Text-to-Image Diffusion Models are Zero Shot Classifiers

Priyank Jaini*
Google DeepMind

Kevin Clark*
Google DeepMind

Abstract

The excellent generative capabilities of text-to-image diffusion models suggest they learn informative representations of image-text data. However, what knowledge their representations capture is not fully understood, and they have not been thoroughly explored on downstream tasks. We investigate diffusion models by proposing a method for evaluating them as zero-shot classifiers. The key idea is using a diffusion model's ability to denoise a noised image given a text description of a label as a proxy for that label's likelihood. We apply our method to Stable Diffusion and Imagen, using it to probe fine-grained aspects of the models' knowledge and comparing them with CLIP's zero-shot abilities. They perform competitively with CLIP on a wide range of zero-shot image classification datasets. Additionally, they achieve state-of-the-art results on shape/texture bias tests and can successfully perform attribute binding while CLIP cannot. Although generative pre-training is prevalent in NLP, visual foundation models often use other methods such as contrastive learning. Based on our findings, we argue that generative pre-training should be explored as a compelling alternative for vision-language tasks.

Large models pre-trained on internet-scale data can adapt effectively to a variety of downstream tasks. Increasingly, they are being used as zero-shot learners with no task-specific training, such as with CLIP [24] for images and GPT-3 [2] for text. In natural language processing, many successful pre-trained models are generative (i.e., language models). However, generative pre-training is less commonly used for visual tasks. Until recently, the usual practice for vision problems was to pre-train models on labeled datasets such as Imagenet [6], or JFT [31]. Later research in visual and vision-language problems has led to image-text models pre-trained primarily using either contrastive losses [24, 14, 34] or autoencoding tasks [32, 10].

On the other hand, generative text-to-image models based on denoising diffusion probabilistic models [12] such as Imagen [28], Dalle-2 [26], and Stable Diffusion [27] can generate realistic high-resolution images and generalize to diverse text prompts. Their strong performance suggests that they learn effective representations of image-text data. However, their ability to transfer to downstream discriminative tasks and how they compare to other pre-trained models has not been explored thoroughly.

In this paper, we investigate these questions by transferring Imagen and Stable Diffusion (SD) to discriminative tasks. While previous studies have used representations from diffusion models for downstream tasks [1, 3, 35], we instead propose a way of using text-to-image diffusion models directly as zero-shot image classifiers. Our method essentially runs the models as generative classifiers [19], using a re-weighted version of the variational lower bound to score images since diffusion models do not produce exact likelihoods. More specifically, the method repeatedly noises and denoises the input image while conditioning the model on a different text prompt for each possible class. The class whose text prompt results in the best denoising ability is predicted. This procedure is expensive because it requires denoising many times per class (with different noise levels). To make it usable in practice, we present improvements that increase the method's sample efficiency by up to 1000x, such as pruning obviously-incorrect classes early. While still requiring too much compute to be an easily-deployable classifier, our method allows us to quantitatively study fine-grained aspects of a diffusion model's learned knowledge through evaluation on classification tasks (as opposed to qualitatively examining model generations).

We compare Imagen and SD against CLIP[1] [24], a widely used model for zero-shot image-text tasks trained with contrastive learning. A high-level goal of the experiments is to see the strengths and weaknesses of generative and contrastive pre-training for computer vision. First, we demonstrate that diffusion models have strong zero-shot classification accuracies (competitive with CLIP) on several diverse vision datasets. Next, we show both Imagen and SD performs remarkably well on the Cue-Conflict dataset [8], where images have been stylized with textures conflict-

[1] We use ViT-L/14, the largest public CLIP model

ing with their labels. For example, Imagen achieves >50% error reduction over CLIP and even outperforms the much larger ViT-22B [5] model. This finding is particularly interesting because, unlike supervised classifiers, humans are known to be much more reliant on shape than texture when identifying images. Lastly, we study attribute binding using the synthetic data from [18], and find that, unlike CLIP, diffusion models can successfully bind together attributes in some settings.

1. Zero-Shot Classification using Imagen

In this section, we show how to convert the generation process of a text-to-image diffusion model into a zero-shot classifier to facilitate quantitative evaluation on downstream tasks.

Diffusion Generative Classifier: We begin with a dataset, $\{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^n, y^n)\} \subseteq \mathbb{R}^{d_1 \times d_2} \times [\mathsf{y}_K]$ of n images[2] where each image belongs to one of K classes $[\mathsf{y}_K] := \{\mathsf{y}_1, \mathsf{y}_2, \cdots, \mathsf{y}_K\}$. Given an image $\boldsymbol{x}$, our goal is to predict the most probable class assignment

$$\tilde{y} = \arg\max_{\mathsf{y}_k} p(y = \mathsf{y}_k | \boldsymbol{x}) = \arg\max_{\mathsf{y}_k} \ \log p(\boldsymbol{x} | y = \mathsf{y}_k).$$

where we assume a uniform prior $p(y_i = \mathsf{y}_k) = \frac{1}{k}$ that can be dropped from the $\arg\max$.[3] A generative classifier [19] uses a conditional generative model with parameters θ to estimate the likelihood as $p_\theta(\boldsymbol{x}|y = \mathsf{y}_k)$.

Using a text-to-image diffusion model as a generative classifier requires two modifications. First, the models are conditioned on text prompts rather than class labels. Thus we convert each label, y_k, to text using a mapping T with a dataset-specific template (e.g. $\mathsf{y}_k \rightarrow$ A photo of a y_k). Second, diffusion models do not produce exact log-likelihoods (*i.e.* we cannot compute $\log p_\theta(\boldsymbol{x}|y = \mathsf{y}_k)$ directly). Our key idea for a solution is to use the VLB (more specifically $\mathcal{L}_{\mathsf{Diffusion}}$ as Imagen and SD are not trained with the other losses) as a proxy. Thus we have:

$$\begin{aligned}\tilde{y} &= \arg\max_{\mathsf{y}_k} \ \log p_\theta(\boldsymbol{x}|y = \mathsf{y}_k) \approx \arg\min_{\mathsf{y}_k} \ \mathcal{L}_{\mathsf{Diffusion}}(\boldsymbol{x}, \mathsf{y}_k) \\ &= \arg\min_{\mathsf{y}_k \in [\mathsf{y}_K]} \ \mathbb{E}_{\epsilon,t}\Big[\boldsymbol{w}_t \|\boldsymbol{x} - \tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}\big(\boldsymbol{x}_t, \mathrm{T}(\mathsf{y}_k), t\big)\|_2^2\Big] \qquad (1)\end{aligned}$$

Note that for SD, $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}$ are latent representations, with $\boldsymbol{x}$ obtained by encoding the image using a VAE. With Imagen on the other hand, $\boldsymbol{x}$ consists of the raw image pixels.

Estimating the Expectation: We approximate the expectation in Eq.1 using Monte-Carlo estimation. At each step, we sample a $t \sim \mathcal{U}([0, 1])$ and then a $\boldsymbol{x}_t$ according to the forward diffusion process: $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)$. Next, we denoise this noisy image using the model (*i.e.* we use it to predict $\boldsymbol{x}$ from $\boldsymbol{x}_t$), obtaining $\hat{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}\big(\boldsymbol{x}_t, \mathrm{T}(\mathsf{y}_k), t\big)$. We call the squared error of the prediction, $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2$, a *score* for $(\boldsymbol{x}, \mathsf{y}_k)$. We score each class N times, obtaining a $K \times N$ *scores matrix*[4] for the image. Finally, we weight the scores according to the corresponding $\boldsymbol{w}_t$ and take the mean, resulting in an estimate of $\mathcal{L}_{\mathsf{Diffusion}}$ for each class. Our key idea for a solution is to use the diffusion model's variational lower bound (VLB) as a proxy. In particular, we use $\mathcal{L}_{\mathsf{Diffusion}}$, the portion of the VLB corresponding to denoising images, as Imagen is not trained with the other loss terms. The predicted class is:

$$\begin{aligned}\tilde{y} &= \arg\max_{\mathsf{y}_k} \ \log p_\theta(\boldsymbol{x}|y = \mathsf{y}_k) \approx \arg\min_{\mathsf{y}_k} \ \mathcal{L}_{\mathsf{Diffusion}}(\boldsymbol{x}, \mathsf{y}_k) \\ &= \arg\min_{\mathsf{y}_k \in [\mathsf{y}_K]} \ \mathbb{E}_{\epsilon,t}\Big[\boldsymbol{w}_t \|\boldsymbol{x} - \tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}\big(\boldsymbol{x}_t, \boldsymbol{c}_{\boldsymbol{\phi}}(\mathrm{T}(\mathsf{y}_k)), t\big)\|_2^2\Big] \qquad (2)\end{aligned}$$

Estimating the Expectation: We approximate the expectation in eq. (2) using a Monte-Carlo estimatation. At each step, we sample a $t \sim \mathcal{U}([0, 1])$ and then $\boldsymbol{x}_t$ according to the forward diffusion process: $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)$. Next, we denoise this noisy image using Imagen (*i.e.* we use Imagen to predict $\boldsymbol{x}$ from $\boldsymbol{x}_t$), obtaining $\hat{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}\big(\boldsymbol{x}_t, \boldsymbol{c}_{\boldsymbol{\phi}}(\mathrm{T}(\mathsf{y}_k)), t\big)$. We predict the class with the lowest average weighted squared error $\boldsymbol{w}_t \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2$ across steps.

The choice of weighting function, $\boldsymbol{w}_t$, is crucial to the overall performance of the classification algorithm. Here, we chose $\boldsymbol{w}_t := \mathsf{exp}(-7t)$ which we found to work well across many datasets and use it in our experiments. Furthermore, the algorithm presented here is computationally expensive because $\mathcal{L}_{\mathsf{Diffusion}}$ has a fairly high variance. We propose efficiency techniques that reduce the compute cost of computing argmin over classes in the appendix.

2. Empirical Analysis and Results

Here we evaluate Imagen and Stable Diffusion as a zero-shot classifier on a variety of tasks. We compare with CLIP [24], which is widely used as a zero-shot classifier. Our main aim is to study the strengths and weaknesses of image-text representation learning via generative training as in Imagen and SD and contrastive training as used for CLIP. See Appendix **??** for details on the experimental setup.

Image Classification: We first evaluate the performance of at zero-shot classification. on 13 datasets from [24] as reported in Table 1. We use the prompt templates and class labels used by [24], which renames some classes that confuse models (e.g. "crane → "crane bird"" in Imagenet) [21].

[2] For simplicity, we use $\boldsymbol{x}$ in place of $\boldsymbol{x}_0$ to refer to an image.

[3] We can't use a learned prior in the zero-shot setting.

[4] Later we discuss how we can avoid computing the full matrix for efficiency.

We use the first prompt from the list, except for Imagenet, where we use “A bad photo of a *label* ” since this is a good prompt for both Imagen and CLIP [22].

Results are shown in Table 1. The first eight datasets (up through EuroSAT) on the top block of the table are all originally of resolution 64×64 or less. On these datasets, Imagen generally outperforms CLIP and Stable Diffusion on classification accuracy under the same evaluation setting *i.e*., the models are conditioned on the same text prompts, etc. Imagen significantly outperforms CLIP on SVHN and SD on digit recognition datasets like MNIST and SVHN, which requires recognizing text in an image. [29] observe that Imagen is particularly good at generating text, while SD generally performs poorly. This demonstrates that Imagen's areas of strength in generation carry over to downstream tasks and suggests that classification on OCR datasets could be used as a quantitative metric to study a model's text-generation abilities. SD generally performs poorly on the low-resolution datasets, perhaps because it is only trained on high-resolution images.[5] To better understand how much low-resolution is to blame, we evaluated SD on ImageNet after down-sampling the images to 32×32 and 64×64 resolution. SD's accuracy drops from 61.9% to 15.5% and 34.6% respectively. The next five datasets use higher-resolution images. For some of these, taking advantage of CLIP's higher input resolution substantially improves results. SD performs comparably to Imagen on all these datasets (although of course it has an advantage in terms of input resolution). To our knowledge, these results are the first instance of a generative model achieving classification accuracy competitive with strong transformer-based discriminative methods.

Robustness We next study the robustness of text-to-image diffusion models by evaluating them on the cue conflict dataset from [8]. The dataset consists of Imagenet images altered to have a shape-texture conflict. While (for example) changing an image of a cat so it has a texture similar to elephant skin doesn't confuse humans, it could cause a model to classify the image as an elephant. Imagen achieves 84.4 % accuracy compared to 51.56% by CLIP and 79% top-5 accuracy by a supervised trained ResNet50.

Evaluating Attribute Binding on Synthetic Data We next test attribute binding in Imagen and CLIP on synthetic datasets. Attribute binding is a key piece of compositional reasoning: to understand novel combinations of concepts, one must bind the concepts together and treat them as a whole. While other work has examined attribute binding in text-to-image models by qualitatively examining model generations [20, 33], our Imagen classifier offers a way of studying the question quantitatively.

Dataset Construction: We use a setup similar to [18], where images are generated based on the CLEVR [15] dataset. CLEVR images contain various object (cubes, cylinders, and spheres) with various attributes (different sizes, colors, and materials). We use a modified version of the CLEVR rendering script that generates images containing two objects of different shapes. From these images, we construct four binary classification tasks as shown below:

Recognition Results: Results are shown in Table 3. Imagen, SD and CLIP are able to accurately identify shapes and colors that occur in the image. Imagen is slightly worse at shape identification; we find most of these are due to it mixing up “cylinder” and “cube” when the objects are small.

Binding Results: CLIP performs no better than random chance for the attribute binding tasks, showing it is unable to map attributes to objects on this data. In contrast, Imagen and SD perform excellently at the pair tasks, and better than chance on two of the three single tasks. Part of Imagen's advantage might be in its text encoder, the pre-trained T5 [25] model. [29] find that instead using CLIP's text encoder for Imagen decreased its performance on generations involving specific colors or spatial positions. Similarly, [26] find that DALLE-2, which uses a CLIP text encoder, is worse at attribute binding than GLIDE, which uses representations from a jointly-trained transformer processing the text. However, a perhaps more significant advantage of diffusion based models over CLIP is their use of cross attention to allow interaction between textual and visual features.

One mistake we observed frequently in Color$|$Shape is Imagen preferring the color of the larger object in the image; e.g. scoring “A gray sphere” over “A yellow sphere”. We hypothesize that it is helpful for denoising at high noise levels when the text conditioning provides the color for a large region of the image, even when the color is associated with the wrong shape. In the pair task, the full color information for both objects is always provided, which avoids this issue.

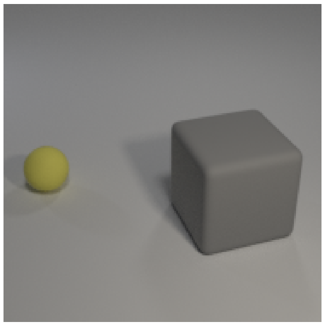
Spatial Positioning Results: Previous work has qualitatively found that large image generation models sometimes struggle with spatial positioning [33]. We find this to be mostly true for Imagen and SD, which perform poorly at associating objects with their position. CLIP performs even worse, performing no better than random chance. We found it prefers the caption with “right” in it over “left” 85% of the time, with it mostly ignoring the rest of the description.

[5] while low-resolution images were incorporated in CLIP's training, doing so with SD would run the risk of the model producing blurry images during generation

Model	CIFAR10	CIFAR100	STL10	MNIST	DTD	Camelyon	SVHN	EuroSAT
Imagen	**96.6**	**84.3**	**99.6**	**79.2**	37.3	**60.3**	**62.7**	**60.3**
Stable Diffusion	72.1	45.3	92.8	19.1	**44.6**	51.3	13.4	12.4
CLIP/ViT-L/14	94.7	68.6	99.6	74.3	36.0	58.0	21.5	58.0

Model	Stanford Cars	Imagenet	Caltech 101	Oxford Pets	Food101
Imagen	**81.0**	62.7	68.9	66.5	68.4
Stable Diffusion	77.8	61.9	73.0	72.5	71.6
CLIP/ViT-L/14	62.8/75.8	63.4/**75.1**	70.2/**84.1**	76.0/**89.9**	83.9/**93.3**

Table 1: **Percent accuracies for zero-shot image classification**. For CLIP where two numbers are reported, the accuracy correspond to two settings: downsizing the images to 64x64 and then resizing the images up to 224x224 (so CLIP does not have an advantage in input resolution over the 64x64 Imagen model) and resizing the images directly to 224x224 (so CLIP has the advantage of higher resolution). Variances in accuracy are <1% across different random seeds. The top set of results are on low-resolution datasets (which is why SD performs poorly).



Control tasks test if the model can identify basic image features by scoring an attribute in the image against one not present. For example:
Shape: A sphere. vs. A cylinder. Color: A gray object. vs. A red object.,

Binding tasks test if the model binds a given attribute to the correct object. For example: Color|Shape: A yellow sphere. vs. A gray sphere.
Color|Position: On the right is a gray object vs. On the right is a yellow object.

Pair binding tasks are easier binding tasks where information about both objects is provided. For example:
Shape,Size: A small sphere and a large cube. vs. A large sphere and a small cube.
Color,Size: A small yellow object and a large gray object. vs. A large yellow object and a small gray object.

Figure 1: Examples of the synthetic-data attribute binding tasks. We explored more sophisticated prompts than in the figure (e.g., "A blender rendering of two objects, one of which is a yellow sphere."), but they didn't substantially change results.

Imagen	**Stable Diffusion**	**CLIP**	**ViT-22B**
84.4	72.5	51.6	68.7

Table 2: Percent shape accuracy for zero-shot classification on the Cue-Conflict Imagenet dataset.

3. Conclusion

While previous fine-grained analysis of diffusion models usually studies generated images qualitatively, our framework provides a new way of quantitatively studying them through evaluation on controlled classification tasks. We find Imagen is an effective and robust image classifier and is capable of performing attribute binding (while CLIP can't).

We hope our findings will inspire future work in using diffusion models for tasks other than generation. One direction is fine-tuning diffusion models on downstream tasks, e.g. evaluating Imagen as a classifier after further supervised training on the dataset. Our main comparison against CLIP is not direct in that the model architectures and parameter counts are different. As models become larger, a key question is how do the scaling laws [11, 16] of contrastive vs generative pre-training compare, which we leave for future work.

Ultimately, our method does not produce a very practical classifier, as it requires substantial compute when scoring many classes. Instead, we see the main value of this work is in revealing more about the abilities of large pretrained diffusion models: our results suggest that generative pre-training may be a useful alternative to contrastive pre-training for text-image self-supervised learning.

References

[1] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Denoising Pretraining for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4175–4186, 2022. 1

Tasks	**Imagen**	**Stable Diffusion**	**CLIP**
Shape (control task)	**85**	**91**	**91**
Color (control task)	**96**	**85**	**94**
Shape,Color / Shape|Color / Color|Shape	**100 / 66 / 73**	**85 / 65 / 59**	54 / 52 / 53
Shape,Size / Shape|Size / Size|Shape	**99** / 48 / 51	**63** / 48 / 52	52 / 51 / 50
Shape,Position / Shape|Position / Position|Shape	**74** / 51 / 52	49 / 50 / 50	50 / 48 / 51
Color,Size / Color|Size / Size|Color	**86** / 54 / 54	**59** / 52 / 48	48 / 51 / 48
Color,Position / Color|Position / Position|Color	**72** / 49 / 49	53 / 51 / 49	49 / 50 / 49
Size,Position / Size|Position / Position|Size	**69** / 50 / 54	54 / 49 / 49	51 / 50 / 48

Table 3: Percent accuracy for models on zero-shot synthetic-data tasks investigating attribute binding. Bold results are significant ($p < 0.01$) according to a two-sided binomial test. CLIP is unable to bind attributes, while Imagen and SD sometimes can.

[2] 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

[3] 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. 1

[4] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983. 9

[5] 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. 2

[6] 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. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. 7

[8] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations*, 2019. 1, 3

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on Machine Learning*, pages 1321–1330. PMLR, 2017. 9

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1

[11] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017. 4

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 7

[13] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded Diffusion Models for High Fidelity Image Generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022. 9

[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1

[15] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 3

[16] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 4

[17] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021. 7

[18] Martha Lewis, Qinan Yu, Jack Merullo, and Ellie Pavlick. Does CLIP Bind Concepts? Probing Compositionality in Large Image Models. *arXiv preprint arXiv:2212.10537*, 2022. 2, 3, 8

[19] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14, 2001. 1, 2

[20] 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. In *International Conference on Machine Learning*, 2021. 3

[21] OpenAI. Prompts for Datasets. *Github*, 2021. 2

[22] OpenAI. Prompt Engineering for Imagenet. *Github*, 2021. 3

[23] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 9

[24] 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. 1, 2

[25] 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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 3, 7

[26] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 3

[27] 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. 1

[28] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 1

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *Advances in Neural Information Processing Systems*, 2022. 3

[30] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 7

[31] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 1

[32] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 1

[33] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 3

[34] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 1

[35] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing Text-to-Image Diffusion Models for Visual Perception. *arXiv preprint arXiv:2303.02153*, 2023. 1

A. Model Details.

Imagen details: Imagen is a text-conditioned diffusion model that comprises of a frozen T5 [25] language encoder that encodes an input prompt into a sequence of embeddings, a 64 × 64 image diffusion model, and two cascaded super-resolution diffusion models that generate 256 × 256 and 1024 × 1024 images. Unlike Stable Diffusion, it operates directly on pixels instead of in a latent space. For our experiments, we use the 64 × 64 model which has 2B parameters and is trained using a batch size of 2048 for 2.5M training steps on a combination of internal datasets, with around 460M image-text pairs, and the publicly available Laion dataset [30], with 400M image-text pairs.

Stable Diffusion details. We use Stable diffusion v1.4 which is a latent text-to-image diffusion model. It uses the pre-trained text encoder from CLIP to encode text and a pre-trained variational autoencoder to map images to a latent space. The model has 890M parameters and takes 512x512-resolution images as input. It was trained on various subsets of Laion-5B, including a portion filtered to only contain aesthetic images, for 1.2M steps using batch size of 2048.

CLIP details: CLIP encodes image features using a ViT-like transformer [7] and uses a causal language model to get the text features. After encoding the image and text features to a latent space with identical dimensions, it evaluates a similarity score between these features. CLIP is pre-trained using contrastive learning. Here, we compare to the largest CLIP model (with a ViT-L/14@224px as the image encoder). The model is smaller than Imagen (400M parameters), but is trained for longer (12.8B images processed vs 5.B). While Imagen was trained primarily as a generative model, CLIP was primarily engineered to be transferred effectively to downstream tasks.

B. Weighting Functions Details.

Learned Weighting Function: While for most experiments we use a heuristic weighting function for $\boldsymbol{w}_t$, we also explored learning an effective weighting function (although this is not truly zero-shot). To do this, we aggregate scores for each image $\boldsymbol{x}$ and class y_k into 20 buckets, with each bucket covering a small slice of timestep values:

$$\boldsymbol{b}_i(\boldsymbol{x}, \mathrm{y}_k) = \mathbb{E}_{\epsilon, t\sim\mathcal{U}[0.05i, 0.05(i+1)]} \|\boldsymbol{x} - \tilde{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \mathrm{T}(\mathrm{y}_k), t)\|_2^2$$

where we estimate the expectation with Monte Carlo sampling (typically around 100 samples). We then learn a 20-feature linear model with parameters $[\boldsymbol{v}_0, ..., \boldsymbol{v}_{19}]$ over these buckets:

$$p_{\boldsymbol{v}}(y = \mathrm{y}_k|\boldsymbol{x}) = \frac{\exp(\sum_{i=0}^{19} -\boldsymbol{v}_i \boldsymbol{b}_i(\boldsymbol{x}, \mathrm{y}_k))}{\sum_{\mathrm{y}_j \in [\mathrm{y}_K]} \exp(\sum_{i=0}^{19} -\boldsymbol{v}_i \boldsymbol{b}_i(\boldsymbol{x}, \mathrm{y}_j))}$$

trained with standard maximum likelihood over the data. At test-time we use the weighting

$$\boldsymbol{w}_t = \boldsymbol{v}_{\lfloor t/0.05 \rfloor}$$

We generally found that (1) learned weighting functions are pretty similar across datasets, and (2) the weighting functions are transferable: the $\boldsymbol{v}$s learned on one dataset get good accuracy when evaluated on other ones. On average, learned weights produced around 1% higher accuracy on zero-shot classification tasks, but we omitted the results from the main paper because using learned weights is not truly zero-shot.

Comparison of Weighting Functions. We compare the learned weighting functions with several heuristic functions on the Caltech101 dataset. We chose Caltech101 because it is high-resolution (SD performs poorly on low-resolution datasets), contains a diversity of image classes, was not used when we developed the heuristic weighting function, and only has 100 classes, so it is much faster to evaluate models on than ImageNet. We compare the following functions:

- **VDM**: $\boldsymbol{w}_t = \mathrm{SNR}'(t)$, the derivative of the signal to noise ratio with respect to t. This weighting scheme from Variational Diffusion Models [17] directly trains the model on a lower bound of the likelihood.

- **Simple**: $\boldsymbol{w}_t = \mathrm{SNR}(t)$. This "simple" loss from [12] results in a model that produces better images according to human judgements and FID scores, even though it results in worse likelihoods.

- **Heuristic**: $\boldsymbol{w}_t = \exp(-6t)$. Our hand-engineered weighting function; we found this by searching for a simple weighting function that works well on CIFAR-100, although we found empirically it generalizes very well to other datasets.

- **Learned**: Learning an effective weighting function on a held-out set of examples as described above.

Results are shown in Table 4. The heuristic weighting function outperforms Simple and VDM for both models. Interestingly, SD appears to be more robust to the choice of weighting function than Imagen. Mechanistically, the reason is that "Simple" and "VDM" weighting put more weight on earlier timesteps than "Heuristic" and Imagen tends to be an inaccurate classifier at very small noise levels. We intuitively believe this is a consequence of pixel vs latent diffusion. The learned weighting only does slightly better than heuristic weighting despite not being truly zero-shot. We found similar results to hold on other datasets.

Weighting	Imagen	Stable Diffusion
VDM	62.0	71.9
Simple	56.1	71.4
Heuristic	68.9	73.0
Learned	70.2	73.1

Table 4: Percent accuracy for models on Caltech101 with different weighting schemes

C. Variances in Classification Accuracies.

Due to our reduced-size evaluation sets, variances in accuracy on zero-shot classification tasks across different random splits are roughly $\pm 0.4\%$ for CLIP, $\pm 0.7\%$ for Imagen, and $\pm 0.6\%$ for Stable Diffusion. The diffusion models have higher variance due to the inherent randomness in noising images (while CLIP is deterministic). Overall, we are not interested in small accuracy differences anyway, as the comparison between models is non-direct in various ways; instead we are trying go get a broad understanding of the models' abilities.

D. Details on Attribute Binding Tasks and Prompts

We use the relational dataset from [18] for the attribute binding experiments. Each image consists of two objects of different shapes and colors; for tasks involving size we filter out examples where both objects are the same size. Each image contains two objects with different attributes $\mathsf{shape} \in \{\text{cube}, \text{sphere}, \text{cylinder}\}$, $\mathsf{color} \in \{\text{blue}, \text{cyan}, \text{blue}, \text{brown}, \text{gray}, \text{green}, \text{purple}, \text{red}, \text{yellow}\}$, $\mathsf{size} \in \{\text{small}, \text{large}\}$, and $\mathsf{position} \in \{\text{left}, \text{right}\}$.

Given a task (e.g. Shape|Size), we construct a task-specific description for an object as follows:

"On the $\{\mathsf{position}\}$ *is a "* if Position tasks else *"A "*+
"size " if Size task else ""+
"color " if Color task else ""+
"shape." if Shape task else *"object."*

For recognition and binding tasks, we randomly select one of the two objects in the image to be the positive example and then use its description as the positive prompt. For pair tasks, we join the descriptions for both objects together with "and" (removing the period from the first description and lowercasing the second one) for the positive prompt.

To construct a negative example for recognition tasks, we replace the positive attribute with a random attribute not in the image. For binding tasks, we replace one of positive description's attributes with the other object's attribute (e.g., for Shape|Color, we replace shape).

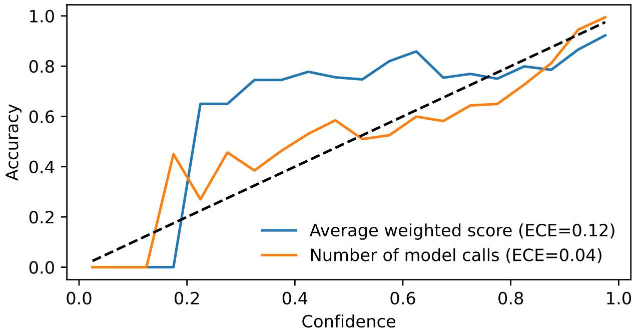


Figure 2: Model reliability diagram comparing confidence measures of Imagen on CIFAR-100. The number of model calls used in Algorithm **??** produces better-calibrated confidences than using the actual scores for different classes.

For pair tasks, there is a choice in how the two objects are ordered (e.g. "On the left is a cube and on the right is a sphere" vs "On the right is a sphere and on the left is a cube". We follow the preference of stating the leftmost position/shape/color/size first in that order. For example, this means we will always start with "On the left..." rather than "On the right...". Similarly, the negative example for Color,Size in Figure 1 is "A large yellow object and small gray object" rather than "A small gray object and a large yellow object" because we prefer to first put the leftmost color over the leftmost size.

We experimented with a variety of other prompts, but found none to work substantially better than these simple ones.

E. Calibration

It is desirable for classifiers, especially when used in the zero-shot setting with possibly out-of-domain examples, to be well calibrated. In other words, if a classifier predicts a label $\tilde{y}_i$ with probability p, the true label should be $\tilde{y}_i$ roughly $100 \cdot p\%$ of the time. However, the diffusion model classifier does not directly produce probabilities for classes. While $p(y_i = \mathrm{y}_k | \boldsymbol{x}_i)$ should roughly be proportional to the expectation in Equation (2) when exponentiated (i.e. we can apply a softmax to the average weighted scores to get probabilities), in practice our estimates of the expectations are very noisy and do not provide well-calibrated scores.

We propose a simple alternative that takes advantage of early pruning: we use the total number of diffusion model calls used for the image as a calibration measure. The intuition is that a harder example will require more scores to determine the $\arg\min$ class with good statistical significance.

More details on the two calibration methods are below:

Temperature-scaled raw scores. We use $s_{\mathrm{y}_k}(\boldsymbol{x})$ to denote the weighted average squared error for class y_k on image $\boldsymbol{x}$, i.e., the Monte-Carlo estimate for the re-weighted VLB in equation 2. We turn these scores into an estimated probability by applying a softmax with temperature:

$$p_\theta(y = \mathrm{y}_k|\boldsymbol{x}) = \frac{\exp(-s_{\mathrm{y}_k}(\boldsymbol{x})/\tau)}{\sum_{\mathrm{y}_j \in [\mathrm{y}_K]} \exp\left(-s_{\mathrm{y}_j}(\boldsymbol{x})/\tau\right)}$$

Note that this approach requires good score estimates for each class, so it is not compatible with the class pruning method presented in Section **??**.

Platt-scaled number of scores. Our other confidence method relies on the total number of scores needed to eliminate all other classes as candidates. Let $\tilde{y}(\boldsymbol{x})$ denote the predicted class for example $\boldsymbol{x}$ and $n(\boldsymbol{x})$ be the total number of calls to $\tilde{\boldsymbol{x}}_\theta$ used to obtain the prediction when running Algorithm **??**. Then we estimate

$$p_\theta(y = \tilde{y}(\boldsymbol{x})|\boldsymbol{x}) = \text{sigmoid}(-n(\boldsymbol{x})/\tau + \beta)$$

We learn τ (and β for Platt scaling) on a small held-out set of examples.

We show reliability diagrams [4] and report Expected Calibration Error [9] (ECE) for the methods in Figure 2. Using a small held-out set of examples, we apply temperature scaling [9] for the score-based confidences and Platt scaling [23] for the number-of-scores confidences, (see Appendix E for details). Number of scores is fairly well-calibrated, showing it is possible to obtain reasonable confidences from diffusion model classifiers despite them not providing a probability distribution over classes.

F. Imagen's Super-resolution Models

Imagen is a cascaded diffusion model [13] consisting of a 64 × 64 low-resolution model and two super-resolution models, one that upsample the image from 64 × 64 to 256 × 256 and one that upsamples from 256 × 256 to 1024 × 1024. However, we found only the 64 × 64 model to work well as a zero-shot classifier. The super-resolution models condition on a low-resolution input image, which means they denoise effectively regardless of the input prompt and thus aren't as sensitive to the class label. For example, unlike with Figure **??**, high-resolution denoising with different text prompts produces images imperceptibly different to the human eye because they all agree with the same low resolution image. Imagen's super-resolution models are trained with varying amount of Gaussian noise added to the low-resolution input image (separate from the noise added to the high-resolution image being denoised). We were able to alleviate the above issue somewhat by using a large amount of such noise, but ultimately did not achieve very strong results with the high-resolution models. For example, the 64 × 64 to 256 × 256 model achieves an accuracy of 16.1% on ImageNet.

We further experimented with combining the low-resolution model's scores with the 64 × 64 to 256 × 256 model's. To do this, we used the learned weighting scheme detailed in Appendix B, but with learning 40 weights: 20 for the low resolution model and 20 for the super-resolution model. However, we found the learned weighting scheme put almost no weight on the super-resolution model's scores, and did not perform significantly better than the low-resolution model did on its own.