Your Diffusion Model is Secretly a Zero-Shot Classifier

Alexander C. Li Mihir Prabhudesai Shivam Duggal Ellis Brown Deepak Pathak

Carnegie Mellon University

Abstract

The recent wave of large-scale text-to-image diffusion models has dramatically increased our text-based image generation abilities. These models can generate realistic images for a staggering variety of prompts and exhibit impressive compositional generalization abilities. Almost all use cases thus far have solely focused on sampling; however, diffusion models can also provide conditional density estimates, which are useful for tasks beyond image generation. In this paper, we show that the density estimates from large-scale text-to-image diffusion models like Stable Diffusion can be leveraged to perform zero-shot classification ***without any additional training****. Our generative approach to classification, which we call* ***Diffusion Classifier****, attains strong results on a variety of benchmarks and outperforms alternative methods of extracting knowledge from diffusion models. Although a gap remains between generative and discriminative approaches on zero-shot recognition tasks, we find that our diffusion-based approach has stronger multimodal relational reasoning abilities than competing discriminative approaches. Finally, we use Diffusion Classifier to extract standard classifiers from class-conditional diffusion models trained on ImageNet. Even though these models are trained with weak augmentations and no regularization, they approach the performance of SOTA discriminative classifiers. Overall, our results are a step toward using generative over discriminative models for downstream tasks. Results and visualizations on our website:* diffusion-classifier.github.io/

1. Introduction

To Recognize Shapes, First Learn to Generate Images [31]—in this seminal paper, Geoffrey Hinton emphasizes generative modeling as a crucial strategy for training artificial neural networks for discriminative tasks like image recognition. Although generative models tackle the more challenging task of accurately modeling the underlying data distribution, they can create a more complete representation of the world that can be utilized for various downstream tasks. As a result, a plethora of implicit and explicit generative modeling approaches [26, 42, 45, 21, 76, 69, 78] have been proposed over the last decade. However, the primary focus of these works has been content creation [18, 8, 39, 40, 75, 34] rather than their ability to perform discriminative tasks. In this paper, we revisit this classic generative vs. discriminative debate in the context of diffusion models, the current state-of-the-art generative model family. In particular, we examine *how diffusion models compare against the state-of-the-art discriminative models on the task of image classification.*

Diffusion models are a recent class of likelihood-based generative models that model the distribution of the data via an iterative noising and denoising procedure [69, 35]. They have recently achieved state-of-the-art performance [20] on several text-based content creation and editing tasks [24, 66, 34, 65, 58]. Diffusion models operate by performing two iterative processes—the fixed *forward process*, which destroys structure in the data by iteratively adding a small amount of noise, and the learned *backward process*, which attempts to recover the structure in the noised data. These models are trained via a variational objective, which maximizes an evidence lower bound (ELBO) [5] of the data log-likelihood. For most diffusion models, computing the ELBO simply consists of repeatedly adding noise ϵ to a sample, using the neural network to predict the added noise, and measuring the prediction error.

Conditional generative models like diffusion models can be easily converted into classifiers [53]. Given an input $\mathbf{x}$ and a finite set of classes $\mathbf{c}$ that we want to choose from, we can use the model to compute class-conditional likelihoods $p_\theta(\mathbf{x} \mid \mathbf{c})$. Then, by selecting an appropriate prior distribution $p(\mathbf{c})$ and applying Bayes' theorem, we can get predicted class probabilities $p(\mathbf{c} \mid \mathbf{x})$. For conditional diffusion models that use an auxiliary input, like a class index for class-conditioned models or prompt for text-to-image models, we can do this by leveraging the ELBO as an approximate class-conditional log-likelihood $\log p(\mathbf{x} \mid \mathbf{c})$. In practice, obtaining a diffusion model classifier through Bayes' theorem consists of repeatedly adding noise and computing a Monte Carlo estimate of the expected noise reconstruction losses (also called ϵ-prediction loss) for every class. We call

Correspondence to: Alexander Li <alexanderli@cmu.edu>

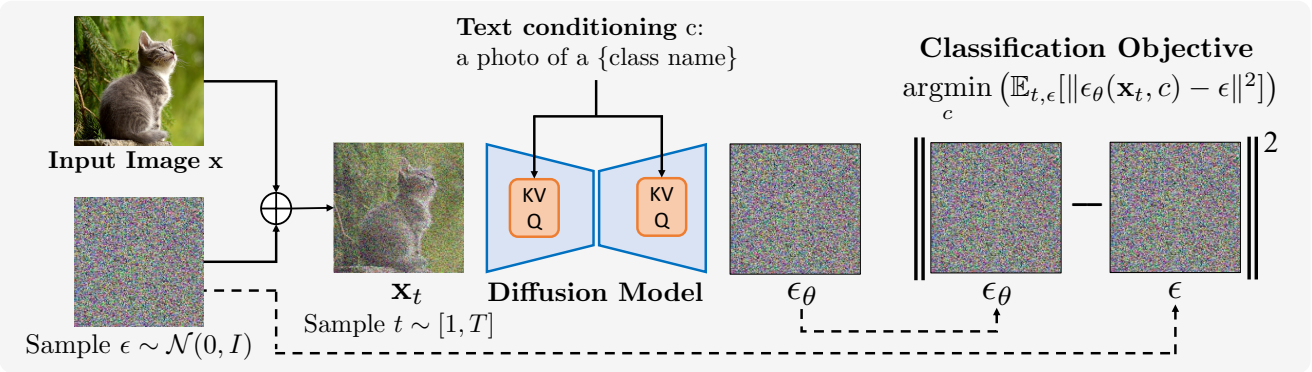

Figure 1. **Overview of our Diffusion Classifier approach:** Given an input image $\mathbf{x}$ and a set of possible conditioning inputs (e.g., text for Stable Diffusion or class index for DiT), we use a diffusion model to choose the one that best fits this image. Diffusion Classifier is theoretically motivated through the variational view of diffusion models and uses the ELBO to approximate $\log p_\theta(\mathbf{x} \mid \mathbf{c})$. Diffusion Classifier chooses the conditioning $\mathbf{c}$ that best predicts the noise added to the input image. *Diffusion Classifier can be used to extract a zero-shot classifier from Stable Diffusion and a standard classifier from DiT without any additional training.*

this approach **Diffusion Classifier**. Diffusion Classifier can extract zero-shot classifiers from text-to-image diffusion models and standard classifiers from class-conditional diffusion models, *without any additional training*. We develop techniques for appropriately choosing diffusion timesteps to compute errors at, reducing variance in the estimated probabilities, and speeding up classification inference.

We highlight the surprising effectiveness of our proposed Diffusion Classifier on zero-shot and supervised classification tasks by comparing against multiple baselines on ten different datasets, including the challenging ObjectNet [4], ImageNetV2 [63], and ImageNet-A [30] datasets. To the best of our knowledge, *our approach is among the first generative modeling approaches to achieve competitive zero-shot classification accuracy with state-of-the-art methods such as CLIP (Table 1).* Finally, our supervised classification experiments (Figure 8) highlight that *our generative approach is catching up to SOTA discriminative classifiers on ImageNet, both in-distribution and out-of-distribution.*

2. Related Work

Generative Models for Discriminative Tasks: Machine learning algorithms designed to solve common classification or regression tasks generally operate under two paradigms: *discriminative* approaches directly learn to model the decision boundary of the underlying task, while *generative approaches* learn to model the distribution of the data and then address the underlying task as a maximum likelihood estimation problem. Algorithms like naive Bayes [53], VAEs [42], GANs [26], EBMs [23, 45], and diffusion models [69, 35] fall under the category of generative models. The idea of modeling the data distribution to better learn the discriminative feature has been highlighted by several seminal works [31, 53, 62]. These works train deep belief networks [32] to model the underlying image data as latents, which are later used for image recognition tasks. Recent works on generative modeling have also learned efficient representations for both global and dense prediction tasks like classification [28, 33, 13, 8, 19] and segmentation [46, 82, 10, 3, 9]. Moreover, such models [27, 50, 37] have been shown to *generalize better, be more robust, and be better calibrated.* However, most of the aforementioned works either train jointly for discriminative and generative modeling or fine-tune generative representations for downstream tasks. Directly utilizing generative models for discriminative tasks is a relatively less-studied problem, and in this work, we particularly highlight the *efficacy of directly using recent diffusion models as zero-shot image classifiers.*

Diffusion Models: Diffusion models [35, 69] have recently gained significant attention from the research community due to their ability to generate high-fidelity and diverse content like images [66, 54, 24], videos [68, 34, 77], 3D [58, 49], and audio [43, 51] from various input modalities like text. Diffusion models are also closely tied to EBMs [45, 23], denoising score matching [71, 79], and stochastic differential equations [72, 83]. In this work, we investigate to what extent the impressive high-fidelity generative abilities of these diffusion models can be utilized for discriminative tasks (namely classification). We take advantage of the variational view of diffusion models for efficient and parallelizable density estimates. The prior work of Dhariwal & Nichol [20] proposed using a classifier network to modify the output of an unconditional generative model to obtain class-conditional samples. Our goal is the reverse: using diffusion models as classifiers.

Zero-Shot Image Classification: Classifiers thus far have usually been trained in a supervised setting where the train and test sets are fixed and limited. CLIP [60] showed that exploiting large-scale image-text data can result in zero-shot generalization to various new tasks. Since then there has been a surge towards building a new category

of classifiers, known as zero-shot or open-vocabulary classifiers, that are capable of detecting a wide range of class categories [25, 47, 48, 1]. These methods have been shown to learn robust representations that generalize to various distribution shifts [38, 16, 73]. Note that in spite of them being called "zero-shot," it is still unclear whether evaluation samples lie in their training data distribution. In contrast to the discriminative approaches above, we propose extracting a zero-shot classifier from a large-scale *generative* model.

3. Method: Classification via Diffusion Models

We describe our approach for calculating class conditional density estimates in a practical and efficient manner using diffusion models. We first provide an overview of diffusion models (Sec. 3.1), discuss the motivation and derivation of our Diffusion Classifier method (Sec. 3.2), and finally propose techniques to improve its accuracy (Sec. 3.3).

3.1. Diffusion Model Preliminaries

Diffusion probabilistic models ("diffusion models" for short) [69, 35] are generative models with a specific Markov chain structure. Starting at a clean sample $\mathbf{x}_0$, the fixed forward process $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ adds Gaussian noise, whereas the learned reverse process $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$ tries to denoise its input, optionally conditioning on a variable $\mathbf{c}$. In our setting, $\mathbf{x}$ is an image and $\mathbf{c}$ represents a low-dimensional text embedding (for text-to-image synthesis) or class index (for class-conditional generation). Diffusion models define the conditional probability of $\mathbf{x}_0$ as:

$$p_\theta(\mathbf{x}_0 \mid \mathbf{c}) = \int_{\mathbf{x}_{1:T}} p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c}) \, \mathrm{d}\mathbf{x}_{1:T} \quad (1)$$

where $p(\mathbf{x}_T)$ is typically fixed to $\mathcal{N}(0, I)$. Directly maximizing $p_\theta(\mathbf{x}_0)$ is intractable due to the integral, so diffusion models are instead trained to minimize the variational lower bound (ELBO) of the log-likelihood:

$$\log p_\theta(\mathbf{x}_0 \mid \mathbf{c}) \geq \mathbb{E}_q \left[\log \frac{p_\theta(\mathbf{x}_{0:T}, \mathbf{c})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right] \quad (2)$$

Diffusion models parameterize $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$ as a Gaussian and train a neural network to map a noisy input $\mathbf{x}_t$ to a value used to compute the mean of $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$. Using the fact that each noised sample $\mathbf{x}_t = \sqrt{\bar{\alpha}_{t_i}}\mathbf{x} + \sqrt{1 - \bar{\alpha}_{t_i}}\epsilon_i$ can be written as a weighted combination of a clean input $\mathbf{x}$ and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, diffusion models typically learn a network $\epsilon_\theta(\mathbf{x}_t, \mathbf{c})$ that estimates the added noise. Using this parameterization, the ELBO can be written as:

$$-\mathbb{E}_\epsilon \left[\sum_{t=2}^{T} w_t \|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})\|^2 - \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{c}) \right] + C \quad (3)$$

Algorithm 1 `Diffusion Classifier`

Input: test image $\mathbf{x}$, conditioning inputs $\{\mathbf{c}_i\}_{i=1}^n$ (e.g., text embeddings or class indices), number of stages N_{stages}, list `KeepList` of number of $\mathbf{c}_i$ to keep after each stage, list `TrialList` of number of trials done by each stage
Initialize $\texttt{Errors}[\mathbf{c}_i] = \text{list}()$ for each $\mathbf{c}_i$
$\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^n$
`PrevTrials` = 0
for stage i $= 1, \ldots, N_{\text{stages}}$ **do**
  for trial $j = 1, \ldots, \texttt{TrialList}[i] - \texttt{PrevTrials}$ **do**
    Sample $t \sim [1, 1000]$
    Sample $\epsilon \sim \mathcal{N}(0, I)$
    $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x} + \sqrt{1 - \bar{\alpha}_t}\epsilon$
    for conditioning $\mathbf{c}_k \in \mathcal{C}$ **do**
      $\texttt{Errors}[\mathbf{c}_k].\texttt{append}(\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_k)\|^2)$
    end for
  end for
  // Keep best $\texttt{KeepList}[i]$ $\mathbf{c}_k$ with the lowest errors
  $\mathcal{C} \leftarrow \underset{\substack{\mathcal{S} \subset C; \\ |\mathcal{S}| = \texttt{KeepList}[i]}}{\arg\min} \sum_{c_k \in \mathcal{S}} \texttt{mean}(\texttt{Errors}[\mathbf{c}_k])$
  `PrevTrials` = `TrialList`[i]
end for
return $\underset{\mathbf{c}_i \in \mathcal{C}}{\arg\min}\, \texttt{mean}(\texttt{Errors}[\mathbf{c}_i])$

where C is a constant term that does not depend on $\mathbf{c}$. Since $T = 1000$ is large and $\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{c})$ is typically small, we choose to drop this term. Finally, [35] find that removing w_t improves sample quality metrics, and many follow-up works also choose to do so. We found that deviating from the uniform weighting used at training time hurts accuracy, so we set $w_t = 1$. Thus, this gives us our final expression for the ELBO:

$$-\mathbb{E}_{t,\epsilon} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})\|^2 \right] + C \quad (4)$$

3.2. Classification with diffusion models

In general, classification using a conditional generative model can be done by using Bayes' theorem on the model predictions and the prior $p(\mathbf{c})$ over labels $\{\mathbf{c}_i\}$:

$$p_\theta(\mathbf{c}_i \mid \mathbf{x}) = \frac{p(\mathbf{c}_i)\, p_\theta(\mathbf{x} \mid \mathbf{c}_i)}{\sum_j p(\mathbf{c}_j)\, p_\theta(\mathbf{x} \mid \mathbf{c}_j)} \quad (5)$$

A uniform prior over $\{\mathbf{c}_i\}$ (i.e., $p(\mathbf{c}_i) = \frac{1}{N}$) is natural and leads to all of the $p(\mathbf{c})$ terms cancelling. For diffusion models, computing $p_\theta(\mathbf{x} \mid \mathbf{c})$ is intractable, so we use the ELBO in place of $\log p_\theta(\mathbf{x} \mid \mathbf{c})$ and use Eq. 4 and Eq. 5 to obtain a

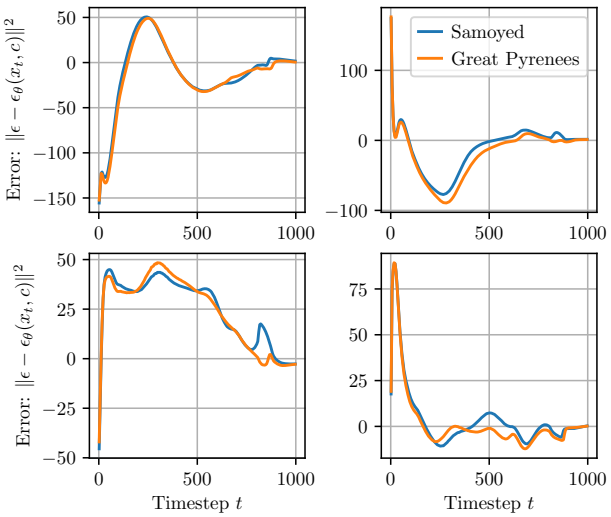

Figure 2. We show the ϵ-prediction error for a fixed image of a Great Pyrenees dog and two prompts. Each subplot corresponds to a single ϵ, with the error evaluated for every $1 \leq t \leq 1000$. Errors are normalized to be zero-mean at each timestep across the 4 plots, and lower is better. Variance in ϵ-prediction error is high across different ϵ, but the variance in relative error between prompts at each t is much smaller for the same ϵ.

posterior distribution over $\{\mathbf{c}_i\}_{i=1}^N$:

$$p_\theta(\mathbf{c}_i \mid \mathbf{x}) \approx \frac{\exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_i)\|^2] + C\}}{\sum_j \exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_j)\|^2] + C\}} \tag{6}$$

$$= \frac{\exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_i)\|^2]\}}{\sum_j \exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_j)\|^2]\}} \tag{7}$$

We compute an unbiased Monte Carlo estimate of each expectation by sampling N (t_i, ϵ_i) pairs, with $t_i \sim [1, 1000]$ and $\epsilon \sim \mathcal{N}(0, I)$, and computing

$$\frac{1}{N}\sum_{i=1}^{N} \left\|\epsilon_i - \epsilon_\theta(\sqrt{\bar{\alpha}_{t_i}}\mathbf{x} + \sqrt{1-\bar{\alpha}_{t_i}}\epsilon_i, \mathbf{c}_j)\right\|^2 \tag{8}$$

By plugging Eq. 8 into Eq. 7, we can extract a classifier from any conditional diffusion model. We call this method **Diffusion Classifier**. *Diffusion Classifier is a powerful, hyperparameter-free approach to extracting classifiers from pretrained diffusion models without any additional training.* Diffusion Classifier can be used to extract a zero-shot classifier from a text-to-image model like Stable Diffusion [64], to extract a standard classifier from a class-conditional diffusion model like DiT [57], and so on. We show an overview of our method in Figure 1.

3.3. Variance Reduction via Difference Testing

At first glance, it seems that accurately estimating $\mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})\|^2\right]$ for each class $\mathbf{c}$ requires prohibitively many samples. Indeed, a Monte Carlo estimate

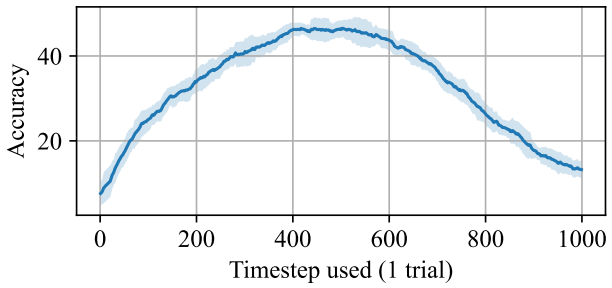

Figure 3. **Pets accuracy, evaluating only a single timestep per class**. Small t corresponds to less noise added, and large t corresponds to significant noise. Accuracy is highest when an intermediate amount of noise is added ($t = 500$).

even using thousands of samples is not precise enough to distinguish classes reliably. However, a key observation is that classification only requires the *relative* differences between the prediction errors, not their *absolute* magnitudes. We can rewrite the approximate $p_\theta(\mathbf{c}_i \mid \mathbf{x})$ from Eq. 7 as:

$$\frac{1}{\sum_j \exp\{\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_i)\|^2 - \|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_j)\|^2]\}} \tag{9}$$

Eq. 9 makes apparent that we only need to estimate the *difference* in prediction errors across each conditioning value. Practically, instead of using different random samples of (t_i, ϵ_i) to estimate the ELBO for each conditioning input $\mathbf{c}$, we simply sample a fixed set $S = \{(t_i, \epsilon_i)\}$ and use the same samples to estimate the ϵ-prediction error for every $\mathbf{c}$. This is reminiscent of paired difference tests in statistics, which increase their statistical power by matching conditions across groups and computing differences.

In Figure 2, we sample 4 fixed ϵ_i's and evaluate $\|\epsilon_i - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x} + \sqrt{1-\bar{\alpha}_t}\epsilon_i, \mathbf{c})\|^2$ for every $t \in 1, \ldots, 1000$, two prompts ("Samoyed dog" and "Great Pyrenees dog"), and a fixed input image of a Great Pyrenees. Even for a fixed prompt, the ϵ-prediction error varies wildly across the specific ϵ used. However, the error difference between each prompt is much more consistent. *Thus, by using the same (t_i, ϵ_i) for each conditioning input, our estimate of $p_\theta(\mathbf{c}_i \mid \mathbf{x})$ is much more accurate.*

4. Practical Considerations

Our Diffusion Classifier method requires repeated error prediction evaluations for every class in order to classify an input image. These evaluations naively require significant inference time, even with the technique presented in Sec 3.3. In this section, we present further insights and optimizations that reduce our method's runtime.

4.1. Effect of timestep

Diffusion Classifier, which is a theoretically principled method for estimating $p(\mathbf{c}_i \mid \mathbf{x})$, uses a uniform distribution over the timestep t for estimating the ϵ-prediction error. Here, we check if alternate distributions over t yield more accurate results. Figure 3 shows the Pets accuracy when using only a single timestep evaluation per class. Perhaps intuitively, accuracy is highest when using intermediate timesteps ($t \approx 500$). This begs the question: can we improve accuracy by oversampling intermediate timesteps and undersampling low or high timesteps?

We try a variety of timestep sampling strategies, including repeatedly trying $t = 500$ with many random ϵ, trying N evenly spaced timesteps, and trying the middle $t - N/2, \ldots, t + N/2$ timesteps. The tradeoff between different strategies is whether to try a few t_i repeatedly with many ϵ or to try many t_i once. Figure 4 shows that all strategies improve when taking using average error of more samples, but simply using evenly spaced timesteps is best. We hypothesize that repeatedly trying a small set of t_i scales poorly since this biases the ELBO estimate.

4.2. Efficient Classification

A naive implementation of our method requires $C \times N$ trials to classify a given image, where C is the number of classes and N is the number of (t, ϵ) samples to evaluate for each conditional ELBO. However, we can do better. Since we only care about $\arg\max_{\mathbf{c}} p(\mathbf{c} \mid \mathbf{x})$, we can stop computing the ELBO for classes we can confidently reject. Thus, one option to classify an image is to use an upper confidence bound algorithm [2] to allocate most of the compute to the top candidates. However, this would require making the assumption that the distribution of $\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_j)\|^2$ is the same across timesteps t. We found that a simpler method works just as well. We split our evaluation into a series of stages, where in each stage we try each remaining c_i some number of times and then remove the ones that have the highest average error. This allows us to efficiently eliminate classes that are almost certainly not the final output and allocate more compute to reasonable classes. As an example, on the Pets dataset, we have $N_{\text{stages}} = 2$ stages. We try each class 25 times in the first stage, then prune to the 5 classes with the smallest average error. Finally, in the second stage we try each of the 5 remaining classes 225 additional times. In Algorithm 1, we write this as $\texttt{KeepList} = (5, 1)$ and $\texttt{TrialList} = (25, 250)$. With this evaluation strategy, classifying one Pets image requires 15 seconds on a single 3090 GPU. As our work focuses on understanding diffusion model capabilities, and does not propose a practical inference algorithm, we do not significantly tune the evaluation strategies. Future work could focus on further speeding up inference time. Further details are in Appendix B.1.

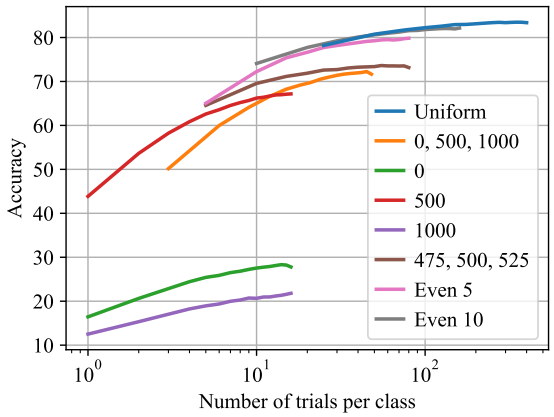

Figure 4. **Zero-shot scaling curves for different timestep sampling strategies**. We evaluate a variety of strategies for choosing the timesteps at which we evaluate the ϵ-prediction error. Each strategy name indicates which timesteps it uses— e.g., "0" only uses the first timestep, "$0, 500, 1000$" uses only the first, middle and last, "Even 10" uses 10 evenly spaced timesteps. We allocate more ϵ evaluations at the chosen timesteps as the number of trials increases. Strategies that repeatedly sample from a restricted set of timesteps, like "475, 500, 525", scale poorly with trials. Using timesteps uniformly from the full range [1, 1000] scales best.

5. Experimental Details

We provide setup details, baselines & datasets for zero-shot and supervised classification.

5.1. Zero-shot Classification

Diffusion Classifier Setup: We build Diffusion Classifier on top of Stable Diffusion [64], a text-to-image latent diffusion model trained on a filtered subset of LAION-5B [67]. For more details on Stable Diffusion model, refer to [64].

Baselines: We provide results using two strong discriminative zero-shot models: (a) CLIP ResNet-50 [59] and (b) OpenCLIP ViT-H/14 [11]. We provide these for reference only, as these models have very different architectures from ours and cannot be compared apples-to-apples. We further compare our approach against two alternative ways to extract class labels from diffusion models: (c) **Synthetic-Labeled-SD**: We train a ResNet-50 classifier on synthetic data generated using Stable Diffusion (with class-names as prompts), (d) **Real-Labeled-SD**: This baseline is not a zero-shot classifier, as it requires a **labeled dataset** of real-world images and class-names. Inspired by Label-DDPM [3], we extract Stable Diffusion features (mid-layer U-Net features at a resolution $[8 \times 8 \times 1024]$ at timestep $t = 100$), and then fit a ResNet-50 classifier on the extracted features and corresponding ground-truth labels. Details are in Appendix B.3.

Datasets: We evaluate the zero-shot classification performance across eight datasets: Food-101 [6], CIFAR-10 [44], FGVC-Aircraft [52], Oxford-IIIT Pets [56], Flowers102

	Zero-shot?	Food101	CIFAR10	FGVC	Oxford Pets	Flowers102	STL10	ImageNet	ObjectNet
Synthetic SD Data	✓	12.6	35.3	9.4	31.3	22.1	38.0	18.9	5.2
SD Features	✗	73.0	**84.0**	**35.2**	75.9	**70.0**	87.2	56.6	10.2
Diffusion Classifier (ours)	✓	**77.9**	76.3	24.3	**85.7**	56.8	**94.2**	**58.4**	**38.3**
CLIP ResNet-50	✓	81.1	75.6	19.3	85.4	65.9	94.3	58.2	40.0
OpenCLIP ViT-H/14	✓	92.7	97.3	42.3	94.6	79.9	98.3	76.8	69.2

Table 1. **Zero-shot classification performance**. Our zero-shot Diffusion Classifier method (which utilizes Stable Diffusion) significantly outperforms the zero-shot diffusion model baseline that trains a classifier on synthetic SD data. Diffusion Classifier also generally outperforms the baseline trained on Stable Diffusion features, especially on complex datasets like ImageNet, in spite of the fact that "SD Features" uses the entire training set to train a classifier. Finally, although it is difficult to make an apples-to-apples comparison due to architecture, our generative approach surprisingly matches CLIP ResNet-50 performance and is competitive with OpenCLIP ViT-H.

[55], STL-10 [12], ImageNet [17] and ObjectNet [4]. We also evaluate zero-shot compositional reasoning ability on the Winoground benchmark [74] .

5.2. Supervised Classification

Diffusion Classifier Setup: We build Diffusion Classifier on top of Diffusion Transformer (DiT) [57], a class-conditional latent diffusion model trained on ImageNet. Other details are the same as those in the zero-shot setting.

Baselines: We compare against these discriminative models trained with cross-entropy on ImageNet: ResNet-18, ResNet-34, ResNet-50, and ResNet-101 [29], as well as ViT-L/32, ViT-L/16, and ViT-B/16 [22].

Datasets: We evaluate the in-distribution (ImageNet) and out-of-distribution (remaining datasets) generalization of Diffusion Classifier and the discriminative baselines on four different datasets: ImageNet [17], ImagenNet-A [30], ImageNetV2 [63], and ObjectNet [4]. We evaluate on 125 shared classes (with 5 data samples per class) between the ImageNet, ImageNetV2, and ObjectNet datasets and 27 common classes of the ImageNet-A dataset.

6. Experimental Results

In this section, we conduct detailed experiments aimed at addressing the following questions:

1. How does our model compare against zero-shot state-of-the-art classifiers such as CLIP?
2. How does our method compare against alternative approaches for classification with diffusion models?
3. How well does our method compare to discriminative models trained on the same dataset?
4. How robust is our model compared to state-of-art classifiers over various distribution shifts?

6.1. Zero-shot Classification Results

Table 1 shows that Diffusion Classifier significantly outperforms Synthetic-SD-Data baseline, an alternate zero-shot approach of extracting information from diffusion models. Our method also achieves comparable performance to SD-Features, which is a classifier trained *supervised* using the entire *labeled training set* for each dataset. In contrast, our method requires no additional training or labels. Furthermore, while it is difficult to make a fair comparison due to architectural differences, our method matches CLIP ResNet-50 performance and is competitive with OpenCLIP ViT-H. This is a major advancement in the performance of generative approaches, and there are clear avenues for improvement. First, we perform no manual prompt tuning and simply use the prompts used by the CLIP authors. Tuning the prompts to the Stable Diffusion training distribution should improve its recognition abilities.

Second, we suspect that Stable Diffusion classifier accuracy could improve with a wider training distribution. Stable Diffusion was trained on a subset of LAION-5B [67] filtered aggressively to remove low-resolution, potentially NSFW, or unaesthetic images. This decreases the likelihood that it has seen relevant data for many of our datasets. CIFAR10 and STL10, the datasets where Diffusion Classifier has the largest relative gap with CLIP, use images that are too small to pass the 256×256 size requirement.

Finally, another factor that affects performance is the fact that the diffusion model optimization objective is chosen for sample quality over good log-likelihoods. Ho *et al*. [35] found that uniform weighting of the ϵ-prediction error over timesteps improves Inception score and FID at the cost of lower log-likelihoods. Stable Diffusion [64], having been trained on this uniformly weighted objective, thus have worse log-likelihood estimation capabilities (and hence potentially worse at classification) than if they had been trained on the weighted variational objective.

6.2. Analyzing Diffusion Classifier for Zero-Shot Classification

We now analyze why our proposed diffusion-based density estimator, Diffusion Classifier, works well.

Experiment Setup: Given an input image, we first perform DDIM inversion [70, 41] (with 50 timesteps) using Stable Diffusion 2.1 and different captions as prompts: BLIP [47] generated caption, human-refined BLIP gener-

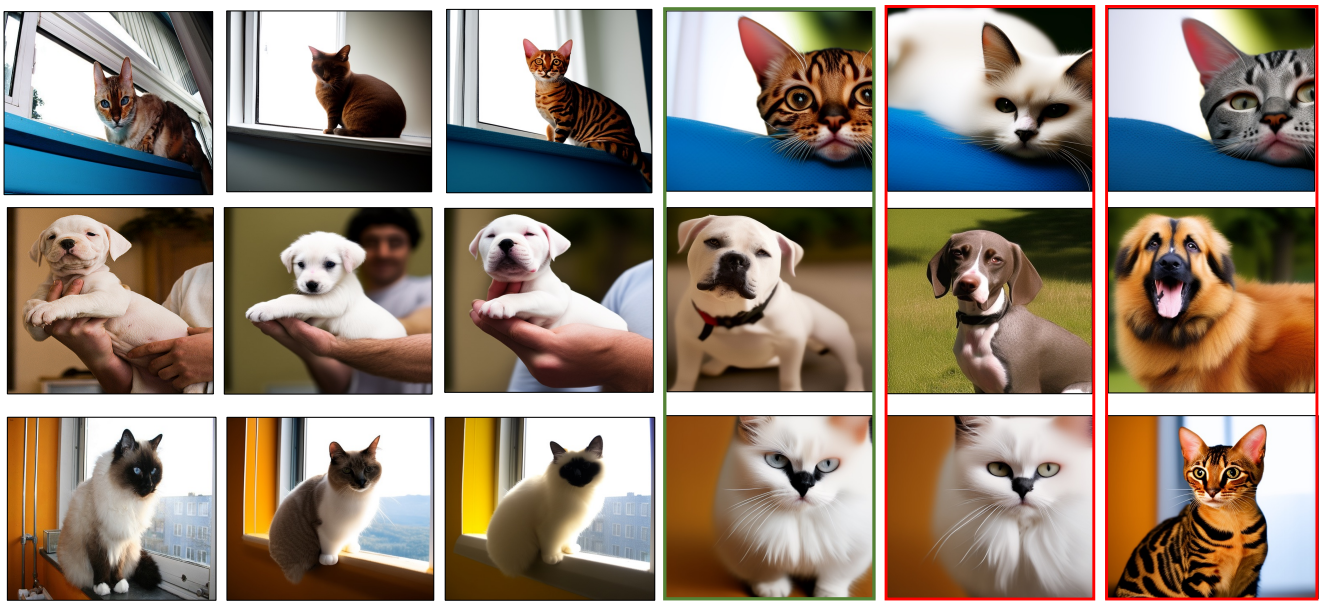

Figure 5. **Analyzing Diffusion Classifier for Zero-Shot Classification:** We analyze the role of different text/captions (BLIP, Human-modified BLIP, correct class-name, incorrect class-name) for zero-shot classification using text-based diffusion models. To do so, we invert the input image using the corresponding caption and then reconstruct it using deterministic DDIM sampling. The image inverted and reconstructed using a human-modified BLIP caption aligns the most with the input image since this caption is the most descriptive. The images reconstructed using correct class names as prompts (column 4) align much better with the input image in terms of class-descriptive features of the underlying object than the images reconstructed using incorrect class names as prompts (columns 5 and 6). Row 3 (columns 4 and 5) demonstrates an example where the base Stable Diffusion does not distinguish the two cat breeds, Birman and Ragdoll, and hence cannot invert/sample them differently. As a result, our classifier also fails.

ated caption, "a photo of {*correct-class-name*}, a type of pet" and "a photo of {*incorrect-class-name*}, a type of pet.". Next, we leverage the inverted DDIM latent and the corresponding prompt to attempt to reconstruct the original image (using a deterministic diffusion scheduler [70]). The underlying intuition behind this experiment is that the inverted image should look more similar to the original image when a correct and appropriate/descriptive prompt is used for DDIM inversion and sampling.

Experimental Evaluation: Figure 5 shows the results of this experiment for the Oxford-IIIT Pets dataset. The image inverted using a human-modified BLIP caption (column 3) is the most similar to the original image (column 1). This aligns with our intuition as this caption is most descriptive of the input image. The human-modified caption (column 2 in Figure 5) only adds the class name (Bengal Cat, American Bull Dog, Birman Cat) ahead of the BLIP predicted "cat or dog" token for the foreground object and slightly enhances the description for the background. *Comparing the BLIP-caption results (column 2) with the human-modified BLIP-caption results (column 3)*, we can see that by just using the class-name as the extra token, the diffusion model can inherit class-descriptive features (Bengal cat has stripes, American Bulldog has a wider chin, Birman cat has a black patch on the face) into the reconstructed image. *This backs our proposal of diffusion-based generative models as strong zero-shot classifiers.*

Compared to the image generated using the oracle (human-generated) caption as a prompt, the images reconstructed using only class names as prompts (columns 4,5,6) align less with the input image (column 1). *This is expected as class names by themselves are not dense descriptions of the input images.* Comparing the results of column 4 (correct class names as prompt) with those of column 5,6 (incorrect class names as prompt), we can see that the foreground object has similar class-descriptive features (brown and black stripes in row 1, white, and black face patches in row 3) to the input image for the correct-prompt reconstructions. This strongly highlights the fact that although using class names as approximate prompts will not lead to absolute perfect denoising or density estimation (Eq. 8), *for the global prediction task of classification, the correct class names should provide enough descriptive features for denoising, relative to the incorrect class names.*

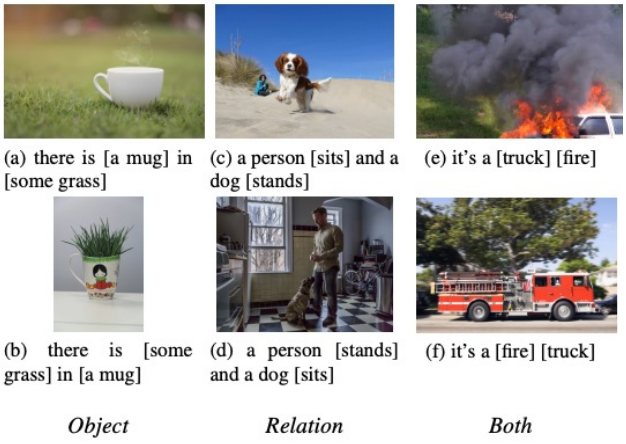

Figure 6. **Example visualizations of Winoground swap types.** Each category corresponds to a different type of linguistic swap in the caption. Object swaps noun phrases, Relation swaps verbs, adjectives, or adverbs, and Both can swap entities of both kinds.

Model	Object	Relation	Both	Average
Random Chance	25.0	25.0	25.0	25.0
CLIP ViT-L/14	27.0	25.8	57.7	28.2
OpenCLIP ViT-H/14	39.0	**26.6**	57.7	33.0
Diffusion Classifier (ours)	**41.8**	25.3	**69.2**	**34.0**

Table 2. **Zero-shot reasoning results on Winoground Object, Relation and Both benchmarks.** Diffusion Classifier improves text score (Eq 10) whenever object swaps are involved (Both also swaps the object). However, performance on Relation still remains roughly at random chance for all three methods.

Row 3 of Figure 5 further highlights an example where the base Stable Diffusion model generates very similar-looking inverted images for correct Birman and incorrect Ragdoll text prompts. As a result, our model also incorrectly classifies Birman cat with Ragdoll, although getting the perfect zero-shot top-2 classification metric. This happens because Ragdolls and Birmans look extremely similar (even to humans). Finally, we fine-tuned the Stable Diffusion diffusion model on a dataset of Ragdoll/Birman cats (175 images in total). Diffusion Classifier using this fine-tuned model achieves a classification accuracy of 85%, significantly higher than the initial zero-shot accuracy of 45%.

6.3. Improved Relational Reasoning Abilities

Large text-to-image diffusion models are capable of generating samples with impressive compositional generalization. In this section, we test whether this generative ability translates to improved compositional *reasoning*.

Winoground Benchmark: We compare Diffusion Classifier to contrastive models like CLIP [59] on Winoground [74], a popular benchmark for evaluating the visuo-linguistic compositional reasoning abilities of

Figure 7. Results on selected Winoground examples.

vision-language models. Each example in Winoground consists of 2 (image, caption) pairs. Notably, both captions within an example contain the exact same set of words, just in a different order. Vision-language multimodal models are scored on Winoground by their ability to match captions C_i to their corresponding images I_i. Given a model that computes a score for each possible pair $score(C_i, I_j)$, the *text score* of a particular example $((C_0, I_0), (C_1, I_1))$ is 1 if and only if it independently prefers caption C_0 over caption C_1 for image I_0 and vice-versa for image I_1. Precisely, the model's text score on an example is:

$$\mathbb{I}[score(C_0, I_0) > score(C_1, I_0) \text{ AND } score(C_1, I_1) > score(C_0, I_1)] \tag{10}$$

Achieving a high text score is extremely challenging. Humans (via Mechanical Turk) achieve 89.5% accuracy on this benchmark, but even the best models do barely above chance. Models can only do well if they understand compositional structure within each modality. CLIP has been found to do poorly on this benchmark since its embeddings tend to be more like a "bag of concepts" that fail to bind subjects to attributes or verbs [80].

Each example is tagged by the type of linguistic swap (object, relation and both) between the two captions:

1. Object: reorder elements like noun phrases that typically refer to real-world objects/subjects.
2. Relation: reorder elements like verbs, adjectives, prepositions, and/or adverbs that modify objects.
3. Both: a combination of the previous two types.

We show examples of each swap type in Figure 6.

Results Table 2 compares Diffusion Classifier to OpenCLIP ViT-H/14 (whose text embeddings Stable Diffusion conditions on) and CLIP ViT-L/14. For the "Relation" swaps, all three models do about the same as a purely random baseline. However, *Diffusion Classifier clearly does better than both discriminative approaches when object swaps are involved (Object and Both).* This indicates that Diffusion Classifier exhibits better compositional generalization than these contrastive methods. Since Stable Dif-

Method	ID	OOD		
	IN	IN-v2	IN-A	ObjectNet
ResNet-18	74.1	57.3	15.0	26.6
ResNet-34	78.1	59.8	10.5	31.6
ResNet-50	79.7	61.6	9.8	35.6
ResNet-101	82.2	63.2	19.5	38.2
ViT-L/32	79.0	61.6	26.3	29.9
ViT-L/16	81.0	66.6	25.6	36.7
ViT-B/16	83.4	66.6	30.1	37.8
Diffusion Classifier	78.9	62.1	22.6	32.3

Table 3. **Diffusion Classifier performs well ID and OOD.** We compare our generative Diffusion Classifier approach to discriminative models trained on ImageNet. We highlight cells where Diffusion Classifier does better.

fusion uses the same text encoder as OpenCLIP ViT-H/14, this improvement must come from better cross-modal binding of concepts to images. Overall, we find it surprising that Stable Diffusion, trained with only sampling in mind, can be repurposed into such a good classifier and reasoner.

In Figure 7, we show examples of some successes and failures. As can be seen from Figure 7 (row 2, column 2), Diffusion Classifier is better able to reason about the spatial and the compositional understanding of the underlying images and hence performs better on the Winoground benchmark. Figure 7 (row 2, column 1) shows a challenging example where all the baselines and our approach fail.

6.4. Supervised Classification Results

In this section, we compare the robustness of our Diffusion Classifier with a variety of strong discriminative models. We compare Diffusion Classifier, leveraging the Imagenet-trained DiT model [57], to variants of ViTs [22] and ResNets [29] trained on ImageNet. In Figure 8 and Table 3 we show that Diffusion Classifier is strongly competitive with state-of-the-art discriminative classifiers on various natural distribution shifts. Diffusion Classifier matches the in-distribution accuracy of a ViT-L/32 model and consistently does better OOD than half of the discriminative methods. Notably, to the best of our knowledge, we are the first to show that a generative model can achieve ImageNet classification accuracy comparable with highly competitive discriminative methods like ViTs [22]. This is especially impressive since DiT was trained with *only random horizontal flips*, unlike typical classifiers that use RandomResizedCrop, Mixup [81], RandAugment [14], and other tricks. Furthermore, [57] reports stable training with fixed learning rate (no warmup or decay) and no regularization other than weight decay. These results indicate that it may be time to revisit a generative approach to classification. Explicitly training diffusion models to maximize their classification accuracy is an exciting avenue for future work.

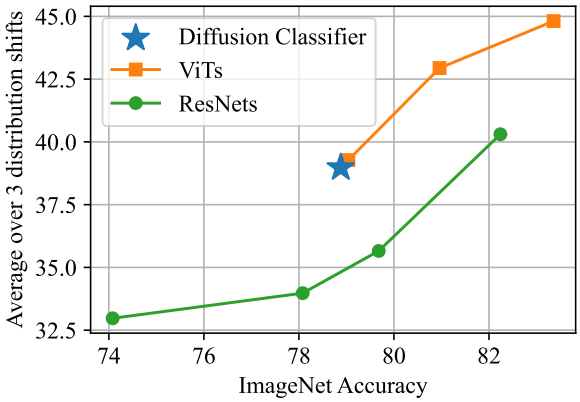

Figure 8. **ImageNet ID vs OOD accuracy** Even with weak augmentations, Diffusion Classifier generalizes OOD better than ResNets and the same as ViTs.

7. Conclusion and Discussion

We investigated diffusion models for zero-shot and supervised classification by leveraging diffusion models as conditional density estimators. By performing a simple unbiased Monte Carlo estimate of the ϵ-predictions at various timesteps of diffusion sampling, we extract **Diffusion Classifier**—a *powerful, zero-shot, and hyper-parameter-free classifier without any additional training.* We find that this classifier narrows the gap with SOTA discriminative approaches on zero-shot and standard classification and outperforms them on multimodal reasoning.

Role of Diffusion Model Design Decisions Since we don't change the base diffusion model of our Diffusion Classifier, we believe the exact choices made during diffusion training affect the classifier. For instance, Stable Diffusion [64] conditions the image generation on the text embeddings from OpenCLIP [38]. However, the language model in OpenCLIP is much weaker than open-ended large-language models like T5-XXL [61] because it is only trained on text data available from image-caption pairs, a minuscule subset of total text data on the Internet. Hence, we believe that diffusion models trained on top of T5-XXL embeddings, such as Imagen [66], should display better zero-shot classification results, but these are not open-source to empirically validate. Other design choices, such as whether to perform diffusion in latent space (e.g. Stable Diffusion) or in pixel space (e.g. DALLE 2), can also affect the adversarial robustness of the classifier and present interesting avenues for future work.

In conclusion, while generative models have previously fallen short of discriminative ones for classification, today's pace of advances in generative modeling means that they may catch up in the near future. Our strong classification, multimodal reasoning, and generalization and results represent an encouraging step in this direction.

Acknowledgements We thank Patrick Chao for helpful discussions and Christina Baek, Rishi Veerapaneni for paper feedback. Stability.AI contributed compute to run some of our experiments. AL is supported by the NSF GRFP DGE1745016 and DGE2140739. This work is supported by NSF IIS-2024594 and ONR MURI N00014-22-1-2773.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 3

[2] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002. 5

[3] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2022. 2, 5, 15

[4] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Joshua B. Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Neural Information Processing Systems*, 2019. 2, 6

[5] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2017. 1

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 5

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 13

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2

[9] Ryan Burgert, Kanchana Ranasinghe, Xiang Li, and Michael S Ryoo. Peekaboo: Text to image diffusion models are zero-shot segmentors. *arXiv preprint arXiv:2211.13224*, 2022. 2

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. 2016. 2

[11] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022. 5

[12] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. 6

[13] Danilo Croce, Giuseppe Castellucci, and Roberto Basili. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. 2

[14] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 9

[15] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022. 14

[16] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023. 3

[17] 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*, pages 248–255. Ieee, 2009. 6

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *preprint arXiv:1810.04805*, 2018. 1

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019. 2

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 1, 2

[21] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *CoRR*, abs/1605.08803, 2016. 1

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *preprint arXiv:2010.11929*, 2020. 6, 9

[23] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *ArXiv*, abs/1903.08689, 2019. 2

[24] Aditya Ramesh et al. Hierarchical text-conditional image generation with clip latents, 2022. 1, 2

[25] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*, 2022. 3

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and

Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2

[27] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020. 2

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021. 2

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 9

[30] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 2, 6

[31] Geoffrey E. Hinton. To recognize shapes, first learn to generate images. *Progress in brain research*, 2007. 1, 2

[32] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 2006. 2

[33] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. 2

[34] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022. 1, 2

[35] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 3, 6

[36] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 13

[37] Yujia Huang, James Gornet, Sihui Dai, Zhiding Yu, Tan Nguyen, Doris Tsao, and Anima Anandkumar. Neural networks with recurrent generative feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020. 2

[38] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. If you use this software, please cite it as below. 3, 9

[39] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 1

[40] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12):4217–4228, 2021. 1

[41] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2426–2435, June 2022. 6

[42] Diederik Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 12 2013. 1, 2

[43] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. 2

[44] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 5

[45] Yann Lecun, Sumit Chopra, and Raia Hadsell. *A tutorial on energy-based learning*. 01 2006. 1, 2

[46] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[47] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 3, 6

[48] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 3

[49] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 2

[50] Hao Liu and P. Abbeel. Hybrid discriminative-generative training via contrastive learning. *ArXiv*, abs/2007.09070, 2020. 2

[51] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023. 2

[52] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 5

[53] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. MIT Press, 2001. 1, 2

[54] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. *CoRR*, abs/2112.10741, 2021. 2

[55] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 6

[56] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 5

[57] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 4, 6, 9, 14

[58] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 5, 8, 15

[60] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2

[61] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 2020. 9

[62] Marc'Aurelio Ranzato, Joshua Susskind, Volodymyr Mnih, and Geoffrey Hinton. On deep generative models with applications to recognition. In *CVPR 2011*, 2011. 2

[63] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, 2019. 2, 6

[64] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 4, 5, 6, 9, 13

[65] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022. 1

[66] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1, 2, 9

[67] 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. *arXiv preprint arXiv:2210.08402*, 2022. 5, 6

[68] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[69] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1, 2, 3

[70] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 6, 7

[71] Yang Song and Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. 2019. 2

[72] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[73] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. 2020. 3

[74] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022. 6, 8

[75] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016. 1

[76] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. JMLR Workshop and Conference Proceedings. JMLR.org, 2016. 1

[77] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual description. *ArXiv*, abs/2210.02399, 2022. 2

[78] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. 1

[79] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008. 2

[80] Yutaro Yamada, Yingtian Tang, and Ilker Yildirim. When are lemons purple? the concept association bias of clip. *arXiv preprint arXiv:2212.12043*, 2022. 8

[81] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 9

[82] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021. 2

[83] Roland S Zimmermann, Lukas Schott, Yang Song, Benjamin A Dunn, and David A Klindt. Score-based generative classifiers. *arXiv preprint arXiv:2110.00473*, 2021. 2

Appendix

A. Techniques that did not help

Diffusion Classifier requires many samples to accurately estimate the ELBO. In addition to using the techniques in Section 3 and 4, we tried several other options for variance reduction. None of the following methods worked, however. We list negative results here for completeness, so others do not have to retry them.

Classifier-free guidance Classifier-free guidance [36] improves the match between a prompt and generated image, at the cost of mode coverage. This is done by training a conditional $\epsilon_\theta(\mathbf{x}_t, \mathbf{c})$ and unconditional $\epsilon_\theta(\mathbf{x}_t)$ denoising network and combining their predictions at sampling time:

$$\tilde{\epsilon}(\mathbf{x}_t, \mathbf{c}) = (1+w)\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) - w\epsilon_\theta(\mathbf{x}_t) \tag{11}$$

where w is a guidance weight that is typically in the range $[0, 10]$. Most diffusion models are trained to enable this trick by occasionally replacing the conditioning $\mathbf{c}$ with an empty token. Intuitively, classifier-free guidance "sharpens" $\log p_\theta(x \mid \mathbf{c})$ by encouraging the model to move away from regions that unconditionally have high probability. We test Diffusion Classifier to see if using the $\tilde{\epsilon}$ from classifier-free guidance can improve confidence and classification accuracy. Our new ϵ-prediction metric is now $\|\epsilon - (1+w)\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) - w\epsilon_\theta(\mathbf{x}_t)\|^2$. However, Figure 9 shows that $w = 0$ (i.e., no classifier-free guidance) performs best. We hypothesize that this is because Diffusion Classifier fails on uncertain examples, which classifier-free guidance affects unpredictably.

Error map cropping The ELBO $\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})\|^2]$ depends on accurately estimating the added noise at every location of the $64 \times 64 \times 4$ image latent. We try to reduce the impact of edge pixels (which are less likely to contain the subject) by computing $\mathbf{x}_t$ as normal, but only measuring the ELBO on a center crop of ϵ and $\epsilon_\theta(\mathbf{x}_t, \mathbf{c})$. We compute:

$$\|\epsilon_{[i:-i,i:-i]} - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})_{[i:-i,i:-i]}\|^2 \tag{12}$$

where i is the number of latent "pixels" to remove from each edge. However, Figure 10 shows that any amount of cropping reduces accuracy.

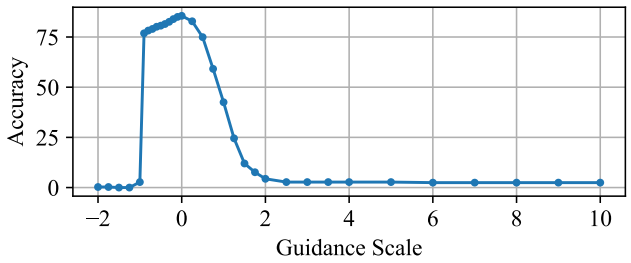

Figure 9. Accuracy plot of classifier-free guidance on Pets.

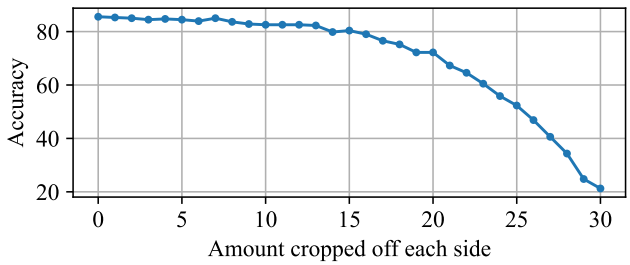

Figure 10. Cropping ϵ and $\epsilon_\theta(\mathbf{x}_t, \mathbf{c})$ reduces accuracy on Pets.

Importance sampling Importance sampling is a common method for reducing the variance of a Monte Carlo estimate. Instead of sampling $\epsilon \sim \mathcal{N}(0, I)$, we sample ϵ from a narrower distribution. We first tried fixing $\epsilon = 0$, which is the mode of $\mathcal{N}(0, I)$, and only varying the timestep t. We also tried the truncation trick [7] which samples $\epsilon \sim \mathcal{N}(0, I)$ but continually resamples elements that fall outside the interval $[a, b]$. Finally, we tried sampling $\epsilon \sim \mathcal{N}(0, I)$ and rescaling them to the expected norm ($\epsilon \rightarrow \frac{\epsilon}{\|\epsilon\|_2}\mathbb{E}_{\epsilon'}[\|\epsilon'\|_2]$) so that there are no outliers. Table 4 shows that none of these importance sampling strategies improve accuracy. This is likely because the noise ϵ sampled with these strategies are completely out-of-distribution for the noise prediction model. For computational reasons, we performed this experiment on a 10% subset of Pets.

B. Additional Implementation Details

B.1. Zero-shot classification using Diffusion Classifier

Training Data For our zero-shot Diffusion Classifier, we utilize Stable Diffusion 2.1 [64]. This model was trained on a subset of the LAION-5B dataset, filtered so that the training data is aesthetic and appropriately safe-for-work. LAION

Sampling distribution for ϵ	Pets accuracy
$\epsilon = 0$	41.3
TruncatedNormal, $[-1, 1]$	49.9
TruncatedNormal, $[-2.5, 2.5]$	81.5
Expected norm	86.9
$\epsilon \sim \mathcal{N}(0, I)$	87.5

Table 4. Every importance sampling strategy underperforms sampling the noise ϵ from a standard normal distribution.

Dataset	Prompts kept per stage	Evaluations per stage	Avg. evaluations per class	Total evaluations
Food101	20 10 5 1	20 50 100 500	50.7	5120
CIFAR10	5 1	100 500	300	3000
FGVC Aircraft	20 10 5 1	20 50 100 500	51	5100
Pets	5 1	25 250	51	1890
Flowers102	20 10 5 1	20 50 100 500	50.4	5140
STL10	5 1	100 500	300	3000
ImageNet	500 50 10 1	50 100 500 1000	100	100000
ObjectNet	25 10 5 1	50 100 500 1000	118.6	13400

Table 5. Evaluation strategy for each zero-shot dataset.

classifiers were used to remove samples that are too small (less than 256×256), potentially pornographic (punsafe ≥ 0.1), or unaesthetic (aesthetic score < 4.5). These thresholds are relatively conservative, since false negatives (leaving NSFW or undesirable images in the training set) is worse than removing extra images from a large starting dataset. As discussed in Section 6.1, these filtering criteria bias the distribution of Stable Diffusion training data and likely negatively affect Diffusion Classifier's performance on datasets whose images do not satisfy these criteria. The checkpoint we use was trained for 550k steps at resolution 256×256 on this subset, followed by an additional 850k steps at resolution 512×512 on images that are at least that large. This checkpoint can be downloaded online through the diffusers repository at `stabilityai/stable-diffusion-2-1-base`.

Inference Details We use FP16 and Flash Attention [15] to improve inference speed. This enables efficient inference with a batch size of 32, which works across a variety of GPUs, from RTX 2080Ti to A6000. We found that adding these two tricks did not affect test accuracy compared to using FP32 without Flash Attention. Given a test image, we resize the shortest edge to 512 pixels using bicubic interpolation, take a 512×512 center crop, and normalize the pixel values to $[-1, 1]$. We then use the Stable Diffusion autoencoder to encode the $512 \times 512 \times 3$ RGB image into a $64 \times 64 \times 4$ latent. We finally classify the test image by applying the method described in Sections 3 and 4 to estimate ϵ-prediction error in this latent space.

Sampling Strategy Table 5 shows the evaluation strategy used for each zero-shot dataset. We hand-picked the strategies based on the number of classes in each dataset. Further gains in accuracy may be possible with more evaluations.

B.2. ImageNet classification using Diffusion Classifier

For this task, we use the recent Diffusion Transformer (DiT) [57] as the backbone of our Diffusion Classifier. DiT [57] was trained on ImageNet-1k, which contains about 1.28 million images from 1,000 unique classes. While it was originally trained to produce high-quality samples with strong FID scores, we repurpose the model and compare it against discriminative models with the same ImageNet-1k training data. Notably, DiT achieves strong performance while using much weaker data augmentations than what discriminative models are usually trained with. During training time for our 256×256 checkpoint, the smaller edge of the input image is resized to 256 pixels. Then, a 256×256 center crop is taken, followed by a random horizontal flip, followed by embedding with the Stable Diffusion autoencoder. At test time, we follow the same preprocessing pipeline, but omit the random horizontal flip. Diffusion Classifier performance in this setting may improve if stronger augmentations, like RandomResizedCrop or color jitter, are used during the diffusion model training process.

Arch	Conv1	Conv2	Conv3 x2	Conv4 x2	Conv5 x2
ResNet-18	7x7x64	3x3 max-pool	3x3x128	3x3x256	3x3x512
ResNet-18 (Real-Labeled-SD)	3x3x1280	-	3x3x1280	3x3x2560	3x3x2560

Table 6. Comparison of Real-Labeled-SD's ResNet-18 classifier architecture with the original ResNet-18

B.3. Baselines for Zero-Shot Classification

Synthetic-SD: We provide the implementation details of the "Synthetic-SD" baseline (row 1 of Table 1) for the task of zero-shot image classification. Our Diffusion Classifier approach builds on the intuition that a model capable of generating examples of desired classes should be able to directly discriminate between them. In contrast, this baseline takes the simple approach of using our generative model, Stable Diffusion, as intended to generate *synthetic training data* for a discriminative model. For a given dataset, we use pre-trained Stable Diffusion 2.1 with default settings to generate $10{,}000$ synthetic 512×512 pixel images per class as follows: we use the English class name and randomly sample a template from those provided by the CLIP [59] authors to form the prompt for each generation. We then train a supervised ResNet-50 classifier using the synthetic data and the labels corresponding to the class name that was used to generate each image. We use batch size $= 256$, weight decay $= 1e-4$, learning rate $= 0.1$ with a cosine schedule, the AdamW optimizer, and use random resized crop & horizontal flip transforms. We create a validation set using the synthetic data by randomly selecting 10% of the images for each class; we use this for early stopping to prevent over-fitting. Finally, we report the accuracy on the target dataset's proper test set.

Real-Labeled-SD: We provide the implementation details of the "Real-Labeled-SD" baseline (row 2 of Table 1) for the task of image classification. This baseline is inspired by Label-DDPM [3], a recent work on diffusion-based semantic segmentation. Unlike Label-DDPM, which leverages a category-specific diffusion model, we directly build on top of the open-sourced Stable Diffusion model (trained on the LAION dataset). We then approach the task of classification as follows: given the pre-trained Stable Diffusion model, we extract the intermediate U-Net features corresponding to the input image. These features are then passed through a ResNet-based classifier to predict the corresponding class name. To extract the intermediate U-Net features, we add a noise equivalent to the $100th$ timestep noise to the input image and evaluate the corresponding noisy latent using the forward diffusion process. We then pass the noisy latent through the U-Net model, conditioned on timestep $t = 100$ and text conditioning (y) as an empty string, and extract out the features from the mid-layer of the U-Net at a resolution of $[8 \times 8 \times 1024]$. Next, we train a supervised classifier on top of these features. *Thus, this baseline is not zero-shot.* The architecture of our classifier is similar to ResNet-18, with small modifications to make it compatible with an input size of $[8 \times 8 \times 1024]$. Table 6 defines these modifications. We set batch size $= 16$, learning rate $= 1e-4$, and use AdamW optimizer. During training, we do augmentations similar to the original ResNet (Random Crop and Flip). We do early stopping using the validation set to prevent over-fitting. We use the official train-test splits for each dataset, except ImageNet and ObjectNet. For these two datasets, we perform class sub-sampling and use the same train-test split as our model. We do this to achieve fair comparisons with the other baselines.