Your Diffusion Model is Secretly a Zero-Shot Classifier

Alexander C. Li [1] **Mihir Prabhudesai** [1] **Shivam Duggal** [1] **Ellis Brown** [1] **Deepak Pathak** [1]

Abstract

The recent wave of large-scale text-to-image diffusion models has dramatically increased our text-based image generation abilities. However, almost all use cases so far have solely focused on sampling. 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. We also 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.

1. Introduction

To Recognize Shapes, First Learn to Generate Images (Hinton, 2007)—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 generative modeling approaches have been proposed over the last decade (Goodfellow et al., 2014; Kingma & Welling, 2013; LeCun et al., 2006; Dinh et al., 2016; Van Den Oord et al., 2016; Sohl-Dickstein et al., 2015; Vincent, 2011). In this paper, we revisit the 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 (Sohl-Dickstein et al., 2015; Ho et al., 2020). They have recently achieved state-of-the-art performance (Dhariwal & Nichol, 2021) on several text-based content creation and editing tasks (et al, 2022; Saharia et al., 2022; Ho et al., 2022; Ruiz et al., 2022; Poole et al., 2022). Diffusion models are trained via a variational objective, which maximizes an evidence lower bound (ELBO) of the data log-likelihood.

Conditional generative models like diffusion models can be easily converted into classifiers (Ng & Jordan, 2001). Given an input $\mathbf{x}$ and a set of classes $\mathbf{c}$ to choose from, we use the model to compute class-conditional likelihoods $p_\theta(\mathbf{x} \mid \mathbf{c})$. With an appropriate prior $p(\mathbf{c})$ and Bayes' theorem, we can predict class probabilities $p(\mathbf{c} \mid \mathbf{x})$. We propose to do this with conditional diffusion models that use an auxiliary input, like a class index for class-conditional models or prompt for text-to-image models, by leveraging the ELBO as an approximate class-conditional log-likelihood $\log p(\mathbf{x} \mid \mathbf{c})$. We call 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 choosing how to perform a Monte Carlo estimate of the ELBO, reducing variance in the estimated probabilities, and speeding up classification inference.

We highlight the surprising effectiveness of our proposed Diffusion Classifier approach on zero-shot and supervised classification tasks by comparing against multiple baselines on ten different 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 (Table 3) highlight that *our generative approach is catching up to SOTA*

[1]Carnegie Mellon University. Correspondence to: Alexander C. Li <alexanderli@cmu.edu>.

Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

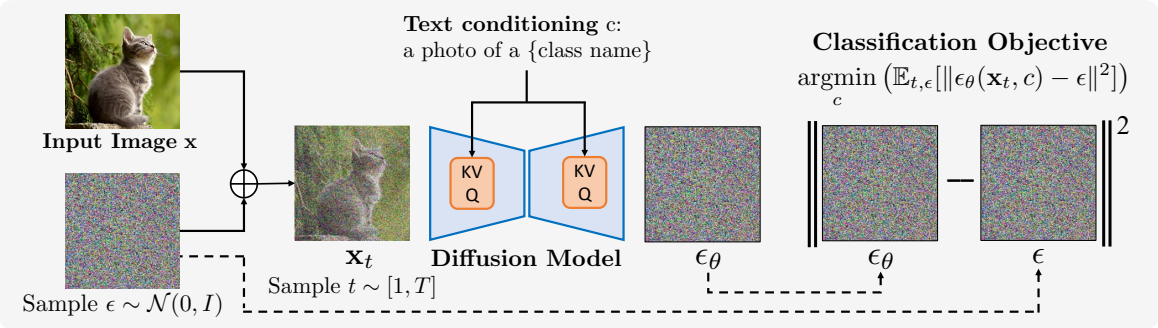

Figure 1. **Overview of our Diffusion Classifier approach:** Given an input image $\mathbf{x}$ and a set of possible conditioning inputs (e.g., text for Stable Diffusion or class index for DiT), we use a diffusion model to choose the one that best fits this image. Diffusion Classifier is theoretically 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.*

discriminative classifiers on ImageNet, both in- and out-of-distribution.

2. Method: Classification via Diffusion Models

2.1. Diffusion Model Preliminaries

Diffusion probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020) 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:

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

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, previous works (Ho et al., 2020) find that setting $w_t = 1$ improves sample quality metrics. 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 ELBO:

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

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

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

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

$$\frac{1}{N} \sum_{i=1}^{N} \left\| \epsilon_i - \epsilon_\theta(\sqrt{\bar{\alpha}_{t_i}}\mathbf{x} + \sqrt{1 - \bar{\alpha}_{t_i}}\epsilon_i, \mathbf{c}_j) \right\|^2 \quad (7)$$

By plugging Eq. 7 into Eq. 6, 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 (Rombach et al., 2022), to extract a standard classifier from a class-conditional model like DiT (Peebles & Xie, 2022), and so on. We show an overview of our method in Fig. 1.

2.3. 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 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. 6 as:

$$\frac{1}{\sum_j \exp\left\{\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_i)\|^2 - \|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_j)\|^2]\right\}} \tag{8}$$

Eq. 8 shows 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. We show our overall algorithm in Alg. 1 and additional practical details in Appendix B.

3. Experimental Details

3.1. Zero-shot Classification

Diffusion Classifier Setup: We build Diffusion Classifier on top of Stable Diffusion 2.1 (Rombach et al., 2022), a text-to-image latent diffusion model trained on a filtered subset of LAION-5B (Schuhmann et al., 2022).

Baselines: We provide results using two strong discriminative zero-shot models: (a) CLIP ResNet-50 (Radford et al., 2021) and (b) OpenCLIP ViT-H/14 (Cherti et al., 2022). We provide these for reference only, as these models are trained on different datasets with very different architectures from ours and thus 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** trains a ResNet-50 classifier on synthetic data generated using Stable Diffusion (with class-names as prompts), (d) **Real-Labeled-SD** trains a ResNet-50 classifier on top of Stable Diffusion features (mid-layer U-Net features at a resolution $[8 \times 8 \times 1024]$ at timestep $t = 100$) using ground-truth labels. This baseline is not zero-shot, as it requires a *labeled dataset* of real-world images and class-names. Details are in Appendix G.3.

3.2. Supervised Classification

Diffusion Classifier Setup: We repurpose Diffusion Transformer (DiT) (Peebles & Xie, 2022), a class-conditional diffusion model trained solely on ImageNet.

Baselines: We compare against discriminative models trained from scratch on ImageNet: ResNet-18, ResNet-34, ResNet-50, and ResNet-101 (He et al., 2016), as well as ViT-L/32, ViT-L/16, and ViT-B/16 (Dosovitskiy et al., 2020).

4. Experimental Results

4.1. 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 *supervised* classifier trained on a *labeled training set*. 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's training data was 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.

4.2. 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 models like CLIP (Radford et al., 2021) on Winoground (Thrush et al., 2022), a popular benchmark for evaluating the reasoning abilities of vision-language models. Each example in Winoground consists of 2 (image,

	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**	**87.1**	24.3	**86.2**	59.4	**95.3**	**58.9**	**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.

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 whenever object swaps are involved (Both also swaps the object). However, performance on Relation still remains roughly at random chance for all three methods.

caption) pairs. Notably, both captions within an example contain the same set of words, just in a different order. Multimodal models are scored on Winoground by their ability to match captions C_i to their corresponding images I_i. Models can only do well if they understand compositional structure within each modality. Each example is tagged by the type of linguistic swap (object, relation, and both) between the two captions. Fig. 6 shows examples of each swap type.

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).* Since Stable Diffusion uses the same text encoder as OpenCLIP ViT-H/14, Diffusion Classifier's compositional reasoning ability comes from better cross-modal binding of concepts to images. Figure 7 visualizes examples of some successes and failures.

4.3. Supervised Classification Results

We compare Diffusion Classifier, leveraging the ImageNet-trained DiT model (Peebles & Xie, 2022), to variants of ViTs (Dosovitskiy et al., 2020) and ResNets (He et al., 2016) trained on ImageNet. Table 3 shows 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 (Dosovitskiy et al., 2020). This is surprising since DiT was trained with *only random horizontal flips*, unlike typical classifiers that use RandomResizedCrop, Mixup (Zhang et al., 2017), RandAugment (Cubuk et al., 2020), and other tricks.

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.

5. Conclusion

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. 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 results are an encouraging step in this direction.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

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

Baranchuk, D., Voynov, A., Rubachev, I., Khrulkov, V., and Babenko, A. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=SlxSY2UZQT.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Burgert, R., Ranasinghe, K., Li, X., and Ryoo, M. S. Peekaboo: Text to image diffusion models are zero-shot segmentors. *arXiv preprint arXiv:2211.13224*, 2022.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., and Jitsev, J. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022.

Croce, D., Castellucci, G., and Basili, R. 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.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A., Caron, M., Geirhos, R., Alabdulmohsin, I., et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *CoRR*, abs/1605.08803, 2016. URL http://dblp.uni-trier.de/db/journals/corr/corr1605.html#DinhSB16.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *preprint arXiv:2010.11929*, 2020.

Du, Y. and Mordatch, I. Implicit generation and generalization in energy-based models. *ArXiv*, abs/1903.08689, 2019.

et al, A. R. Hierarchical text-conditional image generation with clip latents, 2022.

Gadre, S. Y., Wortsman, M., Ilharco, G., Schmidt, L., and Song, S. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*, 2022.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

Hinton, G. E. To recognize shapes, first learn to generate images. *Progress in brain research*, 2007.

Hinton, G. E., Osindero, S., and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Comput.*, 2006.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bklr3j0cKX.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., and Salimans, T. Imagen video: High definition video generation with diffusion models, 2022.

Huang, Y., Gornet, J., Dai, S., Yu, Z., Nguyen, T., Tsao, D., and Anandkumar, A. Neural networks with recurrent generative feedback. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.

Ilharco, G., Wortsman, M., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., et al. Openclip. *Zenodo*, 4:5, 2021.

Kim, G., Kwon, T., and Ye, J. C. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2426–2435, June 2022.

Kingma, D. and Welling, M. Auto-encoding variational bayes. *ICLR*, 12 2013.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Li, D., Yang, J., Kreis, K., Torralba, A., and Fidler, S. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022a.

Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965–10975, 2022b.

Lin, C.-H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.-Y., and Lin, T.-Y. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022.

Liu, H. and Abbeel, P. Hybrid discriminative-generative training via contrastive learning. *ArXiv*, abs/2007.09070, 2020.

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Ng, A. and Jordan, M. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14, 2001.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. *CoRR*, abs/2112.10741, 2021. URL https://arxiv.org/abs/2112.10741.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Ranzato, M., Susskind, J., Mnih, V., and Hinton, G. On deep generative models with applications to recognition. In *CVPR 2011*, 2011.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Gontijo-Lopes, R., Ayan, B. K., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=08Yk-n5l2Al.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., Parikh, D., Gupta, S., and Taigman, Y. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nJfylDvgzlq.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

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

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

Thrush, T., Jiang, R., Bartolo, M., Singh, A., Williams, A., Kiela, D., and Ross, C. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5238–5248, 2022.

Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

Villegas, R., Babaeizadeh, M., Kindermans, P.-J., Moraldo, H., Zhang, H., Saffar, M. T., Castro, S., Kunze, J., and Erhan, D. Phenaki: Variable length video generation from open domain textual description. *ArXiv*, abs/2210.02399, 2022.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

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

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J.-F., Barriuso, A., Torralba, A., and Fidler, S. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021.

Zimmermann, R. S., Schott, L., Song, Y., Dunn, B. A., and Klindt, D. A. Score-based generative classifiers. *arXiv preprint arXiv:2110.00473*, 2021.

Appendix

A. Extended 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 (Ng & Jordan, 2001), VAEs (Kingma & Welling, 2013), GANs (Goodfellow et al., 2014), EBMs (Du & Mordatch, 2019; LeCun et al., 2006), and diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) 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 (Hinton, 2007; Ng & Jordan, 2001; Ranzato et al., 2011). These works train deep belief networks (Hinton et al., 2006) 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 (He et al., 2021; Hjelm et al., 2019; Croce et al., 2020; Brown et al., 2020; Devlin et al., 2019) and segmentation (Li et al., 2021; Zhang et al., 2021; Chen et al., 2016; Baranchuk et al., 2022; Burgert et al., 2022). Moreover, such models (Grathwohl et al., 2020; Liu & Abbeel, 2020; Huang et al., 2020) 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 (Ho et al., 2020; Sohl-Dickstein et al., 2015) have recently gained significant attention from the research community due to their ability to generate high-fidelity and diverse content like images (Saharia et al., 2022; Nichol et al., 2021; et al, 2022), videos (Singer et al., 2023; Ho et al., 2022; Villegas et al., 2022), 3D (Poole et al., 2022; Lin et al., 2022), and audio (Kong et al., 2021; Liu et al., 2023) from various input modalities like text. Diffusion models are also closely tied to EBMs (LeCun et al., 2006; Du & Mordatch, 2019), denoising score matching (Song & Ermon, 2019; Vincent et al., 2008), and stochastic differential equations (Song et al., 2020; Zimmermann et al., 2021). 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 (Dhariwal & Nichol, 2021) 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 (Radford et al., 2019) 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 (Gadre et al., 2022; Li et al., 2022a;b; Alayrac et al., 2022). These methods have been shown to learn robust representations that generalize to various distribution shifts (Ilharco et al., 2021; Dehghani et al., 2023; Taori et al., 2020). 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.

B. Practical Considerations for Diffusion Classifier

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 2.3. In this section, we present further insights and optimizations that reduce our method's runtime.

B.1. Importance of matched ϵ and t

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.*

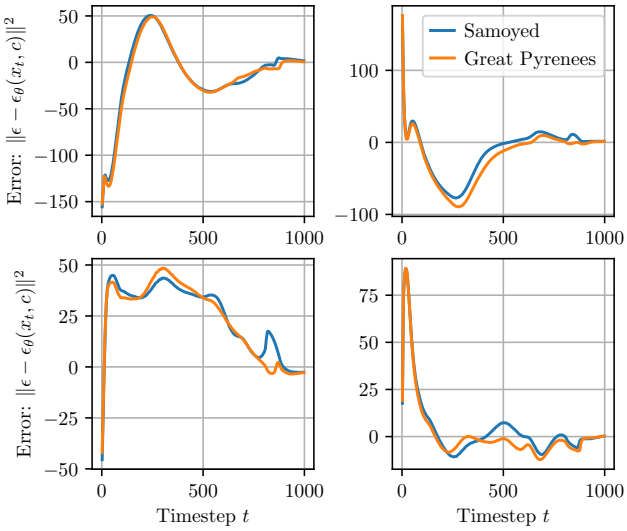

Figure 2. We show the ϵ-prediction error for a fixed image of a Great Pyrenees dog and two prompts. Each subplot corresponds to a single ϵ, with the error evaluated for every $1 \leq t \leq 1000$. Errors are normalized to be zero-mean at each timestep across the 4 plots, and lower is better. Variance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

B.2. 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?

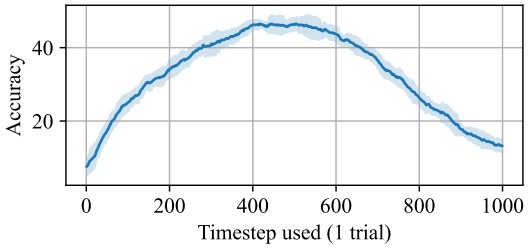

Figure 3. **Pets accuracy, evaluating only a single timestep per class**. Small t corresponds to less noise added, and large t corresponds to significant noise. Accuracy is highest when an intermediate amount of noise is added ($t = 500$).

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.

B.3. 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 (Auer, 2002) 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

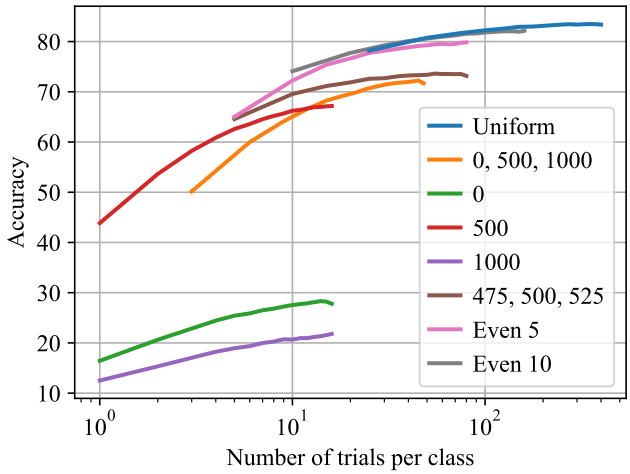

Figure 4. **Zero-shot scaling curves for different timestep sampling strategies**. We evaluate a variety of strategies for choosing the timesteps at which we evaluate the ϵ-prediction error. Each strategy name indicates which timesteps it uses— e.g., "0" only uses the first timestep, "0, 500, 1000" uses 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.

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 G.1.

C. Analyzing Diffusion Classifier for Zero-Shot Classification

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

Experiment Setup: Given an input image, we first perform DDIM inversion (Song et al., 2021; Kim et al., 2022) (with 50 timesteps) using Stable Diffusion 2.1 and different captions as prompts: BLIP (Li et al., 2022a) generated caption, human-refined BLIP generated 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 (Song et al., 2021)). 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. 7), *for the global prediction task of classification, the*

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 $\underset{\mathbf{c}_i \in \mathcal{C}}{\arg\min}\, \texttt{mean}(\texttt{Errors}[\mathbf{c}_i])$

correct class names should provide enough descriptive features for denoising, relative to the incorrect class names.

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 finetuned model achieves a classification accuracy of 85%, significantly higher than the initial zero-shot accuracy of 45%.

D. Additional Details and Figures for Multimodal Reasoning

Figure 6 shows examples of each type of Winoground swap:

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.

E. Effect of Loss Function

	Food101	CIFAR10	FGVC	Oxford Pets	Flowers102	STL10	ImageNet	ObjectNet
Squared ℓ_2	**77.9**	76.3	**24.3**	85.7	56.8	94.2	58.4	**38.3**
ℓ_1	74.3	**87.1**	18.3	**86.2**	**59.4**	**95.3**	58.0	32.2
Huber	**77.9**	76.9	23.7	85.5	57.5	94.2	**58.9**	38.1

Table 4. **Diffusion Classifier performance with different loss functions**.

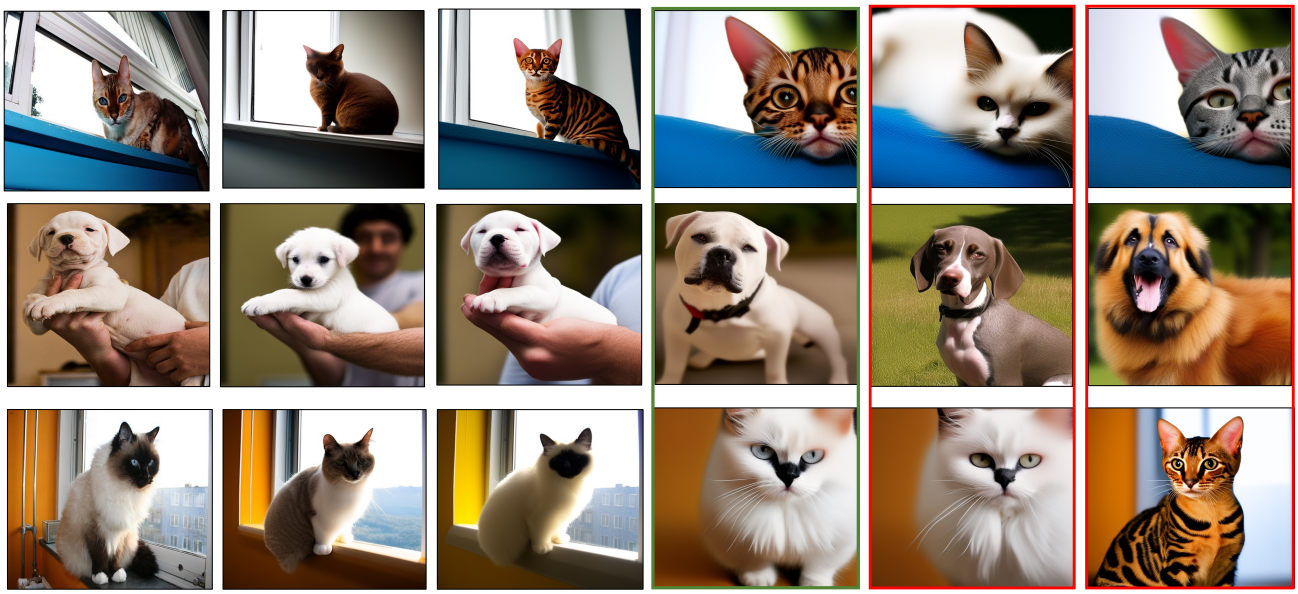

Figure 5. **Analyzing Diffusion Classifier for Zero-Shot Classification:** We analyze the role of different text/captions (BLIP, Human-modified BLIP, correct class-name, incorrect class-name) for zero-shot classification using text-based diffusion models. To do so, we invert the input image using the 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.

F. Techniques that did not help

Diffusion Classifier requires many samples to accurately estimate the ELBO. In addition to using the techniques in Section 2 and B, 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 (Ho & Salimans, 2022) is a technique that 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:

$$\tilde{\epsilon}(\mathbf{x}_t, \mathbf{c}) = (1 + w)\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) - w\epsilon_\theta(\mathbf{x}_t) \tag{9}$$

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 8 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.

Downsampled Latent 256x256 instead of 512x512

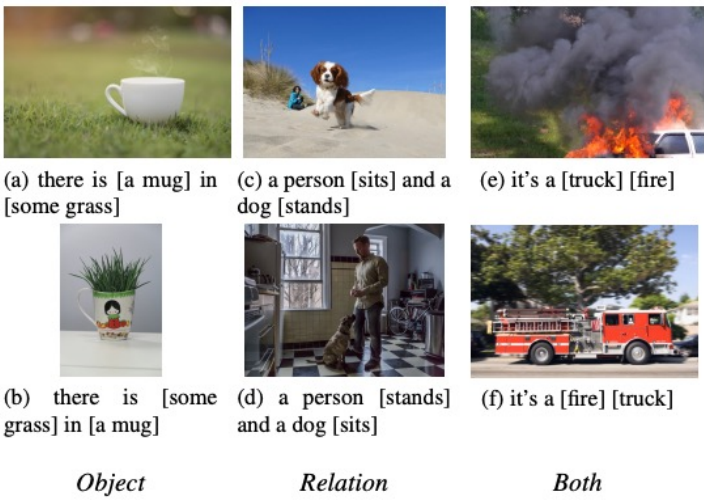

Figure 6. **Example visualizations of Winoground swap types.** Each category corresponds to a different type of linguistic swap in the caption. Object swaps noun phrases, Relation swaps verbs, adjectives, or adverbs, and Both can swap entities of both kinds.

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:

$$\|\epsilon_{[i:-i,i:-i]} - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})_{[i:-i,i:-i]}\|^2 \tag{10}$$

where i is the number of latent "pixels" to remove from each edge. However, Figure 9 shows that any amount of cropping reduces accuracy.

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 (Brock et al., 2018) 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 ($\epsilon \to \frac{\epsilon}{\|\epsilon\|_2}\mathbb{E}_{\epsilon'}[\|\epsilon'\|_2]$) so that there are no outliers. Table 5 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.

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

G. Additional Implementation Details

G.1. Zero-shot classification using Diffusion Classifier

Training Data For our zero-shot Diffusion Classifier, we utilize Stable Diffusion 2.1 (Rombach et al., 2022). 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.

Figure 7. **Results on selected Winoground examples.** The example in the bottom right shows that Diffusion Classifier can better reason about the spatial and the compositional understanding of the underlying images. THe bottom left example shows a challenging example where all the baselines and our approach fail.

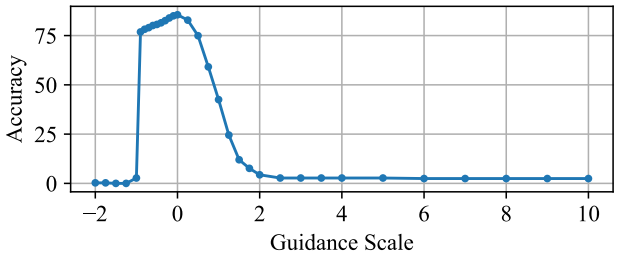

Figure 8. Accuracy plot of classifier-free guidance on Pets.

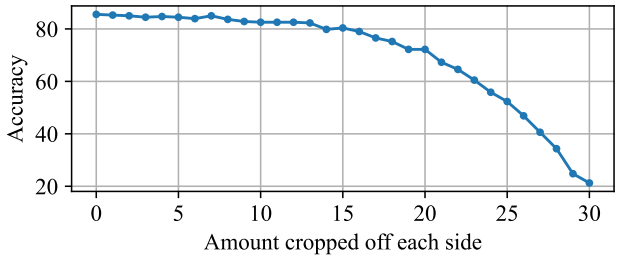

Figure 9. Cropping ϵ and $\epsilon_\theta(\mathbf{x}_t, \mathbf{c})$ reduces accuracy on Pets.

LAION 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 (Dao et al., 2022) 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 6 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.

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 6. Evaluation strategy for each zero-shot dataset.

G.2. ImageNet classification using Diffusion Classifier

For this task, we use the recent Diffusion Transformer (DiT) (Peebles & Xie, 2022) as the backbone of our Diffusion Classifier. DiT 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.

G.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 (Radford et al., 2021) 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 (Baranchuk et al., 2022), 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 7 defines these modifications. We set batch size $= 16$, learning rate

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

$= 1e-4$, 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.