Ambient Diffusion Omni: Training Good Models with Bad Data

Giannis Daras *
Massachusetts Institute of Technology
gdaras@mit.edu

Adrian Rodriguez-Munoz *
Massachusetts Institute of Technology
adrianrm@mit.edu

Adam Klivans
The University of Texas at Austin
klivans@utexas.edu

Antonio Torralba
Massachusetts Institute of Technology
torralba@mit.edu

Constantinos Daskalakis
Massachusetts Institute of Technology
costis@csail.mit.edu

Abstract

We show how to use low-quality, synthetic, and out-of-distribution images to improve the quality of a diffusion model. Typically, diffusion models are trained on curated datasets that emerge from highly filtered data pools from the Web and other sources. We show that there is immense value in the lower-quality images that are often discarded. We present Ambient Diffusion Omni, a simple, principled framework to train diffusion models that can extract signal from all available images during training. Our framework exploits two properties of natural images – spectral power law decay and locality. We first validate our framework by successfully training diffusion models with images synthetically corrupted by Gaussian blur, JPEG compression, and motion blur. We then use our framework to achieve state-of-the-art ImageNet FID and we show significant improvements in both image quality and diversity for text-to-image generative modeling. The core insight is that noise dampens the initial skew between the desired high-quality distribution and the mixed distribution we actually observe. We provide rigorous theoretical justification for our approach by analyzing the trade-off between learning from biased data versus limited unbiased data across diffusion times.

1 Introduction

Large-scale, high-quality training datasets have been a primary driver of recent progress in generative modeling. These datasets are typically assembled by filtering massive collections of images sourced from the web or proprietary databases [16, 31, 37]. The filtering process—which determines which data is retained—is crucial to the quality of the resulting models [8, 18, 16]. However, filtering strategies are often heuristic and inefficient, discarding large amounts of data [35, 31, 16, 8]. We demonstrate that the data typically rejected as low-quality holds significant, underutilized value.

Extracting meaningful information from degraded data requires algorithms that explicitly model the degradation process. In generative modeling, there is growing interest in approaches that learn to generate directly from degraded inputs [12, 11, 9, 10, 5, 33, 27, 36, 3, 1, 2, 39, 49]. A key limitation

*Equal contribution.

Preprint.

of existing methods, however, is their reliance on knowing the exact form of the degradation. In real-world scenarios, image degradations—such as motion blur, sensor artifacts, poor lighting, and low resolution—are often complex and lack a well-defined analytical description, making this assumption unrealistic. Even within the same dataset, from ImageNet to internet scale text-to-image datasets, there are samples of heterogeneous qualities [19], as shown in Figures 4, 25, 28, 26. Given access to this mixed-bag of datapoints, we would like to sample from a tilted continuous measure of high-quality images, without sacrificing the diversity present in the training points.

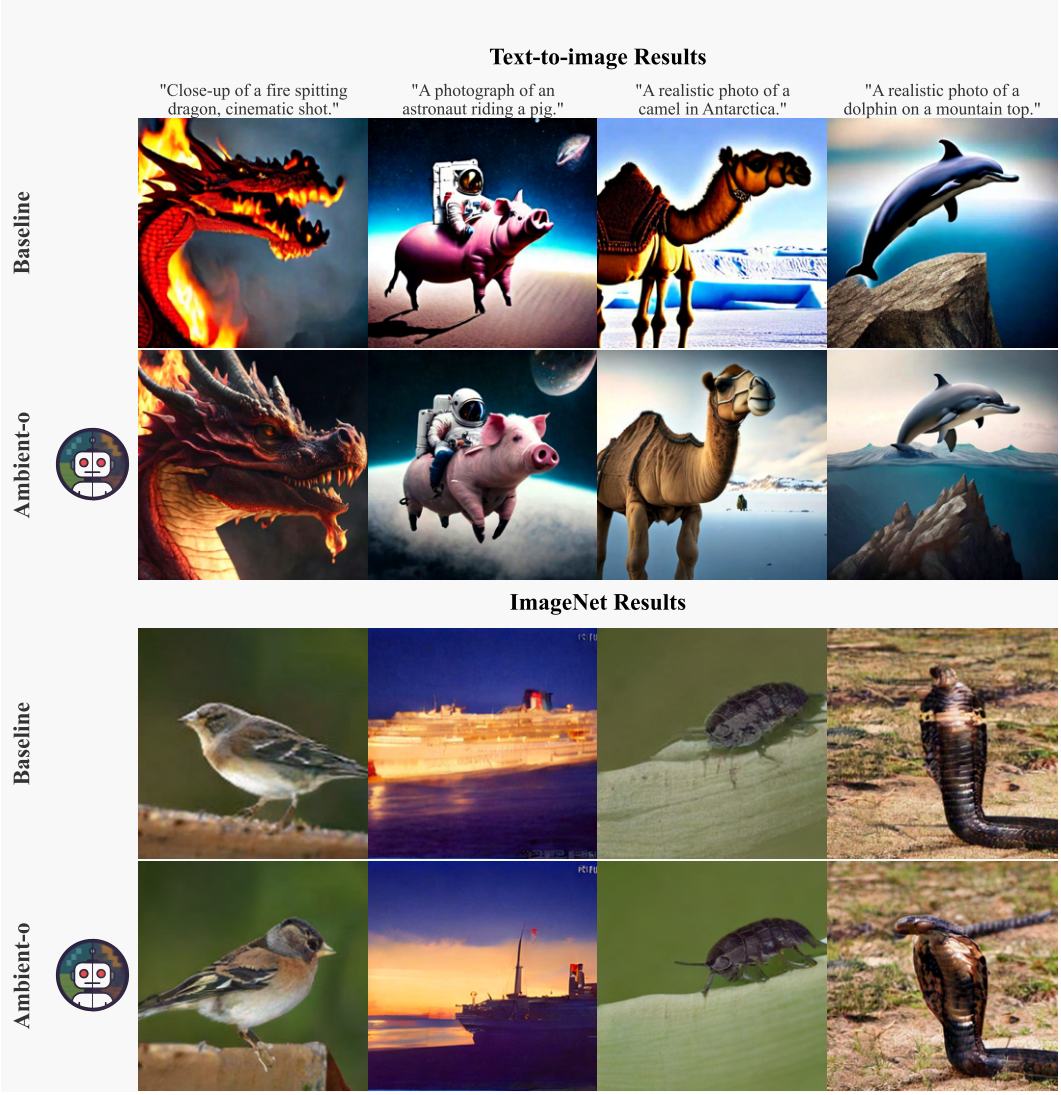


Figure 1: **Effect of using Ambient-o for (a) training a text-to-image model (Micro-Diffusion [38]) and (b) a class-conditional model for ImageNet (EDM-2 [25]).** All generations are initialized with the same noise. The baseline models are trained using all the data equally. Ambient-o changes the way the data is used during the diffusion process based on its quality. This leads to significant visual improvements without sacrificing diversity, as would happen with a filtering approach (see Fig. 7).

The training objective of diffusion models [21, 43, 44] naturally decomposes sampling from a target distribution into a sequence of supervised learning tasks. Due to the power-law structure of natural image spectra [46], high diffusion times focus on generating globally coherent, semantically meaningful content [15], while low diffusion times emphasize learning high-frequency details.

Our first key theoretical insight is that low-quality samples can still be valuable for training in the high-noise regime. As noise increases, the diffusion process contracts distributional differences (see

Theorem 4.2), reducing the mismatch between the high-quality target distribution and the available mixed-quality data. At the same time, incorporating low-quality data increases the sample size, reducing the variance of the learned estimator. Our analysis formalizes this bias–variance trade-off and motivates a principled algorithm for training denoisers at high diffusion times using noisy, heterogeneous data.

For low diffusion times, our algorithm leverages a second key property of natural images: locality. We show a direct relationship between diffusion time and the optimal receptive field size for denoising. Specifically, small image crops suffice at lower noise levels. This allows us to borrow high-frequency details from out-of-distribution or synthetic images, as long as the marginal distributions of the crops match those of the target data.

We introduce Ambient Diffusion Omni (Ambient-o), a simple and principled framework for training diffusion models using arbitrarily corrupted and out-of-distribution data. Rather than filtering samples based on binary 'good' or 'bad' labels, Ambient-o retains all data and modulates the training process according to each sample's utility. This enables the model to generate diverse outputs without compromising image quality. Empirically, Ambient-o advances the state of the art in unconditional generation on ImageNet and enhances diversity in text-conditional generation without sacrificing fidelity. Theoretically, it achieves improved bounds for distribution learning by optimally balancing the bias–variance trade-off: low-quality samples introduce bias, but their inclusion reduces variance through increased sample size.

We will release all our code and trained models in the following URL: https://github.com/giannisdaras/ambient-omni.

2 Background and Related Work

Diffusion Modeling. Diffusion models transform the problem of sampling from p_0 into the problem of learning *denoisers* for smoothed versions of p_0 defined as $p_t = p_0 \circledast \mathcal{N}(0, \sigma^2(t)\mathrm{I})$. We typically denote with $X_0 \sim p_0$ the R.V. distributed according to the distribution of interest and $X_t = X_0 + \sigma(t)Z$, the R.V. distributed according to p_t. The target is to estimate the set of optimal l_2 denoisers, i.e., the set of the conditional expectations: $\{\mathbb{E}[X_0|X_t = \cdot]\}_{t=1}^T$. Typically, this can be achieved through supervised learning by minimizing the following loss (or a re-parametrization of it):

$$J(\theta) = \mathbb{E}_{t\in\mathcal{U}[0,T]}\mathbb{E}_{x_0,x_t|t}\left[||h_\theta(x_t,t) - x_0||^2\right], \tag{2.1}$$

that is optimized over a function family $\mathcal{H} = \{h_\theta : \theta \in \Theta\}$ parametrized by network parameters θ. For sufficiently expressive families, the minimizer is indeed: $h_{\theta^*}(x,t) = \mathbb{E}[X_0|X_t = x]$.

Learning from noisy data. The diffusion modeling framework described above assumes access to samples from the distribution of interest p_0. An interesting variation of this problem is to learn to sample from p_0 given access to samples from a tilted measure $\tilde{p}_0$ and a known degradation model. In Ambient Diffusion [12], the goal is to sample from p_0 given pairs (Ax_0, A) for a matrix $A : \mathbb{R}^{m\times n}, m < n$, that is distributed according to a known density $p(A)$. The techniques in this work were later generalized to accommodate additive Gaussian Noise [10, 11, 1] in the measurements. More recently there have been efforts to further broaden the family of degradation models considered through Expectation-Maximization approaches that involve multiple training runs [36, 3].

Recent work from [11] has shown that, at least for the Gaussian corruption model, leveraging the low-quality data can tremendously increase the performance of the trained generative models. In particular, the authors consider the setting where we have access to a few samples from p_0, let's denote them $\mathcal{D}_0\{x_0^{(i)}\}_{i=1}^{N_1}$ and many samples from p_{t_n}, let's denote them $\mathcal{D}_{t_n}\{x_{t_n}^{(i)}\}_{i=1}^{N_2}$, where $p_{tn} = p_0 \circledast \mathcal{N}(0, \sigma^2(t_n)\mathrm{I})$ is a smoothed version of p_0 at a known noise level t_n. The clean samples are used to learn denoisers for all noise levels $t \in [0, T]$ while the noisy samples are used to learn denoisers only for $t \geq t_n$, using the training objective:

$$J_{\text{ambient}}(\theta) = \mathbb{E}_{t\in\mathcal{U}(t_n,T]}\sum_{i=1}^{N_2}\mathbb{E}_{x_t|x_{t_n}^{(i)}}\left[\left|\left|\alpha(t)h_\theta(x_t,t) + (1-\alpha(t))x_t - x_{t_n}^{(i)}\right|\right|^2\right], \tag{2.2}$$

with $\alpha(t) = \frac{\sigma^2(t)-\sigma^2(t_n)}{\sigma^2(t)}$. Note that the objective of equation 2.2 only requires samples from p_{t_n} (instead of p_0) and can be used to train for all times $t \geq t_n$. This algorithm uses $N_1 + N_2$ datapoints

to learn denoisers for $t > t_n$ and only N_1 datapoints to learn denoisers for $t \leq t_n$. The authors show that even for $N_1 << N_2$, the model performs similarly to the setting of training with $(N_1 + N_2)$ clean datapoints. The main limitation of this method and its related works is that the degradation process needs to be known. However, in many applications, we have data from heterogeneous sources and various qualities, but there is no analytic form or any prior on the corruption model.

Data filtering. One of the most crude, but widely used, approaches for dealing with heterogeneous data sources is to remove the low-quality data and train only the high-quality subset [31]. While this yields better results than naively training on the entire distribution, it leads to a decrease in diversity and relies on heuristics for optimizing the filtering. An alternative strategy is to train on the entire distribution and then fine-tune on high-quality data [8, 38]. This approach better trades the quality-diversity trade-off but still incurs a loss of diversity and is hard to calibrate.

3 Method

We propose a new framework that extends beyond [11] to enable training generative models directly from arbitrarily corrupted and out-of-distribution data, without requiring prior knowledge of the degradation process. We begin by formalizing the setting of interest.

Problem Setting. We are given a dataset $\mathcal{D} = \{w_0^{(i)}\}_{i=1}^N$ consisting of N datapoints. Each point in $\mathcal{D}$ is drawn from a mixture distribution $\tilde{p}_0$, which mixes p_0 (the distribution of interest) and an alternative distribution q_0 that may contain various forms of degradation or out-of-distribution content. We assume access to two labeled subsets, S_G, S_B, where points in S_G are known to come from the clean distribution p_0, and points in S_B from the corrupted distribution q_0. While this assumption simplifies the initial exposition, we relax it in Section E.1. We focus on the practically relevant regime where $|S_G| \ll |\mathcal{D}|$—i.e., access to high-quality data is severely limited. The objective is to learn a generative model that (approximately) samples from the clean distribution p_0, leveraging both clean and corrupted samples in its training.

We now describe how degraded and out-of-distribution samples can be effectively leveraged during training in both the high-noise and low-noise regimes of the diffusion process.

3.1 Learning in the high-noise regime (leveraging low-quality data)

Addition of gaussian noise contracts distribution distances. The first key idea of our method is that, at high diffusion times t, the noised target distribution p_t and the noised corrupted distribution $\tilde{p}_t$ become increasingly similar (Theorem 4.2), effectively attenuating the discrepancy introduced by corruption. This effect is illustrated in Figure 2 (top), where we compare a clean image and its degraded counterpart (in this case, corrupted by Gaussian blur). As the diffusion time t increases, the noised versions of both samples become visually indistinguishable. Consequently, samples from $\tilde{p}_0$ can be leveraged to learn (the score of) p_t, for $t > t_n^{\min}$. We formalize this intuition in Section 4, and we also quantify that for large t there are statistical efficiency benefits for using a large sample from $\tilde{p}_0$ versus a small sample from p_0 .

Heuristic selection of the noise level. From the discussion so far, it follows that to use samples from $\tilde{p}_0$, we need to assign them to a noise level $t_n^{\min}$. One can select this noise level empirically, i.e. we can ablate this parameter by training different models and selecting the one that maximizes the generative performance. However, this approach requires multiple trainings, which can be costly. Instead, we can find the desired noise level in a principled way as detailed below.

Training a classifier under additive Gaussian noise. To identify the appropriate noise level, we train a time-conditional classifier to distinguish between the noised distributions p_t and q_t across various diffusion times t. We use a single neural network $c_\theta^{\text{noise}}(x_t, t)$ that is conditioned on the diffusion time t, following the approach of time-aware classifiers used in classifier guidance [14]. The classifier is trained using labeled samples from S_G (clean) and S_B (corrupted) via the following objective:

$$J_{\text{noise}}(\theta) = \sum_{x_0 \in S_G} \mathbb{E}_{x_t|x_0} \left[-\log c_\theta^{\text{noise}}(x_t, t) \right] + \sum_{y_0 \in S_B} \mathbb{E}_{y_t|y_0} \left[-\log(1 - c_\theta^{\text{noise}}(y_t, t)) \right].$$

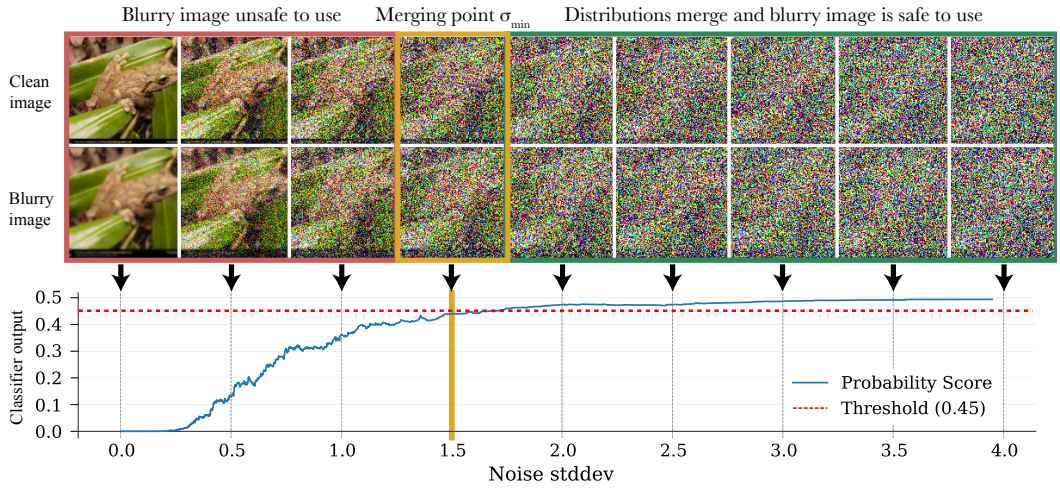


Figure 2: **A time-dependent classifier trained to distinguish noisy clean and blurry images** (blur kernel standard deviation $\sigma_B = 0.6$). At low noise the classifier is able to perfectly identify the blurry images, and outputs a probability close to 0. As the noise increases and the information in the image is destroyed, the clean and blurry distributions converge and the classifier outputs a prediction close to 0.5. The red line plots the threshold (selected at $\tau = 0.45$), which is crossed at $\sigma_t = 1.64$.

Annotation. Once the classifier is trained, we use it to determine the minimal level of noise that must be added to the low-quality distribution q_0 so that it closely approximates a smoothed version of the high-quality distribution p_0. Formally, we compute:

$$t_n^{\min} = \inf \left\{ t \in [0,T] : \frac{1}{|S_B|} \sum_{y_0 \in S_B} \mathbb{E}_{y_t|y_0} \left[c_\theta^{\text{noise}}(y_t, t) \right] > \tau \right\}, \tag{3.1}$$

for $\tau = 0.5 - \epsilon$ and for some $\epsilon > 0$. Subsequently, we form the annotated dataset $\mathcal{D}_{\overline{\text{annot}}} = \{(w_0^{(i)} + \sigma_{t_n^{\min}} Z^{(i)}, t_n^{\min})\}_{i=1}^N \cup \{(x_0, 0) | x_0 \in S_G\}$, where the random variables $Z^{(i)}$ are i.i.d. standard normals. In particular, our annotated dataset indicates that we should only use the samples from $\mathcal{D}$ for diffusion times $t \geq t_n^{\min}$, for which the distributions have approximately merged and hence it is safe to use them. In fact, the optimal classifier assigns time t_n that corresponds to the first time for which $\mathrm{d_{TV}}(p_t, q_t) \leq \epsilon$.

Sample dependent annotation. One potential issue with the aforementioned annotation approach is that all the samples in $\mathcal{D}$ are treated equally. But, as we noted, the points in $\mathcal{D}$ could be drawn from a distribution $\tilde{p}_0$ that mixes p_0 and q_0. In this case, all the samples in $\mathcal{D}$ that came from the p_0 component, will still get a high annotation time, leading to information loss. Instead, we can opt-in for a sample-wise annotation scheme, where each sample $w_0^{(i)}$ gets assigned a time $t_i^{\min}$ based on: $t_i^{\min} = \inf\{t \in [0,T] : \mathbb{E}_{w_t|w_0^{(i)}} \left[c_\theta^{\text{noise}}(w_t, t) \right] > \tau\}$, for $\tau = 0.5 - \epsilon$ and for some $\epsilon > 0$.

From arbitrary corruption to additive Gaussian noise. The afore-described approach reduces our problem of learning from data with arbitrary corruption to the setting of learning from data corrupted with additive Gaussian noise. The price we pay for this reduction is the information loss due to the extra noise we add to the samples during the annotation stage. We can now extend the objective function (2.2) to train our diffusion model. Suppose our annotated dataset is comprised of samples $\{(x^{(i)}_{t_i^{\min}}, t_i^{\min})\}$. Then our objective becomes:

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

where $\alpha(t, t_i^{\min}) = \frac{\sigma^2(t) - \sigma^2(t_i^{\min})}{\sigma^2(t)}$.

Learning something from nothing? The proposed framework comes with limitations worth considering. First, unless the diffusion noise level tends to infinity, the distributions p_t and q_t never fully converge—there is always a bias when treating samples from q_t as if they were from p_t. Moreover, the method is particularly well-suited to certain types of corruptions but is less effective for others. Because the addition of Gaussian noise suppresses high-frequency components—due to the spectral power law of natural images—our approach is most effective for corruptions that primarily degrade high frequencies (e.g., blur). In contrast, degradations that affect low-frequency content—such as color shifts, contrast reduction, or fog-like occlusions—are more challenging. This limitation is illustrated in Figure 3: masked images, for example, require significantly more noise to become usable compared to high-frequency corruptions like blur. In the extreme, the method reduces to a filtering approach, as infinite noise nullifies all information in the corrupted samples.

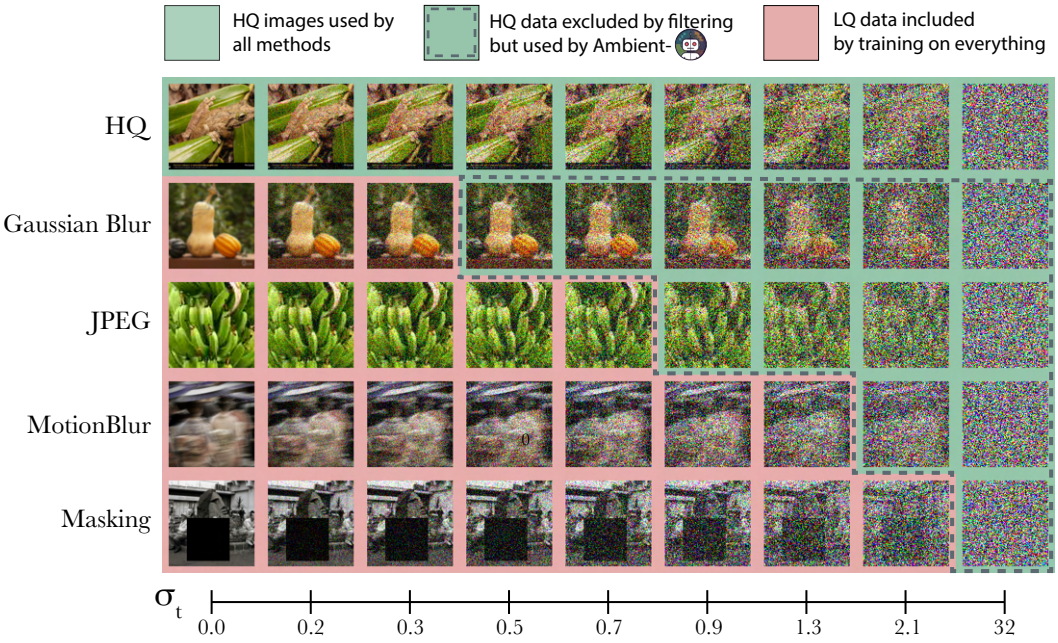


Figure 3: **Visual summary of our method for using low-quality data at high-noise.** We see how the various corrupted images become indistinguishable from the High Quality (HQ) after a minimum noise level. These noisy versions of Low Quality (LQ) images are actually high-quality data, which filtering approaches discard, but Ambient Omni uses.

3.2 Learning in the low-noise regime (synthetic and out-of-distribution data)

So far, our algorithm implicitly results in varying amounts of training data across diffusion noise levels. At high noise, the model can leverage abundant low-quality data, whereas at low noise levels, it must rely solely on the limited set of high-quality samples. We now extend the algorithm to enable the use of synthetic and out-of-distribution data for learning denoisers at low-noise diffusion times.

To achieve this, we leverage another fundamental property of natural images: *locality*. At low diffusion times, the denoising task can be solved using only a small local region of the image, without requiring full spatial context. We validate this hypothesis experimentally in the Experiments Section (Figures 15, 16, 17, 18), where we show that there is a mapping between diffusion time t and the crop size needed to perform the denoising optimally at this diffusion time. Intuitively, the higher the noise, the more context is required to accurately reconstruct the image. Conversely, for lower noise, the local information within a small neighborhood suffices to achieve effective denoising. We use $\mathrm{crop}(t)$ to denote the minimal crop size needed to perform optimal denoising at time t. If there are two distributions p_0 and $\tilde{p}_0$ that agree on their marginals (i.e. crops), they can be used interchangeably for low-diffusion times. Note that the distributions don't have to agree globally, they only have to agree on a local (patch) level. Formally, let $A(t)$ be a random patch selector of size $\mathrm{crop}(t)$. Let also

$p_0, \tilde{p}_0$ two distributions that satisfy:

$$A(t)\#p_0 = A(t)\#\tilde{p}_0, \tag{3.2}$$

where $A(t)\#p_0$ denotes the pushforward measure[2] of p_0 under $A(t)$. Then, the cropped portions of the tilted distributions provide equivalent information to the crops of the original distribution for denoising.

Training a crops classifier. Note that the condition of Equation (3.2) can be trivially satisfied if $A(t)$ masks all the pixels or even if $A(t)$ just selects a single pixel. We are interested in finding what is the maximum crop size for which this condition is approximately true. Once again, we can use a classifier to solve this task. The input to the classifier, c_θ^{crops}, is a crop of an image that either arises from p_0 or $\tilde{p}_0$, and the classifier needs to classify between these two cases.

Annotation and training using the trained classifier. Once the classifier is trained, we are now interested in finding the biggest crop size for which the distributions $p_0, \tilde{p}_0$ cannot be confidently distinguished. Formally,

$$t_n^{\max} = \sup \left\{ t \in [0, T] : \frac{1}{|S_B|} \sum_{y_0 \in S_B} [c_\theta^{\text{crops}}(A(t)(y_t))] > \tau \right\}, \tag{3.3}$$

for $\tau = 0.5 - \epsilon$ and for some $\epsilon > 0$. For times $t \leq t_n^{\max}$, the out-of-distribution images from $\tilde{p}_0$ can be used with the regular diffusion objective as images from p_0, as for these times the denoiser only looks at crops and at the crop level the distributions have converged.

The donut paradox. Each sample can be used for $t \geq t_i^{\min}$ and for $t \leq t_i^{\max}$, but not for $t \in (t_i^{\max}, t_i^{\min})$. We call this the *donut paradox* as there is a hole in the middle of the diffusion trajectory for which we have fewer available data. These times do not have enough noise for the distributions to merge globally, but also the required receptive field for denoising is big enough so that there are differences on a crop level. We show an example of this effect in Figure 13.

Table 1: ImageNet results with and without classifier-free guidance.

ImageNet-512	Train FID ↓				Test FID ↓				Model Size	
	FID		FIDv2		FID		FIDv2		Mparams	NFE
	no CFG	w/ CFG	no CFG	w/ CFG	no CFG	w/ CFG	no CFG	w/ CFG		
EDM2-XS	**3.57**	2.91	**103.39**	79.94	3.77	3.68	115.16	93.86	125	63
Ambient-o-XS	3.59	**2.89**	107.26	**79.56**	**3.69**	**3.58**	**115.02**	**92.96**	125	63
EDM2-XXL	1.91 (1.93)	1.81	42.84	33.09	2.88	2.73	56.42	46.22	1523	63
Ambient-o-XXL	1.99	1.87	43.38	33.34	2.81	2.68	56.40	46.02	1523	63
Ambient-o-XXL+crops	**1.91**	**1.80**	**42.84**	**32.63**	**2.78**	**2.53**	**56.39**	**45.78**	1523	63

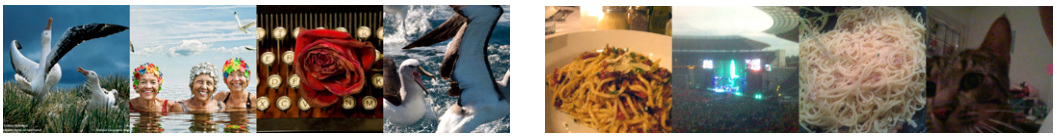


Figure 4: Results using CLIP to obtain the high-quality and the low-quality sets of ImageNet.

4 Theory

We study the 1-d case, but all our claims easily extend to any dimension. We compare two algorithms:

Algorithm 1. Algorithm 1 trains a diffusion model using access to n_1 samples from a target density p_0, assumed to be supported in $[0, 1]$ and be λ_1-Lipschitz.

Algorithm 2. Algorithm 2 trains a diffusion model using access to $n_1 + n_2$ samples from a density $\tilde{p}_0$ that is a mixture of the a target density p_0 and another density q_0, assumed to be supported in $[0, 1]$ and be λ_2-Lipschitz: $\tilde{p}_0 = \frac{n_1}{n_1+n_2} p_0 + \frac{n_2}{n_1+n_2} q_0$.

We want to compare how well these algorithms estimate the distribution $p_t := p_0 \circledast \mathcal{N}(0, \sigma_t^2)$. We use $\hat{p}_t^{(1)}, \hat{p}_t^{(2)}$ to denote the estimates obtained for p_t by Algorithms 1 and 2 respectively.

[2]Given measure spaces (X_1, Σ_1) and (X_2, Σ_2), a measurable function $f : X_1 \to X_2$, and a probability measure $p : \Sigma_1 \to [0, \infty)$, the pushforward measure $f\#p$ is defined as $(f\#p)(B) \coloneqq p(f^{-1}(B))\ \forall B \in \Sigma_2$.

Diffusion modeling is Gaussian kernel density estimation. We start by making a connection between the optimal solution to the diffusion modeling objective and kernel density estimation. Given a finite dataset $\{W^{(i)}\}_{i=1}^n$, the optimal solution to the diffusion modeling objective should match the empirical density at time t, which is:

$$\hat{p}_t(x) = \frac{1}{n\sigma_t} \sum_{i=1}^{n} \phi\left(\frac{W^{(i)} - x}{\sigma_t}\right), \tag{4.1}$$

where $\phi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$ is the Gaussian kernel. We observe that equation 4.1 is identical to a Gaussian kernel density estimate, given samples $\{W^{(i)}\}_{i=1}^n$[3].

We establish the following result for Gaussian kernel density estimation.

Theorem 4.1 (Gaussian Kernel Density Estimation)**.** *Let $\{W^{(i)}\}_{i=1}^n$ be a set of n independent samples from a λ-Lipschitz density p. Let $\hat{p}$ be the empirical density, $p_\sigma := p \circledast \mathcal{N}(0, \sigma^2)$ and $\hat{p}_\sigma = \hat{p} \circledast \mathcal{N}(0, \sigma^2)$. Then, with probability at least $1 - \delta$ with respect to the sample randomness,*

$$\mathrm{d_{TV}}(p_\sigma, \hat{p}_\sigma) \lesssim \frac{1}{n} + \frac{1}{\sigma^2 n} + \sqrt{\frac{\log n + \log(1 \vee \lambda) + \log 2/\delta}{\sigma^2 n}}. \tag{4.2}$$

The proof of this result is given in the Appendix.

Comparing the performance of Algorithms 1 and 2. Applying Theorem 4.1 directly to the p_0 density, we immediately get that the estimate $\hat{p}_t^{(1)}(x)$ obtained by Algorithm 1 satisfies:

$$\mathrm{d_{TV}}(p_t, \hat{p}_t^{(1)}) \lesssim \frac{1}{n_1} + \frac{1}{\sigma_t^2 n_1} + \sqrt{\frac{\log n_1 + \log(1 \vee \lambda_1) + \log 2/\delta}{\sigma_t^2 n_1}}. \tag{4.3}$$

Let us now see what we get by applying Theorem 4.1 to Algorithm 2, which uses samples from the tilted distribution $\tilde{p}_0$. Since this distribution is $\left(\frac{n_1}{n_1+n_2}\lambda_1 + \frac{n_2}{n_1+n_2}\lambda_2\right)$-Lipschitz, we get that:

$$\mathrm{d_{TV}}(\tilde{p}_t, \hat{p}_t^{(2)}) \lesssim \frac{1}{(n_1+n_2)} + \frac{1}{\sigma_t^2(n_1+n_2)} + \sqrt{\frac{\log(n_1+n_2) + \log(1 \vee \frac{n_1}{n_1+n_2}\lambda_1 + \frac{n_2}{n_1+n_2}\lambda_2) + \log 2/\delta}{\sigma_t^2(n_1+n_2)}},$$

where $\tilde{p}_t := \tilde{p}_0 \circledast \mathcal{N}(0, \sigma_t^2)$.

Further, we have that: $\mathrm{d_{TV}}(p_t, \hat{p}_t^{(2)}) \le \mathrm{d_{TV}}(\tilde{p}_t, p_t) + \mathrm{d_{TV}}(\tilde{p}_t, \hat{p}_t^{(2)})$. We already have a bound for the second term. To bound the first term, we prove the following theorem.

Theorem 4.2 (Distance contraction under noise)**.** *Consider distributions P and Q supported on a subset of $\mathbb{R}^d$ with diameter D. Then*

$$\mathrm{d_{TV}}(P \circledast \mathcal{N}(0, \sigma^2 \mathrm{I}), Q \circledast \mathcal{N}(0, \sigma^2 \mathrm{I})) \le \mathrm{d_{TV}}(P, Q) \cdot \frac{D}{2\sigma}.$$

Applying this theorem we get that: $\mathrm{d_{TV}}(\tilde{p}_t, p_t) \le \frac{1}{2\sigma_t} \mathrm{d_{TV}}(\tilde{p}_0, p_0) \le \frac{1}{2\sigma_t} \cdot \frac{n_2}{n_1+n_2} \mathrm{d_{TV}}(p_0, q_0)$, where for the second inequality we used that $\mathrm{d_{TV}}(p_0, \tilde{p}_0) \le \frac{n_2}{n_1+n_2} \mathrm{d_{TV}}(p_0, q_0)$.

Putting everything together, Algorithm (2) achieves an estimation error:

$$\mathrm{d_{TV}}(p_t, \hat{p}_t^{(2)}) \lesssim$$
$$\frac{1}{(n_1+n_2)} + \frac{1}{\sigma_t^2(n_1+n_2)} + \sqrt{\frac{\log(n_1+n_2) + \log(1 \vee \frac{n_1}{n_1+n_2}\lambda_1 + \frac{n_2}{n_1+n_2}\lambda_2) + \log 2/\delta}{\sigma_t^2(n_1+n_2)}} + \frac{n_2}{\sigma_t(n_1+n_2)} \mathrm{d_{TV}}(p_0, q_0).$$

Comparing this with the bound obtained in Equation 4.3, we see that if n_2 is sufficiently larger than n_1 or if $\lambda_2 \le \lambda_1$, there is a $t_n^{\min}$ such that for any $t \ge t_n^{\min}$, the upper-bound obtained by Algorithm 2 is better than the upper-bound obtained by Algorithm 1. That implies that for high-diffusion times, using biased data might be helpful for learning, as the bias term (final term) decays with the amount of noise. Going back to equation 4, note that the switching point $t \ge t_n^{\min}$ depends on the distance $\mathrm{d_{TV}}(\tilde{p}_t, p_t)$ that decays as shown in Theorem 4.2. Once this distance becomes small enough, our computations above suggest that we benefit from biased data. The classifier of Section 3.1, if optimal, exactly tracks the distance $\mathrm{d_{TV}}(\tilde{p}_t, p_t)$ and, as a result, tracks the switching point.

[3]This connection has been observed in prior works too, e.g., see [23, 6].

5 Experiments

Controlled experiments to show utility from low-quality data. To verify our method, we first do synthetic experiments on artificially corrupted data. We use EDM [24] as our baseline, and we train networks on CIFAR-10 and FFHQ. For the first experiments, we only use the high-noise part of our Ambient-o method (Section 3.1). We underline that for all of our experiments, we only change the way we use the data, and we keep all the optimization and network hyperparameters as is. We compare against using all the data as equal (despite the corruption) and the filtering strategy of only training on the clean samples. For evaluation, we measure FID [20] with respect to the full uncorrupted dataset (which is not available during training).

For the blurring experiments, we use a Gaussian kernel with standard deviation $\sigma_B = 0.4, 0.6, 0.8, 1.0$, and we corrupt 90% of the data. We show some example corrupted images in Appendix Figure 9a. To perform the annotations for our method, we train a blurry image vs clean image classifier under noise, as explained in Section 3.1. For this experiment, each sample is annotated on its own based on the amount of noise that is needed to confuse the classifier (sample dependent annotation). We present our results in Table 2a. As shown, for all corruption strengths, Ambient Omni, significantly outperforms the two baseline methods. In the one to the last column of Table 2a, we further show the average annotation of the classifier. As expected, the average assigned noise level increases as the corruption intensifies.

Ablations. We ablate the choice of using a fixed annotation vs sample-adaptive annotations in Appendix Table 7. We find that using sample-adaptive annotations achieves improved results. Nevertheless, both annotation methods yield improvements over the training on filtered data and the training on everything baselines. To show that our method works for more corruption types, we perform an equivalent experiment with JPEG compressed data at different compression ratios and we achieve similar results, presented in Appendix Table 3. We ablate the impact of the amount of training data and the number of training iterations on the classifier annotations in Appendix Section D. We further show results for motion blur (Figure 10 and Section B.1) and for the FFHQ dataset (see table 4).

Table 2: In a controlled experiment with restricted access only to 10% of the clean dataset, our method of Ambient-o uses corrupted and out-of-distribution data to improve performance.

(a) Gaussian blurred data at different levels.

Method	Parameters Values (σ_B)	$\bar{\sigma}_{t_n}^{\min}$	FID
Only Clean (10%)	-	-	8.79
All data	1.0	0	45.32
	0.8		28.26
	0.6		11.42
	0.4		2.47
Ambient-o	1.0	2.84	**6.16**
	0.8	1.93	**6.00**
	0.6	1.38	**5.34**
	0.4	0.22	**2.44**

(b) Additional out-of-distribution data.

Source Data	Additional Data	Method	$\bar{\sigma}_{t_n}^{\max}$	FID
Dogs (10%)	None	–	–	12.08
	Cats	Fixed σ	0.2	11.14
	Cats	Fixed σ	0.1	9.85
	Cats	Fixed σ	0.05	10.66
	Cats	Fixed σ	0.025	12.07
	Cats	Classifier	0.09	**8.92**
	Procedural	Classifier	0.042	10.98
Cats (10%)	None	–	–	5.20
	Dogs	Classifier	0.13	5.11
	Wildlife	Classifier	0.08	**4.89**

Controlled experiments to show utility from out-of-distribution images. We now want to validate the method developed in Section 3.2 for leveraging crops from out-of-distribution data. To start with, we want to find the mapping between diffusion times and the size of the receptive field required for an optimal denoising prediction. To do so, we take a pre-trained denoising diffusion model and measure the denoising loss at a given location as we increase the size of the context. We provide the corresponding plot in the Supplemental Figures 17, 15. The main finding is that while providing more context always leads to a decrease in the average loss, for sufficiently small noise levels, the loss nearly plateaus before the full image context is provided. That implies that the perfect denoiser for a given noise level only needs to look at a localized part of the image.

Equipped with the mapping between diffusion times and crop sizes, we now proceed to a fun experiment. Specifically, we show that it is possible to use images of cats to improve a generative model for dogs (!) and vice-versa. The cats here represent out-of-distribution data that can be used to improve the performance in the distribution of interest (in our toy example, dogs distribution). To perform this experiment, we train a classifier that discriminates between cats and dog images by looking at crops of various sizes (Section 3.2). Figure 5 shows the predictions of an 8×8 crops-classifier for an image of a cat, illustrating that there are a number of crops that are misclassified as crops from a dog image. We report results for this experiment in Table 2b and we observe improvements in FID arising from using out-of-distribution data. Beyond natural images, we show that it is even possible to use procedurally generated data from Shaders [4] to (slightly) improve the performance. Figure 21 shows an example of such an image and the corresponding predictions of a crops classifier. Table 2b contains more results and ablations between annotating all the out-of-distribution at a single noise level vs. sample-dependent annotations.

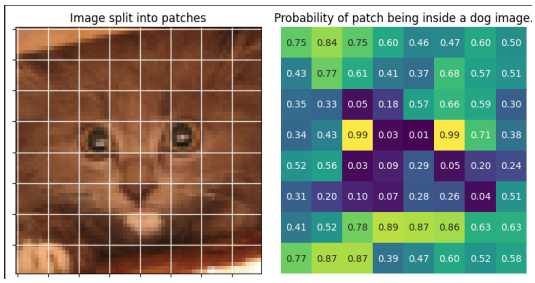


Figure 5: Patch level probabilities for dogness in a cat image.

Takeaway 1: It is possible to use *low-quality in-distribution* images and *high-quality out-of-distribution* images to produce **high-quality in-distribution** images.

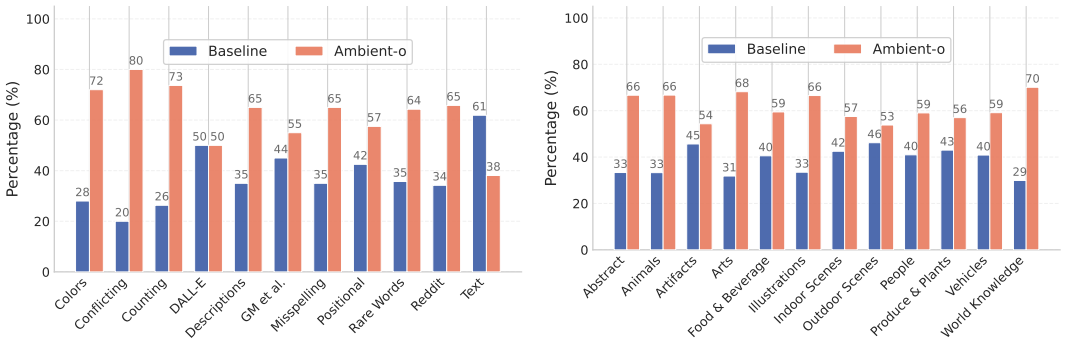
Corruptions of natural datasets – ImageNet results. Up to this point, our corrupted data has been artificially constructed to study our method in a controlled setting. However, it turns out that even in real datasets such as ImageNet, there are images with significant degradations such as heavy blur, low lighting, and low contrast, and also images with fantastic detail, clear lightning, and sharp contrast. Here, the high-quality and the low-quality sets are not given and hence we have to estimate them. We opt to use the CLIP-IQA quality metric [47] to separate ImageNet into high-quality (top 10% CLIP-IQA) and low-quality (bottom 90% CLIP-IQA) sets. Figure 4 shows some of the top and bottom quality images according to our metric. Given the high-quality and low-quality sets, we are now back to the previous setting where we can use the developed Ambient-o methodology.

We use Ambient-o to refer to our method that uses low-quality data at high-diffusion times (Section 5 and Ambient-o+crops to refer to the extended version of our method that uses crops from potentially low-quality images at low-diffusion times. Perhaps surprisingly, there are images in ImageNet that have lower global quality but high-quality crops that we can use for low-noise. We present results in Table 1, where we show the best FID [20] and $\text{FD}_{\text{DINOv2}}$ obtained by different methods. We show the highest and lowest quality crops, alongside their associated full images, of ImageNet according to CLIP in fig. 14.

As shown in the Table, our method leads to state-of-the-art FID scores, improving over the previous state-of-the-art baseline EDM-2 [25] at both the low and high parameter count settings. The benefits are more pronounced when we measure test FID as our method memorizes significantly less due to the addition of noise during the annotation stage of our pipeline (Section 3.1). Beyond FID, we provide qualitative results in Figure 1 (bottom) and Appendix Figures 11, 12. We further show that the quality of the generated images measured by CLIP increased compared to the baseline in Appendix Table 5. The observed improvements are proof that the ability to learn from data with heterogeneous qualities can be truly impactful for realistic settings beyond synthetic corruptions typically studied in prior work.

Takeaway 2: Real datasets contain heterogeneous samples. Ambient-o explicitly accounts for quality variability during training, leading to improved generation quality.

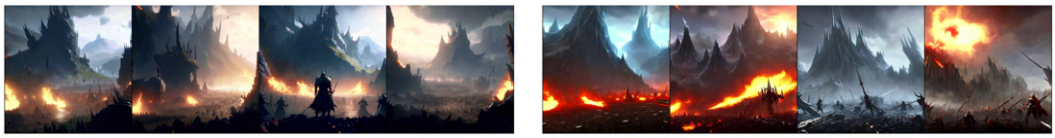
Text-to-image results. For our final set of experiments, we show how Ambient-o can be used to improve the performance of text-to-image diffusion models. We use the code-base of MicroDiffusion [38], as it is open-data and trainable with modest compute ($\approx$ 2 days on 8-H100 GPUs). Sehwag et al. [38] use four main datasets to train their model: Conceptual Captions (12M) [40], Segment Anything (11M) [29], JourneyDB (4.2M) [45], and DiffusionDB (10.7M) [48]. Of these four, DiffusionDB is of significantly lower quality than the others as it contains solely synthetic data from an outdated diffusion model. This presents an opportunity for the use of our method. Can we use this lower-quality data and improve the performance of the trained network?



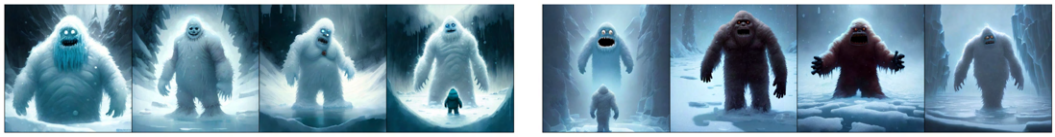
(a) DrawBench (b) PartiPrompts

Figure 6: Assessing image quality with GPT-4o on DrawBench and PartiPrompts.

We set $\sigma_{\min} = 2$ for all samples from DiffusionDB and $\sigma_{\min} = 0$ for all other datasets and we train a diffusion model with Ambient-o. We note that we did not ablate this hyperparameter and it is quite likely that improved results would be obtained by tuning it or by training a high-quality vs low-quality data classifier for the annotation. Despite that, our trained model achieves a remarkable FID of **10.61** in COCO, significantly improving the baseline FID of 12.37 (Table 8). We present qualitative results in Figure 1 and GPT-4o evaluations on DrawBench and PartiPrompt in Figure 6. Ambient-o and baseline generations for different prompts can be found in Figure 1. As an additional ablation, we



(a) "the great battle of middle earth, unreal engine, trending on artstation, masterpiece"



(b) "an abominable snowman trapped in ice by greg rutkowski"

Figure 7: **Examples of mode collapse**. Left: baseline model finetuned on the highest-quality subset. Right: Ambient-o model that uses all the data. As shown, finetuning decreases the diversity of the generations.

compared our method with the recipe of doing a final fine-tuning on the highest-quality subset, as done in the works of [38, 8]. Compared to this baseline, our method obtained slightly worse COCO FID (10.61 vs 10.27) but obtained much greater diversity, as seen visually in fig. 7 and quantitatively through $> 13\%$ increases in DINO Vendi Diversity on prompts from DiffDB (3.22 vs 3.65.). This corroborates our intuition that data filtration leads to decreased diversity. Ambient-o uses all the data but can strike a fine balance between high-quality and diverse generation.

Takeaway 3: Ambient-o achieves superior quality to using all the data in the same way and increased diversity compared to the finetuning on the highest quality subset.

(a) Measuring fidelity and prompt alignment of generated images on COCO dataset.

Method	FID-30K (↓)	Clip-FD-30K (↓)	Clip-score (↑)
Baseline	12.37	10.07	0.345
Ambient-o	**10.61**	**9.40**	**0.348**

(b) Measuring performance on the GenEval benchmark.

Method	Overall	Objects		Counting	Colors	Position	Color attribution
		Single	Two				
Baseline	0.44	0.97	0.33	0.35	0.82	0.06	0.14
Ambient-o	**0.47**	0.97	**0.40**	**0.36**	0.82	**0.11**	0.14

Figure 8: Quantitative benefits of Ambient-o on COCO [32] zero-shot generation and GenEval [17].

6 Limitations and Future Work

Our work opens several avenues for improvement. On the theoretical side, we aim to establish matching lower bounds to demonstrate that learning from the mixture distribution becomes provably optimal beyond a certain noise threshold. Algorithmically, while our method performs well under high-frequency corruptions, it remains an open question whether more effective training strategies could be u.sed for different types of corruptions (e.g., masking). Moreover, real-world datasets often exhibit patch-wise heterogeneity—for example, facial regions are frequently blurred for privacy, leading to uneven corruption across image crops. We plan to investigate patch-level noise annotations to better capture this structure in future work. Finally, we believe the true potential of Ambient-o lies in scientific applications, where data often arises from heterogeneous measurement processes.

7 Conclusion

Is it possible to get good generative models from bad data? Our framework extracts value from low-quality, synthetic and out-of-distribution sources. At a time when the ever-growing data demands of GenAI are at odds with the need for quality control, Ambient-o lights a path for both to be achieved simultaneously.

8 Acknowledgements

This research has been supported by NSF Awards CCF-1901292, DMS-2022448 and DMS-2134108, a Simons Investigator Award, and the Simons Collaboration on the Theory of Algorithmic Fairness. The experiments were run on the Vista GPU Cluster through the Center for Generative AI (CGAI) and the Texas Advanced Computing Center (TACC) at UT Austin.

References

[1] Asad Aali, Marius Arvinte, Sidharth Kumar, and Jonathan I Tamir. "Solving Inverse Problems with Score-Based Generative Priors learned from Noisy Data". In: *arXiv preprint arXiv:2305.01166* (2023) (cit. on pp. 1, 3).

[2] Asad Aali, Giannis Daras, Brett Levac, Sidharth Kumar, Alex Dimakis, and Jon Tamir. "Ambient Diffusion Posterior Sampling: Solving Inverse Problems with Diffusion Models Trained on Corrupted Data". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=qeXcMutEZY (cit. on p. 1).

[3] Weimin Bai, Yifei Wang, Wenzheng Chen, and He Sun. "An Expectation-Maximization Algorithm for Training Clean Diffusion Models from Corrupted Observations". In: *arXiv preprint arXiv:2407.01014* (2024) (cit. on pp. 1, 3).

[4] Manel Baradad, Chun-Fu Chen, Jonas Wulff, Tongzhou Wang, Rogerio Feris, Antonio Torralba, and Phillip Isola. "Procedural Image Programs for Representation Learning". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. 2022. URL: https://openreview.net/forum?id=wJwHTgIoE0P (cit. on p. 10).

[5] Ashish Bora, Eric Price, and Alexandros G Dimakis. "AmbientGAN: Generative models from lossy measurements". In: *International conference on learning representations*. 2018 (cit. on p. 1).

[6] Zdravko I Botev, Joseph F Grotowski, and Dirk P Kroese. "Kernel density estimation via diffusion". In: (2010) (cit. on p. 8).

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. "StarGAN v2: Diverse image synthesis for multiple domains". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8188–8197 (cit. on p. 22).

[8] Xiaoliang Dai et al. *Emu: Enhancing Image Generation Models Using Photogenic Needles in a Haystack*. 2023. arXiv: 2309.15807 [cs.CV] (cit. on pp. 1, 4, 11).

[9] Giannis Daras, Yeshwanth Cherapanamjeri, and Constantinos Costis Daskalakis. "How Much is a Noisy Image Worth? Data Scaling Laws for Ambient Diffusion." In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=qZwtPEw2qN (cit. on p. 1).

[10] Giannis Daras, Yuval Dagan, Alexandros G Dimakis, and Constantinos Daskalakis. "Consistent diffusion models: Mitigating sampling drift by learning to be consistent". In: *arXiv preprint arXiv:2302.09057* (2023) (cit. on pp. 1, 3).

[11] Giannis Daras, Alexandros G Dimakis, and Constantinos Daskalakis. "Consistent Diffusion Meets Tweedie: Training Exact Ambient Diffusion Models with Noisy Data". In: *arXiv preprint arXiv:2404.10177* (2024) (cit. on pp. 1, 3, 4).

[12] Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alex Dimakis, and Adam Klivans. "Ambient Diffusion: Learning Clean Distributions from Corrupted Data". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=wBJBLy9kBY (cit. on pp. 1, 3).

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on p. 22).

[14] Prafulla Dhariwal and Alexander Nichol. "Diffusion models beat gans on image synthesis". In: *Advances in neural information processing systems* 34 (2021), pp. 8780–8794 (cit. on p. 4).

[15] Sander Dieleman. *Diffusion is spectral autoregression*. 2024. URL: https://sander.ai/2024/09/02/spectral-autoregression.html (cit. on p. 2).

[16] 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: *arXiv preprint arXiv:2304.14108* (2023) (cit. on p. 1).

[17] Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. *GenEval: An Object-Focused Framework for Evaluating Text-to-Image Alignment*. 2023. arXiv: 2310.11513 [cs.CV]. URL: https://arxiv.org/abs/2310.11513 (cit. on p. 12).

[18] 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*. 2024, pp. 22702–22711 (cit. on p. 1).

[19] Dan Hendrycks and Thomas Dietterich. "Benchmarking neural network robustness to common corruptions and perturbations". In: *arXiv preprint arXiv:1903.12261* (2019) (cit. on p. 2).

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 9, 10).

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851 (cit. on p. 2).

[22] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. "Adaptive Mixtures of Local Experts". In: *Neural Computation* 3.1 (Mar. 1991). _eprint: https://direct.mit.edu/neco/article-pdf/3/1/79/812104/neco.1991.3.1.79.pdf, pp. 79–87. ISSN: 0899-7667. DOI: 10.1162/neco.1991.3.1.79. URL: https://doi.org/10.1162/neco.1991.3.1.79 (cit. on p. 23).

[23] Mason Kamb and Surya Ganguli. "An analytic theory of creativity in convolutional diffusion models". In: *arXiv preprint arXiv:2412.20292* (2024) (cit. on p. 8).

[24] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. "Elucidating the design space of diffusion-based generative models". In: *arXiv preprint arXiv:2206.00364* (2022) (cit. on pp. 9, 22).

[25] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. "Analyzing and Improving the Training Dynamics of Diffusion Models". In: *Proc. CVPR*. 2024 (cit. on pp. 2, 10, 20, 21, 23).

[26] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410 (cit. on p. 22).

[27] Varun A Kelkar, Rucha Deshpande, Arindam Banerjee, and Mark A Anastasio. "Ambient-Flow: Invertible generative models from incomplete, noisy measurements". In: *arXiv preprint arXiv:2309.04856* (2023) (cit. on p. 1).

[28] Diederik P Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 23).

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. *Segment Anything*. 2023. arXiv: 2304.02643 [cs.CV]. URL: https://arxiv.org/abs/2304.02643 (cit. on pp. 11, 22).

[30] Alex Krizhevsky and Geoffrey Hinton. "Learning multiple layers of features from tiny images". In: (2009) (cit. on p. 22).

[31] Jeffrey Li et al. *DataComp-LM: In search of the next generation of training sets for language models*. 2024. arXiv: 2406.11794 [cs.LG] (cit. on pp. 1, 4).

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV]. URL: https://arxiv.org/abs/1405.0312 (cit. on p. 12).

[33] Haoye Lu, Qifan Wu, and Yaoliang Yu. "SFBD: A Method for Training Diffusion Models with Noisy Data". In: *Frontiers in Probabilistic Inference: Learning meets Sampling*. 2025. URL: https://openreview.net/forum?id=6HN14zuHRb (cit. on p. 1).

[34] William Peebles and Saining Xie. *Scalable Diffusion Models with Transformers*. 2023. arXiv: 2212.09748 [cs.CV]. URL: https://arxiv.org/abs/2212.09748 (cit. on p. 23).

[35] Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. "The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale". In: *arXiv preprint arXiv:2406.17557* (2024) (cit. on p. 1).

[36] François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. "Learning Diffusion Priors from Observations by Expectation Maximization". In: *arXiv preprint arXiv:2405.13712* (2024) (cit. on pp. 1, 3).

[37] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. "Laion-5b: An open large-scale dataset for training next generation image-text models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 25278–25294 (cit. on p. 1).

[38] Vikash Sehwag, Xianghao Kong, Jingtao Li, Michael Spranger, and Lingjuan Lyu. "Stretching Each Dollar: Diffusion Training from Scratch on a Micro-Budget". In: *arXiv preprint arXiv:2407.15811* (2024) (cit. on pp. 2, 4, 11, 22, 23).

[39] Kulin Shah, Alkis Kalavasis, Adam R. Klivans, and Giannis Daras. *Does Generation Require Memorization? Creative Diffusion Models using Ambient Diffusion*. 2025. arXiv: 2502.21278 [cs.LG] (cit. on p. 1).

[40] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 2556–2565. DOI: 10.18653/v1/P18-1238. URL: https://aclanthology.org/P18-1238/ (cit. on pp. 11, 22).

[41] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. *Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer*. 2017. arXiv: 1701.06538 [cs.LG]. URL: https://arxiv.org/abs/1701.06538 (cit. on p. 23).

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising diffusion implicit models”. In: *arXiv preprint arXiv:2010.02502* (2020) (cit. on pp. 22, 23).

[43] Yang Song and Stefano Ermon. “Generative modeling by estimating gradients of the data distribution”. In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 2).

[44] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. “Score-based generative modeling through stochastic differential equations”. In: *arXiv preprint arXiv:2011.13456* (2020) (cit. on p. 2).

[45] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, Jifeng Dai, Yu Qiao, Limin Wang, and Hongsheng Li. *JourneyDB: A Benchmark for Generative Image Understanding*. 2023. arXiv: 2307.00716 [cs.CV]. URL: https://arxiv.org/abs/2307.00716 (cit. on pp. 11, 22).

[46] Antonio Torralba, Phillip Isola, and William T Freeman. *Foundations of computer vision*. MIT Press, 2024 (cit. on p. 2).

[47] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. “Exploring CLIP for Assessing the Look and Feel of Images”. In: *AAAI*. 2023 (cit. on pp. 10, 22).

[48] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. “DiffusionDB: A Large-scale Prompt Gallery Dataset for Text-to-Image Generative Models”. In: *arXiv preprint arXiv:2210.14896* (2022) (cit. on pp. 11, 22).

[49] Yasi Zhang, Tianyu Chen, Zhendong Wang, Ying Nian Wu, Mingyuan Zhou, and Oscar Leong. “Restoration Score Distillation: From Corrupted Diffusion Pretraining to One-Step High-Quality Generation”. In: *arXiv preprint arXiv:2505.13377* (2025) (cit. on p. 1).

A Theoretical Results

A.1 Kernel Estimation

Assumption A.1. The density p is λ lipschitz.

Let $\{X^{(i)}\}_{i=1}^n$ a set of n independent samples from a density p that satisfies Assumption A.1. Let $\hat{p}$ be the empirical density on those samples.

We are interested in bounding the total variation distance between $p_\sigma := p \circledast \mathcal{N}(0, \sigma^2)$ and $\hat{p}_\sigma = \hat{p} \circledast \mathcal{N}(0, \sigma^2)$. In particular,

$$\hat{p}_\sigma(x) = \frac{1}{n\sigma} \sum_{i=1}^{n} \phi\left(\frac{X^{(i)} - x}{\sigma}\right), \tag{A.1}$$

where $\phi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$ is the Gaussian kernel. We want to argue that the TV distance between p_σ and $\hat{p}_\sigma$ is small given sufficiently many samples n. For simplicity, let's fix the support of p to be $[0, 1]$. We have:

$$\mathrm{d_{TV}}(p_\sigma, \hat{p}_\sigma) = \frac{1}{2} \int_0^1 |p_\sigma(x) - \hat{p}_\sigma(x)| \mathrm{d}x = \sum_{l=0}^{L-1} \int_{l/L}^{(l+1)/L} |p_\sigma(x) - \hat{p}_\sigma(x)| \mathrm{d}x \tag{A.2}$$

Now let us look at one of the terms of the summation.

$$\int_{l/L}^{(l+1)/L} |p_\sigma(x) - \hat{p}_\sigma(x)| \mathrm{d}x = \int_{l/L}^{(l+1)/L} |p_\sigma(x) - p_\sigma(l/L) + p_\sigma(l/L) - \hat{p}_\sigma(x)| \mathrm{d}x \tag{A.3}$$

$$\leq \int_{l/L}^{(l+1)/L} |p_\sigma(x) - p_\sigma(l/L)| \mathrm{d}x + \int_{l/L}^{(l+1)/L} |p_\sigma(l/L) - \hat{p}_\sigma(x)| \mathrm{d}x. \tag{A.4}$$

We first work on the first term. Using Lemma A.6:

$$\int_{l/L}^{(l+1)/L} |p_\sigma(x) - p_\sigma(l/L)| \mathrm{d}x \leq \lambda \int_{l/L}^{(l+1)/L} |x - l/L| \mathrm{d}x \tag{A.5}$$

$$= \frac{\lambda}{2L^2}. \tag{A.6}$$

Next, we work on the second term.

$$\int_{l/L}^{(l+1)/L} |p_\sigma(l/L) - \hat{p}_\sigma(x)| \mathrm{d}x = \int_{l/L}^{(l+1)/L} |p_\sigma(l/L) - \hat{p}_\sigma(l/L) + \hat{p}_\sigma(l/L) - \hat{p}_\sigma(x)| \mathrm{d}x \tag{A.7}$$

$$\leq \int_{l/L}^{(l+1)/L} |p_\sigma(l/L) - \hat{p}_\sigma(l/L)| \mathrm{d}x + \int_{l/L}^{(l+1)/L} |\hat{p}_\sigma(l/L) - \hat{p}_\sigma(x)| \mathrm{d}x. \tag{A.8}$$

According to Lemma A.5, we have that $\hat{p}_\sigma$ is $\hat{\lambda} = \frac{1}{\sigma^2\sqrt{2\pi e}}$ Lipschitz. Then, the second term becomes:

$$\int_{l/L}^{(l+1)/L} |\hat{p}_\sigma(l/L) - \hat{p}_\sigma(x)| \mathrm{d}x \leq \hat{\lambda} \int_{l/L}^{(l+1)/L} |l/L - x| \mathrm{d}x = \frac{\hat{\lambda}}{2L^2}. \tag{A.9}$$

It remains to bound the following term

$$\int_{l/L}^{(l+1)/L} |p_\sigma(l/L) - \hat{p}_\sigma(l/L)| \mathrm{d}x = \frac{|p_\sigma(l/L) - \hat{p}_\sigma(l/L)|}{L} \tag{A.10}$$

We will be applying Hoeffding's Inequality, stated below:

Theorem A.2 (Hoeffding's Inequality). *Let $Y_1, ..., Y_n$ be independent random variables in $[a, b]$ with mean μ. Then,*

$$\Pr\left(\left|\frac{1}{n}\sum_{i=1}^{n} Y_i - \mu\right| \geq t\right) \leq 2\exp\left(-2nt^2/(b-a)^2\right). \tag{A.11}$$

Recall that $\hat{p}_\sigma$ can be written as

$$\hat{p}_\sigma(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{\phi((X^{(i)} - x)/\sigma)}{\sigma} = \frac{1}{n}\sum_{i=1}^{n} Y_i, \tag{A.12}$$

in terms of the random variables $Y_i := \frac{\phi((X^{(i)}-x)/\sigma)}{\sigma}$. These random variables are supported in $\left[0, \frac{1}{\sqrt{2\pi\sigma^2}}\right]$. So, for any x, we have that:

$$\Pr\left(|\hat{p}_\sigma(x) - \mathbb{E}[\hat{p}_\sigma(x)]| \geq t\right) \leq 2\exp\left(-4\pi\sigma^2 nt^2\right). \tag{A.13}$$

Taking $t = \sqrt{\frac{\log(2L/\delta)}{4\pi\sigma^2 n}}$ and using the above inequality and the union bound, we have that, with probability at least $1-\delta$, for all $l \in \{0, 1, \ldots, L-1\}$:

$$|\hat{p}_\sigma(l/L) - \mathbb{E}[\hat{p}_\sigma(l/L)]| \leq \sqrt{\frac{\log(2L/\delta)}{4\pi\sigma^2 n}}. \tag{A.14}$$

Let us now compute the expected value of $\hat{p}_\sigma(x)$.

$$\mathbb{E}[\hat{p}_\sigma(x)] = \mathbb{E}\left[\frac{1}{n\sigma}\sum_{i=1}^{n} \phi\left(\frac{X^{(i)} - x}{\sigma}\right)\right] \tag{A.15}$$

$$= \frac{1}{n\sigma}\sum_{i=1}^{n} \mathbb{E}\left[\phi\left(\frac{X^{(i)} - x}{\sigma}\right)\right] \tag{A.16}$$

$$= \frac{1}{\sigma}\int p(u)\phi\left(\frac{x-u}{\sigma}\right)\mathrm{d}u \equiv (p \circledast \mathcal{N}(0, \sigma^2))(x) = p_\sigma(x). \tag{A.17}$$

Combining equation A.14 and equation A.17, we get:

$$|\hat{p}_\sigma(l/L) - p_\sigma(x)| \leq \sqrt{\frac{\log(2L/\delta)}{4\pi\sigma^2 n}}. \tag{A.18}$$

Putting everything together we have:

$$\mathrm{d_{TV}}(p_\sigma, \hat{p}_\sigma) \leq \frac{\lambda}{2L} + \frac{1}{2L\sigma^2\sqrt{2\pi e}} + \sqrt{\frac{\log(2L/\delta)}{4\pi\sigma^2 n}}.$$

Choosing $L = n \cdot \max\{\lambda, 1\}$ we get that:

$$\mathrm{d_{TV}}(p_\sigma, \hat{p}_\sigma) \lesssim \frac{1}{n} + \frac{1}{\sigma^2 n} + \sqrt{\frac{\log n + \log(1 \vee \lambda) + \log 2/\delta}{\sigma^2 n}}.$$

A.2 Evolution of parameters under noise

Proof of theorem 4.2: We will use the following facts:
Fact 1 (Direct corollary of the optimal coupling theorem). There exists a coupling γ of P and Q, which samples a pair of random variables $(X, Y) \sim \gamma$ such that $\Pr_\gamma[X \neq Y] = \mathrm{d_{TV}}(P, Q)$.
Fact 2. For any $x, y \in \mathbb{R}^d$: $\mathrm{d_{TV}}(\mathcal{N}(x, \sigma^2\mathrm{I}), \mathcal{N}(y, \sigma^2\mathrm{I})) \leq \|x - y\|/2\sigma$

Proof. The KL divergence between $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$ is

$$\mathrm{KL}(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) = \frac{1}{2}\left(\mathrm{tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)\Sigma_2^{-1}(\mu_2 - \mu_1) - d + \log\frac{|\Sigma_2|}{|\Sigma_1|}\right).$$

Applying this general result to our case:

$$\mathrm{KL}(\mathcal{N}(x, \sigma^2\mathrm{I}), \mathcal{N}(y, \sigma^2\mathrm{I})) = \frac{1}{2}\left(\frac{\|x-y\|^2}{\sigma^2}\right).$$

We conclude by applying Pinsker's inequality. □

A corollary of Fact 2 and the optimal coupling theorem is the following:
Fact 3. Fix arbitrary $x, y \in \mathbb{R}^d$. There exists a coupling $\gamma_{x,y}$ of $\mathcal{N}(0, \sigma^2\mathrm{I})$ and $\mathcal{N}(0, \sigma^2\mathrm{I})$, which samples a pair of random variables $(Z, Z') \sim \gamma_{x,y}$ such that $\Pr_{\gamma_{x,y}}[x + Z \neq y + Z'] = \|x - y\|/2\sigma$.

Now let us denote by $\tilde{P} = P \circledast \mathcal{N}(0, \sigma^2\mathrm{I})$ and $\tilde{Q} = Q \circledast \mathcal{N}(0, \sigma^2\mathrm{I})$. To establish our claim in the theorem statement, it suffices to exhibit a coupling $\tilde{\gamma}$ of $\tilde{P}$ and $\tilde{Q}$ which samples a pair of random variables $(\tilde{X}, \tilde{Y}) \sim \tilde{\gamma}$ such that: $\Pr_{\tilde{\gamma}}[\tilde{X} \neq \tilde{Y}] \leq \mathrm{d_{TV}}(P, Q) \cdot \frac{D}{2\sigma}$. We define coupling $\tilde{\gamma}$ as follows:

Let us argue the following:

Lemma A.3. *The afore-described sampling procedure $\tilde{\gamma}$ is a valid coupling of $\tilde{P}$ and $\tilde{Q}$.*

Proof. We need to establish that the marginals of $\tilde{\gamma}$ are $\tilde{P}$ and $\tilde{Q}$. □

Lemma A.4. *Under the afore-described coupling $\tilde{\gamma}$: $\Pr_{\tilde{\gamma}}[\tilde{X} \neq \tilde{Y}] \leq \mathrm{d_{TV}}(P, Q) \cdot \frac{D}{2\sigma}$.*

Proof. Event $\tilde{X} \neq \tilde{Y}$ happens, when $X \neq Y$ and, conditioning on this event, when $X + Z \neq Y + Z'$ happens. By Fact 1, $\Pr_{\gamma}[X \neq Y] = \mathrm{d_{TV}}(P, Q)$. By Fact 3, for any realization of (X, Y), $\Pr_{\gamma_{X,Y}}[X + Z \neq Y + Z'] = \frac{\|X-Y\|}{2\sigma} \leq \frac{D}{2\sigma}$, where we used that P and Q are supported on a set with diameter D. Putting the above together, the claim follows. □

□

A.3 Auxiliary Lemmas

Lemma A.5 (Lipschitzness of the empirical density)**.** *For a collection of points $X^{(1)}, \ldots, X^{(n)}$ consider the function $\hat{p}_\sigma(x) = \frac{1}{n\sigma}\sum_{i=1}^n \phi\left(\frac{X^{(i)}-x}{\sigma}\right)$, where $\phi(u) = \frac{1}{\sqrt{2\pi}}e^{-u^2/2}$ is the Gaussian kernel. Then p_σ is $\left(\frac{1}{\sigma^2\sqrt{2\pi e}}\right)$-Lipschitz.*

Proof. Let us compute the derivative of $\hat{p}_\sigma$:

$$\hat{p}'_\sigma(x) = \frac{1}{n\sigma}\sum_{i=1}^n \frac{d}{dx}\phi\left(\frac{x - X^{(i)}}{\sigma}\right) \tag{A.19}$$

$$= \frac{1}{\sqrt{2\pi}n\sigma}\sum_{i=1}^n \exp\left(-(X^{(i)} - x)^2/(2\sigma^2)\right)\frac{X^{(i)} - x}{\sigma^2} \tag{A.20}$$

$$\leq \frac{1}{\sqrt{2\pi}\sigma^2}\max_u \exp(-u^2/2)u \tag{A.21}$$

$$\leq \frac{1}{\sigma^2\sqrt{2\pi e}}. \tag{A.22}$$

□

Lemma A.6 (Lipschitzness of a density convolved with a Gaussian)**.** *Let p be a density that is λ-Lipschitz. Let $p_\sigma = p \circledast \mathcal{N}(0, \sigma^2 I)$. Then, p_σ is also λ-Lipschitz.*

Proof. Let us denote with $\phi_\sigma(\cdot)$ the Gaussian density with variance σ^2. We have that:

$$p_\sigma(x) - p_\sigma(y) = \int (p(x-\tau) - p(y-\tau))\phi_\sigma(\tau)d\tau \Rightarrow \tag{A.23}$$

$$|p_\sigma(x) - p_\sigma(y)| \leq \int |p(x-\tau) - p(y-\tau)|\phi_\sigma(\tau)d\tau \tag{A.24}$$

$$\leq \lambda|x-y| \cdot \int \phi_\sigma(\tau)d\tau \tag{A.25}$$

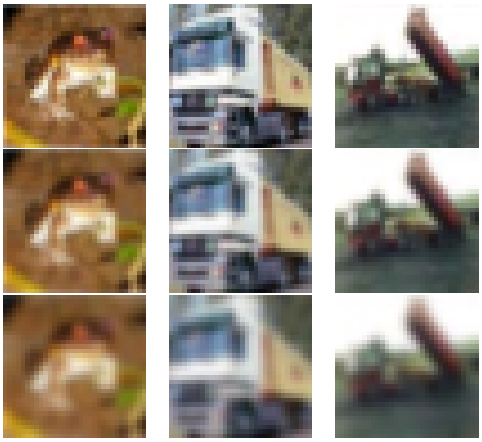
$$= \lambda|x-y|. \tag{A.26}$$

□

B Additional Results

B.1 CIFAR-10 controlled corruptions

Figures 9a, 9b and 10 show gaussian blur, motion blur, and JPEG corrupted CIFAR-10 images respectively at different levels of severity. Table 3 shows results for JPEG compressed data at different levels of compression. We also tested our method for motion blurred data with high severity, visualized in the last row of fig. 10), obtaining a best FID of 5.85 (compared to 8.79 of training on only the clean data).



(a) CIFAR-10 images corrupted with blur at increasing levels ($\sigma_B = 0.4, 0.6, 1.0$).

(b) CIFAR-10 images corrupted with JPEG at compression rates: $25\%, 18\%, 15\%$ respectively.

Table 3: Results for learning from JPEG compressed data on CIFAR-10.

Method	**Dataset**	**Clean (%)**	**Corrupted (%)**	**JPEG Compression (Q)**	$\bar{\sigma}_{t_n}^{\min}$	**FID**
Only Clean	Cifar-10	10	0	–	–	8.79
Ambient Omni	Cifar-10	10	90	15%	1.60	6.67
				18%	1.40	6.43
				25%	1.27	6.34
				50%	1.03	5.94
				75%	0.81	5.57
				90%	0.63	4.72

B.2 FFHQ-64x64 controlled corruptions

In Table 4 we show additional results for learning from blurred data on the FFHQ dataset. Similarly to the main paper, we observe that our Ambient-o algorithm leads to improvements over just using the high-quality data that are inversely proportional to the corruption level.

Figure 10: CIFAR-10 images corrupted with motion blur at increasing levels of corruption.

Table 4: Results for learning from blurred data, FFHQ.

Method	**Dataset**	**Clean (%)**	**Corrupted (%)**	**Parameters Values** (σ_B)	$\bar{\sigma}_{t_n}^{\min}$	**FID**
Only Clean	FFHQ	10	0	-	-	5.12
Ambient Omni	FFHQ	10	90	0.8	2.89	4.95
		10	90	0.6	2.12	4.65
		10	90	0.4	0.63	3.32

B.3 ImageNet results

In the main paper, we used FID as a way to measure the quality of generated images. However, FID is computed with respect to the test dataset that might also have samples of poor quality. Further, during FID computation, quality and diversity are entangled. To disentangle the two, we generate images using the EDM-2 baseline and our Ambient-o model and we use CLIP to evaluate the quality of the generated image (through the CLIP-IQA package). We present results and win-rates in Table 5. As shown, Ambient-o achieves a better per-image quality compared to the baseline despite using exactly the same model, hyperparameters, and optimization algorithm. The difference comes solely from better use of the available data.

Table 5: Additional comparison between EDM-2 XXL and our Ambient-o model using the CLIP IQA metric for image quality assesment. Ambient-o leads to improved scores despite using the exact same architecture, data and hyperparameters. For this experiment, we use the models with guidance optimized for DINO FD since they are the ones producing the higher quality images.

Metric	**EDM-2 [25] XXL**	**Ambient-o XXL crops**
Average CLIP IQA score	0.69	**0.71**
Median CLIP IQA score	0.79	**0.80**
Win-rate	47.98%	**52.02%**

C Ambient diffusion implementation details and loss ablations

Similar to the EDM-2 [25] paper, we use a pre-condition weight to balance the importance of different diffusion times. This weight is set to:

$$\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}}) = \sigma^4/(\sigma^2 - \sigma^2_{\text{min}})^2 \tag{C.1}$$

for our ambient loss based on a similar analysis to [25]. We further use a buffer zone around the annotation time of each sample to ensure that the loss doesn't have singularities due to divisions by 0. We ablate the precondition term and the buffer size in table 6.

Table 6: Ablation study of ambient weight and stability buffer on Cifar-10 with 10% clean data and 90% corrupted data with blur of 0.6.

Method	**FID** $\downarrow$
No ambient preconditioning weight and no buffer:	
$\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}}) = 1$ & $\sigma > \sigma_{\text{min}}$	5.49
Adding ambient preconditioning weight:	
+ Weight $\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}}) = \sigma^4/(\sigma^2 - \sigma^2_{\text{min}})^2$	5.36
Adding stability buffer/clipping:	
+ Clip $\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}})$ at 2.0	5.35
+ Clip $\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}})$ at 4.0	5.69
+ Buffer $\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}})$ at 2.0 i.e. $\sigma > \sqrt{2}\sigma_{\text{min}}$	5.40
+ Buffer $\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}})$ at 4.0 i.e. $\sigma > (2/\sqrt{3})\sigma_{\text{min}}$	**5.34**

For our ablations, we focus on the setting of training with 10% clean data and 90% corrupted data with Gaussian blur of $\sigma_B = 0.6$. Using no ambient pre-conditioning and no buffer, we obtain an FID of 5.56. In the same setting, adding the ambient pre-conditioning weight $\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}})$ improves FID by 0.13 points. Next, we ablate two strategies to mitigate the impact of the singularity of $\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}})$ at $\sigma = \sigma_{\text{min}}$. The first strategy clips the ambient pre-conditioning weight at a specified maximum value $\lambda^{\text{MAX}}_{\text{amb}}$, but still trains for σ arbitrarily close to σ_{min}. The second strategy also specifies a maximum value, but imposes a buffer

$$\sigma > \sqrt{1 + \frac{1}{\lambda^{\text{MAX}}_{\text{amb}} - 1}}\sigma_{\text{min}} \tag{C.2}$$

that restricts training to noise levels σ such that $\lambda_{\text{amb}}(\sigma, \sigma_{\text{min}}) \leq \lambda^{\text{MAX}}_{\text{amb}}$. Clipping the ambient weight to $\lambda^{\text{MAX}}_{\text{amb}} = 2.0$ minimally improves FID to 5.35, but clipping to 4.0 significantly worsens it to 5.69. Adding a buffer at $\lambda^{\text{MAX}}_{\text{amb}} = 2.0$ slightly worsens FID to 5.40, but slackening the buffer to 4.0 minimally improves FID to 5.34. We opt for the buffering strategy in favor of the clipping strategy since performance appears convex in the buffer parameter, and because it obtains the best FID.

D Classifier annotation ablations

Balanced vs unbalanced data: We ablate the impact of classifier training data on the setting of CIFAR-10 with 10% clean data and 90% corrupted data with gaussian blur with $\sigma_B = 0.6$. When annotating with a classifier trained on the same unbalanced dataset we train the diffusion model on we obtained a best FID of 6.04, compared to the 5.34 obtained if we train on a balanced dataset.

Training iterations: We ablate the impact of classifier training iterations on the setting of CIFAR-10 with 10% clean data and 90% corrupted data with JPEG compression at compression rate of 18%, training the classifier with a balanced dataset. We report minute variations in the best FID, obtaining 6.50, 6.58, and 6.49 when training the classifier for 5e6, 10e6, and 15e6 images worth of training respectively.

Table 7: Comparison with baselines for training with data corrupted by Gaussian Blur at different levels. The dataset used in this experiment is CIFAR-10.

Method	Clean (%)	Corrupted (%)	Parameters Values (σ_B)	$\bar{\sigma}_{t_n}^{\min}$	FID
Only Clean	10	0	-	-	8.79
No annotations	10	90	1.0	0	45.32
			0.8		28.26
			0.4		2.47
Single annotation	10	90	1.0	2.32	6.95
			0.8	1.89	6.66
			0.4	0.00	2.47
Classifier annotations	10	90	1.0	2.84	6.16
	10	90	0.8	1.93	6.00
	10	90	0.4	0.22	2.44

E Training Details

E.1 Formation of the high-quality and low-quality sets.

In the theoretical problem setting we assumed the existence of a good set S_G from the clean distribution and a bad set S_B from the corrupted distribution. In practice, we do not actually possess these sets initially, but we can construct them so long as we have access to a measure of "quality". Given a function on images which tells us wether its good enough to generate or not e.g. CLIP-IQA quality [47] greater than some threshold, we can define our good set S_G as the good enough images and S_B as the complement. From this point on we can apply the methodology of ambient-o as developed, either employing classifier annotations as in our pixel diffusion experiments, or fixed annotations as in our large scale ImageNet and text-to-image experiments.

E.2 Datasets

CIFAR-10. CIFAR-10 [30] consists of 60,000 32x32 images of ten classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck).

FFHQ. FFHQ [26] consists of 70,000 512x512 images of faces from Flickr. We used the dataset at 64x64 resolution for our experiments.

AFHQ. AFHQ [7] consists of 5,653 images of cats, 5,239 images of dogs and 5,000 images of wildlife, for a total of 15,892 images.

ImageNet. ImageNet [13] consists of 1,281,167 images of variable resolution from 1000 classes.

Conceptual Captions. Conceptual Captions [40] consists of 12M (image url, caption) pairs.

Segment Anything. Segment Anything [29] consists of 11.1M high-resolution images annotated with segmentation masks. Since the original dataset did not have real captions, we use the same LLaVA generated captions created by the MicroDiffusion [38] paper.

JourneyDB. JourneyDB consists of 4.4M synthetic image-caption pairs from Midjourney [45].

DiffusionDB. DiffusionDB consists of 14M synthetic image-caption pairs, mostly generated from Stable Diffusion models [48]. We use the same 10.7M quality-filtered subset created by the MicroDiffusion paper [38].

E.3 Diffusion model training

CIFAR-10. We use the EDM [24] codebase as a reference to train class-conditional diffusion models on CIFAR-10. The architecture is a Diffusion U-Net [42] with ~55M paramemeters. We use

the Adam optimizer [28] with learning rate 0.001, batch size 512, and no weight decay. While the original EDM paper trained for 200×10^6 images worth of training, when training with corrupted data we saw best results around 20×10^6 images. On a single 8xV100 node we achieved a throughput of 0.8s per 1k images, for an average of 4.4h per training run.

FFHQ. Same as for CIFAR-10, except learning was set to $2e-4$, we trained for a maximum of 100×10^6 images worth of training, and saw best results around 30×10^6 images worth.

AFHQ. Same as FFHQ.

ImageNet. We use the EDM2 [25] codebase as a reference to train class-conditional diffusion models on ImageNet. The architecture is a Diffusion U-Net [42] with ~125M paramemeters. We use the Adam optimizer [28] with reference learning rate 0.012, batch size 2048, and no weight decay. Same as the original codebase, we trained for ~2B worth of images. On 32 H200 GPUs, XS models took ~3 days to train, while XXL models took ~7 days.

MicroDiffusion. We use the MicroDiffusion codebase [38] as a reference to train text-to-image models on an academic budget. We follow their recipe exactly, changing only the standard denoising diffusion loss to the ambient diffusion loss. The architecture is a Diffusion Transformer [34] utilizing Mixture-of-Experiments (MoE) feedforward layers [41, 22], with ~1.1B paramemeters. We use the AdamW optimizer [28] with reference learning rates $2.4e-4/8e-5/8e-5/8e-5$ for each of the four phases and batch size 2048 for all phases. On 8 H200 GPUs, training takes ~2 days to train.

E.4 Classifier training

Classifier training is done using the same optimization recipe (optimizer, learning rate, batch size, etc.) as diffusion model training, except we change the architecture to an encoder-only "Half-Unet", simply by removing the decoder half of the original UNet architecture.

F Additional Figures

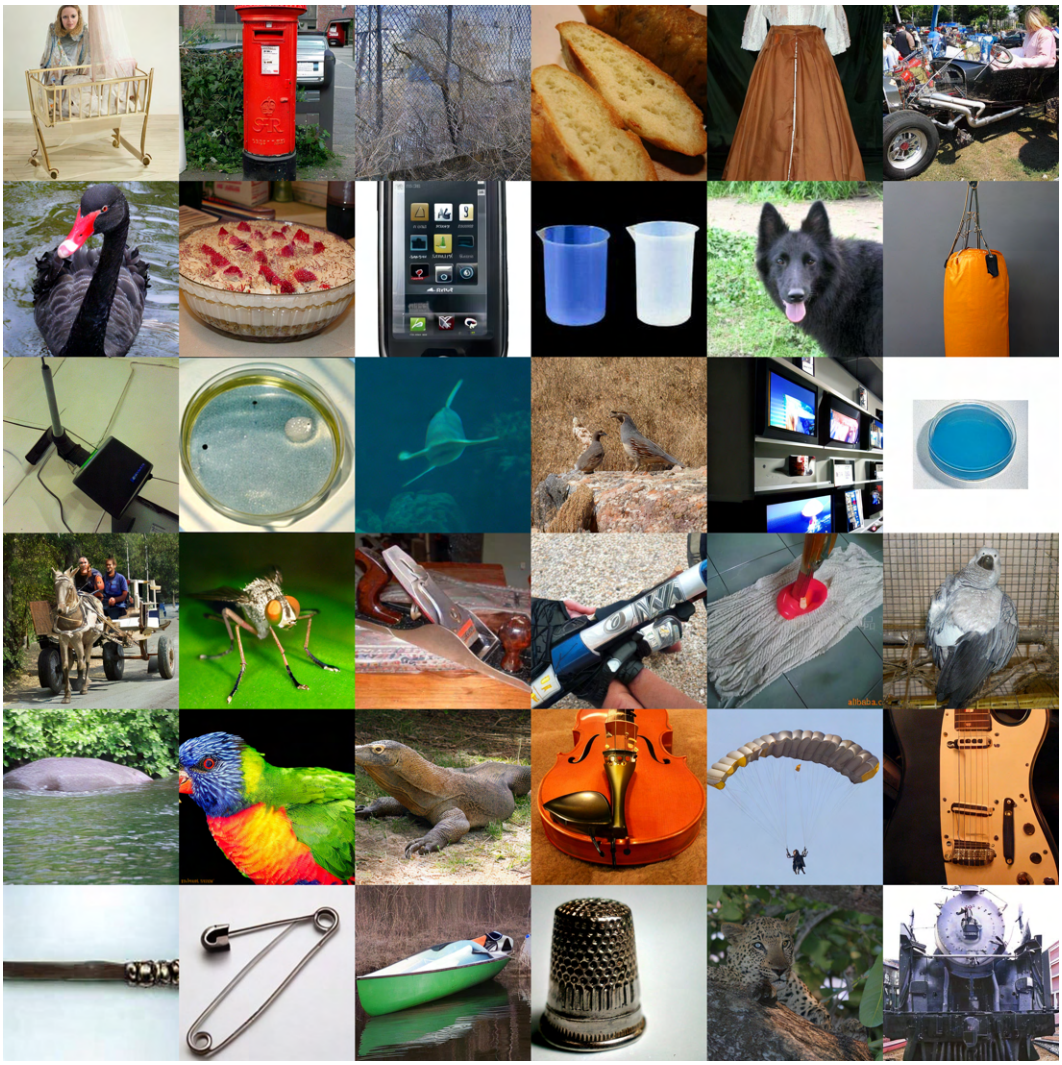


Figure 11: Uncurated generations from our Ambient-o XXL model trained on ImageNet.

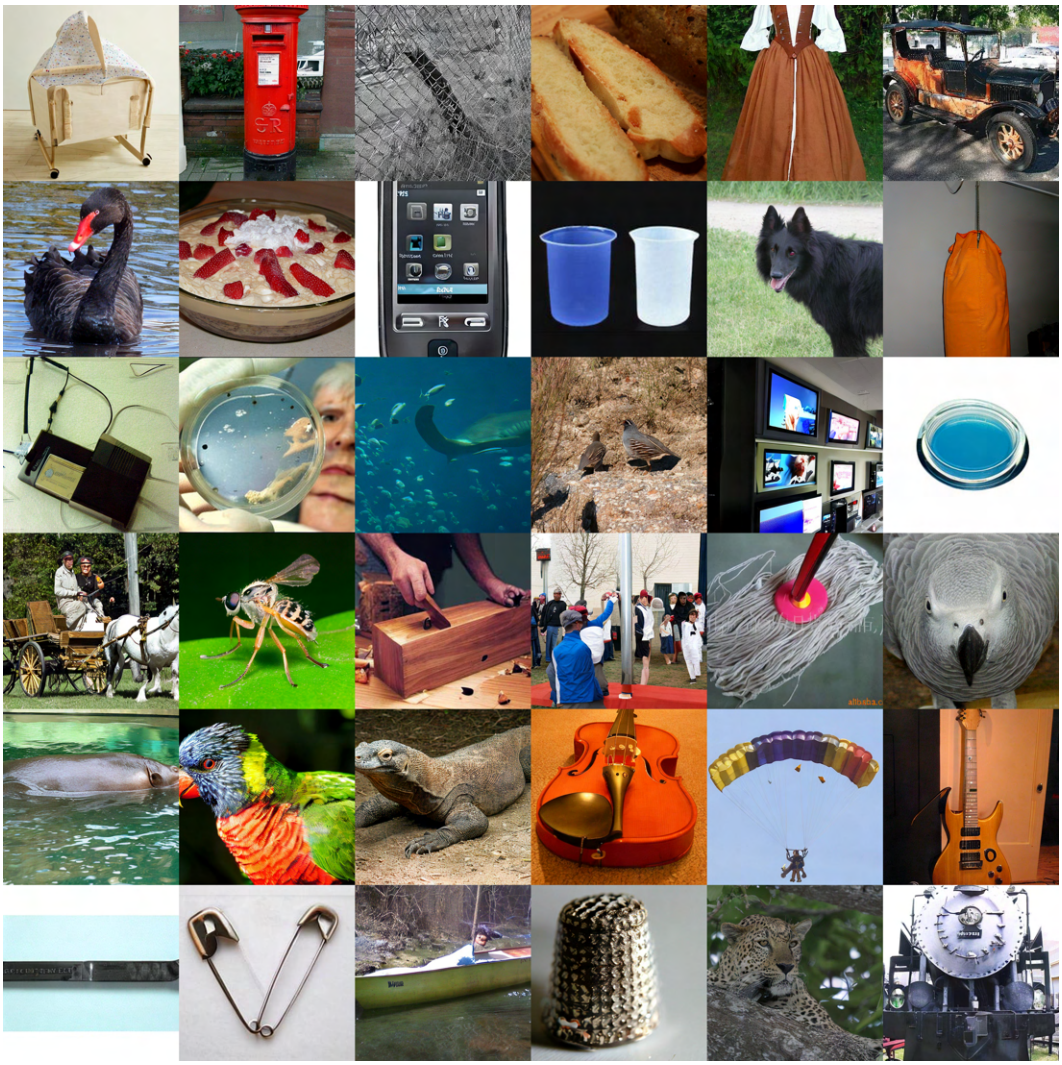


Figure 12: Uncurated generations from our Ambient-o+crops XXL model trained on ImageNet.

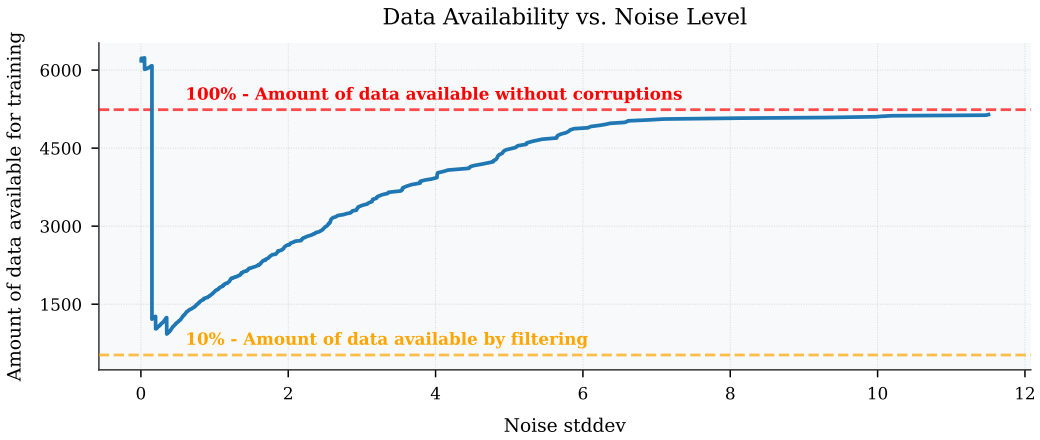


Figure 13: Amount of samples available at each noise level when training a generative model for dogs in the following setting: (1) we have 10% of the dogs dataset uncorrupted, (2) we have the other 90% of the dogs dataset corrupted with gaussian blur with $\sigma_B = 0.6$, and (3) we have 100% of the clean dataset of cats. At low noise levels, we can train on both the high quality dogs and a lot of the cats, resulting in $> 100\%$ of samples available relative to the original dogs dataset size. As the noise level starts to increase, we stop being able to use to the out-of-distribution cat samples, but start gaining some blurry dog samples. As the noise level approaches the maximum all the blurry dogs become available for training, such that the amount of data available approaches 100%.

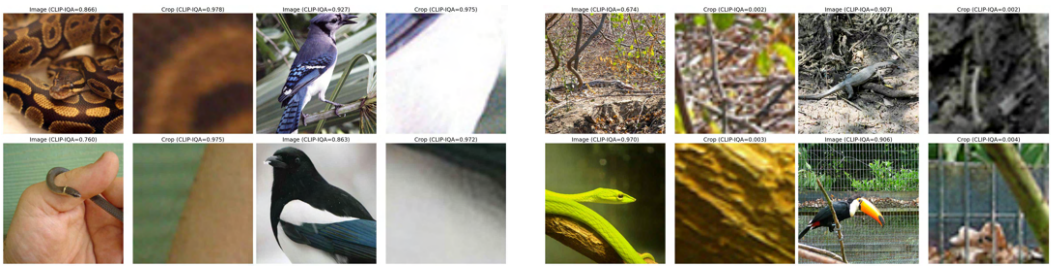


Figure 14: Results using CLIP to obtain the high-quality and the low-quality crops (64x64) of ImageNet.

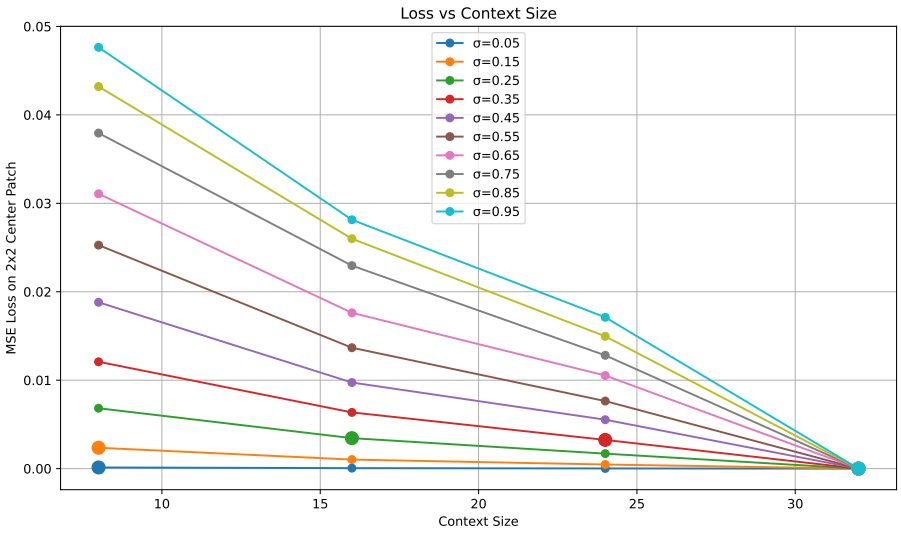


Figure 15: ImageNet-512x512: denoising loss of an optimally trained model, measured at 2×2 center patch, as we increase the context size given to the model (horizontal axis) and the noise level (different curves). As expected, for higher noise, more context is needed for optimal denoising. The large dot on each curve marks the point where the loss nearly plateaus.

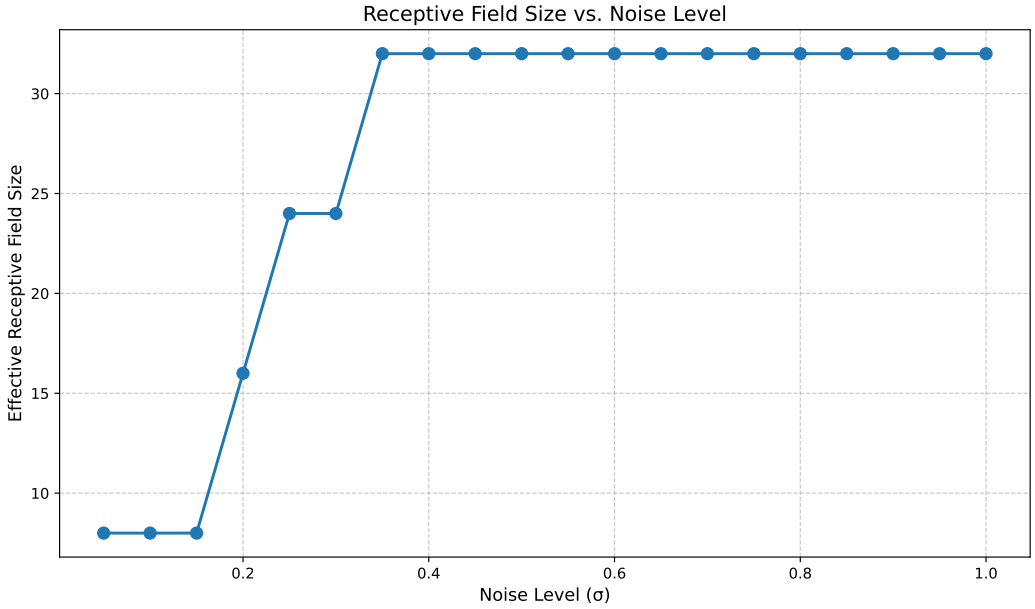


Figure 16: ImageNet-512x512: context size needed to be within $\epsilon = 1e-3$ of the optimal loss for different noise levels. As expected, for higher noise, more context is needed for optimal denoising.

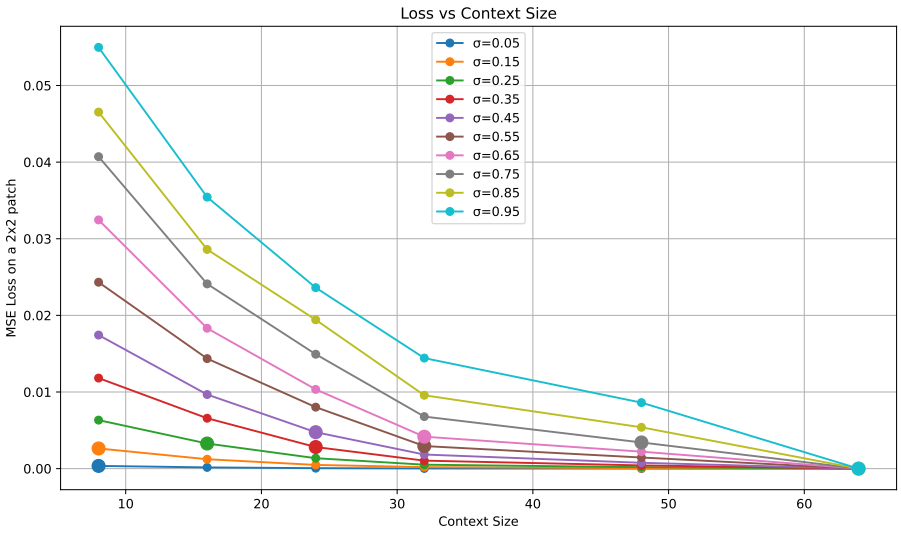


Figure 17: FFHQ: denoising loss of an optimally trained model, measured at 2×2 center patch, as we increase the context size given to the model (horizontal axis) and the noise level (different curves). As expected, for higher noise, more context is needed for optimal denoising. The large dot on each curve marks the point where the loss nearly plateaus.

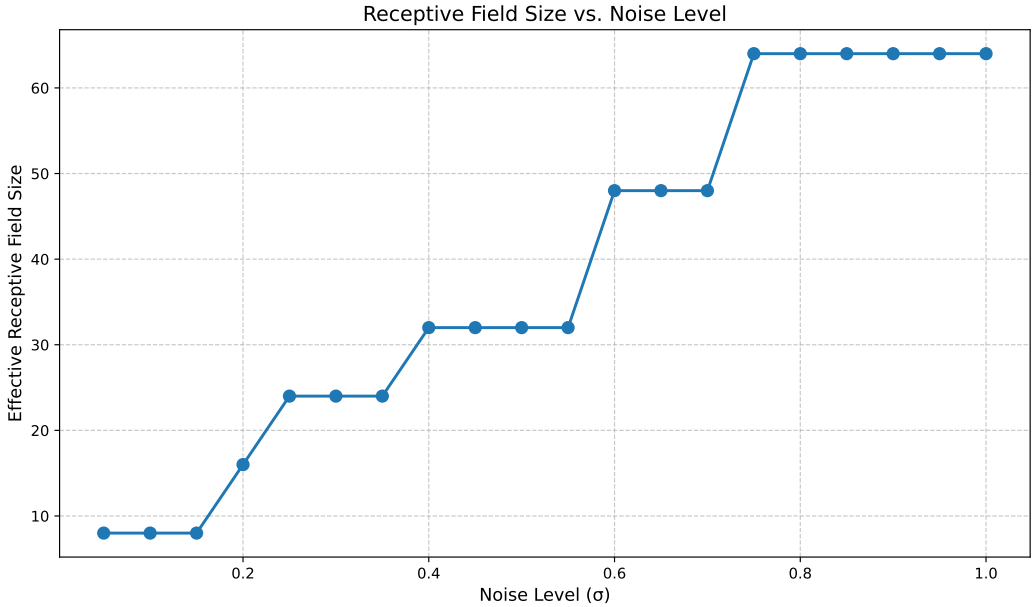
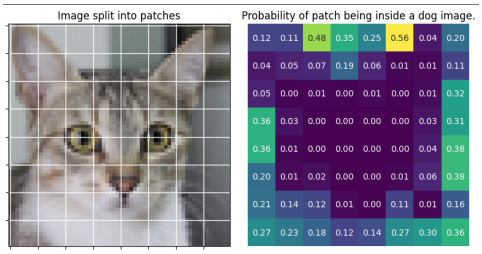
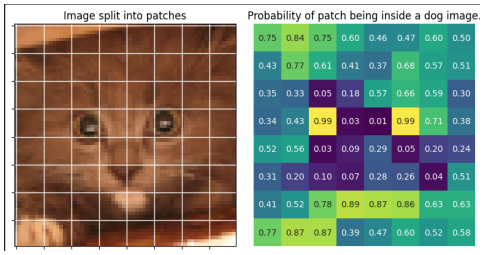


Figure 18: FFHQ: context size needed to be within $\epsilon = 1e-3$ of the optimal loss for different noise levels. As expected, for higher noise, more context is needed for optimal denoising.

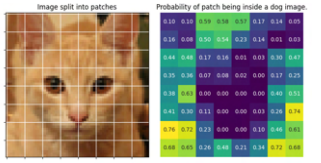


(a) Cat image and classification probabilities over patches.

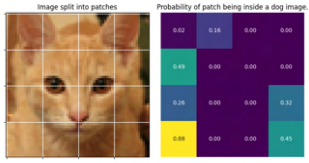


(b) Cat image and classification probabilities over patches.

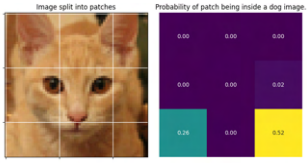
Figure 19: Two examples of cats from the AFHQ dataset. We partition each cat into non overlapping patches and we compute the probabilities of the patch belonging to an image of a dog using a cats vs dogs classifier trained on patches. The cat on the right has a lot more patches that could belong to a dog image according to the classifier, possibly due to the color or the texture of the fur.



(a) Cat annotated by a cats vs. dogs classifier that operates with crops of size 8.



(b) Cat annotated by a cats vs. dogs classifier that operates with crops of size 16.



(c) Cat annotated by a cats vs. dogs classifier that operates with crops of size 24.

Figure 20: Patch-based annotations of a cat image from AFHQ using cats vs. dogs classifiers trained on different patch sizes.

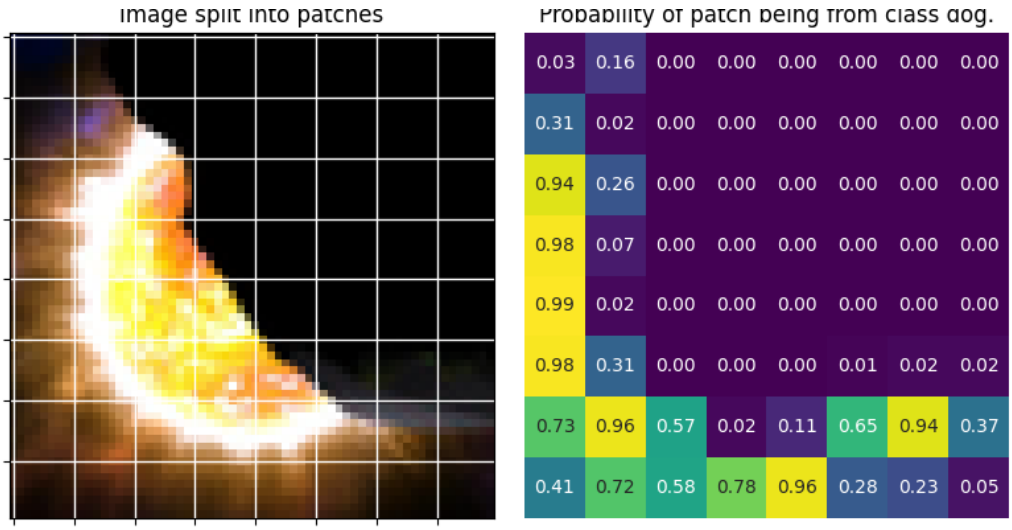
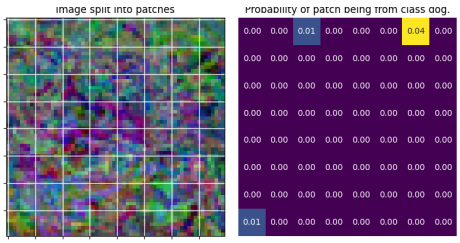
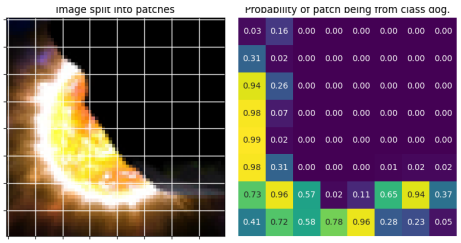


Figure 21: Patch level probabilities for dogness in a synthetic dog image (procedural program). The cat has more useful patches than this non-realistic procedural program.

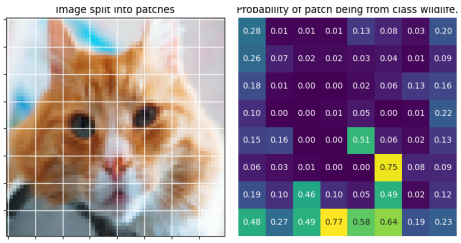


(a) Synthetic image and classification probabilities over patches.

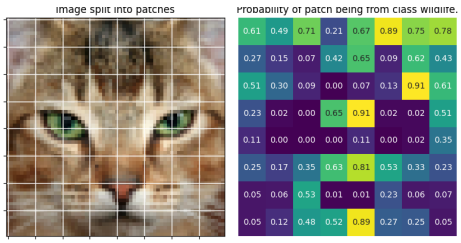


(b) Synthetic image and classification probabilities over patches.

Figure 22: Two examples of procedurally generated images. We partition each image into non overlapping patches and we compute the probabilities of the patch belonging to an image of a dog using a synthetic image vs dogs classifier trained on patches. The image on the right has a lot more patches that could belong to a dog image according to the classifier, possibly due to the color or the texture.



(a) Cat image and classification probabilities over patches.



(b) Cat image and classification probabilities over patches.

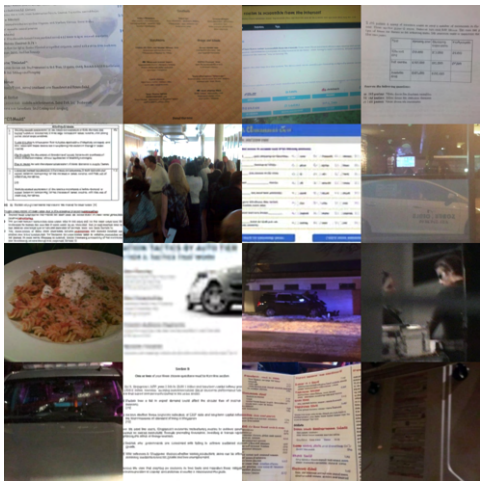
Figure 23: Two examples of cat images. We partition each image into nonoverlapping patches and we compute the probabilities of the patch belonging to an image of wildlife using a cats vs wildlife classifier trained on patches. The image on the right has a lot more patches that could belong to a wildlife image according to the classifier, possibly due to the color or the texture.

(a) Example batch.

(b) Noisy batch.

Figure 24: Example batch.

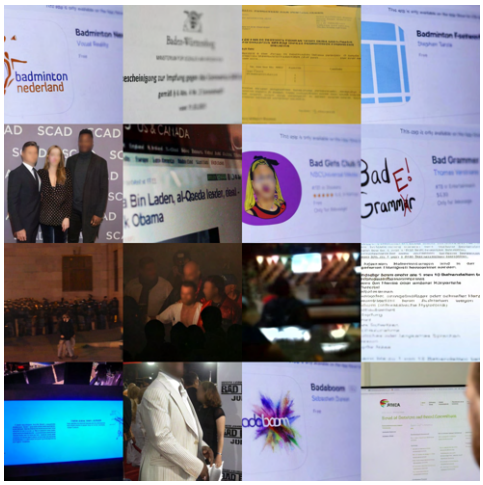
(a) Highest quality images from CC12M according to CLIP.



(b) Lowest quality images from CC12M according to CLIP.

Figure 25: CLIP annotations for quality of images from CC12M.

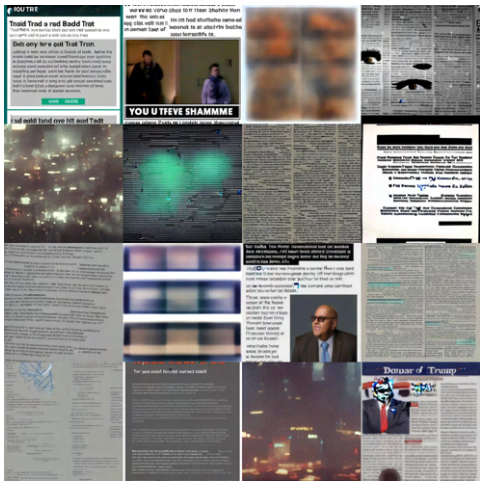
(a) Highest quality images from SA1B according to CLIP.



(b) Lowest quality images from SA1B according to CLIP.

Figure 26: CLIP annotations for quality of images from SA1B.

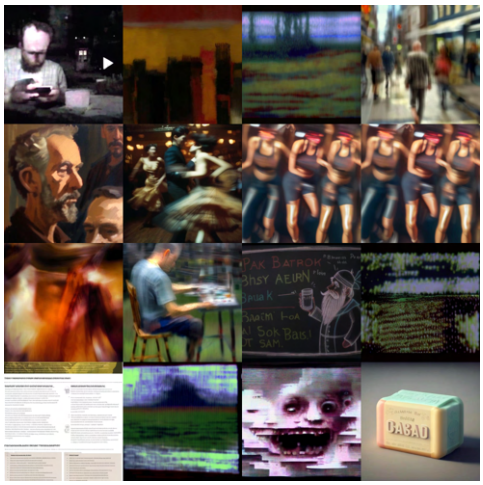
(a) Highest quality images from DiffDB according to CLIP.



(b) Lowest quality images from DiffDB according to CLIP.

Figure 27: CLIP annotations for quality of images from DiffDB.

(a) Highest quality images from JDB according to CLIP.



(b) Lowest quality images from JDB according to CLIP.

Figure 28: CLIP annotations for quality of images from JDB.

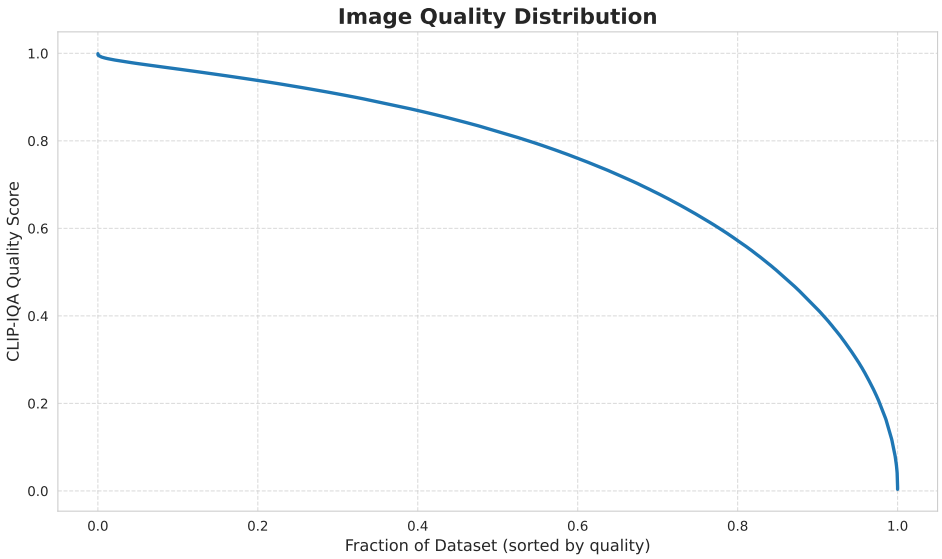


Figure 29: Distribution of image qualities according to CLIP for ImageNet-512.