EMAC-DIFFUSION: LEARNING CLEAN DISTRIBUTIONS FROM UNKNOWN CORRUPTIONS VIA EXPECTATION MAXIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce EMAC, an expectation-maximization (EM) algorithm for learning a clean data distribution from a small clean dataset and abundant observations corrupted by an unknown channel. Our method introduces an auxiliary-corruption channel that enables us to train a conditional diffusion model using EM without access to the original corruption channel. We apply this channel to both clean and already-corrupted images in the dataset and impose specific constraints on its design. We theoretically and experimentally identify limitations of prior work in handling low-frequency corruptions and we establish convergence guarantees for our method under appropriate assumptions. Experimentally, we first validate our method in controlled experiments, show how it can be used in the context of a privacy-related application, and finally test it at scale by improving imperfect data that naturally occur in ImageNet at 512×512 resolution. Qualitative and quantitative analyses show that our method improves over previously proposed baselines on several image-quality metrics.

1 INTRODUCTION

Recent progress in diffusion models has been driven by improved learning methods (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021) and curated, web-scale datasets (Schuhmann et al., 2022; Gadre et al., 2023). These datasets are formed from heterogeneous sources, and filtering decisions can substantially affect model performance (Gadre et al., 2023; Goyal et al., 2024). For instance, while discarding degraded examples can improve training-data quality, it also removes potentially useful information about the world and leads to mode collapse (Daras et al., 2025c). This trade-off between high-quality generations and diversity in the training set becomes particularly important in domains where obtaining high-quality data is expensive while imperfect data is readily available. We therefore ask: **how can abundant imperfect data improve a generative model when only a small clean dataset is available?**

Many prior works have studied instantiations of this question for which the corruption model is known or can be simulated. AmbientGAN (Bora et al., 2018) pioneered this direction for GAN-based generative models and follow-up works generalized the techniques to other modeling classes (Kelkar et al., 2023; Givens et al., 2025). In the context of diffusion, three main lines of work have been proposed: (a) Ambient based methods (Daras et al., 2023; 2025c; 2024; Rodriguez-Munoz et al., 2026; Daras et al., 2025a) that are based on the idea of additional corruption, (b) Expectation-Maximization based approaches (Bai et al., 2024; Rozet et al., 2024; Lu et al., 2025c;a;b; Modi et al., 2026), and (c) techniques based on Stein's Unbiased Risk Estimate (SURE) (Aali et al., 2023; Kawar et al., 2024). One weakness of many of these works is that they assume that the degradation channel is known or can be simulated. This limits the applicability of these approaches to certain applications (e.g., MRI), but prevents a broader adoption of the techniques in domains where more complex or unknown degradation forms appear (e.g., protein folding).

We study generative modeling from a small clean dataset and a large collection of observations corrupted by an *unknown* channel. Our contributions are as follows:

Method. We introduce an EM-based algorithm for learning a clean distribution from a small clean dataset and abundant observations corrupted by an unknown channel.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

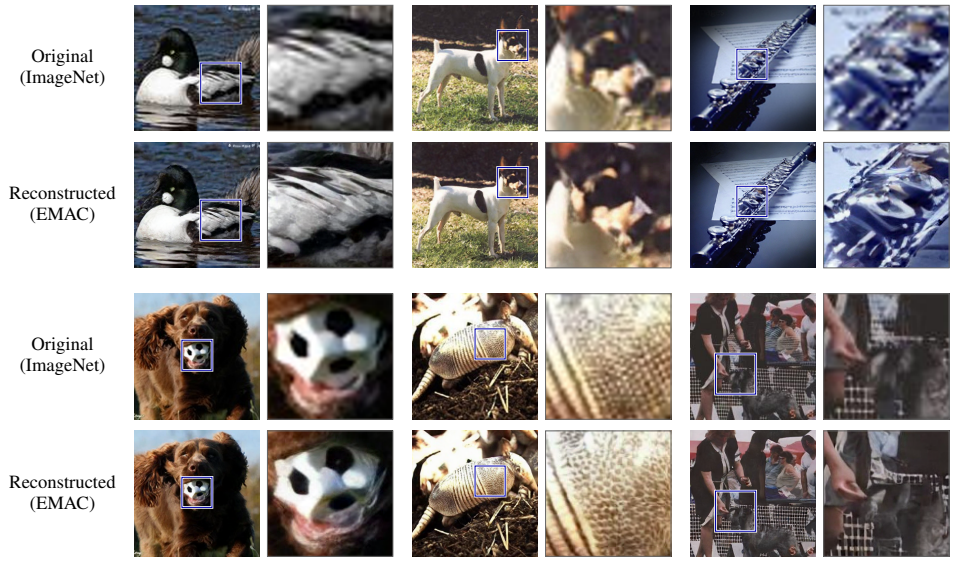


Figure 1: ImageNet-512 examples from Section 5.2. Each example shows an original image (top) and its EMAC reconstruction (bottom), with close-ups of the boxed regions to their right.

Theory.

- We derive lower bounds on the Gaussian noise required to bring clean and corrupted distributions close, helping assess how much information Ambient Omni discards from corrupted data through high-noise supervision.
- We establish convergence to the clean distribution and characterize the convergence rate under appropriate assumptions.

Experiments.

- Controlled CIFAR-10 corruptions to investigate auxiliary-channel design.
- Face-masked CelebA to evaluate learning from censored images.
- ImageNet at 512×512 resolution to evaluate learning from mixed-quality data without additional training images.

2 Preliminaries

2.1 Diffusion Models

We use a variance-exploding formulation in which Gaussian perturbations connect denoising to score estimation. Let $X_0 \sim p_0$ denote a sample from the target distribution. At each fixed time t, the noisy marginal admits the representation

$$X_t = X_0 + \sigma(t)Z, \qquad Z \sim \mathcal{N}(0, I_d), \qquad t \in [0,1], \tag{1}$$

where Z is independent of X_0, and σ is nondecreasing with $\sigma(0) = 0$ and $\sigma(t) > 0$ for $t > 0$. The noisy density is $p_t = p_0 \circledast \mathcal{N}(0, \sigma^2(t) I_d)$. We learn a denoiser $h_\theta(x,t)$ across noise levels. We sample $t \sim \pi$ independently of (X_0, Z), where π is the chosen training distribution on $(0,1]$, and minimize the weighted denoising objective (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021):

$$\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathbb{E}_{t,X_0,Z}\left[\lambda(t)\,\|h_\theta(X_0 + \sigma(t)Z, t) - X_0\|_2^2\right], \tag{2}$$

where $\lambda(t) > 0$ weights different noise levels. For $t > 0$, replacing the conditional mean $\mathbb{E}[X_0 \mid X_t = x]$ in Tweedie's exact formula with the learned denoiser $h_\theta(x,t)$ gives the score approximation $\nabla_x \log p_t(x) \approx (h_\theta(x,t) - x)/\sigma^2(t) = s_\theta(x,t)$. We use this score estimate to generate approximate samples from p_0 (Song et al., 2021).

2.2 Problem Setup

Let P_X denote the target clean distribution. We observe clean samples $\mathcal{D}_G$ drawn from P_X and corrupted samples $\mathcal{D}_B$ obtained by applying an unknown corruption channel $\mathbf{Q}$ to samples from P_X. We also refer to $\mathbf{Q}$ as the *dataset corruption*. The channel may be deterministic or stochastic. Formally,

$$\mathcal{D}_G = \{X^{(i)}\}_{i=1}^{n_G}, \quad X^{(i)} \overset{\text{i.i.d.}}{\sim} P_X, \qquad \mathcal{D}_B = \{Y^{(j)}\}_{j=1}^{n_B}, \quad Y^{(j)} \overset{\text{i.i.d.}}{\sim} P_Y := \mathbf{Q}_{\#}P_X, \quad (3)$$

The subscripts G and B stand for good (clean) and bad (corrupted) data, respectively. In Equation 3, the pushforward notation $P_Y = \mathbf{Q}_{\#}P_X$ denotes the distribution obtained by first sampling $X \sim P_X$ and then $Y \sim \mathbf{Q}(\cdot \mid X)$. We focus on the regime $n_G \ll n_B$, where clean samples are scarce relative to corrupted samples.

Our goal is to use both datasets to learn the clean-distribution score described in Section 2.1. Clean samples yield an unbiased estimate of the clean-data denoising objective, but the small dataset can lead to high variance. The larger corrupted dataset can reduce variance but generally introduces bias by targeting a different distribution. We seek to exploit corrupted data while retaining the clean distribution as the learning target.

2.3 Ambient Diffusion Omni

Ambient Diffusion Omni (Ambient-o) (Daras et al., 2025c;b) addresses the setting in Section 2.2. The key idea is to use corrupted data only for certain noise levels during the training of the diffusion model. In standard diffusion training, a clean sample X_0 is perturbed at different noise levels to obtain $X_t = X_0 + \sigma(t)Z$, which is used to predict X_0 through the denoising objective in Equation 2. Adding Gaussian noise with variance $\sigma^2(t)$ to samples from both the clean and corrupted distributions reduces the total variation distance between the resulting noisy distributions. For distributions with bounded support, Theorem 4.2 in the Ambient-o paper bounds this distance by a quantity proportional to $1/\sigma(t)$. This motivates using abundant corrupted samples at sufficiently high noise levels, where the distributional mismatch is reduced but not necessarily eliminated.

Ambient-o assigns each corrupted sample a fixed minimum noise level above which it can be used for training. This threshold can be chosen empirically and shared across samples, or assigned per sample using a separately trained model. Training on these samples requires an objective that does not use their unobserved clean targets. Unlike the standard denoising loss in Equation 2, Ambient-o uses the Gaussian ambient-denoising construction of Daras et al. (2024), adapted to observations at their assigned noise levels. We give the full objective in Appendix A.

2.4 Expectation-Maximization methods

When the corruption channel $\mathbf{Q}$ is known or can be estimated, expectation-maximization (EM) provides an alternative approach to learning a clean distribution from corrupted observations. The algorithm of Bai et al. (2024) initializes a diffusion model using a small clean dataset, then alternates between two steps:

E-step: for each observation $Y^{(i)}$, approximately sample a clean reconstruction $\hat{X}^{(i)}$ from $p_\theta(X \mid Y^{(i)})$ using the current diffusion model.

M-step: update the diffusion model $p_\theta(X)$ using the reconstructions $\{\hat{X}^{(i)}\}$ sampled in the E-step.

The method maintains an unconditional diffusion prior, represented by a learned score estimate $s_\theta(x_t, t)$. This network approximates $\nabla_{x_t} \log p_t(x_t)$, where p_t is defined as in Section 2.1. In the E-step, the method uses a variant of Diffusion Posterior Sampling (DPS) (Chung et al., 2023) to sample conditionally on the observations. This requires approximating the conditional score $\nabla_{x_t} \log p_t(x_t \mid y)$, which decomposes into an unconditional score and a likelihood-score term by Bayes' rule:

$$\nabla_{x_t} \log p_t(x_t \mid y) = \nabla_{x_t} \log p_t(x_t) + \nabla_{x_t} \log p_t(y \mid x_t). \quad (4)$$

The likelihood-score term can be written as $\nabla_{x_t} \log \int p(y \mid x_0)\, p(x_0 \mid x_t)\, \mathrm{d}x_0$, but is generally intractable. For observations $Y = Q(X_0) + \varepsilon$, where Q is a known deterministic corruption function

and $\varepsilon \sim \mathcal{N}(0, \sigma_y^2 I)$ is independent of X_0, Bai et al. (2024) use the approximation $p_t(y \mid x_t) \approx \mathcal{N}\big(y;\, Q\big(\mathbb{E}[X_0 \mid X_t = x_t]\big),\, \sigma_y^2 I\big)$. With the conditional mean approximated by the denoiser output, the second term on the right-hand side of Equation 4 therefore has a simple analytical approximation, evaluated by differentiating through the denoiser and the forward operator.

DiEM (Rozet et al., 2024) improves this estimate using $p_t(y \mid x_t) \approx \mathcal{N}(y; Q\mathbb{E}_\theta[X_0 \mid X_t = x_t], \Sigma_\theta(x_t, t))$, where the conditional mean and covariance are estimated from the learned model. It assumes a known corruption channel $\mathbf{Q}(X) = QX + \varepsilon$, where Q is linear and $\varepsilon \sim \mathcal{N}(0, \sigma_y^2 I)$ is independent of X.

DiffEM (Hosseintabar et al., 2025) extends this line of work by directly learning the conditional score $\nabla_{x_t} \log p_\theta(x_t \mid y)$ within the EM framework, enabling posterior sampling with a conditional diffusion model. Its E-step includes an additional step that uses black-box access to the corruption channel to corrupt the generated samples. It does not require a separate approximation of $p_t(y \mid x_t)$ like previous methods. It also accommodates nonlinear corruption, although it still requires a known corruption channel. Its stronger convergence guarantee for recovering the clean distribution requires an additional constraint on the corruption channel (Hosseintabar et al., 2025). Since DiffEM does not require clean samples, some restriction on the channel is necessary for unique recovery: otherwise, distinct clean distributions can produce the same observations, making the recovery problem ill-posed. Modi et al. (2026) and Lu et al. (2025c) address related settings with black-box corruption access using self-consistent stochastic interpolants and bridge models, respectively, with iterative structures similar to DiffEM.

3 Motivation: Limitations of High-Noise Supervision

We consider the problem formulated in Section 2.2. DiffEM cannot be applied directly because the corruption channel is unknown. In this section, we explain why Ambient Omni's performance may be suboptimal, motivating an alternative approach within the EM framework.

Overfitting in the low-noise regime. Ambient Omni relies on the small clean dataset for low-noise supervision, creating a risk of overfitting (Figure 2). We support this hypothesis with three complementary observations. First, Figure 9 illustrates this behavior in a toy experiment: the sampling trajectories bend sharply toward the clean training samples upon entering the low-noise regime. Details of the toy experiment are given in Appendix A. Second, to assess this behavior quantitatively, we followed the CIFAR-10 experiment setup in Ambient Omni (Daras et al., 2025c) and evaluated standard EDM loss on held-out clean images at six fixed noise levels during training. Unlike the Ambient objective used for optimization, this loss measures clean-data denoising performance. Figure 3 shows that the low-noise validation losses increase after initially decreasing, while higher-noise losses remain stable or continue to improve. This pattern is consistent with overfitting under limited low-noise supervision. Figure 2 also compares Ambient Omni and EMAC at different fixed noise levels using their denoising validation losses divided by the loss of an oracle EDM model trained on 100% clean data. This comparison is also consistent with Ambient Omni overfitting. Experimental details are given in Appendix A. Third, table 3 provides evidence from generation metrics: Ambient Omni has lower recall, coverage, and Vendi score than our method, indicating reduced diversity consistent with overfitting.

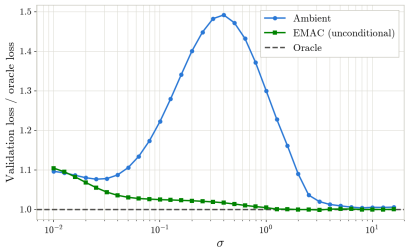


Figure 2: Validation loss relative to an oracle across noise levels for Ambient Omni and unconditional EMAC (Section 3).

Distributional alignment may require substantial noise. Even simple corruptions can require substantial Gaussian noise to bring clean and corrupted distributions close. At these noise levels, useful image structure may be obscured, limiting the contribution of corrupted data to learning.

Theorem 1 (Necessary noise level under a mean discrepancy)**.** *Let P and Q be distributions with finite second moments and mean gap $\Delta = \|\mu_P - \mu_Q\|_2 > 0$. Adding Gaussian noise with standard deviation σ gives $P_\sigma = P \circledast \mathcal{N}(0, \sigma^2 I)$ and $Q_\sigma = Q \circledast \mathcal{N}(0, \sigma^2 I)$. For the direction $u = (\mu_P - \mu_Q)/\Delta$, define $v_P = \mathrm{Var}_{X \sim P}(\langle u, X \rangle)$ and $v_Q = \mathrm{Var}_{Y \sim Q}(\langle u, Y \rangle)$, and set $M_2 = v_P + v_Q + \Delta^2/2$.*

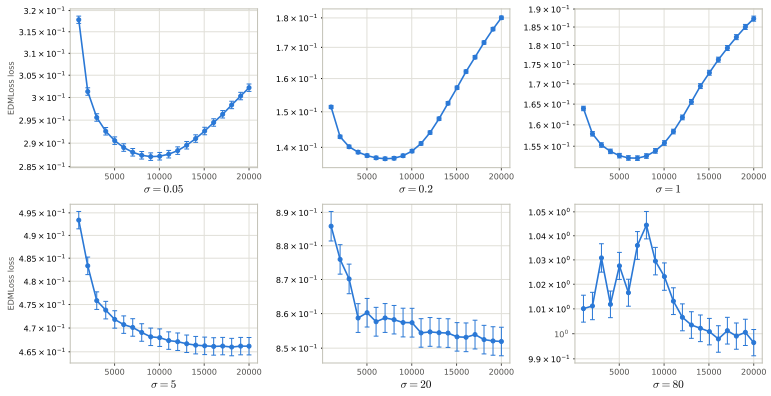


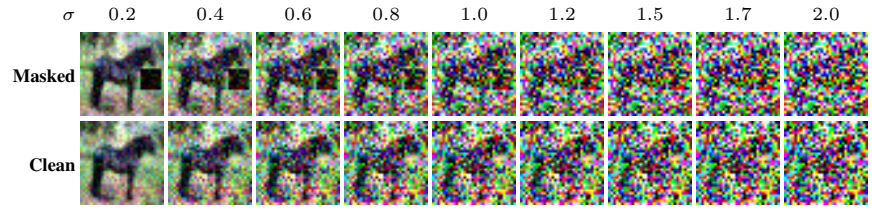
Figure 3: EDM validation loss evaluated every 1 Mimg during 20 Mimg of Ambient-o training, shown at six noise levels. The low-noise profiles turn upward late in training, whereas the higher-noise profiles remain stable or continue to improve. See Section 3 and Appendix A for discussion.

For $0 < \delta < 1$, making the resulting distributions δ-close in total variation requires at least the following noise level:

$$\mathrm{d_{TV}}(P_\sigma, Q_\sigma) \le \delta \quad \Longrightarrow \quad \sigma \ge \sqrt{\left[\frac{\Delta^2}{4\delta} - \frac{M_2}{2}\right]_+},$$

where $[a]_+ = \max\{a, 0\}$. A sharper high-noise bound further implies $\sigma = \Omega(\Delta/\delta)$ as $\delta \to 0$. See Theorem 2 in Appendix C.1 for the proof.

When the mean gap Δ is comparable to the distributions' standard deviations (more precisely, those along the direction of the mean difference), achieving a small TV distance (e.g., $\delta = 0.01$) can require a noise standard deviation σ comparable to or larger than these scales. Such noise can cause substantial loss of image information. Figure 4 illustrates this trade-off when the dataset corruption is a randomly positioned square mask of fixed size, which can introduce a mean discrepancy. In this experiment, the noised clean and corrupted distributions become close according to FID only at high noise levels. However, this high noise obscures image features, as indicated by the decrease in Inception Score.



FID	44.52	37.90	19.85	9.66	4.79	2.59	1.24	0.84	0.52
IS	4.22	2.61	1.94	1.66	1.49	1.40	1.32	1.29	1.26

Figure 4: Masked and clean images under increasing Gaussian noise (Section 3). FID compares the two noised image sets; IS is Inception Score. Both metrics use image sets, not the displayed pair.

4 METHOD

We introduce an EM-based method for learning the clean data distribution from abundant corrupted data and a small clean dataset, without access to the original corruption channel.

As a starting point, we recall how DiffEM (Hosseintabar et al., 2025) operates when the corruption channel is known. It requires no clean samples. In the E-step, the current conditional diffusion model $p_\theta(x \mid z)$ samples reconstructions of the corrupted observations. Applying the known channel $\mathbf{Q}$ to

these reconstructions produces new training pairs. The M-step updates the conditional model using these pairs, and the procedure repeats.

Because the dataset corruption is unknown, one idea is to introduce an auxiliary corruption channel $\widehat{\mathbf{Q}}$ and use it in the EM procedure instead of $\mathbf{Q}$. Not every auxiliary channel is suitable; we discuss the properties it should satisfy later in this section. We now describe the resulting EM procedure. In the E-step, we first apply $\widehat{\mathbf{Q}}$ to the dataset-corrupted observations $\{Y^{(j)}\}$, then sample reconstructions conditioned on the resulting inputs, and apply $\widehat{\mathbf{Q}}$ to the reconstructions to form training pairs. In the M-step, we train the conditional diffusion model on these pairs. We call this procedure EMAC-Diffusion, where EMAC stands for Expectation Maximization with Auxiliary Corruption. Algorithm 1 summarizes the procedure. If an unconditional model is desired, we train it on reconstructions of the dataset generated by the learned conditional model.

Algorithm 1 EMAC-Diffusion

Require: Clean dataset $\mathcal{D}_G = \{X^{(i)}\}_{i=1}^{n_G}$; Corrupted dataset $\mathcal{D}_B = \{Y^{(j)}\}_{j=1}^{n_B}$;
Auxiliary corruption channel $\widehat{\mathbf{Q}}$, initial parameters $\theta^{(0)}$;
EM iterations K, fine-tuning steps T, clean-pair probability schedule $\{\rho_k\}_{k=0}^{K-1}$.

for $k = 0, 1, \ldots, K-1$ **do**
  $\widetilde{\mathcal{D}}_B \leftarrow \{\overline{Y}^{(j)} : Y^{(j)} \in \mathcal{D}_B,\ \overline{Y}^{(j)} \sim \widehat{\mathbf{Q}}(\cdot \mid Y^{(j)})\}$
  // Generate reconstructions
  $\widehat{X}^{(j)} \sim p_{\theta^{(k)}}(\cdot \mid \overline{Y}^{(j)}), \qquad j = 1, \ldots, n_B$
  // Construct training pairs
  $\mathcal{P}^{(k)}_{\text{reconstructed}} \leftarrow \{(\widehat{X}^{(j)}, \widehat{Y}^{(j)}) : \widehat{Y}^{(j)} \sim \widehat{\mathbf{Q}}(\cdot \mid \widehat{X}^{(j)})\}$
  $\mathcal{P}^{(k)}_{\text{clean}} \leftarrow \{(X, \widehat{Y}) : X \in \mathcal{D}_G,\ \widehat{Y} \sim \widehat{\mathbf{Q}}(\cdot \mid X)\}$
  $\theta^{(k+1)} \leftarrow \theta^{(k)}$
  for $t = 1, 2, \ldots, T$ **do**
    $(X, Z) \sim \begin{cases} \mathrm{Unif}(\mathcal{P}^{(k)}_{\text{clean}}), & \text{with probability } \rho_k, \\ \mathrm{Unif}(\mathcal{P}^{(k)}_{\text{reconstructed}}), & \text{with probability } 1 - \rho_k \end{cases}$
    $\theta^{(k+1)} \leftarrow \textsc{SGDStep}(\theta^{(k+1)}; X, Z)$
  end for
end for
return $p_{\theta^{(K)}}(X \mid Z)$

E-step (lines 2–5) **M-step** (lines 6–10)

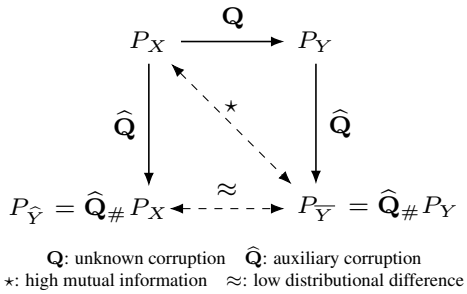
Let us view each EM iteration as a function f that takes the current parameters $\theta^{(k)}$ and returns the updated parameters $\theta^{(k+1)} = f(\theta^{(k)})$. A fixed point $\bar{\theta}$ satisfies $f(\bar{\theta}) = \bar{\theta}$, so further updates leave the model unchanged. This fixed-point view motivates our first requirement, the *distributional alignment constraint* (*alignment* for short): $\widehat{\mathbf{Q}}_{\#}P_X = \widehat{\mathbf{Q}}_{\#}P_Y$, which makes the auxiliary outputs of clean and corrupted inputs indistinguishable in distribution. To see why, consider the extreme case where the supports $\mathrm{supp}(\widehat{\mathbf{Q}}_{\#}P_X) = \mathcal{S}_X$ and $\mathrm{supp}(\widehat{\mathbf{Q}}_{\#}P_Y) = \mathcal{S}_Y$ are disjoint. For $X \sim P_X$ and $Y \sim P_Y$, let $\widehat{Y} \sim \widehat{\mathbf{Q}}(\cdot \mid X)$ and $\overline{Y} \sim \widehat{\mathbf{Q}}(\cdot \mid Y)$. A model can then learn

$$p_{\bar{\theta}}(\cdot \mid z) = \begin{cases} P_{X \mid \widehat{Y}}(\cdot \mid z), & z \in \mathcal{S}_X, \\ P_{Y \mid \overline{Y}}(\cdot \mid z), & z \in \mathcal{S}_Y. \end{cases} \tag{5}$$

Under exact sampling and fitting, this is an undesirable fixed point: the E-step reproduces P_Y rather than P_X, and the M-step preserves these separate conditionals instead of learning to recover clean images. We therefore seek indistinguishable auxiliary-output distributions, so the model cannot tell whether an input originated from the clean or corrupted dataset. In corollary 1, we prove that, under the stated assumptions, alignment gives EM a unique conditional fixed point: the distribution of clean samples given their auxiliary-corrupted observations. Figure 5 illustrates the alignment constraint.

Our second requirement is *information preservation*. Mapping every image to zero satisfies alignment, but makes reconstruction independent of the corrupted observations. We therefore seek high input–output mutual information subject to alignment. The resulting optimization problem in Equa-

tion 6 guides our channel selection; we do not solve it directly, but explore empirically how to design suitable auxiliary corruptions.



$$\widehat{\mathbf{Q}}^* = \arg\max_{\widehat{\mathbf{Q}}} I_{X \sim P_X}\Big(X; \widehat{\mathbf{Q}}(X)\Big) \quad \text{subject to} \quad \widehat{\mathbf{Q}}_{\#} P_X = \widehat{\mathbf{Q}}_{\#} P_Y. \tag{6}$$

Figure 5: Auxiliary-channel design in Section 4: the distributional diagram (left) and the optimization problem (right, Equation 6). Solid arrows denote corruption channels; dashed arrows indicate the information-preservation and distributional-alignment goals. Notation follows Section 2.2.

5 Experiments

5.1 CIFAR-10 experiments

We begin with controlled experiments on CIFAR-10, a small, well-established benchmark, to explore auxiliary-corruption design. Our goal in this section is to build intuition about alignment and information preservation that could guide us in designing auxiliary corruptions for later experiments. We construct a synthetically corrupted dataset following the setup of Ambient Omni (Daras et al., 2025c), applying Gaussian blur with kernel standard deviation $\sigma_B = 0.8$ to 90% of the images, leaving the remaining 10% clean. We denote Gaussian blur with kernel standard deviation σ by $\mathcal{B}_\sigma$. We compare against clean-only training, training on all available data, and Ambient Omni; tables 1 and 2 summarize the conditional and unconditional FID results, respectively. Full sweeps are in Appendix F. We evaluate four auxiliary corruption functions to identify the best-performing choice among them, with additional evaluation metrics reported in table 3.

Table 1: Conditional FID comparison for Section 5.1.

Auxiliary corruption	Best FID ↓
Masking	9.18
Gaussian blur	10.51
Gaussian noise	8.46
Gaussian random blur	**1.96**

Table 2: Unconditional FID comparison for Section 5.1.

Method	Best FID ↓
All-data baseline	28.26
Clean-only baseline	8.79
Ambient Omni	6.00
EMAC (Unconditional)	**3.76**

Random Masking and Gaussian Noise. We first consider independent pixel masking as an auxiliary corruption, masking each pixel with probability p_{mask}. The masking sweep in Figure 16 shows that higher masking probabilities generally improve FID, although none of the tested configurations outperforms the clean-only baseline. Blurring correlates neighboring pixels. Hiding this difference through masking may require high masking probabilities, which also discard information useful for reconstruction, highlighting the trade-off between alignment and information preservation. A similar trade-off arises when using additive Gaussian noise as the auxiliary corruption (Figure 18).

Table 3: Additional CIFAR-10 evaluation metrics for Omni and EMAC (Section 5.1).

Method	Coverage	Density	Precision	Recall	Vendi score
Omni	0.32	**0.87**	0.70	0.48	200.72
EMAC	**0.49**	0.76	**0.72**	**0.61**	**211.27**

Gaussian Blur. Seeking to preserve more information than masking, we next consider fixed Gaussian blur as the auxiliary channel, $\widehat{\mathbf{Q}}(X) = \mathcal{B}_\sigma(X)$. In the invertible case, the model can realize the undesirable fixed point in Equation 5 simply by learning $\mathcal{B}_\sigma^{-1}$: applied to $\mathcal{B}_\sigma(Y)$, it returns

the corrupted observation Y rather than a clean reconstruction. Its best conditional FID is 10.508 (Figure 17), worse than the clean-only baseline. This motivates stochastic auxiliary corruptions.

Random Gaussian Blur. Randomizing the blur strength removes the single fixed inverse available with deterministic blur, encouraging the model to learn a clean-data prior for reconstruction. For each input image X, we independently sample $\sigma \sim \mathcal{U}[\sigma_{\min}, \sigma_{\max}]$ and output $\mathcal{B}_\sigma(X)$, without supplying σ to the conditional model. In the sweep reported in Figure 15, randomized blur achieves a best conditional FID of 1.964, outperforming the other tested auxiliary-channel families.

5.2 IMAGENET EXPERIMENT

We evaluate EMAC on ImageNet, where low-quality images occur naturally rather than being produced by synthetic corruption (Figure 22). To define the clean and corrupted subsets, we rank images using CLIP-IQA (Wang et al., 2023), which assigns each image a quality score in $[0, 1]$, with higher scores indicating better predicted quality. We treat the highest-scoring 10% as clean and the remaining 90% as corrupted.

Table 4: Image-quality scores for real data and model outputs in the ImageNet experiment (Section 5.2), computed using 50,000 randomly selected samples.

Metric	Real data	EDM2	Ambient Omni	EMAC (unconditional)	EMAC (conditional)
`arniqa`	0.63 ± 0.12	0.65 ± 0.11	0.64 ± 0.11	$\mathbf{0.69 \pm 0.06}$	0.68 ± 0.07
`fgresq`	0.52 ± 0.05	0.51 ± 0.05	0.51 ± 0.05	$\mathbf{0.51 \pm 0.03}$	0.51 ± 0.04
`topiq_nr`	0.50 ± 0.12	0.48 ± 0.12	0.46 ± 0.12	$\mathbf{0.50 \pm 0.09}$	0.52 ± 0.09
`qalign`	3.55 ± 0.73	3.29 ± 0.72	3.16 ± 0.70	$\mathbf{3.30 \pm 0.52}$	3.43 ± 0.54

We design the ImageNet auxiliary channel to preserve information while reducing the distributional mismatch between clean and corrupted inputs. Our CIFAR-10 experiments (Section 5.1) suggest that randomization can encourage learning a data prior, whereas an invertible auxiliary mapping can create a reconstruction shortcut.

Table 5: Mean and median CLIP-IQA scores for the ImageNet experiment.

Method	Mean ↑	Median ↑
EDM2	0.69	0.79
Ambient-o	0.71	0.80
EMAC (unconditional)	**0.77**	**0.81**
EMAC (conditional)	0.80	0.86

Guided by these observations, we use randomized Gaussian blur followed by quantization. Blurring preserves image structure, but a fixed, invertible blur can be undone without learning a data prior. We therefore sample the blur level as $\sigma \sim \mathcal{U}[2, 4]$ to discourage reliance on a fixed inverse. We then quantize pixel values to multiples of 10 on the $[0, 255]$ scale to reduce the effect of small differences in pixel values.

We implement EMAC using the EDM2 codebase (Karras et al., 2024). After EM training, we reconstruct the corrupted subset with the conditional model. We then train an unconditional model on these reconstructions, starting from a pretrained EDM2 checkpoint. Figure 1 shows original images and their conditional reconstructions, and Figures 23 to 29 provide additional examples.

We evaluate image quality using pretrained no-reference models, which score each image without requiring a paired clean target. We use ARNIQA (Agnolucci et al., 2024), FGResQ (Sheng et al., 2026), TOPIQ-NR (Chen et al., 2023), and Q-Align (Wu et al., 2024). Table 4 reports the mean and standard deviation of their scores over 50,000 randomly selected images per evaluated set, comparing real ImageNet images, EDM2 and Ambient Omni generations, and EMAC's unconditional generations and conditional reconstructions. Table 5 separately reports mean and median CLIP-IQA scores.

5.3 CELEBA EXPERIMENT

Our CelebA experiment studies learning from censored images when the masking mechanism is unknown to the learner. We construct the censored dataset using the pretrained `jonathandinu/face-parsing` model (Dinu, 2022), based on SegFormer (Xie et al., 2021),

to mask facial regions in 90% of the images, leaving the rest clean. We assume that the corruption family (pixel masking) is known, but the rule selecting the masked pixels is not. We therefore use masking-based auxiliary channels, but the learner has no access to the segmentation model. The clean-only baseline achieves a best FID of 4.085 (Figure 13).

We first consider an auxiliary channel $\widehat{\mathbf{Q}}$ that applies a fixed circular mask of radius r at the image center (Figure 19). Let us examine how r affects the alignment and information-preservation constraints introduced in Section 4. A small radius masks fewer pixels and therefore preserves more information, as quantified by $I_{X\sim P_X}(X;\widehat{\mathbf{Q}}(X))$, but may leave a large distributional mismatch, $d_{\mathrm{TV}}(\widehat{\mathbf{Q}}_{\#}P_X, \widehat{\mathbf{Q}}_{\#}P_Y)$. This mismatch can arise because images from $\widehat{\mathbf{Q}}_{\#}P_X$ have circular masks, whereas those from $\widehat{\mathbf{Q}}_{\#}P_Y$ may have non-circular masks because the original face mask extends beyond the circle (Figures 7 and 11). A larger radius can cover these residual masked regions, reducing the mismatch at the cost of discarding more information. The radius sweep in Figure 6 is consistent with this trade-off: conditional FID improves up to $r = 25$, where it reaches 0.922, and worsens at the larger tested radius (Figure 14).

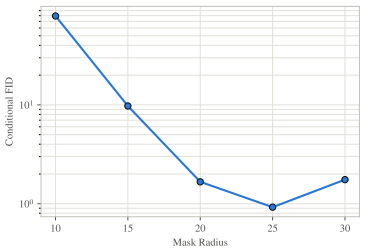


Figure 6: Conditional FID versus circular-mask radius on CelebA; the FID axis is logarithmic.

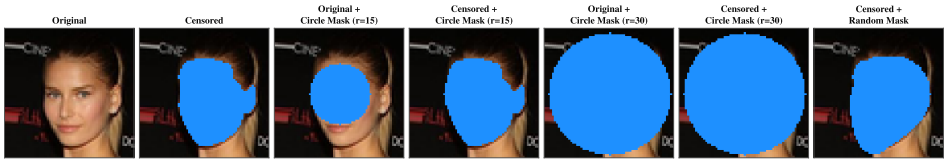


Figure 7: CelebA dataset and auxiliary corruptions (Section 5.3).

We plot the mean-discrepancy lower bound in Figure 8 using empirical moments of the clean and face-masked distributions. The bound $\sigma \geq \sqrt{[\Delta^2/(4\delta) - M_2/2]_+}$ from theorem 2 (Appendix C.1) indicates that achieving a small TV distance δ requires very high noise levels. This suggests that Ambient Omni effectively discards the corrupted dataset's useful reconstruction information to achieve alignment, leaving little advantage over clean-only training. To test this prediction, we evaluate Ambient Omni at fixed noise thresholds $\sigma_{t_n} \in \{3, 8, 10, 12\}$. Its FID improves as the threshold increases, but remains worse than the clean-only baseline at every tested setting (table 6).

Motivated by our CIFAR-10 experiments (Section 5.1), we hypothesize that randomizing the mask shape can reduce the distributional mismatch associated with fixed circular boundaries discussed above. We therefore evaluate an auxiliary channel $\widehat{\mathbf{Q}}$ that masks the face region with a random shape. This channel improves both conditional and unconditional FID over the fixed circular-mask variant (table 6). Details of the random-shape corruption are given in Section B.

Table 6: CelebA FID (Section 5.3). Sweeps are in Figures 12 to 14.

Method		FID ↓
Clean-only		4.08
Ambient Omni	$\sigma_{t_n} = 3$	81.55
	$\sigma_{t_n} = 8$	47.35
	$\sigma_{t_n} = 10$	21.31
	$\sigma_{t_n} = 12$	15.54
EMAC (Conditional)	$\widehat{\mathbf{Q}}$ =circle mask	0.92
	$\widehat{\mathbf{Q}}$ =random-shape mask	0.52
EMAC (Unconditional)	$\widehat{\mathbf{Q}}$ =circle mask	3.46
	$\widehat{\mathbf{Q}}$ =random-shape mask	**3.23**

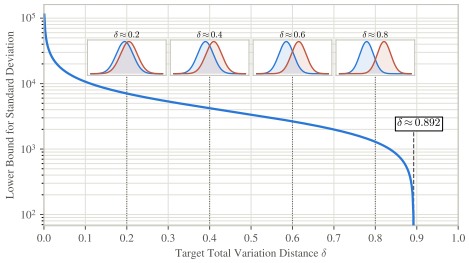


Figure 8: Mean-discrepancy lower bound on Gaussian noise needed for TV distance δ on CelebA (Section 5.3), using empirical moments. Gaussian insets illustrate mismatch.

AI USE STATEMENT

We used generative AI tools to assist with mathematical statements and proofs, reviewing related literature, and drafting and editing manuscript text. We also used these tools to prepare LaTeX code, tables, and figure layouts. Everything has been human-verified rigorously. The authors retain responsibility for all claims, results, citations, and other content in the paper.

ETHICS STATEMENT

Reconstructing corrupted images raises privacy and misuse concerns, particularly for face imagery. Generated reconstructions need not match the hidden original content and should not be treated as evidence of an individual's appearance or identity. Models may inherit biases from their training data and pretrained components, and aggregate image-quality metrics do not establish fairness or privacy protection. Use and redistribution of datasets, models, and example images should respect their applicable licences and access conditions. Our experiments do not establish that masking provides reliable anonymization.

REFERENCES

Asad Aali, Marius Arvinte, Sidharth Kumar, and Jonathan I. Tamir. Solving inverse problems with score-based generative priors learned from noisy data. In *57th Asilomar Conference on Signals, Systems, and Computers*, 2023.

Lorenzo Agnolucci, Leonardo Galteri, Marco Bertini, and Alberto Del Bimbo. ARNIQA: Learning distortion manifold for image quality assessment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. URL https://github.com/miccunifi/ARNIQA.

Weimin Bai, Yifei Wang, Wenzheng Chen, and He Sun. An expectation-maximization algorithm for training clean diffusion models from corrupted observations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 19447–19471, 2024.

Ashish Bora, Eric Price, and Alexandros G. Dimakis. AmbientGAN: Generative models from lossy measurements. In *International Conference on Learning Representations*, 2018.

Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. TOPIQ: A top-down approach from semantics to distortions for image quality assessment. *arXiv preprint arXiv:2308.03060*, 2023. URL https://arxiv.org/abs/2308.03060.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alexandros G. Dimakis, and Adam Klivans. Ambient diffusion: Learning clean distributions from corrupted data. In *Advances in Neural Information Processing Systems*, volume 36, pp. 288–313, 2023.

Giannis Daras, Alexandros G. Dimakis, and Constantinos Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10091–10108. PMLR, 2024.

Giannis Daras, Yeshwanth Cherapanamjeri, and Constantinos Costis Daskalakis. How much is a noisy image worth? data scaling laws for ambient diffusion. In *International Conference on Learning Representations*, 2025a.

Giannis Daras, Jeffrey Ouyang-Zhang, Krithika Ravishankar, William Daspit, Costis Daskalakis, Qiang Liu, Adam Klivans, and Daniel J. Diaz. Ambient proteins: Training diffusion models on low quality structures. *bioRxiv*, 2025b. doi: 10.1101/2025.07.03.663105. URL https://www.biorxiv.org/content/early/2025/07/05/2025.07.03.663105.

Giannis Daras, Adrian Rodriguez-Munoz, Adam Klivans, Antonio Torralba, and Constantinos Daskalakis. Ambient diffusion omni: Training good models with bad data. *arXiv preprint arXiv:2506.10038*, 2025c.

Jonathan Dinu. Face parsing. Hugging Face model repository, 2022. URL https://huggingface.co/jonathandinu/face-parsing. Pretrained model. Accessed September 12, 2026.

Dan Friedman and Adji Bousso Dieng. The Vendi Score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2210.02410.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. DataComp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Josh Givens, Song Liu, and Henry WJ Reeve. Score matching with missing data. *arXiv preprint arXiv:2506.00557*, 2025.

Sachin Goyal, Pratyush Maini, Zachary C. Lipton, Aditi Raghunathan, and J. Zico Kolter. Scaling laws for data filtering—data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22702–22711, 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 *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Danial Hosseintabar, Fan Chen, Giannis Daras, Antonio Torralba, and Constantinos Daskalakis. DiffEM: Learning from corrupted data with diffusion models via expectation maximization. *arXiv preprint arXiv:2510.12691*, 2025.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24174–24184, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Karras_Analyzing_and_Improving_the_Training_Dynamics_of_Diffusion_Models_CVPR_2024_paper.html.

Bahjat Kawar, Noam Elata, Tomer Michaeli, and Michael Elad. GSURE-based diffusion model training with corrupted data. *Transactions on Machine Learning Research*, 2024.

Varun A Kelkar, Rucha Deshpande, Arindam Banerjee, and Mark A Anastasio. Ambientflow: Invertible generative models from incomplete, noisy measurements. *arXiv preprint arXiv:2309.04856*, 2023.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1904.06991.

Haoye Lu, Darren Lo, and Yaoliang Yu. Sfbd flow: A continuous-optimization framework for training diffusion models with noisy samples. *arXiv preprint arXiv:2506.02371*, 2025a.

Haoye Lu, Qifan Wu, and Yaoliang Yu. Stochastic forward-backward deconvolution: Training diffusion models with finite noisy datasets. *arXiv preprint arXiv:2502.05446*, 2025b.

Haoye Lu, Yaoliang Yu, and Darren Lo. SFBD-OMNI: Bridge models for lossy measurement restoration with limited clean samples. *arXiv preprint arXiv:2512.17051*, 2025c.

Chirag Modi, Jiequn Han, Eric Vanden-Eijnden, and Joan Bruna. Generative modeling from black-box corruptions via self-consistent stochastic interpolants. In *International Conference on Learning Representations*, 2026.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 7176–7185, 2020. URL https://proceedings.mlr.press/v119/naeem20a.html.

Adrian Rodriguez-Munoz, William Daspit, Adam Klivans, Antonio Torralba, Constantinos Daskalakis, and Giannis Daras. Ambient dataloops: Generative models for dataset refinement. *arXiv preprint arXiv:2601.15417*, 2026.

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016. URL https://arxiv.org/abs/1606.03498.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 25278–25294, 2022.

Xiangfei Sheng, Xiaofeng Pan, Zhichao Yang, Pengfei Chen, and Leida Li. Fine-grained image quality assessment for perceptual image restoration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8914–8922, 2026. URL https://sxfly99.github.io/FGResQ-Home/.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 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. In *International Conference on Learning Representations*, 2021.

Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2555–2563, 2023. doi: 10.1609/aaai.v37i2.25353.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtao Zhai, and Weisi Lin. Q-Align: Teaching LMMs for visual scoring via discrete text-defined levels. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2312.17090.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12077–12090, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/64f1f27bf1b4ec22924fd0acb550c235-Abstract.html.

Appendix Contents

A Analysis of Ambient-o Training **14**

B CelebA Experimental Details **15**

C Theory **18**

C.1 Lower bounds after Gaussian smoothing 18

C.2 Population setup and notation 20

C.3 General likelihood convergence 21

C.4 Convergence under auxiliary alignment 23

C.5 Discussion 25

D Evaluation Metrics **26**

D.1 FID 26

D.2 Precision, recall, density, and coverage 26

D.3 Vendi Score 27

D.4 Inception Score 27

D.5 No-reference image-quality metrics 27

E Conditional Model Architecture **28**

F CIFAR-10 Experimental Sweeps **30**

A Analysis of Ambient-o Training

To examine how Ambient-o distributes supervision across noise levels, we first recall its training objective using the notation of Daras et al. (2025c, Equation 2.2). There, $x_0^{(i)}$ denotes a sample from the dataset (that is either clean or corrupted), $x_{t_n}^{(i)}$ a noisy sample at diffusion-time t_n, and x_t a further-noised input at diffusion-time $t > t_n$. Ambient-o first assigns each training example a minimum diffusion time $t_i^{\min}$. The annotation aims to identify when adding Gaussian noise makes the clean and corrupted distributions sufficiently close, as expressed by $d_{\mathrm{TV}}\big(P_X \circledast \mathcal{N}(0, \sigma^2(t) I_d), P_Y \circledast \mathcal{N}(0, \sigma^2(t) I_d)\big) < \epsilon$ for a small tolerance $\epsilon > 0$. Its annotated dataset consists of pairs $(x_{t_i^{\min}}^{(i)}, t_i^{\min})$, where $x_{t_i^{\min}}^{(i)}$ is the observed image after adding Gaussian noise at its assigned threshold. Clean examples have $t_i^{\min} = 0$ and receive no additional noise at annotation; corrupted examples have positive thresholds. Conditioning on this annotated dataset gives the core objective from Daras et al. (2025c, Section 3.1):

$$J_{\text{ambient-o}}(\theta) = \mathop{\mathbb{E}}_{t \in \mathcal{U}[0,1]} \sum_{i : t_i^{\min} < t} \mathop{\mathbb{E}}_{x_t \mid x_{t_i^{\min}}^{(i)}} \left[\left\| \alpha(t, t_i^{\min}) h_\theta(x_t, t) + \big(1 - \alpha(t, t_i^{\min})\big) x_t - x_{t_i^{\min}}^{(i)} \right\|_2^2 \right], \quad (7)$$

where $\alpha(t, t_i^{\min}) = (\sigma^2(t) - \sigma^2(t_i^{\min}))/\sigma^2(t)$ and $x_t \mid x_{t_i^{\min}}^{(i)} \sim \mathcal{N}\Big(x_{t_i^{\min}}^{(i)}, (\sigma^2(t) - \sigma^2(t_i^{\min})) I_d\Big)$.

Under this objective, each example contributes only for $t > t_i^{\min}$. For a clean datapoint with $t_i^{\min} = 0$ and $\sigma(0) = 0$, its loss term becomes the squared denoising error in Equation 2. At low diffusion times, the model is effectively trained only on the small clean subset. This motivates examining denoising accuracy separately at low and high noise levels.

We evaluated clean-data denoising performance throughout Ambient-o training using the CIFAR-10 setup described in Section 3. We trained for 20 Mimg (1 Mimg $= 2^{20}$ images). At every 1-Mimg checkpoint, we evaluated the standard EDM loss at fixed noise levels on a held-out validation set containing 10,000 clean images, $\mathcal{D}_{\text{val}} = \{x_0^{(i)}\}_{i=1}^m$. Clean-data evaluation uses the ordinary denoising score-matching objective:

$$J_{\text{val}}(\theta; t) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{x_t \mid x_0^{(i)}} \left[\left\| h_\theta(x_t, t) - x_0^{(i)} \right\|_2^2 \right], \quad (8)$$

where $x_t \mid x_0^{(i)} \sim \mathcal{N}(x_0^{(i)}, \sigma^2(t) I_d)$. This is obtained by setting $t_i^{\min} = 0$, so that $\alpha = 1$, in Equation 7 and averaging over the clean validation set at fixed t. This evaluates denoising against clean targets, rather than the noisy targets in the Ambient training objective. Figure 3 shows the resulting validation-loss profiles: at low noise levels, the loss initially decreases and then rises, consistent with overfitting, whereas at higher noise levels it remains stable or continues to decrease.

To visualize how sampling trajectories behave under sparse low-noise supervision, we also ran a two-dimensional toy experiment. We sampled 100 points uniformly from a unit-length line segment in $\mathbb{R}^2$. None of the points were actually corrupted. However, when training Ambient-o, we designated 95% of them as corrupted by assigning them a minimum noise level $\sigma(t_i^{\min}) = 0.1$. The remaining five points were designated as clean and assigned a threshold of zero, making them the only source of low-noise supervision. The assigned noise standard deviation, 0.1, is one tenth of the segment length, and none of the underlying samples are actually corrupted. This lets us examine restricted low-noise supervision without introducing dataset corruption. We use only 100 samples so that the clean subset contains just five points, creating a setting in which learning from the remaining samples is important. Figure 9 shows the resulting DDIM trajectories: upon entering the low-noise regime, they bend sharply toward these five points and terminate near them rather than covering the target segment, consistent with overfitting to the clean subset.

The validation losses reach their lowest values at different training checkpoints across noise levels. Noise-dependent regularization or changes to the training schedule may help mitigate this imbalance; we leave their investigation to future work.

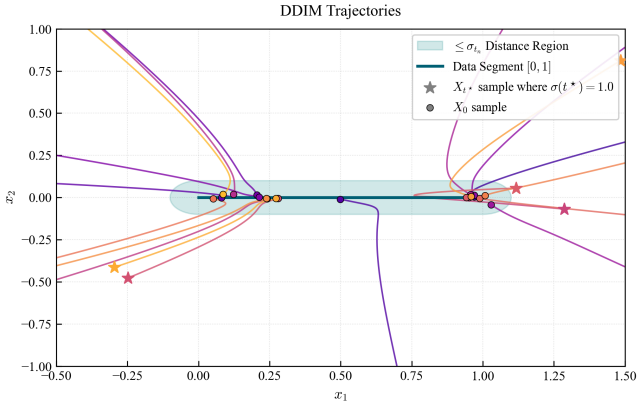


Figure 9: DDIM trajectories in the two-dimensional toy experiment illustrating overfitting under sparse low-noise supervision, as discussed in Section 3 and Appendix A.

B CelebA Experimental Details

Our random-shape auxiliary corruption uses a broad family of mask shapes to encourage distributional alignment. Figure 11 supports this qualitatively. Convex masks such as circles provide a simple starting point, but a fixed boundary can leave the mismatch discussed in Section 5.3. Enlarging the mask can reduce this mismatch, but also discards more information from the image. We therefore randomize the boundary to give the auxiliary channel support over many different shapes. Any smooth mask can be seen as a positive radial function $u(\theta)$. Let us express $\log u(\theta)$ as a Fourier series (since it is 2π-periodic). We retain the first four terms and randomize their sine and cosine coefficients. This will give us a diverse family of mask shapes. The overall scale is set separately by the mask diameter below. We use Gaussian coefficients so that each coefficient's distribution has support on all of $\mathbb{R}$. Exponentiation ensures $u(\theta) > 0$.

For a $w \times h$ image, we place the coordinate origin at the image center $(w/2, h/2)$. We first sample the initial radial function u as follows:

$$\begin{aligned} A_k, B_k &\overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \left(\frac{0.4}{k^{1.5}}\right)^2\right), \quad k = 1, \ldots, 4, \\ u(\theta) &= \exp\left(\sum_{k=1}^{4} [A_k \cos(k\theta) + B_k \sin(k\theta)]\right), \quad \theta \in [0, 2\pi). \end{aligned} \tag{9}$$

We stretch the boundary defined by u vertically by a factor s to obtain v, then recenter and uniformly rescale v to diameter $D \min(h, w)$ to obtain the final curve r. Writing $\bar{v}$ and $\operatorname{diam}(v)$ for the centroid and diameter of the boundary points $\{v(\theta_i)\}$, we define

$$\begin{aligned} s \sim \mathcal{U}[1, 1.8], \qquad D \sim \mathcal{U}[0.6, 0.8], \\ v(\theta) = u(\theta)\big(\cos\theta,\ s \sin\theta\big), \\ r(\theta) = \min(h, w)\, \frac{D\,[v(\theta) - \bar{v}]}{\operatorname{diam}(v)}. \end{aligned} \tag{10}$$

Figure 10 illustrates different random mask realizations applied to the same image.



Figure 10: Random-shape auxiliary masks constructed as described in Appendix B. The top-left image is the original; the remaining images show fifteen random mask realizations applied to it.

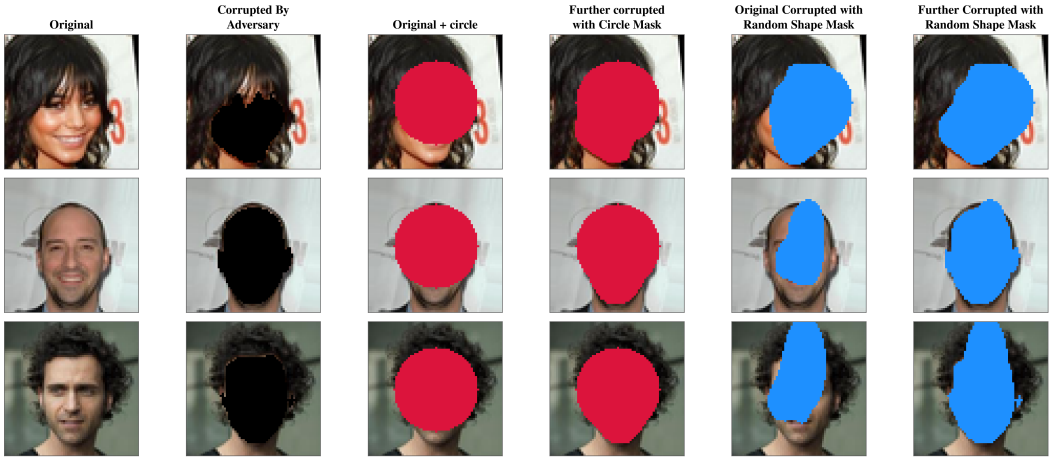


Figure 11: Three CelebA examples from Section 5.3. From left to right, the columns show clean images, face-redacted observations, circular auxiliary masks applied to clean and redacted images, and random-shape auxiliary masks applied to clean and redacted images.

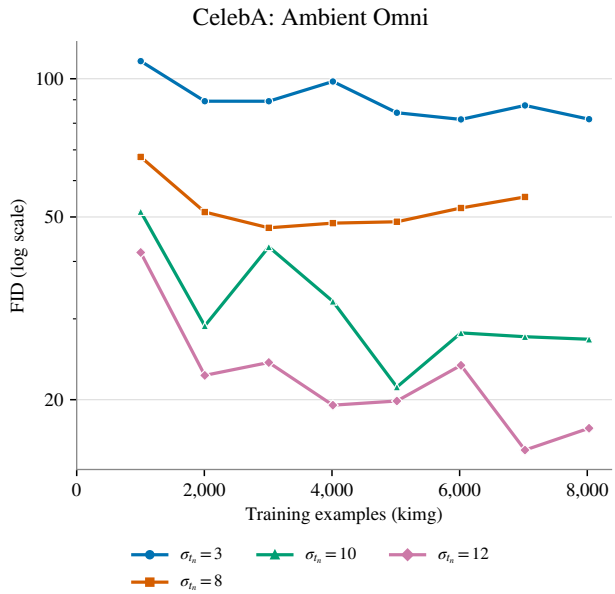


Figure 12: CelebA FID across training checkpoints for Ambient Omni with fixed noise thresholds σ_{t_n} (Section 5.3). The lowest FID for each threshold is reported in table 6.

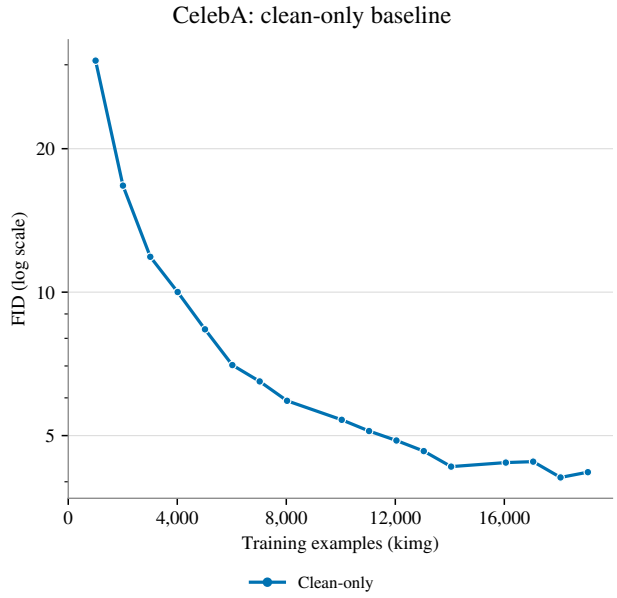


Figure 13: CelebA clean-only baseline FID across training checkpoints (Section 5.3). The lowest FID is reported in table 6.

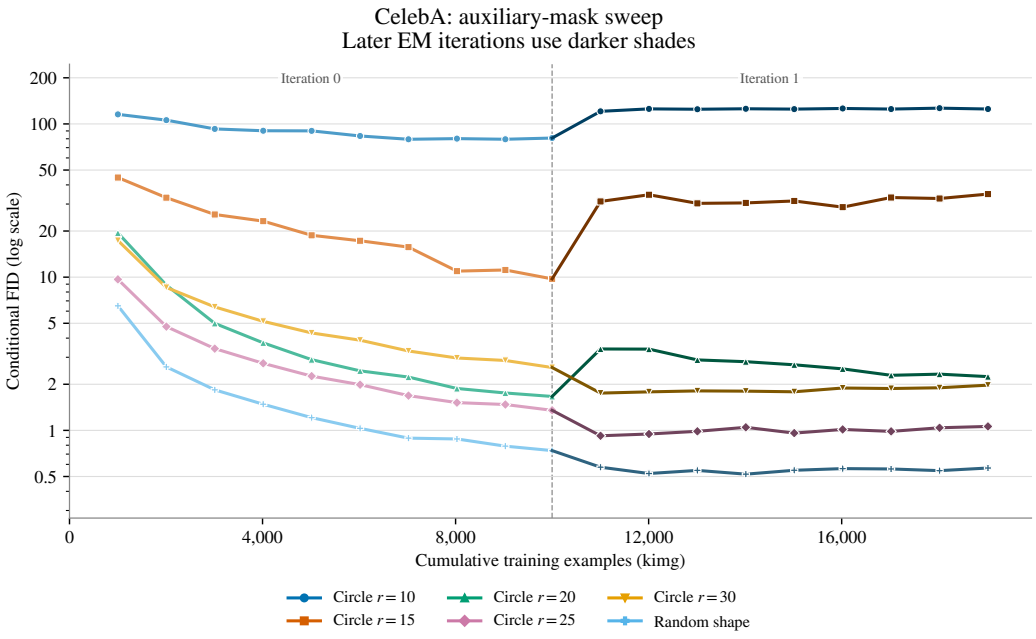


Figure 14: CelebA conditional FID for fixed center-circle masks of different radii and random-shape masks (Section 5.3). EM iterations continue along a cumulative training axis; darker shades denote later iterations, separated by the dashed line.

C THEORY

C.1 LOWER BOUNDS AFTER GAUSSIAN SMOOTHING

Let P and Q be probability distributions on $\mathbb{R}^d$. For $\sigma > 0$, define

$$P_\sigma = P \circledast \mathcal{N}(0, \sigma^2 I_d), \qquad Q_\sigma = Q \circledast \mathcal{N}(0, \sigma^2 I_d), \tag{11}$$

and use the convention

$$\mathrm{d_{TV}}(P, Q) = \sup_A |P(A) - Q(A)| = \frac{1}{2} \int |p - q| \tag{12}$$

whenever P and Q admit densities p and q.

For the mean-discrepancy bounds, assume that P and Q have finite second moments, with means μ_P and μ_Q. Suppose

$$\Delta := \|\mu_P - \mu_Q\|_2 > 0, \qquad u := \frac{\mu_P - \mu_Q}{\Delta}, \tag{13}$$

and define the directional variances

$$v_P := \mathrm{Var}_{X \sim P}(u^\top X), \qquad v_Q := \mathrm{Var}_{Y \sim Q}(u^\top Y). \tag{14}$$

Set

$$M_2 := v_P + v_Q + \frac{\Delta^2}{2}. \tag{15}$$

Lemma 1 (Global lower bound from a mean discrepancy)**.** *Under the setup above, for every $\sigma > 0$,*

$$\mathrm{d_{TV}}(P_\sigma, Q_\sigma) \geq \frac{\Delta^2}{2M_2 + 4\sigma^2}. \tag{16}$$

Proof. First project along u. Let $R = u^\top X + \sigma G$ and $S = u^\top Y + \sigma G$, where $G \sim \mathcal{N}(0, 1)$ is independent of X, Y. Since total variation contracts under measurable maps,

$$\mathrm{d_{TV}}(P_\sigma, Q_\sigma) \geq \mathrm{d_{TV}}(\mathrm{law}(R), \mathrm{law}(S)). \tag{17}$$

The means of R and S differ by Δ, while $\mathrm{Var}(R) = v_P + \sigma^2$ and $\mathrm{Var}(S) = v_Q + \sigma^2$.

We begin with a bound that is valid globally. Write $\varepsilon = \mathrm{d_{TV}}(\mathrm{law}(R), \mathrm{law}(S))$. Let p_R, p_S be the densities of R, S, and set $m = (\mathbb{E}[R] + \mathbb{E}[S])/2$. By Cauchy–Schwarz,

$$\Delta^2 = \left| \int (x - m)(p_R(x) - p_S(x)) \, \mathrm{d}x \right|^2 \tag{18}$$

$$\leq \left(\int (x - m)^2 |p_R - p_S| \, \mathrm{d}x \right) \left(\int |p_R - p_S| \, \mathrm{d}x \right). \tag{19}$$

The inequality applies Cauchy–Schwarz to $(x - m)\sqrt{|p_R - p_S|}$ and $\mathrm{sgn}(p_R - p_S)\sqrt{|p_R - p_S|}$. Since both means are at distance $\Delta/2$ from their midpoint m, each second moment about m equals its variance plus $\Delta^2/4$:

$$\int (x - m)^2 (p_R + p_S) \, \mathrm{d}x = \left(\mathrm{Var}(R) + \frac{\Delta^2}{4} \right) + \left(\mathrm{Var}(S) + \frac{\Delta^2}{4} \right).$$

Using $|p_R - p_S| \leq p_R + p_S$ and $\int |p_R - p_S| = 2\varepsilon$ in the preceding bound therefore gives

$$\Delta^2 \leq 2\varepsilon \left(\mathrm{Var}(R) + \mathrm{Var}(S) + \frac{\Delta^2}{2} \right). \tag{20}$$

Rearranging and substituting the variances of R and S yields

$$\varepsilon \geq \frac{\Delta^2}{\Delta^2 + 2(v_P + v_Q + 2\sigma^2)} = \frac{\Delta^2}{2M_2 + 4\sigma^2}. \tag{21}$$

Combining this with equation 17 proves the claim. □

Lemma 2 (Gaussian-specific lower bound from a mean discrepancy)**.** *Under the same setup, for every $\sigma > 0$,*

$$\mathrm{d_{TV}}(P_\sigma, Q_\sigma) \geq \left[\frac{\Delta}{\sigma\sqrt{2\pi}} - \frac{M_2}{\sigma^2\sqrt{2\pi e}}\right]_+. \tag{22}$$

Proof. Let $X \sim P$, $Y \sim Q$, and $G \sim \mathcal{N}(0,1)$ be independent. Set $c = (\mu_P + \mu_Q)/2$ and define $A = u^\top(X - c)$ and $B = u^\top(Y - c)$. Then

$$\mathbb{E}[A] = \frac{\Delta}{2}, \qquad \mathbb{E}[B] = -\frac{\Delta}{2}, \qquad \mathbb{E}[A^2] + \mathbb{E}[B^2] = M_2. \tag{23}$$

The map $z \mapsto u^\top(z - c)$ sends P_σ, Q_σ to the laws of $A + \sigma G, B + \sigma G$, respectively. Since total variation contracts under this map, it suffices to prove the bound for these projected laws.

Let g_σ be the one-dimensional $\mathcal{N}(0, \sigma^2)$ density. The projected densities are obtained by convolving the laws of A and B with g_σ. When these laws admit densities,

$$p_\sigma(x) = \int_{\mathbb{R}} g_\sigma(x - a)\, p_A(a)\, \mathrm{d}a = \mathbb{E}[g_\sigma(x - A)],$$
$$q_\sigma(x) = \int_{\mathbb{R}} g_\sigma(x - b)\, p_B(b)\, \mathrm{d}b = \mathbb{E}[g_\sigma(x - B)].$$

where p_A and p_B are the densities of A and B, respectively. Taylor's theorem gives

$$g_\sigma(x - z) = g_\sigma(x) - z g'_\sigma(x) + R_z(x), \qquad R_z(x) = z^2 \int_0^1 (1 - t) g''_\sigma(x - tz)\, \mathrm{d}t. \tag{24}$$

Taking expectations in equation 24, with $R_P(x) := \mathbb{E}[R_A(x)]$ and $R_Q(x) := \mathbb{E}[R_B(x)]$, yields

$$p_\sigma(x) = \mathbb{E}[g_\sigma(x - A)] = g_\sigma(x) - \mathbb{E}[A] g'_\sigma(x) + R_P(x),$$
$$q_\sigma(x) = \mathbb{E}[g_\sigma(x - B)] = g_\sigma(x) - \mathbb{E}[B] g'_\sigma(x) + R_Q(x).$$

Subtracting these expansions and using equation 23 gives

$$p_\sigma - q_\sigma = -\Delta g'_\sigma + R_P - R_Q. \tag{25}$$

We now bound the remainders using their integral representations. The triangle inequality gives

$$\begin{aligned}
\|R_P\|_1 &= \int_{\mathbb{R}} \left| \mathbb{E}\left[A^2 \int_0^1 (1 - t) g''_\sigma(x - tA)\, \mathrm{d}t \right] \right| \mathrm{d}x \\
&\leq \mathbb{E}\left[A^2 \int_0^1 (1 - t) \left(\int_{\mathbb{R}} |g''_\sigma(x - tA)|\, \mathrm{d}x \right) \mathrm{d}t \right] \\
&= \frac{\mathbb{E}[A^2]}{2} \|g''_\sigma\|_1.
\end{aligned}$$

The same calculation for B, together with $\mathbb{E}[A^2] + \mathbb{E}[B^2] = M_2$, yields

$$\|R_P\|_1 + \|R_Q\|_1 \leq \frac{M_2}{2} \|g''_\sigma\|_1.$$

Using the TV–L^1 identity and the reverse triangle inequality,

$$\begin{aligned}
\mathrm{d_{TV}}(\mathrm{law}(A + \sigma G), \mathrm{law}(B + \sigma G)) &= \frac{1}{2} \|p_\sigma - q_\sigma\|_1 \\
&= \frac{1}{2} \| - \Delta g'_\sigma + R_P - R_Q\|_1 \\
&\geq \frac{\Delta}{2} \|g'_\sigma\|_1 - \frac{1}{2} \big(\|R_P\|_1 + \|R_Q\|_1 \big) \\
&\geq \frac{\Delta}{2} \|g'_\sigma\|_1 - \frac{M_2}{4} \|g''_\sigma\|_1.
\end{aligned} \tag{26}$$

For $G \sim \mathcal{N}(0,1)$, direct calculation yields

$$\|g'_\sigma\|_1 = \frac{\sqrt{2/\pi}}{\sigma}, \qquad \|g''_\sigma\|_1 = \frac{\mathbb{E}|G^2 - 1|}{\sigma^2} = \frac{4}{\sigma^2\sqrt{2\pi e}}. \tag{27}$$

Substituting equation 27 into equation 26, using nonnegativity of total variation, and applying the projection argument proves the claim. □

Theorem 2 (Two-regime lower bound from a mean discrepancy)**.** *Let P and Q have finite second moments and $\Delta > 0$, with Δ and M_2 defined above. Then, for every $\sigma > 0$,*

$$\mathrm{d_{TV}}(P_\sigma, Q_\sigma) \geq \max\left\{ \frac{\Delta^2}{2M_2 + 4\sigma^2}, \left[\frac{\Delta}{\sigma\sqrt{2\pi}} - \frac{M_2}{\sigma^2\sqrt{2\pi e}} \right]_+ \right\}. \tag{28}$$

The first term is positive for every σ. For fixed P and Q, the second term becomes positive once $\sigma > M_2/(\Delta\sqrt{e})$ and yields

$$\mathrm{d_{TV}}(P_\sigma, Q_\sigma) \geq \frac{\Delta}{\sigma\sqrt{2\pi}} - O(\sigma^{-2}) \qquad \text{as } \sigma \to \infty. \tag{29}$$

Thus, a nonzero mean discrepancy yields a total-variation lower bound of order $1/\sigma$ in the high-noise regime.

Proof. Both bounds hold for every $\sigma > 0$ by lemmas 1 and 2. Taking their maximum gives equation 28; the asymptotic statement follows from the second term. □

C.2 Population setup and notation

We study convergence of Algorithm 1 in the population setting, assuming exact conditional fitting and sampling. We fix the clean-pair probability at $\rho_k = \rho$, with $0 < \rho \leq 1$ unless stated otherwise. We show that the EM objective converges and that, under alignment, the iterates approach the clean distribution.

Let us first introduce the notation for the population EM iterations. Throughout the EM analysis, P denotes a probability distribution and $P(\cdot)$ its density (or probability mass function in the discrete case).

Let P_X be the clean distribution and

$$P_Y = \mathbf{Q}_\# P_X \tag{30}$$

the observed distribution under the unknown corruption channel $\mathbf{Q}$. For the known auxiliary channel $\widehat{\mathbf{Q}}$, define

$$P_{\widehat{Y}} := \widehat{\mathbf{Q}}_\# P_X, \qquad P_{\overline{Y}} := \widehat{\mathbf{Q}}_\# P_Y. \tag{31}$$

We impose auxiliary alignment (Assumption 1), as in Section 4:

$$P_{\widehat{Y}} = P_{\overline{Y}} =: P_Z. \tag{32}$$

Let $P^k_{X|Z}(x \mid z) := p_{\theta^{(k)}}(x \mid z)$ denote the learned conditional at iteration k. Define the E-step reconstruction density by

$$R^k_X(x) := \int p_{\theta^{(k)}}(x \mid z)\, P_Z(z)\, \mathrm{d}z. \tag{33}$$

Let P^{k+1}_X denote the distribution of inputs used in the next M-step:

$$P^{k+1}_X(x) := \rho P_X(x) + (1-\rho) R^k_X(x). \tag{34}$$

For $k \geq 1$, P^k_X is this training-mixture distribution, whereas $P^k_{X|Z}(x \mid z)$ is the conditional learned by the network.

Set $P^k_Z := \widehat{\mathbf{Q}}_\# P^k_X$ and let $P^k(x, z)$ denote the joint training density. Exact conditional fitting means that

$$P^k(x, z) := P^k_X(x)\widehat{\mathbf{Q}}(z \mid x) = p_{\theta^{(k)}}(x \mid z) P^k_Z(z). \tag{35}$$

The implicit marginal P^k_X satisfies

$$P^k_X(x) = \int p_{\theta^{(k)}}(x \mid z)\, P^k_Z(z)\, \mathrm{d}z, \qquad P^k_Z = \widehat{\mathbf{Q}}_\# P^k_X. \tag{36}$$

Here P^k_Z is the model-induced marginal; using the observed marginal P_Z instead gives R^k_X.

Assume the initial distribution P^0_X and conditional satisfy equation 35, and $P^0_Z(Z) > 0$ with probability one for $Z \sim P_Z$. For an arbitrary initial conditional, the analysis starts after the first exact M-step.

C.3 General likelihood convergence

For this subsection only, we also allow Assumption 1 to be dropped. The reconstruction in Equation 33 then uses $P_{\overline{Y}}$ in place of P_Z. We assume the initial conditional is compatible with P_X^0 as in equation 35 and that $P_Z^k(Z) > 0$ with probability one for $Z \sim P_{\overline{Y}}$ at every iterate, so the population E-steps remain well defined.

Definition 1 (Population EM objective)**.** For a candidate clean distribution Π, set $\Pi_Z := \widehat{\mathbf{Q}}_{\#}\Pi$ and define the population log-likelihood by

$$\ell(\Pi) := \rho\, \mathbb{E}_{X\sim P_X}[\log \Pi(X)] + (1-\rho)\, \mathbb{E}_{Z\sim P_{\overline{Y}}}[\log \Pi_Z(Z)]\,. \tag{37}$$

Define the corresponding KL objective by

$$\mathcal{F}(\Pi) := \rho D_{\mathrm{KL}}(P_X \| \Pi) + (1-\rho) D_{\mathrm{KL}}\left(P_{\overline{Y}} \middle\| \widehat{\mathbf{Q}}_{\#}\Pi\right). \tag{38}$$

The exact population updates above are EM updates for ℓ. Minimizing $\mathcal{F}$ is equivalent to maximizing ℓ. Here $\mathcal{F}$ is a distribution-level objective, not the denoising loss.

Theorem 3 (General population likelihood convergence)**.** *Let*

$$P_X^\star \in \arg\min_{\Pi} \mathcal{F}(\Pi) \tag{39}$$

be a population minimizer. Without auxiliary alignment, $P_X^\star$ need not equal the true clean distribution P_X. We assume that $P_X^\star$ exists, that the relevant KL divergences are finite, and that the exact population EM update is well defined.

The exact population iterates satisfy

$$\mathcal{F}(P_X^K) - \mathcal{F}(P_X^\star) \le \frac{D_{\mathrm{KL}}(P_X^\star \| P_X^0)}{K}. \tag{40}$$

Thus, the population likelihood converges to its optimal value at rate $O(1/K)$ without requiring auxiliary alignment or identifiability.

Proof. At a population optimum, the EM update leaves $P_X^\star$ unchanged. Write $P_{X|Z}^\star$ for its posterior under the auxiliary channel and $R_X^\star$ for its reconstruction distribution. The full fixed-point condition includes both clean samples and reconstructions:

$$\begin{aligned} R_X^\star(x) &:= \int P_{X|Z}^\star(x \mid z) P_{\overline{Y}}(z)\, \mathrm{d}z, \\ P_X^\star(x) &= \rho P_X(x) + (1-\rho) R_X^\star(x). \end{aligned}$$

Reapplying the auxiliary channel gives the joint training density $P_X^\star(x)\widehat{\mathbf{Q}}(z \mid x)$, whose posterior is

$$P_{X|Z}^\star(x \mid z) = \frac{P_X^\star(x)\widehat{\mathbf{Q}}(z \mid x)}{P_Z^\star(z)} \tag{41}$$

where $P_Z^\star := \widehat{\mathbf{Q}}_{\#}P_X^\star$ is the model-induced observation distribution, not necessarily $P_{\overline{Y}}$. Bayes' rule gives

$$\log \frac{P_{X|Z}^\star(x \mid z)}{P_{X|Z}^k(x \mid z)} = \log \frac{P_X^\star(x)}{P_X^k(x)} + \log \frac{P_Z^k(z)}{P_Z^\star(z)}. \tag{42}$$

Averaging Equation 42 over $Z \sim P_{\overline{Y}}$ and $X \mid Z \sim P_{X|Z}^\star(\cdot \mid Z)$ (so marginally $X \sim R_X^\star$) gives

$$\begin{aligned} &\mathbb{E}_{Z\sim P_{\overline{Y}}} D_{\mathrm{KL}}\Big(P_{X|Z}^\star(\cdot \mid Z) \Big\| P_{X|Z}^k(\cdot \mid Z)\Big) \\ &\quad = \int R_X^\star(x) \log \frac{P_X^\star(x)}{P_X^k(x)}\, \mathrm{d}x + \int P_{\overline{Y}}(z) \log \frac{P_Z^k(z)}{P_Z^\star(z)}\, \mathrm{d}z. \end{aligned} \tag{43}$$

By the definition of $\mathcal{F}$ in Equation 38,

$$\mathcal{F}(P_X^k) - \mathcal{F}(P_X^\star) = \rho \int P_X(x) \log \frac{P_X^\star(x)}{P_X^k(x)}\, \mathrm{d}x + (1-\rho) \int P_{\overline{Y}}(z) \log \frac{P_Z^\star(z)}{P_Z^k(z)}\, \mathrm{d}z.$$

Now multiply Equation 43 by $1-\rho$. Substituting the fixed-point relation $(1-\rho)R_X^\star(x) = P_X^\star(x) - \rho P_X(x)$ and the objective gap above gives

$$\begin{aligned}(1-\rho)\mathbb{E}_{Z\sim P_{\overline{Y}}} D_{\mathrm{KL}}\left(P_{X|Z}^\star(\cdot \mid Z)\middle\| P_{X|Z}^k(\cdot \mid Z)\right) \\ = D_{\mathrm{KL}}(P_X^\star \| P_X^k) - \left[\mathcal{F}(P_X^k) - \mathcal{F}(P_X^\star)\right].\end{aligned} \tag{44}$$

The fixed-point relation and the M-step update have the same clean component. Joint convexity of KL divergence therefore gives

$$\begin{aligned}D_{\mathrm{KL}}(P_X^\star \| P_X^{k+1}) &= D_{\mathrm{KL}}\big(\rho P_X + (1-\rho)R_X^\star \big\| \rho P_X + (1-\rho)R_X^k\big) \\ &\le \rho D_{\mathrm{KL}}(P_X \| P_X) + (1-\rho) D_{\mathrm{KL}}(R_X^\star \| R_X^k) \\ &\le (1-\rho)\mathbb{E}_{Z\sim P_{\overline{Y}}} D_{\mathrm{KL}}\left(P_{X|Z}^\star(\cdot \mid Z)\middle\| P_{X|Z}^k(\cdot \mid Z)\right).\end{aligned} \tag{45}$$

Here $D_{\mathrm{KL}}(P_X \| P_X) = 0$, and the last inequality follows from lemma 3, proved below. Combining Equations 44 and 45,

$$\begin{aligned}\mathcal{F}(P_X^k) - \mathcal{F}(P_X^\star) \le D_{\mathrm{KL}}(P_X^\star \| P_X^k) \\ - D_{\mathrm{KL}}(P_X^\star \| P_X^{k+1}).\end{aligned} \tag{46}$$

Summing Equation 46 for $k = 0, \ldots, K-1$ yields

$$\sum_{k=0}^{K-1} \left[\mathcal{F}(P_X^k) - \mathcal{F}(P_X^\star)\right] \le D_{\mathrm{KL}}(P_X^\star \| P_X^0). \tag{47}$$

Exact EM makes $\mathcal{F}(P_X^k)$ nonincreasing. Hence,

$$K\left[\mathcal{F}(P_X^K) - \mathcal{F}(P_X^\star)\right] \le D_{\mathrm{KL}}(P_X^\star \| P_X^0), \tag{48}$$

which proves the result. □

Lemma 3 (KL contraction under marginalization)**.** *The reconstruction distributions $R_X^\star$ and R_X^k defined above satisfy*

$$D_{\mathrm{KL}}(R_X^\star \| R_X^k) \le \mathbb{E}_{Z\sim P_{\overline{Y}}} D_{\mathrm{KL}}\Big(P_{X|Z}^\star(\cdot \mid Z)\Big\| P_{X|Z}^k(\cdot \mid Z)\Big).$$

Proof. The bound follows from Jensen's inequality. For each fixed x, write

$$a_x(z) := P_{\overline{Y}}(z) P_{X|Z}^\star(x \mid z), \qquad b_x(z) := P_{\overline{Y}}(z) P_{X|Z}^k(x \mid z).$$

Then $\int a_x(z)\,\mathrm{d}z = R_X^\star(x)$ and $\int b_x(z)\,\mathrm{d}z = R_X^k(x)$. For $R_X^k(x) > 0$, the weights $w_x(z) := b_x(z)/R_X^k(x)$ integrate to one. Applying Jensen's inequality to the convex function $\phi(u) = u \log u$ gives

$$\begin{aligned}R_X^\star(x) \log \frac{R_X^\star(x)}{R_X^k(x)} &= R_X^k(x)\, \phi\left(\int w_x(z) \frac{a_x(z)}{b_x(z)}\,\mathrm{d}z\right) \\ &\le R_X^k(x) \int w_x(z)\, \phi\left(\frac{a_x(z)}{b_x(z)}\right)\,\mathrm{d}z \\ &= \int a_x(z) \log \frac{a_x(z)}{b_x(z)}\,\mathrm{d}z.\end{aligned}$$

Zero-density cases are interpreted using the usual KL conventions. Integrating over x and canceling the common factor $P_{\overline{Y}}(z)$ inside the logarithm yields

$$\begin{aligned}D_{\mathrm{KL}}(R_X^\star \| R_X^k) &\le \iint P_{\overline{Y}}(z) P_{X|Z}^\star(x \mid z) \log \frac{P_{X|Z}^\star(x \mid z)}{P_{X|Z}^k(x \mid z)}\,\mathrm{d}x\,\mathrm{d}z \\ &= \mathbb{E}_{Z\sim P_{\overline{Y}}} D_{\mathrm{KL}}\Big(P_{X|Z}^\star(\cdot \mid Z)\Big\| P_{X|Z}^k(\cdot \mid Z)\Big).\end{aligned}$$

□

C.4 Convergence under auxiliary alignment

We now return to the aligned setting of Assumption 1 and study recovery of the clean distribution itself. Since $P_{\overline{Y}} = P_{\widehat{Y}} = P_Z$, P_X is a fixed point of the population update.

Define

$$\mathcal{E}_k := D_{\mathrm{KL}}(P_X \| P_X^k), \tag{49}$$

and

$$E_k := D_{\mathrm{KL}}(P_Z \| P_Z^k). \tag{50}$$

Here, $\mathcal{E}_k$ measures error in the clean distribution, while E_k measures error after applying the auxiliary channel. The following lemma is the main step in the analysis.

Lemma 4 (Reconstruction error under alignment)**.** *Under Assumption 1, the reconstruction distribution in Equation 33 satisfies*

$$D_{\mathrm{KL}}(P_X \| R_X^k) \le \mathcal{E}_k - E_k. \tag{51}$$

Proof. Let $P_{X|Z}(x \mid z)$ denote the posterior induced by P_X and $\widehat{\mathbf{Q}}$. Starting from the definition,

$$\mathcal{E}_k = D_{\mathrm{KL}}(P_X \| P_X^k) = \int P_X(x) \log \frac{P_X(x)}{P_X^k(x)} \, \mathrm{d}x.$$

Since $\int \widehat{\mathbf{Q}}(z \mid x) \, \mathrm{d}z = 1$ for each x, we can insert it:

$$\begin{aligned}
\mathcal{E}_k &= \int P_X(x) \log \frac{P_X(x)}{P_X^k(x)} \left[\int \widehat{\mathbf{Q}}(z \mid x) \, \mathrm{d}z \right] \mathrm{d}x \\
&= \iint P_X(x) \widehat{\mathbf{Q}}(z \mid x) \log \frac{P_X(x)}{P_X^k(x)} \, \mathrm{d}z \, \mathrm{d}x \\
&= \iint P_X(x) \widehat{\mathbf{Q}}(z \mid x) \log \frac{P_X(x) \widehat{\mathbf{Q}}(z \mid x)}{P_X^k(x) \widehat{\mathbf{Q}}(z \mid x)} \, \mathrm{d}z \, \mathrm{d}x.
\end{aligned}$$

We now factor each joint density in the other direction. Under alignment and exact conditional fitting (Equation 35), Bayes' rule gives

$$\begin{aligned}
P_X(x) \widehat{\mathbf{Q}}(z \mid x) &= P_Z(z) P_{X|Z}(x \mid z), \\
P_X^k(x) \widehat{\mathbf{Q}}(z \mid x) &= P_Z^k(z) P_{X|Z}^k(x \mid z).
\end{aligned}$$

Substituting these identities and splitting the logarithm into two terms gives

$$\begin{aligned}
\mathcal{E}_k &= D_{\mathrm{KL}}(P_X \| P_X^k) \\
&= \iint P_Z(z) P_{X|Z}(x \mid z) \log \frac{P_Z(z) P_{X|Z}(x \mid z)}{P_Z^k(z) P_{X|Z}^k(x \mid z)} \, \mathrm{d}x \, \mathrm{d}z \\
&= \iint P_Z(z) P_{X|Z}(x \mid z) \left[\log \frac{P_Z(z)}{P_Z^k(z)} + \log \frac{P_{X|Z}(x \mid z)}{P_{X|Z}^k(x \mid z)} \right] \mathrm{d}x \, \mathrm{d}z \\
&= \int P_Z(z) \log \frac{P_Z(z)}{P_Z^k(z)} \, \mathrm{d}z \\
&\quad + \int P_Z(z) \left[\int P_{X|Z}(x \mid z) \log \frac{P_{X|Z}(x \mid z)}{P_{X|Z}^k(x \mid z)} \, \mathrm{d}x \right] \mathrm{d}z \\
&= D_{\mathrm{KL}}(P_Z \| P_Z^k) + \mathbb{E}_{Z \sim P_Z} D_{\mathrm{KL}} \left(P_{X|Z}(\cdot \mid Z) \middle\| P_{X|Z}^k(\cdot \mid Z) \right).
\end{aligned} \tag{52}$$

Therefore,

$$\mathbb{E}_{Z \sim P_Z} D_{\mathrm{KL}} \left(P_{X|Z}(\cdot \mid Z) \middle\| P_{X|Z}^k(\cdot \mid Z) \right) = \mathcal{E}_k - E_k. \tag{53}$$

The X-marginal of $P_Z(z) P_{X|Z}(x \mid z)$ is P_X, while the X-marginal of $P_Z(z) P_{X|Z}^k(x \mid z)$ is R_X^k. Applying the same style of marginalization argument as in Lemma 3 gives

$$D_{\mathrm{KL}}(P_X \| R_X^k) \le \mathbb{E}_{Z \sim P_Z} D_{\mathrm{KL}} \left(P_{X|Z}(\cdot \mid Z) \middle\| P_{X|Z}^k(\cdot \mid Z) \right). \tag{54}$$

Combining this inequality with Equation 53 proves the result. □

This reconstruction bound yields linear convergence without an identifiability assumption.

Theorem 4 (Clean-anchor convergence)**.** *Under Assumption 1, the exact population iterates satisfy*

$$D_{\mathrm{KL}}(P_X \| P_X^K) \le (1-\rho)^K D_{\mathrm{KL}}(P_X \| P_X^0). \tag{55}$$

No identifiability condition on $\widehat{\mathbf{Q}}$ is required; see DiffEM (Hosseintabar et al., 2025, Assumption 1) for the definition of the condition used in its linear-convergence analysis.

Proof. Substituting Equation 34 and using convexity of KL divergence in its second argument gives

$$\begin{aligned} \mathcal{E}_{k+1} &= D_{\mathrm{KL}}(P_X \| P_X^{k+1}) \\ &= D_{\mathrm{KL}}\left(P_X \middle\| \rho P_X + (1-\rho) R_X^k\right) \\ &\le (1-\rho) D_{\mathrm{KL}}(P_X \| R_X^k). \end{aligned} \tag{56}$$

Using Lemma 4,

$$\mathcal{E}_{k+1} \le (1-\rho)(\mathcal{E}_k - E_k). \tag{57}$$

Since $E_k \ge 0$, we have $\mathcal{E}_{k+1} \le (1-\rho)\mathcal{E}_k$. Iterating this inequality proves Equation 55. □

Since $(1-\rho)^K \le e^{-\rho K}$, Theorem 4 also implies

$$D_{\mathrm{KL}}(P_X \| P_X^K) \le e^{-\rho K} D_{\mathrm{KL}}(P_X \| P_X^0). \tag{58}$$

Thus, reaching clean-distribution error at most ε requires

$$K = O\left(\frac{1}{\rho} \log \frac{\mathcal{E}_0}{\varepsilon}\right) \tag{59}$$

The following corollary of Theorem 4 establishes uniqueness of the conditional fixed point.

Corollary 1 (Unique conditional fixed point under alignment)**.** *Under the same assumptions as Theorem 4, the posterior $P_{X|Z}(\cdot \mid Z)$ induced by $X \sim P_X$ and $Z \sim \widehat{\mathbf{Q}}(\cdot \mid X)$ is the unique fixed conditional distribution, where equality holds with probability one for $Z \sim P_Z$. No identifiability assumption on $\widehat{\mathbf{Q}}$ is required.*

Proof. The conditional distribution $P_{X|Z}$ itself is a fixed point: under alignment, its reconstruction distribution and the next training distribution satisfy

$$R_X^k(x) = \int P_{X|Z}(x \mid z) P_Z(z)\, \mathrm{d}z = P_X(x),$$

$$P_X^{k+1}(x) = \rho P_X(x) + (1-\rho) R_X^k(x) = P_X(x).$$

Exact conditional fitting therefore returns $p_{\theta^{(k+1)}} = P_{X|Z}$.

For uniqueness, assume that $\widetilde{P}_{X|Z}(x \mid z)$ is any conditional fixed point and define

$$\widetilde{P}_X(x) = \rho P_X(x) + (1-\rho)\widetilde{R}_X(x), \tag{60}$$

where $\widetilde{R}_X(x) = \int \widetilde{P}_{X|Z}(x \mid z) P_Z(z)\, \mathrm{d}z$ is its reconstruction density. Since the conditional is unchanged by exact fitting, it satisfies, for P_Z-almost every z,

$$\widetilde{P}_{X|Z}(x \mid z) = \frac{\widetilde{P}_X(x)\widehat{\mathbf{Q}}(z \mid x)}{\int \widetilde{P}_X(u)\widehat{\mathbf{Q}}(z \mid u)\, \mathrm{d}u}.$$

Initializing at $P_X^0 = \widetilde{P}_X$ therefore gives $P_X^K = \widetilde{P}_X$ for every K. Since $\widetilde{P}_X \ge \rho P_X$,

$$D_{\mathrm{KL}}(P_X \| \widetilde{P}_X) \le \log(1/\rho) < \infty. \tag{61}$$

Applying Equation 58 gives

$$D_{\mathrm{KL}}(P_X \| \widetilde{P}_X) \le e^{-\rho K} D_{\mathrm{KL}}(P_X \| \widetilde{P}_X) \xrightarrow[K\to\infty]{} 0. \tag{62}$$

Hence $\widetilde{P}_X = P_X$, so its posterior $\widetilde{P}_{X|Z}(\cdot \mid Z)$ equals $P_{X|Z}(\cdot \mid Z)$ with probability one for $Z \sim P_Z$. Thus, the conditional fixed point is unique. □

The positive clean-pair weight is essential: when $\rho = 0$ and the auxiliary channel maps every input to the same observation, every prior induces a fixed posterior. Corollary 1 concerns conditional laws in the exact population setting, not uniqueness of network parameters or fixed points of finite-data, finite-step training.

C.5 DISCUSSION

The convergence results clarify the role of auxiliary alignment. Without alignment, minimizing the EM objective need not recover the clean distribution. Under alignment, a positive clean-data weight ensures convergence to P_X and uniqueness of the conditional fixed point, without requiring identifiability of the auxiliary channel. However, alignment alone does not ensure informative observations: a channel that maps every input to a constant also satisfies this condition. The convergence guarantee therefore does not establish a statistical advantage over clean-only training, highlighting the separate motivation for preserving information.

These guarantees assume exact conditional fitting and sampling from the underlying distributions. Finite datasets, imperfect conditional models, and approximate posterior sampling fall outside this analysis. Quantifying their effects requires additional error bounds.

D Evaluation Metrics

We describe some of the metrics used in our experiments. Let $\mathcal{R} = \{x_i\}_{i=1}^{n_r}$ be a reference image set, $\mathcal{S} = \{\widetilde{x}_j\}_{j=1}^{n_g}$ an evaluated image set, and f a feature extractor, typically a pretrained model that maps each image to a lower-dimensional feature space.

D.1 FID

To assess how closely generated images match a reference distribution, we compare their distributions in feature space. A common approach compares their mean vectors and covariance matrices rather than the full distributions. Let (μ_r, Σ_r) and (μ_g, Σ_g) denote these empirical statistics for the reference and generated features, respectively. Their Fréchet distance is defined as follows:

$$\mathrm{FD}_f(\mathcal{R}, \mathcal{S}) = \|\mu_r - \mu_g\|_2^2 + \mathrm{tr}\Big(\Sigma_r + \Sigma_g - 2\big(\Sigma_r^{1/2}\Sigma_g\Sigma_r^{1/2}\big)^{1/2}\Big). \tag{63}$$

This is the squared Wasserstein-2 distance between the corresponding Gaussian approximations. Intuitively, it is the minimum average squared distance that probability mass must travel to transform one Gaussian distribution into the other.

FID (Heusel et al., 2017) computes the distance in Equation 63 using Inception-v3 features. Lower values indicate closer feature statistics. FID summarizes discrepancies in both fidelity and diversity, rather than separating them.

Here, conditional FID is ordinary FID between a clean reference set and reconstructions of the full dataset $\mathcal{D}_G \cup \mathcal{D}_B$, including both clean and corrupted images. For each dataset image u, we draw $z \sim \tilde{\mathbf{Q}}(\cdot \mid u)$ and $\widetilde{x} \sim p_\theta(\cdot \mid z)$, then compute FID over the resulting set of reconstructions.

D.2 Precision, recall, density, and coverage

Neighborhood-based metrics separate sample fidelity from coverage of the reference distribution. Let $r_i = f(x_i)$ and $g_j = f(\widetilde{x}_j)$ denote the reference and generated feature vectors, respectively. For a fixed neighborhood size k, define a_i as the Euclidean distance from r_i to its kth nearest neighbor among the other reference features $\{r_\ell : \ell \neq i\}$. Similarly, b_j is the distance from g_j to its kth nearest neighbor among the other generated features $\{g_\ell : \ell \neq j\}$. Thus, each radius is computed within its own set, not between the reference and generated sets. Define $B(u, a) = \{v : \|v - u\|_2 \leq a\}$, i.e., the Euclidean ball centered at u with radius a. Let $\mathcal{M}_r = \bigcup_i B(r_i, a_i)$ and $\mathcal{M}_g = \bigcup_j B(g_j, b_j)$. The precision and recall of Kynkäänniemi et al. (2019) are

$$\text{Precision} = \frac{1}{n_g}\sum_{j=1}^{n_g} \mathbf{1}\{g_j \in \mathcal{M}_r\}, \qquad \text{Recall} = \frac{1}{n_r}\sum_{i=1}^{n_r} \mathbf{1}\{r_i \in \mathcal{M}_g\}. \tag{64}$$

Precision measures the fraction of generated samples that are close to some reference sample; recall measures the fraction of reference samples that are close to some generated sample. Generating images outside the estimated reference support lowers precision, while missing modes of the reference distribution lowers recall. Both metrics lie in $[0, 1]$, with higher values being better.

Density and coverage refine these comparisons using reference neighborhoods alone (Naeem et al., 2020):

$$\begin{aligned} \text{Density} &= \frac{1}{kn_g}\sum_{j=1}^{n_g}\sum_{i=1}^{n_r} \mathbf{1}\{g_j \in B(r_i, a_i)\}, \\ \text{Coverage} &= \frac{1}{n_r}\sum_{i=1}^{n_r} \mathbf{1}\{\exists j :\ g_j \in B(r_i, a_i)\}. \end{aligned} \tag{65}$$

Density is the average number of reference neighborhoods containing a generated sample, divided by k; coverage is the fraction of reference neighborhoods containing at least one generated sample. Coverage lies in $[0, 1]$, whereas density is nonnegative and can exceed 1.

D.3 VENDI SCORE

The Vendi Score measures diversity within an image set without requiring a reference set (Friedman & Dieng, 2023). The matrix $K \in \mathbb{R}^{n_g \times n_g}$ contains pairwise similarities between image representations. For example, using cosine similarity between nonzero feature vectors $g_j = f(\widetilde{x}_j)$ gives

$$K_{j\ell} = \frac{g_j^\top g_\ell}{\|g_j\|_2 \|g_\ell\|_2}.$$

This is the Gram matrix of the normalized features, so it is positive-semidefinite with $K_{jj} = 1$. Let $\lambda_1, \ldots, \lambda_{n_g}$ be the eigenvalues of K/n_g. Then

$$\mathrm{VS}(\mathcal{S}) = \exp\left(-\sum_{j=1}^{n_g} \lambda_j \log \lambda_j \right), \qquad 0 \log 0 := 0. \tag{66}$$

The score ranges from one for identical representations to n_g for mutually orthogonal representations, and can be interpreted as an effective number of distinct samples under the chosen similarity. Higher values indicate greater diversity.

D.4 INCEPTION SCORE

Inception Score rewards confident image classifications together with diversity across predicted classes (Salimans et al., 2016). Let $p_{\mathrm{Inc}}(c \mid x)$ be the Inception classifier's class probabilities and $\overline{p}(c) = n_g^{-1} \sum_j p_{\mathrm{Inc}}(c \mid \widetilde{x}_j)$. The score is

$$\mathrm{IS}(\mathcal{S}) = \exp\left(\frac{1}{n_g} \sum_{j=1}^{n_g} D_{\mathrm{KL}}(p_{\mathrm{Inc}}(\cdot \mid \widetilde{x}_j) \,\|\, \overline{p}) \right). \tag{67}$$

Higher values indicate confident predictions distributed across more classes. We use it in Figure 4 as a rough proxy for recognizable image structure and class diversity under increasing noise, not as a direct measure of retained information or agreement with clean data.

D.5 NO-REFERENCE IMAGE-QUALITY METRICS

No-reference quality models assign a score to each image without a paired clean target. For a scoring model q, the set-level mean is $\overline{q}(\mathcal{S}) = n_g^{-1} \sum_j q(\widetilde{x}_j)$. Table 4 reports four such metrics on 50,000 randomly selected images per evaluated set. Higher scores indicate better predicted quality for each metric, but their numerical scales are not interchangeable. These learned assessments do not directly measure distributional coverage or reconstruction accuracy.

CLIP-IQA. CLIP-IQA (Wang et al., 2023) compares an image's CLIP embedding with opposing quality prompts. For normalized image embedding $u(x)$, positive and negative text embeddings t_+ and t_-, and logit scale α, its prompt-pair score is $q_{\mathrm{CLIP}}(x) = \exp(\alpha u(x)^\top t_+) / \big[\exp(\alpha u(x)^\top t_+) + \exp(\alpha u(x)^\top t_-)\big]$. We also use CLIP-IQA to rank ImageNet images when defining the clean subset in Section 5.2; this ranking is a quality proxy, not a ground-truth annotation of corruption.

ARNIQA. ARNIQA (Agnolucci et al., 2024) predicts quality from a representation learned to distinguish image distortions. Its encoder is trained self-supervised using synthetically distorted images, and a quality regressor maps the resulting features to a scalar score.

FGResQ. FGResQ (Sheng et al., 2026) assesses the perceptual quality of restored images. Its training combines coarse score regression with fine-grained quality ranking, targeting quality differences between restoration outputs. We report its scalar quality score as a complementary assessment of our ImageNet reconstructions.

TOPIQ-NR. The no-reference variant of TOPIQ (Chen et al., 2023) combines multiscale features through a top-down attention mechanism. High-level semantic features guide the assessment of lower-level distortions, producing an image-quality score without a reference image.

Q-Align. Q-Align (Wu et al., 2024) uses a large multimodal model to predict text-defined quality levels. It converts probabilities over the five levels, ordered from bad to excellent, into a continuous

score $q_{\text{QAlign}}(x) = \sum_{\ell=1}^{5} \ell\, p(\ell \mid x)$. This score reflects the model's predicted quality rating rather than a pixelwise comparison with a clean target.

E CONDITIONAL MODEL ARCHITECTURE

Code Snippet 1 gives the conditional EDM preconditioning wrapper. It adds a conditioning image, optionally scales it using its expected standard deviation, and passes it to a conditional U-Net alongside the noisy sample. Code Snippet 2 shows the corresponding U-Net wrapper, which concatenates the noisy and conditioning inputs before invoking the underlying model.

```
@persistence.persistent_class
class ConditionalEDMPrecond(torch.nn.Module, PyTorchModelHubMixin):
    def __init__(self,
        img_resolution,                 # Image resolution.
        img_channels,                   # Number of color channels of x and y.
        label_dim       = 0,            # Number of class labels; 0 = unconditional.
        use_fp16        = False,        # Execute the model at FP16 precision?
        sigma_min       = 0,            # Minimum supported noise level.
        sigma_max       = float('inf'), # Maximum supported noise level.
        sigma_data      = 0.5,          # Expected standard deviation of training data.
        cond_image_sigma = None,        # Std of y; None leaves y unscaled.
        model_type      = 'SongUNet',   # Underlying model class.
        **model_kwargs,                 # Arguments for the underlying model.
    ):
        super().__init__()
        self.img_resolution = img_resolution
        self.img_channels = img_channels
        self.label_dim = label_dim
        self.use_fp16 = use_fp16
        self.sigma_min = sigma_min
        self.sigma_max = sigma_max
        self.sigma_data = sigma_data
        self.cond_image_sigma = cond_image_sigma
        self.model = ConditionalUNet(
            num_return_channels=img_channels,
            model_type=model_type,
            img_resolution=img_resolution,
            in_channels=img_channels * 2,
            out_channels=img_channels * 2,
            label_dim=label_dim,
            **model_kwargs,
        )

    def forward(self, x, y, sigma, class_labels=None,
                force_fp32=False, **model_kwargs):
        x = x.to(torch.float32)
        y = y.to(torch.float32)
        sigma = sigma.to(torch.float32).reshape(-1, 1, 1, 1)
        class_labels = (
            None if self.label_dim == 0 else
            torch.zeros([1, self.label_dim], device=x.device)
            if class_labels is None else
            class_labels.to(torch.float32).reshape(-1, self.label_dim)
        )
        dtype = (
            torch.float16
            if self.use_fp16 and not force_fp32 and x.device.type == 'cuda'
            else torch.float32
        )
        c_skip = self.sigma_data ** 2 / (
            sigma ** 2 + self.sigma_data ** 2
        )
        c_out = sigma * self.sigma_data / (
            sigma ** 2 + self.sigma_data ** 2
        ).sqrt()
        c_in = 1 / (
            self.sigma_data ** 2 + sigma ** 2
        ).sqrt()
        c_noise = sigma.log() / 4

        cond_image_sigma = getattr(self, 'cond_image_sigma', None)
        if cond_image_sigma is None:
            y_in = y
        else:
            c_in_y = 1 / cond_image_sigma
            y_in = c_in_y * y

        F_x = self.model(
            (c_in * x).to(dtype), y_in.to(dtype),
            c_noise.flatten(), class_labels=class_labels,
            **model_kwargs,
        )
        assert F_x.dtype == dtype
        D_x = c_skip * x + c_out * F_x.to(torch.float32)
        return D_x
```

```
    def round_sigma(self, sigma):
        return torch.as_tensor(sigma)
```

Code Snippet 1: Conditional EDM preconditioning wrapper. Red highlights the conditioning-specific modifications.

```python
@persistence.persistent_class
class ConditionalUNet(torch.nn.Module, PyTorchModelHubMixin):
    def __init__(self,
        num_return_channels,          # Channels retained from the model output.
        model_type = 'SongUNet',      # Underlying model class to wrap.
        **model_kwargs,               # Arguments for the underlying model;
                                      # input/output channels account for
                                      # concatenated x and y, e.g. 3 + 3 = 6.
    ):
        super().__init__()
        self.num_return_channels = num_return_channels
        self.model = globals()[model_type](**model_kwargs)

    def forward(self, x, y, emb, **model_kwargs):
        xy = torch.cat([x, y], dim=1)
        out = self.model(xy, emb, **model_kwargs)
        return out[:, :self.num_return_channels]
```

Code Snippet 2: Conditional U-Net wrapper that concatenates the noisy sample and conditioning image before invoking the underlying model.

F CIFAR-10 Experimental Sweeps

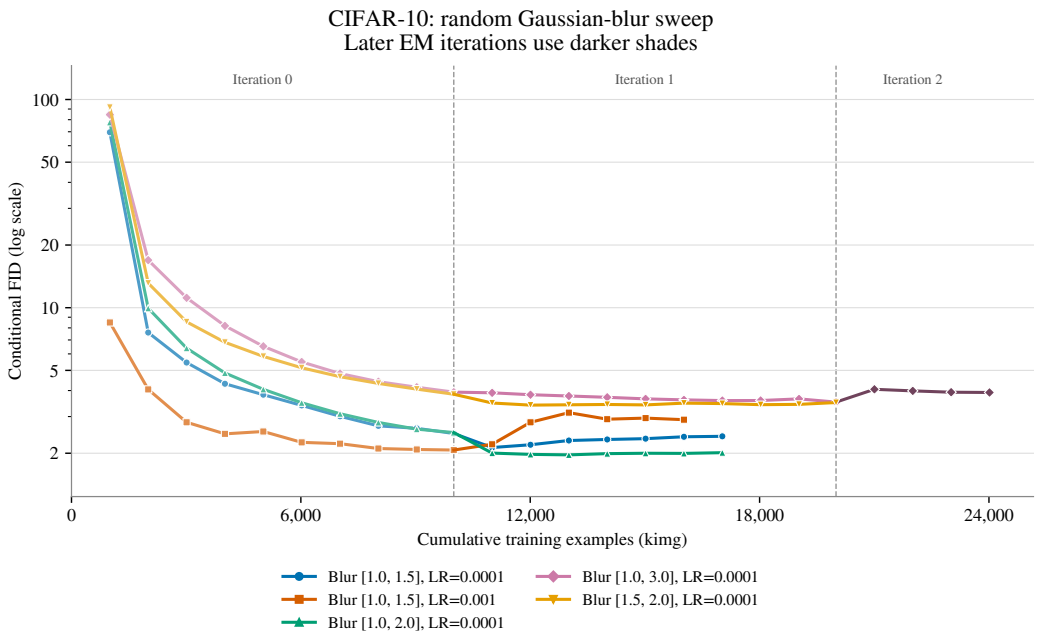


Figure 15: CIFAR-10 conditional FID for random Gaussian-blur auxiliary channels (Section 5.1). Legends specify the uniform blur-strength interval and learning rate. EM iterations continue along a cumulative training axis; darker shades denote later iterations, separated by dashed lines.

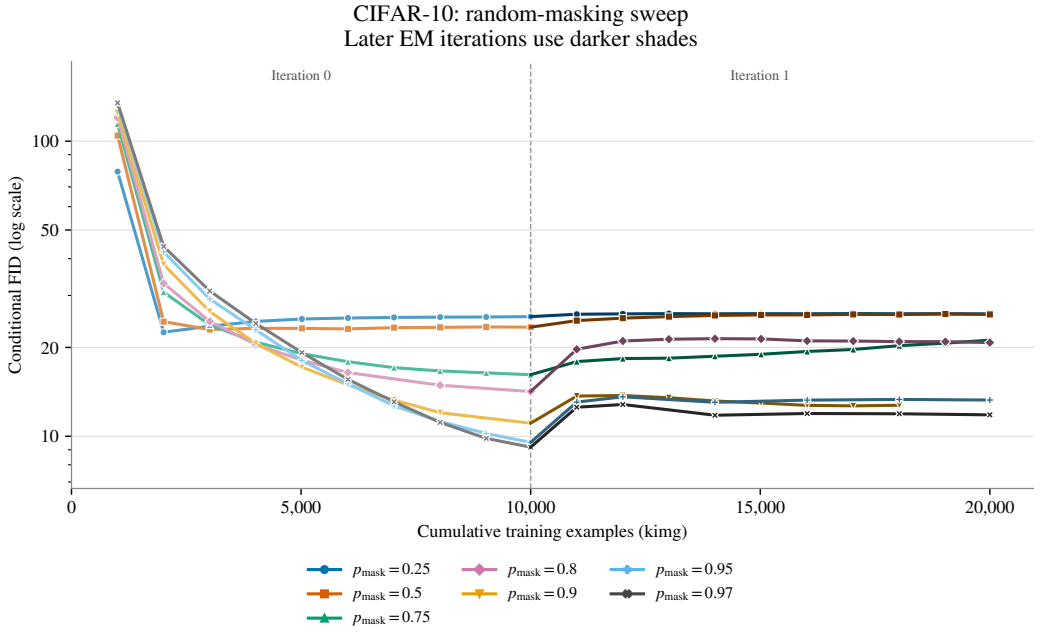


Figure 16: CIFAR-10 conditional FID for random-masking auxiliary channels with different masking probabilities (Section 5.1). EM iterations continue along a cumulative training axis; darker shades denote later iterations, separated by the dashed line.

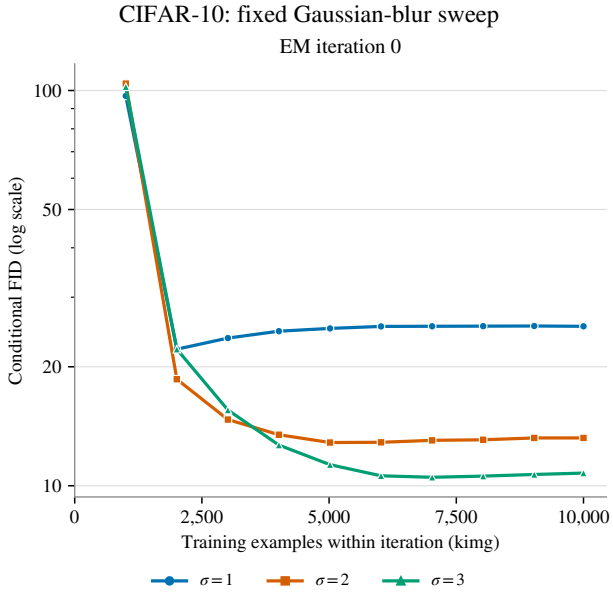


Figure 17: CIFAR-10 conditional FID for fixed Gaussian-blur auxiliary channels with different blur strengths during EM iteration 0 (Section 5.1).

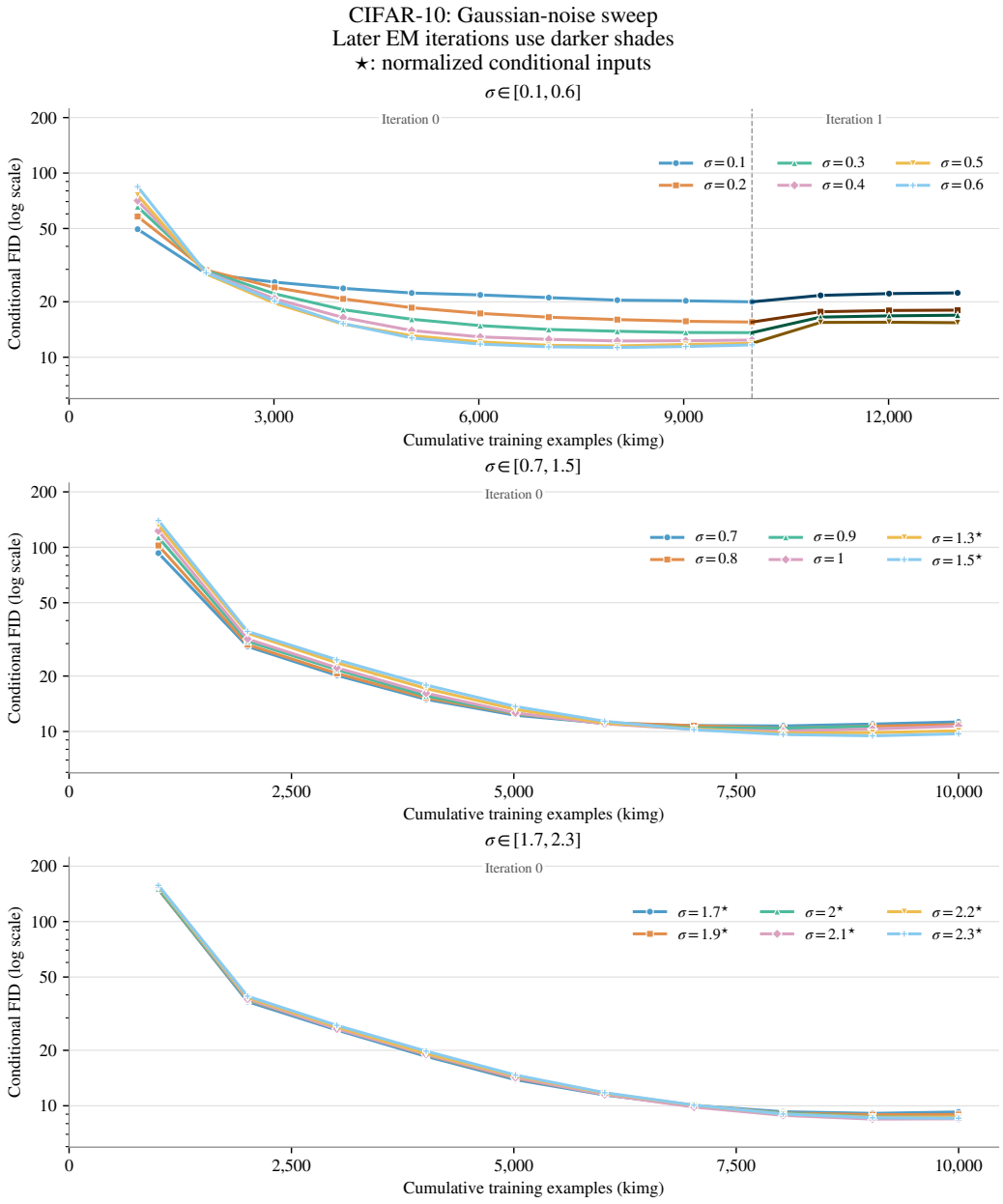


Figure 18: CIFAR-10 conditional FID for Gaussian-noise auxiliary channels (Section 5.1). Panels group runs by noise strength. EM iterations continue along a cumulative training axis; darker shades denote later iterations, separated by the dashed line. Stars denote normalized conditional inputs.



Figure 19: CelebA corruptions used in our experiments. From left to right, the columns show clean images, adversarially face-redacted observations, the fixed circular auxiliary mask applied to clean and redacted images, and the random-shape auxiliary mask applied to clean and redacted images.

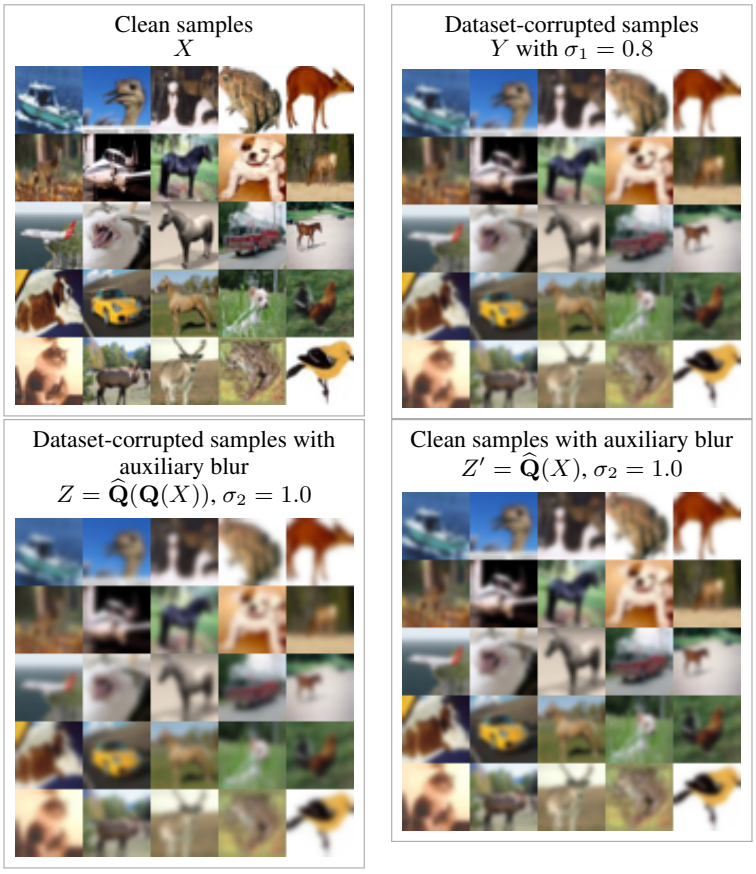


Figure 20: Qualitative fixed-blur comparison corresponding to Section 5.1. The top row shows clean samples X and dataset-corrupted observations Y with blur level $\sigma_1 = 0.8$. The bottom row compares $Z = \widehat{\mathbf{Q}}(\mathbf{Q}(X))$ with $Z' = \widehat{\mathbf{Q}}(X)$ after applying a fixed auxiliary blur with $\sigma_2 = 1.0$. The two bottom grids appear visually similar, even though fixed blur yields poor generative performance.

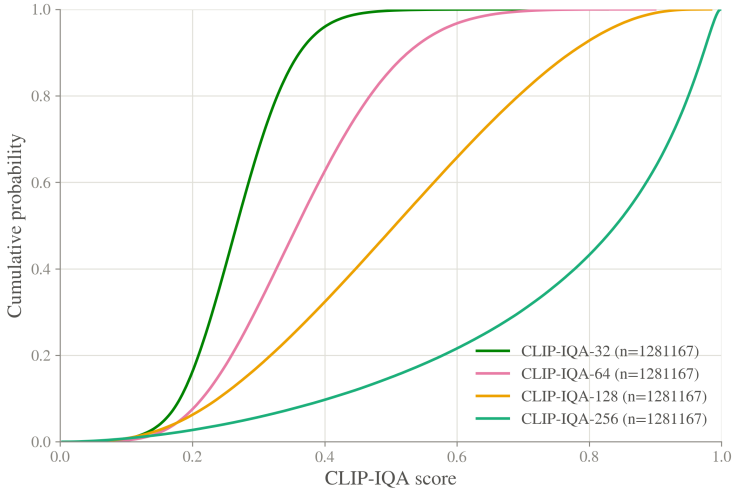
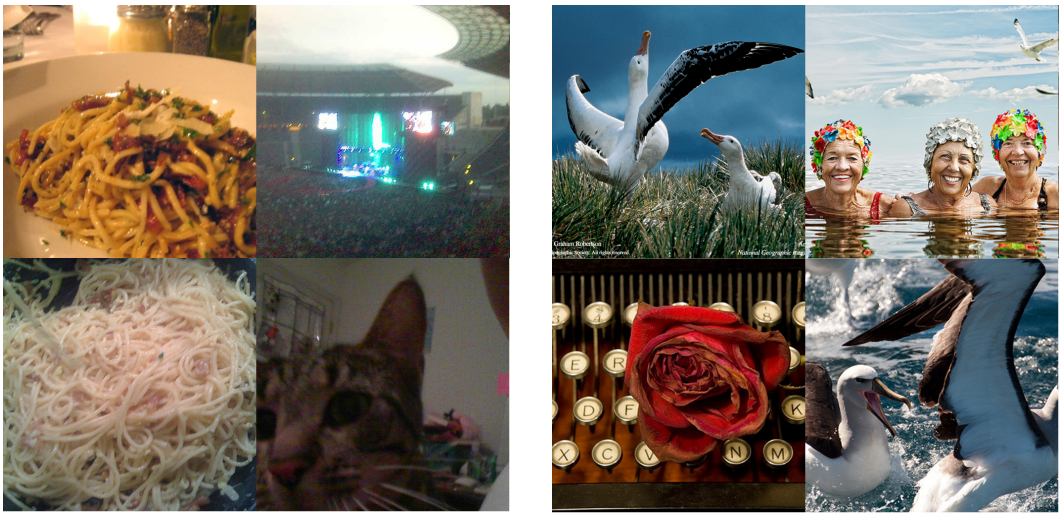


Figure 21: Empirical CDFs of CLIP-IQA scores on ImageNet-256 at input resolutions 32, 64, 128, and 256. Each curve contains 1,281,167 images.



(a) Low CLIP-IQA scores

(b) High CLIP-IQA scores

Figure 22: ImageNet examples assigned low (left) and high (right) image-quality scores by CLIP-IQA, illustrating the natural quality variation discussed in Section 5.2. Figure taken from Ambient Omni (Daras et al., 2025c).

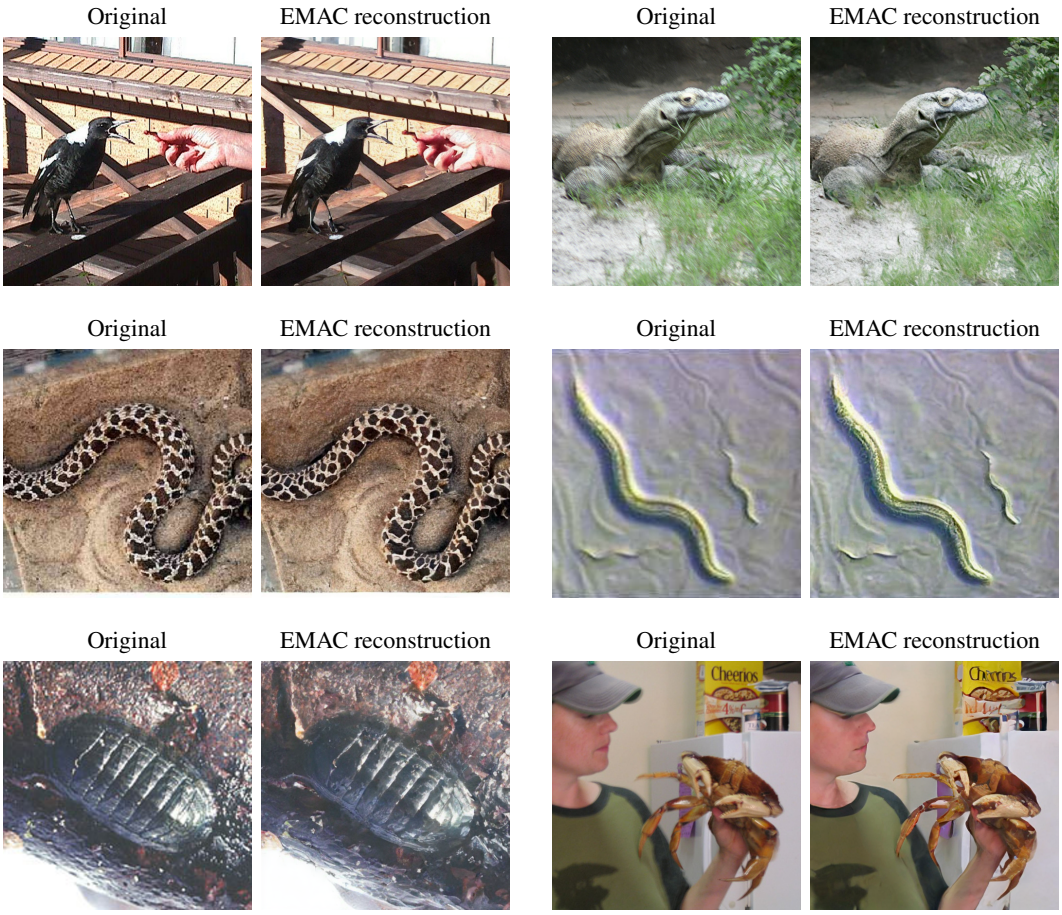


Figure 23: Additional ImageNet examples from Section 5.2 (group 1 of 7). Each pair shows the original image (left) and its EMAC reconstruction (right).



Figure 24: Additional ImageNet examples from Section 5.2 (group 2 of 7). Each pair shows the original image (left) and its EMAC reconstruction (right).

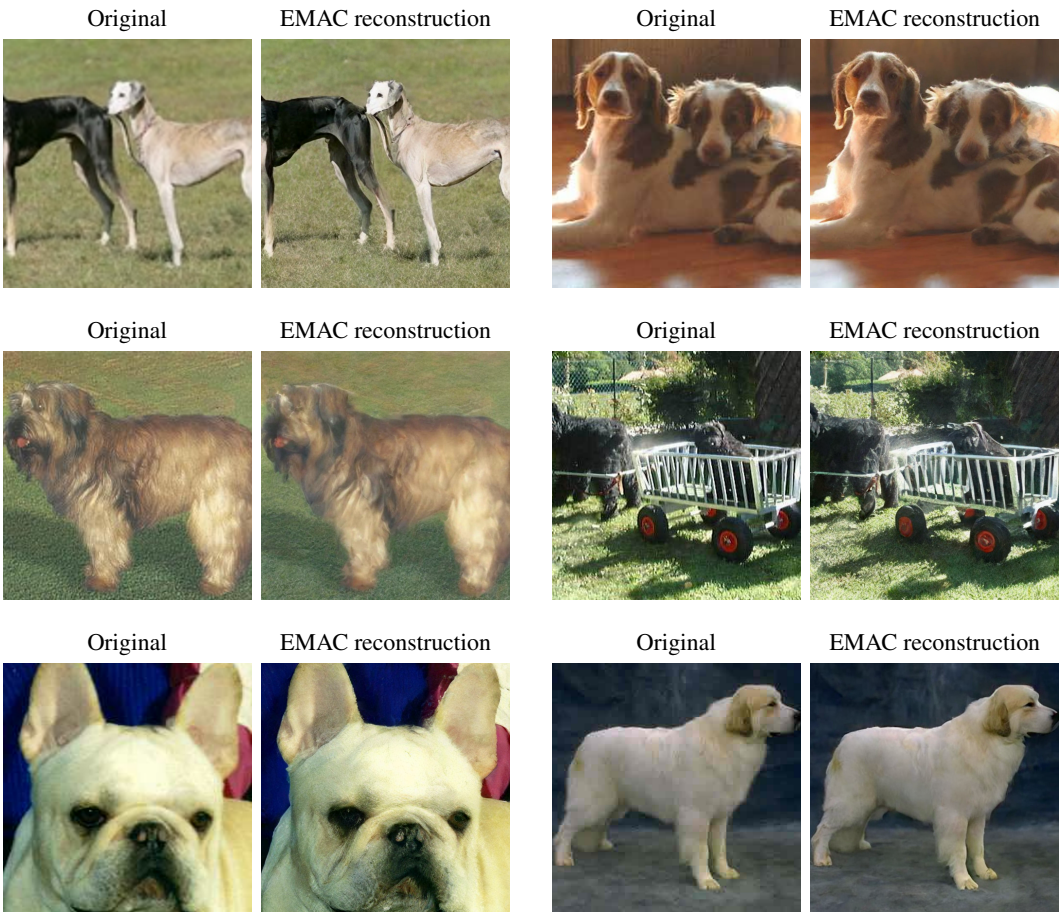


Figure 25: Additional ImageNet examples from Section 5.2 (group 3 of 7). Each pair shows the original image (left) and its EMAC reconstruction (right).

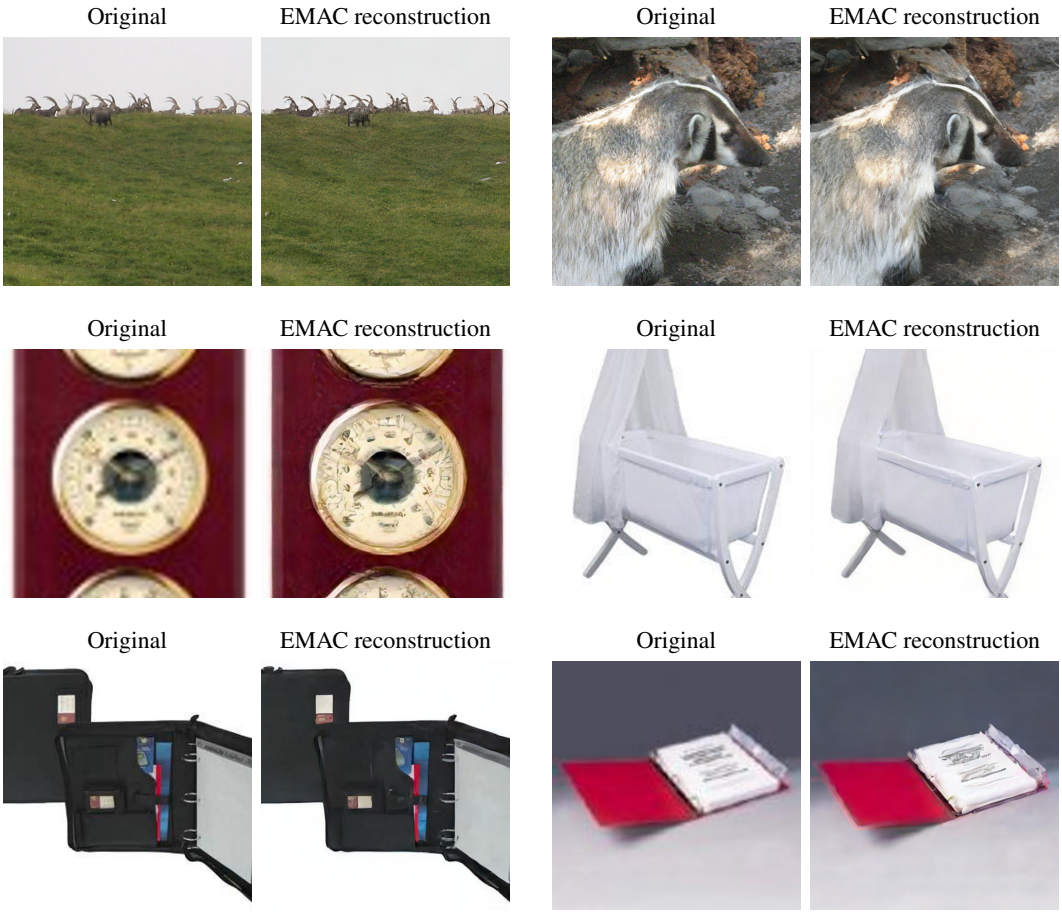


Figure 26: Additional ImageNet examples from Section 5.2 (group 4 of 7). Each pair shows the original image (left) and its EMAC reconstruction (right).

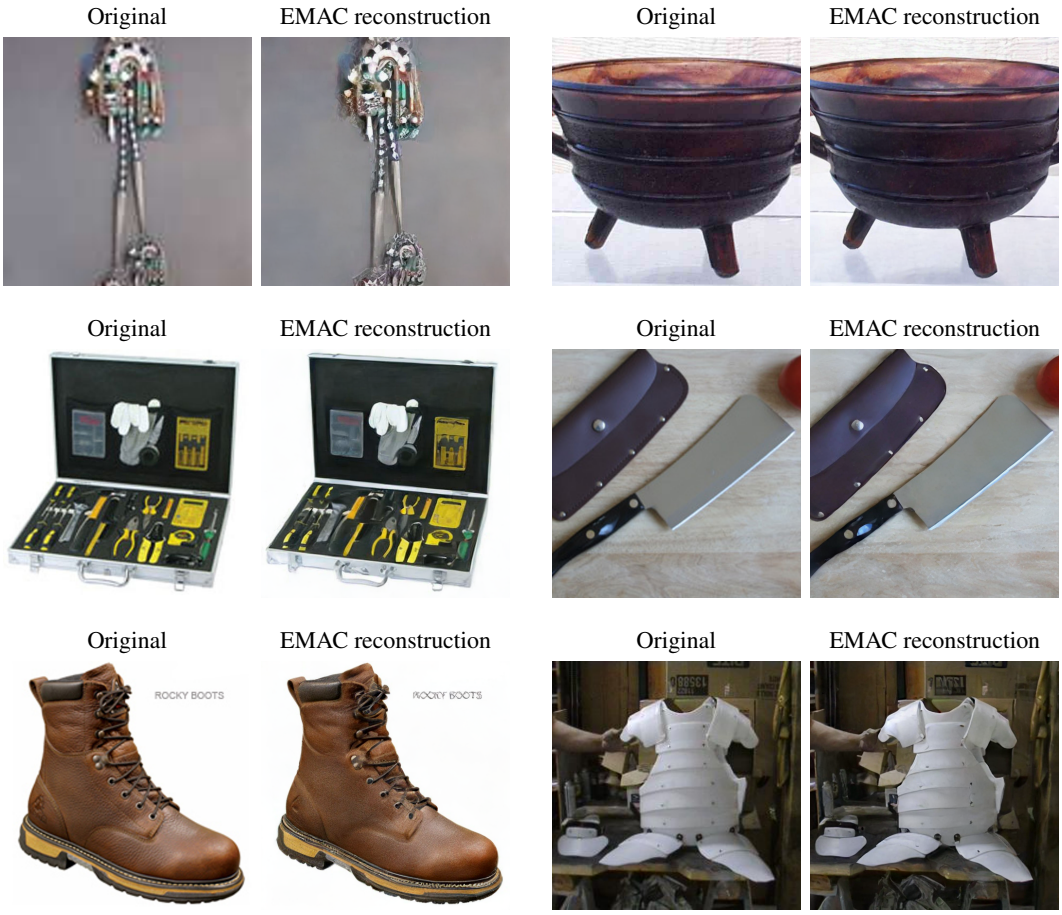


Figure 27: Additional ImageNet examples from Section 5.2 (group 5 of 7). Each pair shows the original image (left) and its EMAC reconstruction (right).

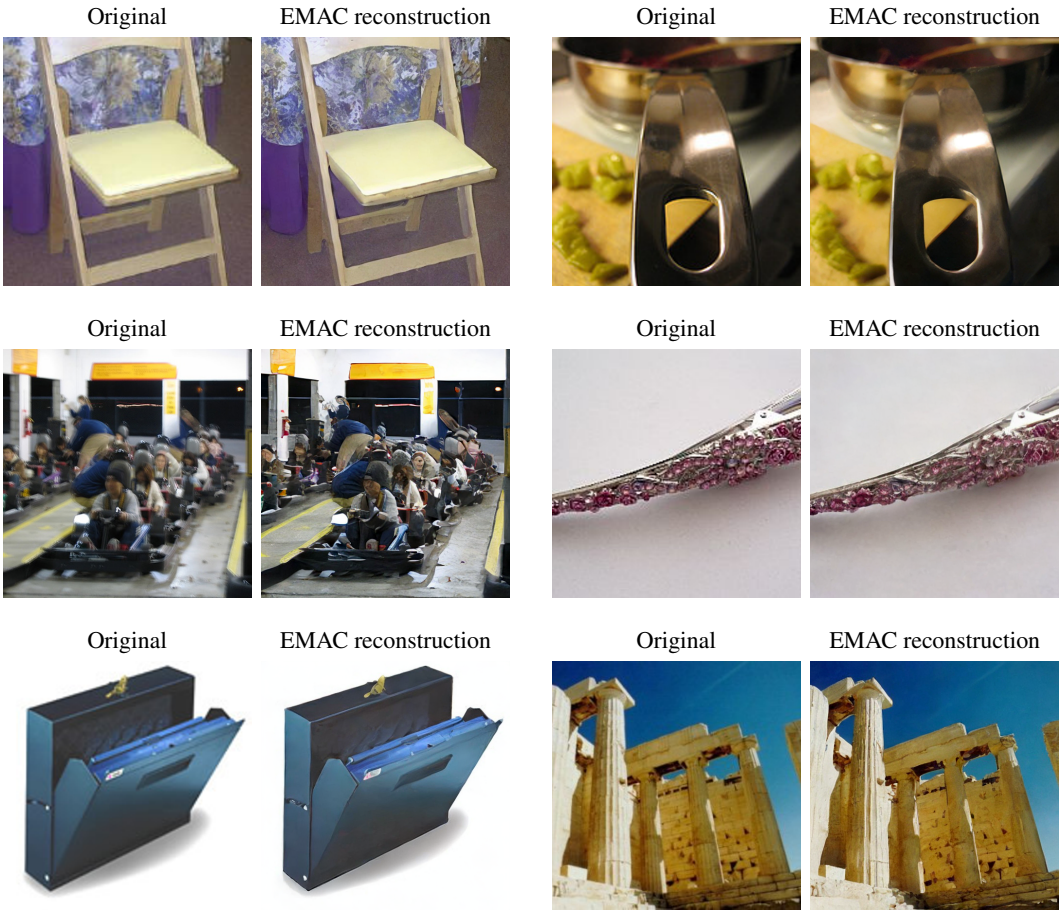


Figure 28: Additional ImageNet examples from Section 5.2 (group 6 of 7). Each pair shows the original image (left) and its EMAC reconstruction (right).

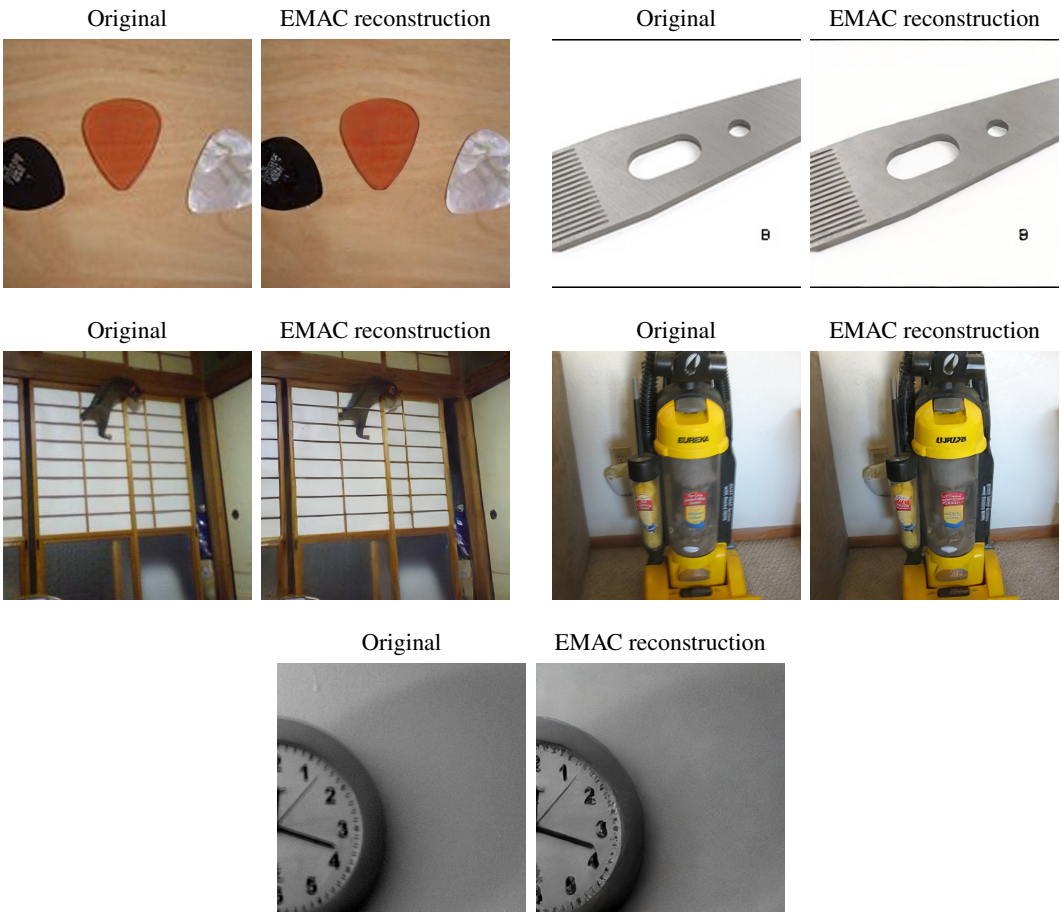


Figure 29: Additional ImageNet examples from Section 5.2 (group 7 of 7). Each pair shows the original image (left) and its EMAC reconstruction (right).