. Therefore, to evaluate the data distribution recovered by DiffEM, we use the reconstructed dataset $\mathcal{D}_{\text{recon}}$ to train a new (unconditional) diffusion model $p_{\theta_{\text{uncond}}}$, which learns to sample from the data distribution induced by q_θ. We then evaluate

[2]The Fréchet distance measures discrepancies at the *distributional* level. Under severe corruption (75% of pixels deleted), the posterior distribution $P^\star_{X|Y}$ may not concentrate around a single ground-truth. As a result, classical reconstruction metrics such as PSNR and LPIPS are less appropriate in this setting (Rozet et al., 2024).

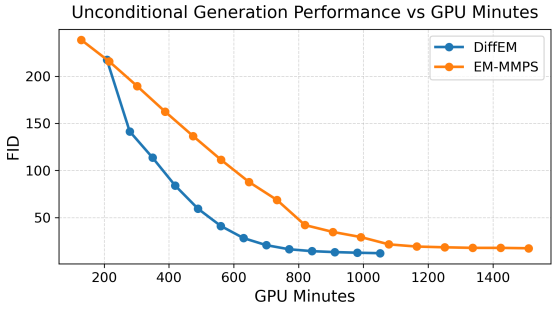

Figure 3: Performance of EM-MMPS and DiffEM (Measured in FID) as a function of GPU time. Both methods are trained on 4×H200 GPUs. DiffEM converges substantially faster and achieves superior performance for any fixed compute budget.

Method	EM-MMPS	DiffEM
K	32	21
T_{init}	43.0 ± 0.8	63.5 ± 0.4
T_{ft}	86.3 ± 0.7	70.3 ± 0.2
T_{u}	N/A	74.54 ± 0.09

Table 3: Comparison of computation time (cf. section 2.2), with $T_{\mathsf{init}}, T_{\mathsf{ft}}, T_{\mathsf{u}}$ measured in minutes using 4× H200. The cost of EM-MMPS (Rozet et al., 2024) can similarly be decomposed as $T_{\mathsf{init}} + K \cdot T_{\mathsf{ft}}$ (it does not incur the cost T_{u}). As shown, DiffEM is more computationally efficient.

the metrics (IS, FID, FD_∞, $\text{FD}_{\text{DINOv2}}$, Precision, Recall, Density, Coverage) of the model $p_{\theta_{\text{uncond}}}$ as our performance on the *unconditional generation* task. We report the metrics in table 1 and table 2.

Discussion and comparison We compare DiffEM to Ambient-Diffusion (Daras et al., 2023b)[3] and EM-MMPS (Rozet et al., 2024) under the above metrics in table 1 (higher IS and lower FID/FD scores indicate better performance) and table 2 (higher recall, precision, and coverage indicate better performance). To evaluate the diffusion prior trained by these baselines, we apply their respective approximate posterior sampling schemes and report the metrics evaluated on the reconstructed dataset. Across all four main metrics in table 1, DiffEM outperforms both Ambient-Diffusion and EM-MMPS, demonstrating the effectiveness of our approach.[4] Figure 6 shows qualitative results comparing the corrupted observations and reconstructions from our model.

We also compare the computational cost of DiffEM and EM-MMPS in table 6 and in the Figure 3 following our discussion in section 2.2.

4.2.1 DiffEM with warm-start

Additionally, we perform experiments on the masked CIFAR-10 dataset with *warm-started* DiffEM. Specifically, we take the diffusion prior trained with 32 iterations of EM-MMPS (Rozet et al., 2024), and perform 10 DiffEM iterations starting from this prior. We evaluate the final posterior sampling performance and unconditional generation quality (reported in table 1 and table 2).

The results show that using a high-quality initial prior accelerates the convergence of DiffEM: only 10 DiffEM iterations are needed. This observation is consistent with our theoretical results (section 3). Furthermore, warm-started DiffEM outperforms DiffEM with an initial Gaussian prior in terms of the scores $\text{FD}_{\text{DINOv2}}$, FD_∞, coverage, and recall, indicating that DiffEM can converge to a better distribution when starting from an informed prior.[5] We also plot the evolution of the IS, FID, $\text{FD}_{\text{DINOv2}}$, and FD_∞ scores in Figure 9, which corroborates the monotonic improvement property of DiffEM (lemma 1).

[3]We note that the Ambient-Diffusion model was trained on a dataset with corruption level $\rho' = 0.6$, an easier setting than ours ($\rho = 0.75$).

[4]We note that Bai et al. (2024) proposed EM-Diffusion and reported FID score 21.08 (corruption level $\rho' = 0.6$ and initialized with a diffusion prior trained on 50 clean images). However, we cannot reproduce their experiments to evaluate other metrics. Given that EM-MMPS (Rozet et al., 2024) achieves a much better FID score than EM-Diffusion (Bai et al., 2024), we believe it is sufficient to compare DiffEM to EM-MMPS.

[5]However, it is worth noting that warm-started DiffEM is computationally more expensive, as the warm-start prior requires training with 32 iterations of EM-MMPS.

4.2.2 Other Corruption Channels

Beyond random masking, we evaluate DiffEM under several additional corruption channels. We consider *Gaussian blurring*, where the linear component of the corruption is no longer diagonal; *JPEG compression*, which applies a nonlinear, discrete, and non-differentiable transformation with substantial information loss; and masking with high Gaussian noise, where random masking is combined with additive Gaussian noise. Details of these experiments are provided in section B.4, section B.6, and section B.9.

In real-world settings, the corruption channel $\mathbf{Q}$ is often not known exactly and must be approximated, leading to inevitable *model mismatch*. To study robustness under such mismatch, we evaluate DiffEM under varying levels of mismatch for the masking experiment. The results show that DiffEM is robust to moderate mismatch, and that overestimating the corruption level leads to more graceful performance degradation than underestimating it. The details are reported in section B.5.

For JPEG compression, the corruption channel is the JPEG algorithm itself, which applies a highly nonlinear and non-differentiable mapping to images. The JPEG algorithm includes a *quality* hyperparameter that controls compression strength, with lower values corresponding to more aggressive compression. We evaluate DiffEM at quality levels of 20%, 50%, and 70%. JPEG compression constitutes a particularly severe corruption, revealing that training for too many iterations degrades quality after reaching some maximum performance. This phenomenon is known as the Model Autophagy Disorder (MAD) effect (Figure 13), which we discuss in greater detail in section B.7.

4.3 Corrupted CelebA

We perform experiments on the CelebA dataset (Liu et al., 2018), with images cropped to 64×64 pixels following Wang et al. (2023); Daras et al. (2023b). We consider the setting in section 4.2 with masking probability $\rho \in \{0.5, 0.75\}$ and isotropic Gaussian noise level $\sigma_Y^2 = 0$. We initialize the first DiffEM iteration with the Gaussian prior (cf. section B). We evaluate the diffusion models trained by DiffEM following the protocol of section 4.2 (table 9). As shown in table 9, DiffEM significantly outperforms EM-MMPS. We also present sample reconstructed images in Figure 18 and an illustration of the pipeline in Figure 1. More details on the experimental settings and results are provided in section B.10.

Broader Impact Statement

While the primary goal of this research is to advance the field of machine learning and inverse problems, we identify several potential societal consequences and ethical considerations:

Positive Applications DiffEM has significant potential in scientific and medical domains where obtaining ground-truth "clean" data is difficult or impossible. Applications include denoising low-dose CT/MRI scans, enhancing under-sampled astronomical signals, and restoring degraded historical archives. By allowing models to learn directly from corrupted data, DiffEM enables high-fidelity generative modeling in fields with limited clean data resources.

Privacy and Security Risks The ability to reconstruct clean signals from corrupted data carries inherent risks regarding de-anonymization. If applied to data that has been intentionally blurred or redacted to protect privacy (e.g., license plates, faces, or sensitive documents), the model could potentially recover or "hallucinate" identifiable information. This necessitates cautious deployment in forensic or surveillance contexts.

Synthetic Media As with all generative diffusion models, the high-fidelity outputs of DiffEM could be leveraged to create synthetic media. While our focus is on data restoration, the improved ability to generate realistic samples from low-quality inputs could be used in the creation of deepfakes or other forms of misinformation.

Research Initiatives To mitigate these risks, we emphasize that the reconstructed outputs should be treated as probabilistic inferences rather than absolute ground truths. We advocate for the development of diagnostic tools that can quantify the uncertainty of reconstructions in sensitive applications.

References

Asad Aali, Marius Arvinte, Sidharth Kumar, and Jonathan I Tamir. Solving inverse problems with score-based generative priors learned from noisy data. *arXiv preprint arXiv:2305.01166*, 2023.

Asad Aali, Giannis Daras, Brett Levac, Sidharth Kumar, Alex Dimakis, and Jon Tamir. Ambient diffusion posterior sampling: Solving inverse problems with diffusion models trained on corrupted data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=qeXcMutEZY.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go mad, 2023. URL https://arxiv.org/abs/2307.01850.

Pierre-Cyril Aubin-Frankowski, Anna Korba, and Flavien Léger. Mirror descent with relative smoothness in measure spaces, with application to sinkhorn and em. *Advances in Neural Information Processing Systems*, 35:17263–17275, 2022.

Weimin Bai, Yifei Wang, Wenzheng Chen, and He Sun. An expectation-maximization algorithm for training clean diffusion models from corrupted observations. *arXiv preprint arXiv:2407.01014*, 2024.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

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. *arXiv preprint arXiv:2209.11215*, 2022.

Sitan Chen, Vasilis Kontonis, and Kulin Shah. Learning general gaussian mixtures with efficient score matching. *arXiv preprint arXiv:2404.18893*, 2024.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.

Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6070–6079, 2020.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Giovanni Conforti, Alain Durmus, and Marta Gentiloni Silveri. Score diffusion models without early stopping: finite fisher information is all you need. *arXiv e-prints*, pp. arXiv–2308, 2023.

Giovanni Conforti, Alain Durmus, and Marta Gentiloni Silveri. Kl convergence guarantees for score diffusion models under minimal data assumptions. *SIAM Journal on Mathematics of Data Science*, 7(1):86–109, 2025.

Giannis Daras, Yuval Dagan, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *arXiv preprint arXiv:2302.09057*, 2023a.

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

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024a.

Giannis Daras, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. *arXiv preprint arXiv:2404.10177*, 2024b.

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

Giannis Daras, Adrian Rodriguez-Munoz, Adam Klivans, Antonio Torralba, and Constantinos Daskalakis. Ambient diffusion omni: Training good models with bad data, 2025b. URL https://arxiv.org/abs/2506.10038.

Zehao Dou, Subhodh Kotekal, Zhehao Xu, and Harrison H Zhou. From optimal score matching to optimal sampling. *arXiv preprint arXiv:2409.07032*, 2024.

Khashayar Gatmiry, Jonathan Kelner, and Holden Lee. Learning mixtures of gaussians using diffusion models. *arXiv preprint arXiv:2404.18869*, 2024.

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 I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

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

Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022.

Bahjat Kawar, Noam Elata, Tomer Michaeli, and Michael Elad. Gsure-based diffusion model training with corrupted data. *arXiv preprint arXiv:2305.13128*, 2023.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

Haoye Lu, Qifan Wu, and Yaoliang Yu. SFBD: A method for training diffusion models with noisy data. In *Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025. URL https://openreview.net/forum?id=6HN14zuHRb.

Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. *Version: 0.3. 0, DOI*, 10, 2021.

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. *arXiv preprint arXiv:2304.07193*, 2023.

Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. On wasserstein two sample testing and related families of nonparametric tests, 2015. URL https://arxiv.org/abs/1509.02237.

William Hadley Richardson. Bayesian-based iterative method of image restoration∗. *J. Opt. Soc. Am.*, 62 (1):55–59, Jan 1972. doi: 10.1364/JOSA.62.000055. URL https://opg.optica.org/abstract.cfm?URI=josa-62-1-55.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. *arXiv preprint arXiv:2405.13712*, 2024.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. URL `https://arxiv.org/abs/1606.03498`.

Kulin Shah, Alkis Kalavasis, Adam R. Klivans, and Giannis Daras. Does generation require memorization? creative diffusion models using ambient diffusion, 2025.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6048–6058, 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *arXiv preprint arXiv:2305.20086*, 2023b.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36:3732–3784, 2023.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23 (7):1661–1674, 2011.

Shanshan Wang, Zhenghang Su, Leslie Ying, Xi Peng, Shun Zhu, Feng Liang, Dagan Feng, and Dong Liang. Accelerating magnetic resonance imaging via deep learning. In *International Symposium on Biomedical Imaging*, 2016. doi: 10.1109/ISBI.2016.7493320.

Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch diffusion: Faster and more data-efficient training of diffusion models. *arXiv preprint arXiv:2304.12526*, 2023.

Andre Wibisono, Yihong Wu, and Kaylee Yingxi Yang. Optimal score estimation via empirical bayes smoothing. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 4958–4991. PMLR, 2024.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastMRI: An Open Dataset and Benchmarks for Accelerated MRI. 2018. URL `http://arxiv.org/abs/1811.08839`.

Kaihong Zhang, Caitlyn H Yin, Feng Liang, and Jingbo Liu. Minimax optimality of score-based diffusion models: Beyond the density lower bound assumptions. *arXiv preprint arXiv:2402.15602*, 2024.

A Proofs from section 3

A.1 Proof of lemma 1

Note that

$$D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)}) - D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k+1)}) = \mathbb{E}_{Y\sim P_Y^\star} \log \frac{P_Y^{(k+1)}(Y)}{P_Y^{(k)}(Y)}.$$

By definition and Bayes' rule,

$$\begin{aligned}
P_Y^{(k+1)}(y) &= \int \mathbf{Q}(y|x)\pi^{(k+1)}(x)\mathrm{d}x = \int \mathbf{Q}(y|x)\pi^{(k)}(x)\cdot\frac{\pi^{(k+1)}(x)}{\pi^{(k)}(x)}\mathrm{d}x \\
&= \int P^{(k)}(x,y)\cdot\frac{\pi^{(k+1)}(x)}{\pi^{(k)}(x)}\mathrm{d}x \\
&= \int P_Y^{(k)}(y)\cdot P^{(k)}(x|y)\cdot\frac{\pi^{(k+1)}(x)}{\pi^{(k)}(x)}\mathrm{d}x \\
&= P_Y^{(k)}(y)\cdot\mathbb{E}_{X\sim q_{\theta^{(k+1)}}(\cdot|y)}\left[\frac{P^{(k)}(X|y)}{q_{\theta^{(k+1)}}(X|y)}\cdot\frac{\pi^{(k+1)}(X)}{\pi^{(k)}(X)}\right].
\end{aligned}$$

Therefore, by Jensen's inequality, we have

$$\begin{aligned}
&D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)}) - D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k+1)}) \\
&= \mathbb{E}_{Y\sim P_Y^\star}\log\frac{P_Y^{(k+1)}(Y)}{P_Y^{(k)}(Y)} \\
&= \mathbb{E}_{Y\sim P_Y^\star}\log\mathbb{E}_{X\sim q_{\theta^{(k+1)}}(\cdot|Y)}\left[\frac{\pi^{(k)}(X|Y)}{q_{\theta^{(k+1)}}(X|Y)}\cdot\frac{\pi^{(k+1)}(X)}{\pi^{(k)}(X)}\right] \\
&\geq \mathbb{E}_{Y\sim P_Y^\star}\mathbb{E}_{X\sim q_{\theta^{(k+1)}}(\cdot|Y)}\left[\log\left(\frac{P^{(k)}(X|Y)}{q_{\theta^{(k+1)}}(X|Y)}\cdot\frac{\pi^{(k+1)}(X)}{\pi^{(k)}(X)}\right)\right] \\
&= \mathbb{E}_{Y\sim P_Y^\star}\mathbb{E}_{X\sim q_{\theta^{(k+1)}}(\cdot|Y)}\log\frac{\pi^{(k+1)}(X)}{\pi^{(k)}(X)} - \mathbb{E}_{Y\sim P_Y^\star}\mathbb{E}_{X\sim q_{\theta^{(k+1)}}(\cdot|Y)}\log\frac{q_{\theta^{(k+1)}}(X|Y)}{P^{(k)}(X|Y)} \\
&= D_{\mathrm{KL}}(\pi^{(k+1)} \parallel \pi^{(k)}) - \mathbb{E}_{Y\sim P_Y^\star} D_{\mathrm{KL}}(q_{\theta^{(k+1)}}(\cdot|Y) \parallel P^{(k)}(\cdot|Y)).
\end{aligned}$$

Rearranging the terms completes the proof. □

A.2 Proof of proposition 1

We first show that: For each $k \geq 0$, it holds that

$$D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k+1)}) \leq D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) - D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k+1)}) + \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}.$$

To simplify the presentation, we define $\widetilde{\pi}^{(k+1)}(x) = \mathbb{E}_{Y\sim P_Y^\star} P^{(k)}(x|Y)$. Then, by definition, we have

$$\begin{aligned}
\widetilde{\pi}^{(k+1)}(x) &= \mathbb{E}_{Y\sim P_Y^\star} P^{(k)}(x|Y) \\
&= \mathbb{E}_{Y\sim P_Y^\star}\left[\frac{\pi^{(k)}(x)\mathbf{Q}(Y|x)}{P_Y^{(k)}(Y)}\right] \\
&= \pi^{(k)}(x)\cdot\mathbb{E}_{Y\sim\mathbf{Q}(\cdot|x)}\left[\frac{P_Y^\star(Y)}{P_Y^{(k)}(Y)}\right].
\end{aligned}$$

Therefore, it follows that

$$\begin{aligned} D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) - D_{\mathrm{KL}}(P_X^\star \parallel \widetilde{\pi}^{(k+1)}) &= \mathbb{E}_{X\sim P_X^\star} \log \frac{\widetilde{\pi}^{(k+1)}(X)}{\pi^{(k)}(X)} \\ &= \mathbb{E}_{X\sim P_X^\star} \log \mathbb{E}_{Y\sim \mathbf{Q}(\cdot|x)} \left[\frac{P_Y^\star(Y)}{P_Y^{(k)}(Y)}\right] \\ &\geq \mathbb{E}_{X\sim P_X^\star} \mathbb{E}_{Y\sim \mathbf{Q}(\cdot|x)} \left[\log \frac{P_Y^\star(Y)}{P_Y^{(k)}(Y)}\right] \\ &= \mathbb{E}_{Y\sim P_Y^\star} \left[\log \frac{P_Y^\star(Y)}{P_Y^{(k)}(Y)}\right] = D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)}). \end{aligned}$$

Furthermore, we have

$$\begin{aligned} & D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k+1)}) - D_{\mathrm{KL}}(P_X^\star \parallel \widetilde{\pi}^{(k+1)}) \\ &= \mathbb{E}_{X\sim P_X^\star} [\log \widetilde{\pi}^{(k+1)}(X) - \log \pi^{(k+1)}(X)] \\ &= \mathbb{E}_{X\sim P_X^\star} \left[\log \mathbb{E}_{Y\sim P_Y^\star} [P^{(k)}(X|Y)] - \log \mathbb{E}_{Y\sim P_Y^\star} [q_{\theta^{(k+1)}}(X|Y)]\right] \\ &\leq \mathbb{E}_{(X,Y)\sim P_X^\star} \left[\log \frac{P^{(k)}(X|Y)}{q_{\theta^{(k+1)}}(X|Y)}\right] = \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}. \end{aligned}$$

Combining the above equations, we have shown that

$$D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)}) \leq D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) - D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k+1)}) + \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}.$$

This is the desired upper bound. Taking the summation over $k = 0, 1, \cdots, K$ completes the proof. For the last-iterate convergence rate, we only need to use the fact that $D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k)}) \leq D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(K)}) + \sum_{\ell=k}^K \varepsilon_{\mathsf{SM}}^{(\ell)}$ (by lemma 1). □

A.3 Proof of proposition 2

By proposition 1, we have

$$D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k+1)}) + D_{\mathrm{KL}}(P_Y^\star \parallel P_Y^{(k+1)}) \leq D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) + \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}.$$

Using assumption 1, we know that as long as $D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) \leq R$, we have

$$(1+\kappa^{-1}) D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k+1)}) \leq D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) + \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}.$$

Denote $\widetilde{\varepsilon}_{\mathsf{SM}} = \max_k \widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}$. Therefore, using the fact that $\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)} \leq \widetilde{\varepsilon}_{\mathsf{SM}} \leq \frac{R}{\kappa}$, we can show by induction that $D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) \leq R$ for each $k \geq 0$, and hence

$$(1+\kappa^{-1}) D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k+1)}) \leq D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) + \widetilde{\varepsilon}_{\mathsf{SM}}.$$

Applying this inequality recursively, we obtain

$$\begin{aligned} D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k)}) &\leq \frac{\kappa}{1+\kappa} D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k-1)}) + \widetilde{\varepsilon}_{\mathsf{SM}} \\ &\leq \left(\frac{\kappa}{1+\kappa}\right)^2 D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(k-2)}) + \left(\frac{\kappa}{1+\kappa}\right) \widetilde{\varepsilon}_{\mathsf{SM}} + \widetilde{\varepsilon}_{\mathsf{SM}} \\ &\leq \cdots \\ &\leq \left(\frac{\kappa}{1+\kappa}\right)^k D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(0)}) + \sum_{i=0}^{k-1} \left(\frac{\kappa}{1+\kappa}\right)^{k-1-i} \widetilde{\varepsilon}_{\mathsf{SM}} \\ &\leq e^{-k/(\kappa+1)} D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(0)}) + (1+\kappa) \widetilde{\varepsilon}_{\mathsf{SM}}, \end{aligned}$$

where the last inequality follows from $\frac{\kappa}{1+\kappa} = 1 - \frac{1}{1+\kappa} \leq \exp\left(-\frac{1}{1+\kappa}\right)$. □

A.4 Relation between the Score-Matching errors

In this section, we provide the following upper bound for $\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}$ in terms of $\varepsilon_{\mathsf{SM}}^{(k)}$. Recall that $\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)}$ is defined as

$$\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)} = \mathbb{E}_{(X,Y)\sim P^\star}\left[\log \frac{P^{(k)}(X|Y)}{q_{\theta^{(k+1)}}(X|Y)}\right],$$

Proposition 3. *Suppose that* $\mathbb{E}_{Y\sim P_Y^\star} D_{\chi^2}(P^\star(\cdot|Y) \parallel q_{\theta^{(k+1)}}(\cdot|Y)) \le C < +\infty$. *Then it holds that* $\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)} \le 2\sqrt{(C+1)\varepsilon_{\mathsf{SM}}^{(k)}}$.

Proof of proposition 3. By definition,

$$\begin{aligned}
\widetilde{\varepsilon}_{\mathsf{SM}}^{(k)} &\le \mathbb{E}_{(X,Y)\sim P_X^\star}\left(\log \frac{P^{(k)}(X|Y)}{q_{\theta^{(k+1)}}(X|Y)}\right)_+ \\
&= \mathbb{E}_{Y\sim P_Y^\star}\mathbb{E}_{X\sim P_X^\star(\cdot|Y)}\left(\log \frac{P^{(k)}(X|Y)}{q_{\theta^{(k+1)}}(X|Y)}\right)_+ \\
&\le \mathbb{E}_{Y\sim P_Y^\star}\sqrt{\left(1 + D_{\chi^2}(P^\star(\cdot|Y) \parallel q_{\theta^{(k+1)}}(\cdot|Y))\right)\cdot \mathbb{E}_{X\sim q_{\theta^{(k+1)}}(\cdot|Y)}\left(\log \frac{P^{(k)}(X|Y)}{q_{\theta^{(k+1)}}(X|Y)}\right)_+^2} \\
&\le \mathbb{E}_{Y\sim P_Y^\star}\sqrt{\left(1 + D_{\chi^2}(P^\star(\cdot|Y) \parallel q_{\theta^{(k+1)}}(\cdot|Y))\right)\cdot 4D_{\mathrm{KL}}(q_{\theta^{(k+1)}}(\cdot|Y) \parallel P^{(k)}(\cdot|Y))} \\
&\le 2\sqrt{(C+1)\varepsilon_{\mathsf{SM}}^{(k)}},
\end{aligned}$$

where we apply lemma 2.

This yields the desired upper bound. □

Lemma 2. *For any distributions* P *and* Q, *it holds that*

$$\mathbb{E}_{X\sim Q}(\log P(X) - \log Q(X))_+^2 \le 4D_{\mathrm{KL}}(Q \parallel P).$$

Proof. Note that $\log x \le 2(\sqrt{x}-1)$ for any $x \ge 1$, and hence $(\log x)_+^2 \le 4(\sqrt{x}-1)^2$. Applying this inequality, we have

$$\begin{aligned}
\mathbb{E}_{X\sim Q}(\log P(X) - \log Q(X))_+^2 &= \mathbb{E}_{X\sim Q}\left(\log \frac{P(X)}{Q(X)}\right)_+^2 \\
&\le 4\mathbb{E}_{X\sim Q}\left(\sqrt{\frac{P(X)}{Q(X)}} - 1\right)^2 = 8D_{\mathrm{H}}^2(P,Q) \le 4D_{\mathrm{KL}}(Q \parallel P).
\end{aligned}$$

This is the desired upper bound.

□

B Experiment Details

Parameterization Following section 2.2, we adopt the denoiser parameterization $d_\theta(x,t|y)$, and the conditional score function $\mathbf{s}_\theta$ is thus given by

$$\mathbf{s}_\theta(x,t|y) = \frac{d_\theta(x,t|y) - x}{\sigma_t^2}.$$

Therefore, the score-matching loss defined in (10) can be equivalently written as

$$L_{\mathrm{SM},k}(\theta) = \int_0^1 \lambda_t' \mathbb{E}_{X_0 \sim \mathcal{D}^{(k)}, Y \sim \mathbf{Q}(\cdot|X)} \mathbb{E}_{X_t \sim \mathsf{N}\left(X_0, \sigma_t^2 \mathbf{I}\right)} \left\| d_\theta(X_t, t|Y) - X_0 \right\|^2 \mathrm{d}t, \tag{12}$$

where $\lambda_t' = \frac{\lambda_t}{\sigma_t^2}$, and λ_t is the weight function from (10).

In our experiments, we adopt the following noise schedule:

$$\sigma_t^2 = \exp\left((1-t)\log(\sigma_0) + t\log(\sigma_1)\right),$$

where $\sigma_0 < \sigma_1$ are appropriate parameters, and the scalar σ_t is encoded as a positional embedding. The input to the denoiser network is the concatenation of X_t, Y, and the positional embedding of the noise σ_t. We also choose $\lambda_t = (\sigma_t^2 + 1) \cdot f(t; \alpha, \beta)$, where $f(t; \alpha, \beta)$ is the density function of the Beta distribution with parameters (α, β).

For the manifold experiment (section B.1), we choose $\alpha = 3.5, \beta = 1.5$, $\sigma_0 = 10^{-3}, \sigma_1 = 10^1$. For the remaining experiments, we set $\alpha = \beta = 3$, $\sigma_0 = 10^{-3}, \sigma_1 = 10^2$.

Initialization As noted in section 3, the convergence rate of DiffEM depends on the quality of the initial prior $\pi^{(0)}$ through the quantity $D_{\mathrm{KL}}(P_X^\star \parallel \pi^{(0)})$, i.e., the KL divergence between the ground-truth data distribution $P_X^\star$ and the initial $\pi^{(0)}$. Therefore, a better initial prior may lead to faster convergence. In our experiments, we consider the following initialization strategies:

(a) **Corrupted prior:** For eq. (2), the observation is $Y = (AX + \epsilon, A)$. When $d_y = d_x$, we can consider the *corrupted prior* $\pi^{(0)}$, which is simply the distribution of $X' = AX + \epsilon$. To sample from $\pi^{(0)}$, we can draw $Y = (AX + \epsilon, A) \sim P_Y^\star$ and set $X' = Y[0 : d_y]$.

(b) **Gaussian prior:** In general, we can fit a Gaussian prior $\pi^{(0)} = \mathsf{N}(\mu_X, \Sigma_X)$ using the observations $\{Y^{[1]}, \cdots, Y^{[N]}\} \sim P_Y^\star$.

(c) **Warm-start:** More generally, we can set $\pi^{(0)}$ to be any pre-trained diffusion prior as the *warm-start* prior. In particular, this can be the diffusion prior trained on corrupted data by existing methods (Daras et al., 2023b; Kawar et al., 2023; Rozet et al., 2024, etc.).

For the experiments (except section 4.2.1), we adopt initialization strategy (b). Following the implementation in Rozet et al. (2024), the Gaussian prior is fitted efficiently through a few closed-form EM iterations. An exception is the experiment on blurred CIFAR-10, where we adopt strategy (a). In section 4.2.1, we perform experiments with strategy (c), applying DiffEM to the warm-start prior trained by EM-MMPS (Rozet et al., 2024), demonstrating that DiffEM can monotonically improve upon the initial prior.

B.1 More details on the experiment on synthetic manifold

We implement the denoiser network $d_\theta(x, t|y)$ using a Multi-Layer Perceptron (MLP). The network architecture and training hyperparameters are detailed in table 4.

To quantify the quality of the learned distribution, we compute the Sinkhorn divergence S_λ (Ramdas et al., 2015) with regularization parameter $\lambda = 10^{-3}$ after each EM iteration. The Sinkhorn divergence is defined as:

$$\begin{aligned} S_\lambda(\mu, \nu) &:= T_\lambda(\mu, \nu) - \frac{1}{2}(T_\lambda(\mu, \mu) + T_\lambda(\nu, \nu)) \\ T_\lambda(\mu, \nu) &:= \min_{\gamma \in \Pi(\mu, \nu)} \int_{(\mathbb{R}^d)^2} \|y - x\|_2^2 d\gamma(x, y) + 2\lambda H(\gamma, \mu \otimes \nu) \end{aligned}$$

Architecture	MLP
Input Shape	$5 + 2 + 5 \times 2 = 17$
Hidden Layers	3
Hidden Layer Sizes	256, 256, 256
Activation	SiLU
Normalization	LayerNorm
Optimizer	Adam
Weight Decay	0
Scheduler	linear
Initial Learning Rate	1×10^{-3}
Final Learning Rate	1×10^{-6}
Gradient Norm Clipping	1.0
Batch Size	1024
Epochs in each iteration	65536
Sampler	Predictor-Corrector
Sampler Steps	4096
Number of EM iterations	32

Table 4: Network architecture and training hyperparameters for the MLP used in the synthetic manifold experiment.

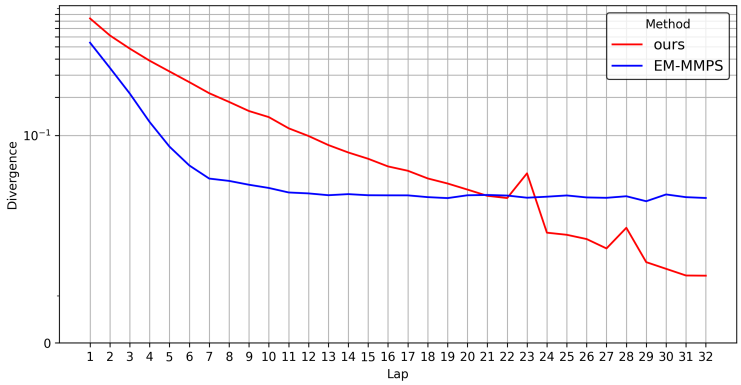

Figure 4: Evolution of Sinkhorn divergence between the ground-truth and reconstructed distributions over EM iterations for the experiment on manifold learning in $\mathbb{R}^5$ described in section 4.1. The red line shows DiffEM, and the blue line shows EM-MMPS.

We plot the evolution of Sinkhorn divergence over the iterations of DiffEM and EM-MMPS (Rozet et al., 2024) in Figure 4. We also plot the 2D marginals of the distributions reconstructed by DiffEM and EM-MMPS in Figure 5.

Figure 4 demonstrates that while EM-MMPS provides effective initialization when the learned distribution is far from the true data distribution, it plateaus quickly and fails to achieve further improvements. This is likely due to the inherent approximation error of the approximate posterior sampling scheme (MMPS). In contrast, DiffEM continues to refine the reconstructed distribution, achieving better concentration around the ground-truth curve.

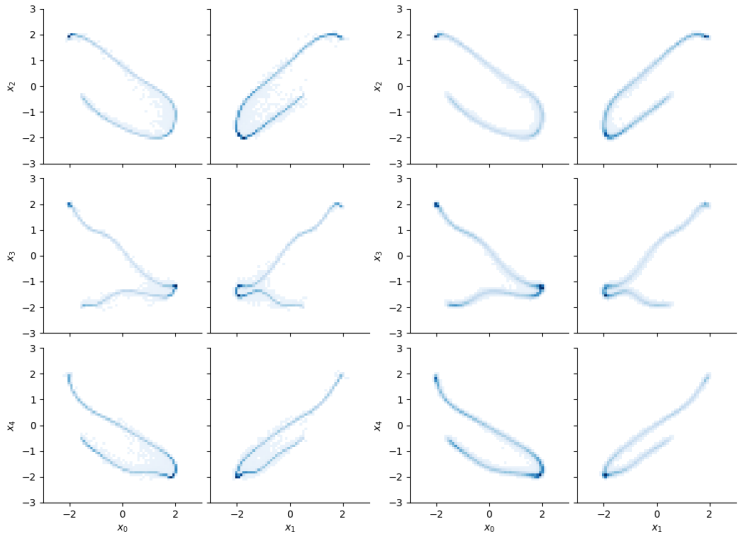

Figure 5: Comparison of 2D marginals of reconstructed distributions after the final iteration for the experiment on learning the manifold in $\mathbb{R}^5$ described in section 4.1. **Left**: EM-MMPS; **Right**: DiffEM. DiffEM achieves better concentration around the ground-truth curve, indicating more accurate posterior learning.

B.2 Details of Masked CIFAR-10 (section 4.2)

In this experiment, the conditional denoiser network d_θ is a U-Net (Ronneberger et al., 2015), and we adopt the same experimental setup as Rozet et al. (2024) for a fair comparison. The only major difference is that our model is conditional: in addition to the sinusoidal time-step embedding, it takes two images as input, X_t and Y, each with shape $(32, 32, 3)$, which are concatenated along the channel dimension to give an input shape of $(32, 32, 6)$. The output has the same shape, but we discard the last three channels during training. The details of network architecture and hyperparameters are presented in table 5.

We apply DiffEM with $K = 21$ iterations to train our conditional diffusion model and evaluate its performance for the posterior sampling task as described in section 4.2. To evaluate the quality of the reconstructed data distribution, we also train an unconditional diffusion model with the same architecture on the reconstructed data. We compute the Inception Score (IS) (Salimans et al., 2016) and the Fréchet Inception Distance (FID) (Heusel et al., 2017) using the torch-fidelity package (Obukhov et al., 2021), and $\text{FD}_{\text{DINOv2}}$ (Oquab et al., 2023; Stein et al., 2023) and FD_∞ (Chong & Forsyth, 2020) using the codebase from (Stein et al., 2023). The results are presented in table 1 and table 2. We also note that the results of EM-MMPS are obtained with 32 iterations, following the original setup of Rozet et al. (2024). To compare the computational cost with EM-MMPS, we report the cost for different levels of performance in Figure 3. Furthermore, we report the values described in eq. (11) in table 6.

As an illustration, we also plot the evolution of the different metrics during DiffEM iterations in Figure 7 and Figure 8, demonstrating that DiffEM monotonically improves the quality of the reconstructed data distribution, in accordance with our theoretical results (lemma 1).

Experiments with higher corruption In addition, we perform experiments on CIFAR-10 with corruption probability $\rho = 0.9$ (i.e., 90% of the pixels are randomly deleted) and present the results in table 7. Under such high corruption, DiffEM also consistently outperforms EM-MMPS (Rozet et al., 2024) on all metrics.

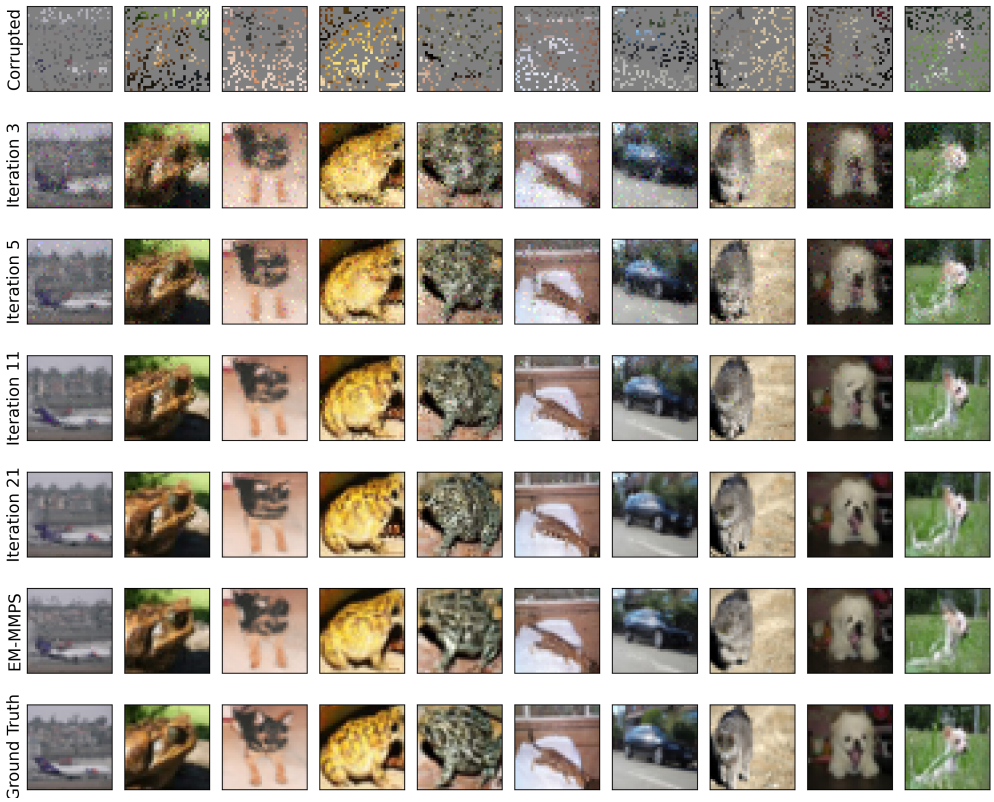

Figure 6: Qualitative comparison of reconstruction results on masked CIFAR-10 images. Top to bottom: corrupted input, EM-MMPS reconstructions, DiffEM reconstructions, and ground truth.

Experiment	CIFAR-10	CelebA
Architecture	U-Net	U-Net
Input Shape	(32, 32, 6)	(64, 64, 6)
Channels Per Level	(128, 256, 384)	(128, 256, 384, 512)
Attention Heads per level	(0, 4, 0)	(0, 0, 0, 4)
Hidden Blocks	(5, 5, 5)	(3, 3, 3, 3)
Kernel Shape	(3, 3)	(3, 3)
Embedded Features	256	256
Activation	SiLU	SiLU
Normalization	LayerNorm	LayerNorm
Optimizer	Adam	Adam
Initial Learning Rate	2×10^{-4}	1×10^{-4}
Final Learning Rate	1×10^{-6}	1×10^{-6}
Weight Decay	0	0
EMA	0.9999	0.999
Dropout	0.1	0.1
Gradient Norm Clipping	1.0	1.0
Batch Size	256	256
Epochs per EM iteration	256	64
Sampler	DDPM	DDPM

Table 5: Network architecture and training hyperparameters for the U-Net models used in the CIFAR-10 and CelebA experiments. Input shape varies by task.

Method	K	T_{init}	T_{ft}	T_{u}
EM-MMPS	32	43.0 ± 0.8	86.3 ± 0.7	N/A
DiffEM	21	63.5 ± 0.4	70.3 ± 0.2	74.54 ± 0.09

Table 6: Comparison of computation time (cf. section 2.2), with $T_{\mathsf{init}}, T_{\mathsf{ft}}, T_{\mathsf{u}}$ measured in minutes using 4× H200. The cost of EM-MMPS (Rozet et al., 2024) can similarly be decomposed as $T_{\mathsf{init}} + K \cdot T_{\mathsf{ft}}$ (it does not incur the cost T_{u}). As shown, DiffEM is more computationally efficient.

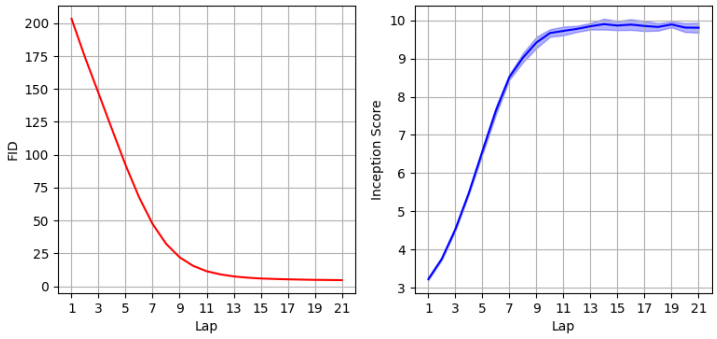

Figure 7: Evolution of evaluation metrics for posterior sampling measured during DiffEM training on CIFAR-10 with random masking. Left: FID, Right: Inception Score.

B.3 DiffEM with warm-start

We plot the evolution of IS, FID, $\mathrm{FD}_{\mathrm{DINOv2}}$, and FD_{∞}scores versus EM iterations during training in Figure 9.

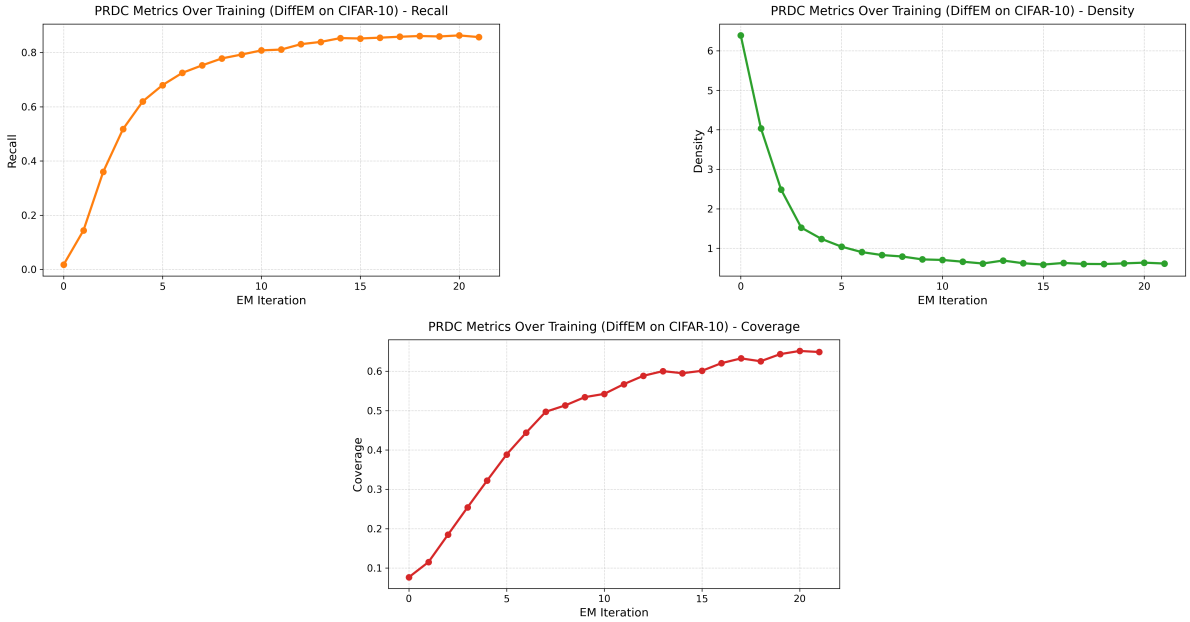

Figure 8: Evolution of Recall, Density, and Coverage metrics for posterior sampling over EM iterations of DiffEM training on CIFAR-10 with random masking.

Task	Method	IS ↑	FID ↓	$\text{FD}_{\text{DINOv2}}$ ↓	FD_{∞} ↓
Posterior sampling	EM-MMPS	5.06	67.97	1045.51	1039.82
	DiffEM	**5.86**	**46.13**	**915.69**	**912.26**
Unconditional generation	EM-MMPS	4.86	73.34	1174.13	1168.66
	DiffEM	**5.46**	**49.10**	**1111.16**	**1107.64**

Table 7: Performance of DiffEM and EM-MMPS on CIFAR-10 with 90% random masking.

B.4 Details of Blurred CIFAR-10

In the Gaussian blur model, each observation $Y \sim \mathsf{N}\big(AX, \sigma_Y^2\big)$ is generated by applying a Gaussian blur kernel to X with standard deviation σ_{kernel} (represented by the matrix A), and then adding isotropic Gaussian noise $\epsilon \sim \mathsf{N}\big(0, \sigma_Y^2 \mathbf{I}\big)$.

In the experiment on CIFAR-10 with Gaussian blur, we set $\sigma_{\text{kernel}} = 2$ and $\sigma_Y^2 = 10^{-6}$. We apply DiffEM for $K = 21$ iterations, with the same initialization, denoiser network architecture, and hyperparameters as in the masked CIFAR-10 experiment (detailed in table 5 and section B.2). Due to time constraints, we do not evaluate EM-MMPS (Rozet et al., 2024), as the moment-matching steps (based on the conjugate gradient method) are very time-consuming in this setting. For comparison, we use the Richardson-Lucy deblurring algorithm (Richardson, 1972) as a baseline, which is a widely used method for image deconvolution. Since Richardson-Lucy only performs posterior sampling and cannot generate unconditionally, we train an unconditional diffusion model on its reconstructions of the corrupted data to enable a fair comparison of unconditional generation performance.

Qualitative study To evaluate the quality of the trained conditional model, we sample a set of blurred images from the CIFAR-10 training set and use the trained model to generate a reconstruction for each image. We present the images in Figure 10.

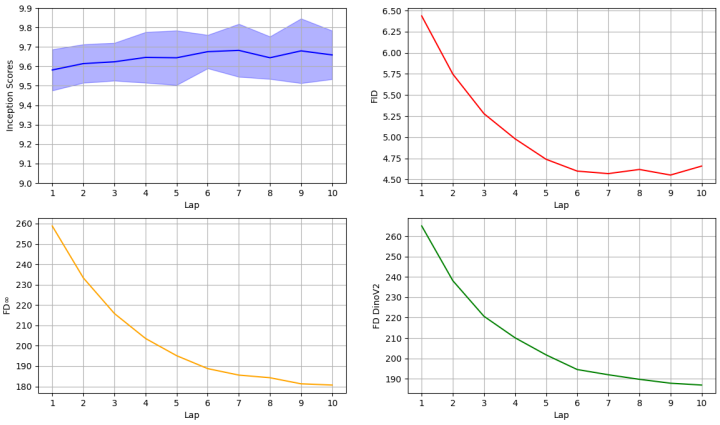

Figure 9: Evolution of IS, FID, DINO, and FD_∞ during the 10 DiffEM iterations with the warm-started prior.

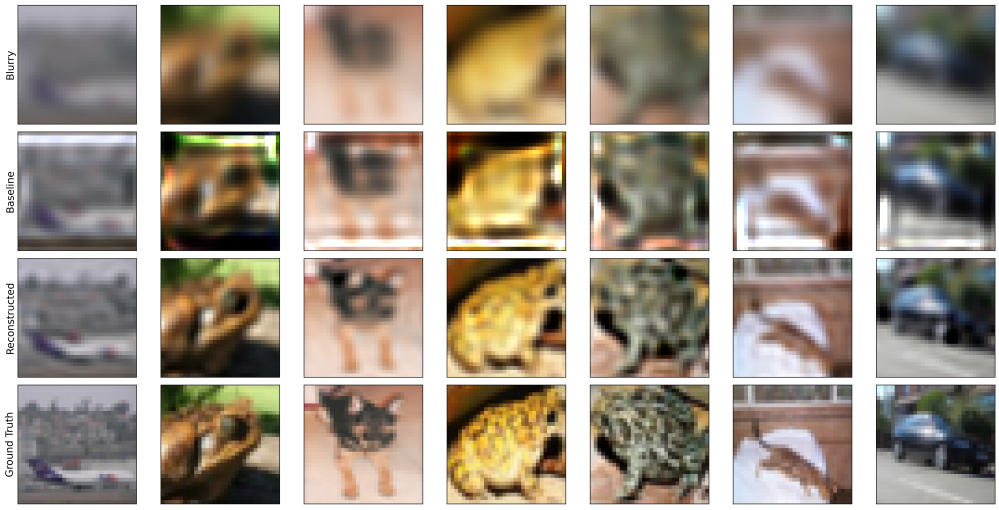

Figure 10: Qualitative results of image reconstruction from Gaussian blur. Top to bottom: blurred image, reconstruction by Richardson-Lucy deconvolution, image reconstructed by DiffEM model, and ground truth. DiffEM effectively recovers image details.

Quantitative comparison For both the baseline and our method, we compute the standard metrics as in previous experiments and report them in table 8. We also plot the evolution of IS and FID during DiffEM iterations in Figure 11, which shows the monotonic improvement consistent with previous experiments.

Task	Method	IS ↑	FID ↓	FD_{DINOv2} ↓	FD_∞ ↓
Posterior Sampling	Richardson-Lucy deconvolution	3.76	131.74	1479.79	1470.78
	DiffEM (Ours)	**6.12**	**43.65**	**404.05**	**400.65**
Unconditional Generation	Richardson-Lucy deconvolution	3.87	135.62	1688.42	1685.31
	DiffEM (Ours)	**11.27**	**51.25**	**772.23**	**768.19**

Table 8: Performance on CIFAR-10 with Gaussian blur ($\sigma_{\text{kernel}} = 2$).

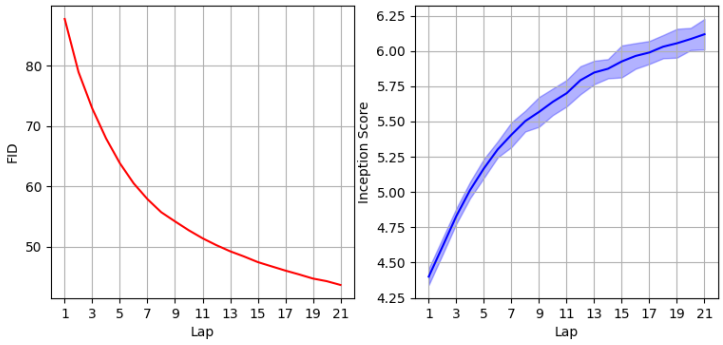

Figure 11: Evolution of evaluation metrics for posterior sampling measured during DiffEM training on CIFAR-10 with Gaussian blur. **Left**: FID, **Right**: Inception Score.

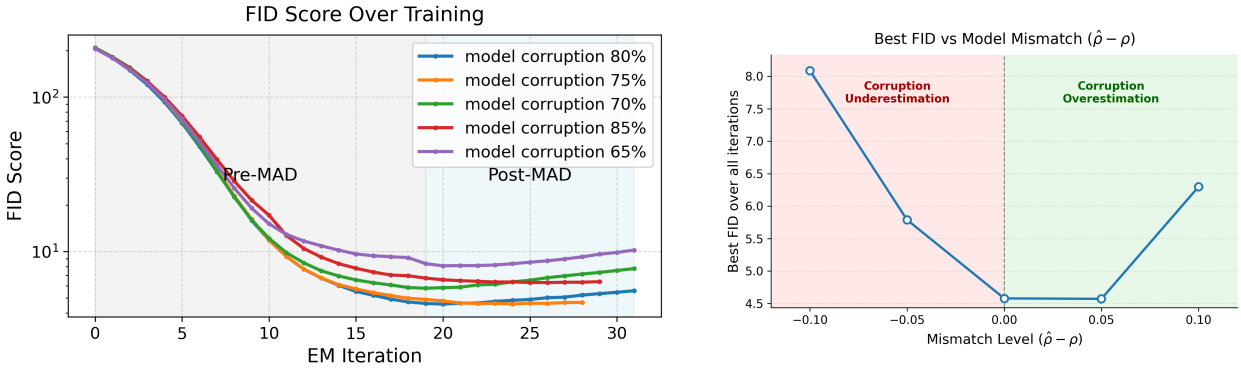

Figure 12: Model corruption mismatch experiments (see section B.5), with the true corruption level fixed at 75%. **Left:** FID across EM iterations for models assuming different corruption levels. **Right:** Minimum FID achieved over all EM iterations as a function of the assumed corruption level. The degradation is strongly asymmetric: underestimating the corruption level is substantially more harmful than overestimating it.

B.5 Corruption Model Mismatch

In many real-world settings, the corruption channel is not known exactly. Instead, one typically works with an estimate $\hat{Q}(\cdot \mid X)$ rather than the true corruption channel $Q(\cdot \mid X)$. Notably, all of our experiments and those in prior work (Rozet et al., 2024; Daras et al., 2023b; 2025b) assume access to the exact corruption channel.

In this section, we investigate the more realistic scenario in which the corruption model assumed during training is misspecified relative to the true one. Concretely, we use CIFAR-10 and apply random masking with true corruption probability $\rho = 0.75$ to generate the observations. However, during training we assume a mismatched corruption probability $\hat{\rho} = \rho + \frac{\delta\rho}{100}$. For $\delta\rho \in \{-10, -5, 0, 5, 10\}$, we train and evaluate DiffEM to study its robustness under corruption-model misspecification. As shown in Figure 12, slightly overestimating the corruption probability (which trains the model on a harder task) yields better results than slightly underestimating it.

B.6 Non-linear Discrete Corruption

In this section, we investigate a corruption function that is neither linear nor continuous, but instead exhibits inherently discrete behavior. A canonical example of such corruption is JPEG compression. JPEG applies a sequence of nonlinear operations—including blockwise discrete cosine transforms (DCT), quantization, and rounding—which introduce structured, non-Gaussian artifacts that cannot be modeled as additive noise.

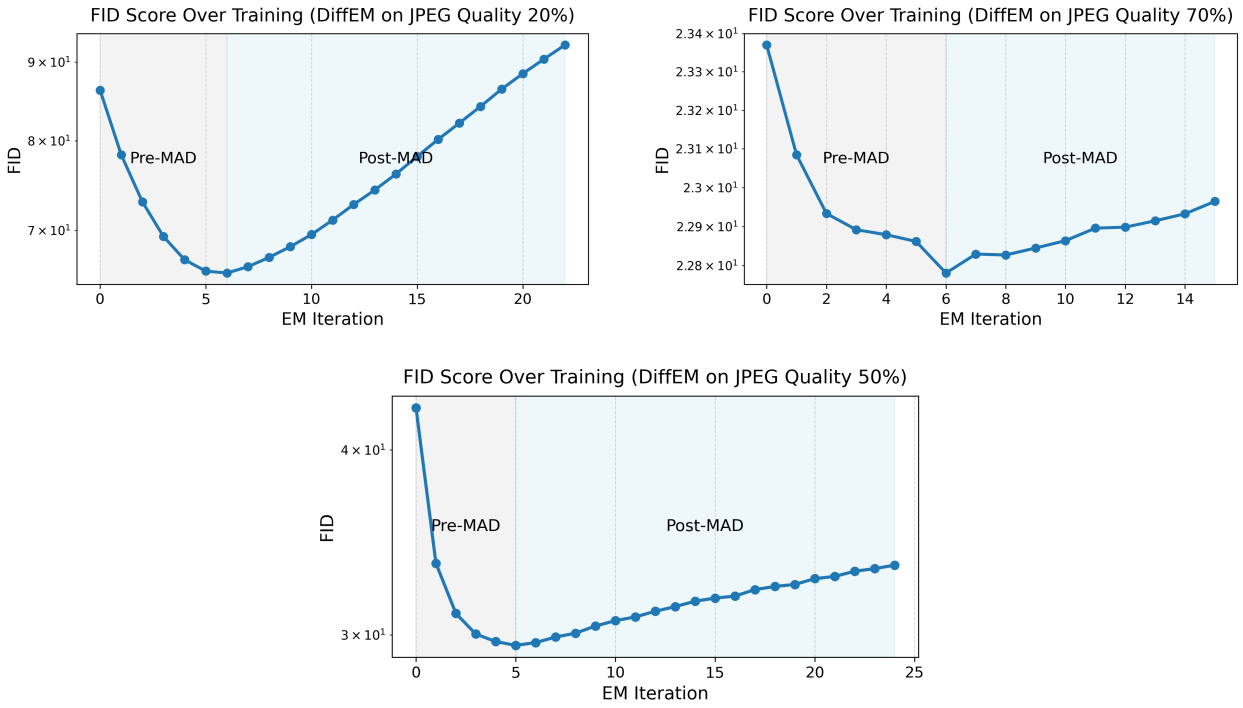

Figure 13: Evolution of FID for DiffEM's conditional model under JPEG corruption on CIFAR-10, for three compression quality levels. **Top left:** quality 20%. **Top right:** quality 70%. **Bottom:** quality 50%. More details are provided in section B.6.

This setting is especially relevant, as many real-world image pipelines (e.g., internet images, mobile devices, and storage-limited datasets) rely heavily on JPEG or similar codecs.

To study the effect of such discrete corruption, we compress and decompress all CIFAR-10 images using JPEG with a quality factor of 20%. At this low quality level, the quantization step is extremely aggressive, removing a substantial portion of the high-frequency content and producing severe compression artifacts. In practice, this corruption destroys a significant amount of the original information in the dataset. We train our diffusion models directly on these JPEG-compressed images to evaluate the robustness of our method under realistic, non-smooth corruptions.

The results in Figure 13 indicate that under this high level of corruption the model converges rapidly and exhibits the MAD (Model Autophagy Disorder) effect much earlier than in our other experiments. Further discussion of MAD is provided in section B.7. Notably, MAD can have a pronounced impact especially under stronger corruption: at JPEG quality 20%, performance after sufficient EM iterations may degrade significantly, with the model at iteration 21 performing worse than after a single iteration.

B.7 MAD: Model Autophagy Disorder

We consistently observe the MAD effect (Alemohammad et al., 2023) across nearly all of our experiments when the EM procedure is continued for sufficiently many iterations. In the case of CIFAR-10 with random masking at corruption level $\rho = 0.75$, evaluating the conditional model after each EM iteration yields the behavior shown in Figure 14. Figure 13 demonstrates the MAD effect for JPEG corruption under three different compression qualities (20%, 50%, and 70%). The results suggest that stronger corruption leads to more pronounced MAD behavior.

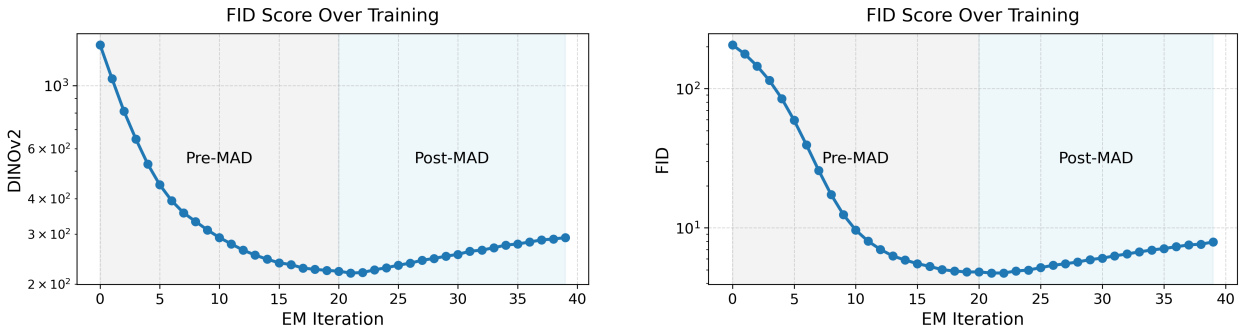

Figure 14: Evolution of $\mathrm{FD}_{\mathrm{DINOv2}}$ and FID across EM iterations for the conditional model on CIFAR-10 with $\rho = 0.75$ and $N = 128$ sampling steps. After an initial phase of improvement, both metrics gradually degrade over later iterations.

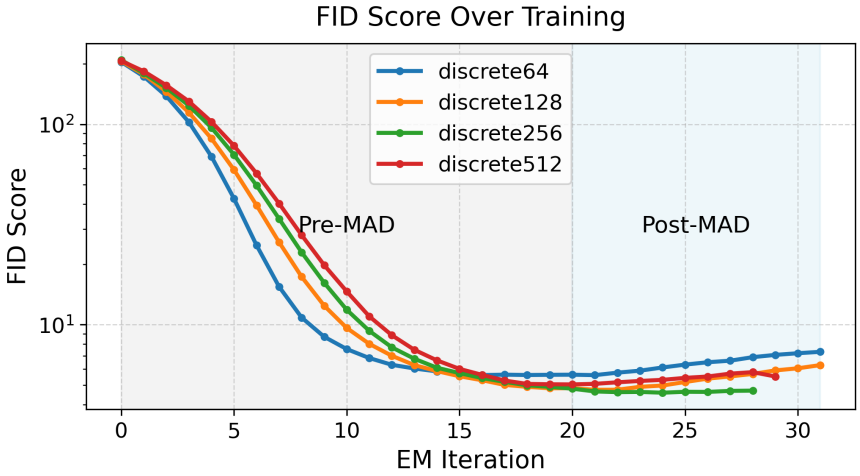

Figure 15: Evolution of the conditional model's performance across EM iterations under different discretization choices for the CIFAR-10 masking experiment with $\rho = 0.75$.

B.8 Analysis of Discretization Error

In section 3, we decomposed the score-matching error $\varepsilon_{\mathrm{KL}}^{(k)}$ into a discretization error and a learning error. In this section, we examine how the model's performance varies under different discretization choices. We train on randomly masked CIFAR-10 with corruption probability $\rho = 0.75$, and for discretization step counts $N \in \{64, 128, 256, 512\}$, we train the model and evaluate it after each EM iteration. The resulting performance curves are shown in Figure 15.

We observe that runs with fewer sampling steps perform better in early EM iterations, while runs with more sampling steps dominate in later iterations, with the performance gap gradually narrowing over time. This is consistent with the decomposition of the score-matching error $\varepsilon_{\mathrm{KL}}^{(k)}$ into a discretization error and a learning error. In early iterations, the learning error is large, making its accumulation over many sampling steps costly; it is therefore preferable to use fewer steps, accepting a larger discretization error in exchange for a smaller accumulated learning error. In later iterations, the learning error becomes negligible, so many sampling steps incur little penalty from learning error accumulation, whereas too few steps would introduce a significant discretization error. Beyond accuracy, this analysis has a direct practical implication: using fewer sampling steps in early iterations substantially reduces the cost of the E-step in each EM iteration, yielding a faster overall training pipeline with better final performance.

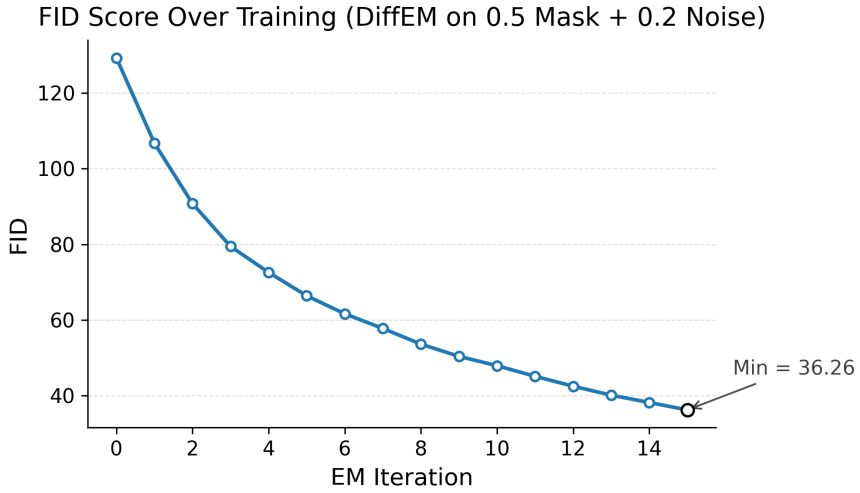

Figure 16: FID evolution of the conditional model over EM iterations for the experiment described in section B.9. The dataset is corrupted by sampling $A(X + \sigma N)$ where $A_{ij} \sim \mathrm{Ber}(0.5)$, $N \sim \mathcal{N}(0, I)$, and $\sigma = 0.2$. Pixel values are normalized to $[-2, 2]$.

Task	ρ	Method	IS ↑	FID ↓	$\mathrm{FD}_{\mathrm{DINOv2}}$ ↓	FD_{∞} ↓
Posterior sampling	0.5	EM-MMPS	3.237	0.61	9.36	6.07
		DiffEM	**3.239**	**0.33**	**5.07**	**2.07**
	0.75	EM-MMPS	2.96	31.22	113.09	109.41
		DiffEM	**3.16**	**1.43**	**39.34**	**36.26**
Unconditional generation	0.5	EM-MMPS	2.50	11.44	**186.16**	**182.90**
		DiffEM	**2.52**	**10.11**	344.60	340.97
	0.75	EM-MMPS	2.35	61.40	**321.90**	**319.58**
		DiffEM	**2.50**	**10.75**	423.95	420.76

Table 9: Performance of DiffEM and EM-MMPS (Rozet et al., 2024) on masked CelebA with masking probability $\rho \in \{0.5, 0.75\}$.

B.9 Random Masking with Gaussian Noise Corruption

We also evaluate our method under a mixed corruption model combining additive Gaussian noise with masking. Specifically, we run the CIFAR-10 experiment with Gaussian noise of standard deviation $\sigma = 0.2$ and a masking probability of $\rho = 0.5$, which is milder than the high-corruption settings considered earlier ($\rho = 0.75$ or $\rho = 0.9$). Note that pixel values are supported on $[-2, 2]$, so $\sigma = 0.2$ corresponds to a meaningful noise level relative to the signal range. The resulting corruption channel is

$$Q(Y \mid X) = A\big(X + \sigma Z\big), \quad Z \sim \mathcal{N}(0, I), \quad A_{ij} \sim \mathrm{Ber}(0.5),$$

where A denotes the random masking matrix. Qualitative samples are shown in Figure 17, and the evolution of evaluation metrics across EM iterations is presented in Figure 16.

B.10 Masked CelebA

As a demonstration, we sample seven masked images from the CelebA training set under the 75% corruption setting. Using the trained model, we generate reconstructions for each image after the 1^{st}, 2^{nd}, 4^{th}, 8^{th}, and 16^{th} iterations. The results are shown in Figure 18. The denoiser architecture is detailed in table 5. For the 50% corruption setting, we trained the conditional diffusion model for 20 EM iterations, while for the 75% corruption setting we trained it for 24 iterations. In both cases, we trained EM-MMPS for 9 iterations,

Figure 17: Samples from the experiment in section B.9. **Top:** corrupted inputs from the dataset. **Bottom:** reconstructions generated by the conditional model.

since each of its iterations is computationally expensive. The computational overhead of Moment Matching Posterior Sampling becomes particularly evident in this experiment, as the CelebA dataset is larger (202,599 images) and each image is higher-dimensional (64×64) compared to CIFAR-10. Evaluation results for the learned distribution are reported in table 9. We observe that each EM iteration of EM-MMPS required 4.85 ± 0.02 hours, whereas each iteration of DiffEM required 1.19 ± 0.03 hours.

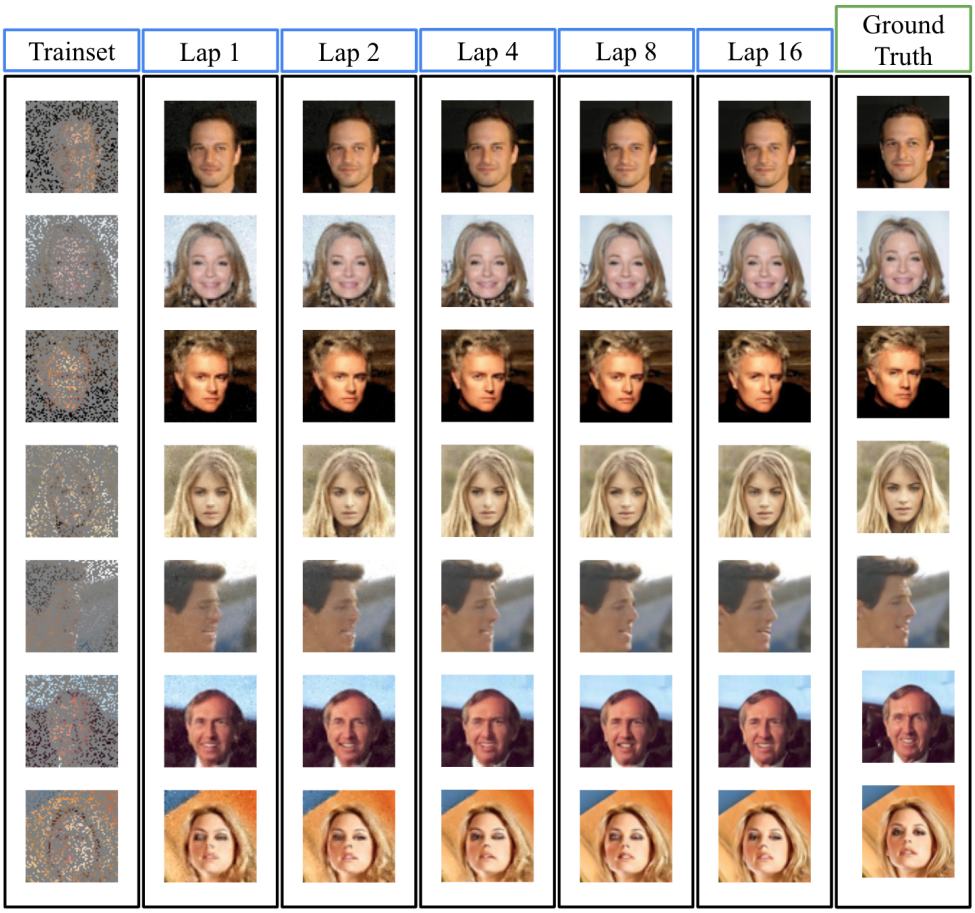

Figure 18: Qualitative results on the CelebA experiment under the 75% corruption setting. The leftmost column shows corrupted inputs. The subsequent columns display reconstructions generated by the conditional diffusion model after EM iterations $k = 1, 2, 4, 8, 16$. The rightmost column shows the ground-truth images.

Figure 19: Unconditional samples from the CelebA experiment with corruption probability $\rho = 0.5$.