Diffusion Classifiers Understand Compositionality, but Conditions Apply

Yujin Jeong[1*] Arnas Uselis[2*] Seong Joon Oh[2] Anna Rohrbach[1]

[1]TU Darmstadt & hessian.AI [2]Tübingen AI Center & University of Tübingen

Abstract

Zero-shot diffusion classifiers have emerged to repurpose diffusion models for discriminative tasks, but these studies are limited and inconclusive. Here, we evaluate diffusion classifiers on a wide range of compositional tasks. We further introduce SELF-BENCH*, a diagnostic benchmark of diffusion-generated images, and reveal the importance of image domain for classification performance. Finally, we explore the importance of low-shot timestep weighting and uncover a relationship between domain gap and timestep sensitivity. Our study spans three Stable Diffusion models (SD 1.5, 2.0, and 3-m), 10 datasets, and over 30 tasks, showing that diffusion classifiers understand compositionality—under the right conditions. Code and datasets are available at* https://github.com/eugene6923/Diffusion-Classifiers-Compositionality.

1. Introduction

Discriminatively trained models like CLIP [19] often struggle with word order, spatial relationships, and compositional reasoning [7, 25, 27]. Diffusion models, trained with dense pixel-level supervision, may be less prone to such issues, prompting the question: can they outperform CLIP on compositional discrimination tasks? Recent work on diffusion classifiers [2, 5, 13, 14] shows promising results—especially on compositional benchmarks like Winoground [24]—leading us to posit **Hypothesis 1 (H1):** *Diffusion models' discriminative compositional abilities are better than CLIP's.*

However, despite strong compositional generative ability [4, 8], newer diffusion models like Stable Diffusion 3-medium (SD3-m) [3] underperform in discriminative accuracy compared to earlier versions in our analysis, suggesting a disconnect between generation and discrimination. We hypothesize this is due to domain-specific biases and style mismatches between training and evaluation datasets. Thus, **Hypothesis 2 (H2)** posits that *Diffusion models understand (through classification) what they generate*. Further, because diffusion models generate images progressively over timesteps [15], different tasks may benefit from leveraging different noise levels. This motivates **Hypothesis 3 (H3):** *The domain gap can be mitigated by the timestep weighting*. We investigate these hypotheses via (1) an evaluation across ten compositional benchmarks, encompassing 33 sub-tasks within four categories: *Object*, *Attribute*, *Position*, and *Counting*, (2) a diagnostic dataset (SELF-BENCH) to probe domain similarity effects, and (3) a low-shot timestep-weight optimization for a specific task.

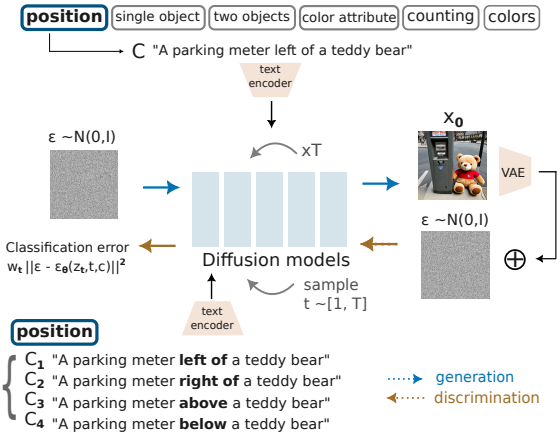

Figure 1. **SELF-BENCH**. (i) Using Geneval's prompts from six categories, generate images from random noise. Filter failed images. (ii) For each image, create discriminative tasks within its category using the prompts from the generation process.

2. Methodology

We begin by discussing the prerequisites for diffusion classifiers and learning an optimal timestep weighting function. Next, we discuss our method for turning Stable Diffusion 3 [3] into a classifier—a first attempt to the best of our knowledge. Last, we outline the collection of SELF-BENCH, our domain-diagnostic dataset.

2.1. Preliminaries

Diffusion Classfiers Given $\mathcal{D}_N = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_N)\}$, a dataset of N images, where each image $\mathbf{x}_i \in \mathbb{R}^{H\times W\times 3}$ is labeled with one of K classes, we aim to learn a classifier that can effectively

*Equal contribution

handle compositional classification tasks. In practice, we work with latent representations $\mathbf{z} \in \mathbb{R}^d$ by encoding the images using a pretrained autoencoder model. Diffusion models [6, 23] are generative models trained to reverse a noise process in latent space $\mathbf{z}$, conditioned on text $\mathbf{c}$, via denoising loss $\mathcal{L}(\mathbf{z}, \mathbf{c})$. This objective approximates the ELBO of $p(\mathbf{z}|y)$, enabling classification by selecting the label y_k that maximizes $\log p(\mathbf{z}|y = y_k)$, estimated via the diffusion loss. Diffusion classifiers [2, 13, 14] use this loss as a classification objective, and we follow this approach for SD1.5 and SD2.0 [21].

Learning the weighting function. In diffusion models, different timesteps capture varying levels of information [15]. This hierarchical information processing is crucial for compositional tasks, where both global structure (e.g., object relationships) and local details (e.g., attributes) matter. Recently, [2] has explored universal timestep weighting in discriminative (yet non-compositional) settings. While we adopt several components from their approach, our setting and low-shot smoothing strategy differ. They rely on computationally expensive, high-variance classification estimates. For instance, they assume 100 trials for a single image-text pair. In contrast, we use fixed timesteps and noise to reduce variance in prediction [14] when computing the reconstruction target. In the low-shot learning setting (5% of the full training set), we use a p-degree polynomial $w_t = \sum_{i=0}^{p} a_i t^i, \quad t \in S_0$ to enforce smoothness in order to prevent overfitting.

2.2. SD3-m as a classifier

SD3-m is a rectified flow model [17] trained with a conditional flow matching (CFM) loss [16], which differs from the standard diffusion objectives used in SD1.5 and SD2.0. As a result, we cannot directly apply the same classifier objective used in earlier versions. By reparameterizing the CFM objective as a noise-prediction loss (see, e.g., [3]), we can obtain

$$\mathcal{L}_{\text{RF}}(\mathbf{x}_0) = \mathbb{E}_{t,\boldsymbol{\epsilon}}[w_t \left\|\epsilon_\Theta(\mathbf{z}_t, t, \mathbf{c}) - \boldsymbol{\epsilon}\right\|^2] \tag{1}$$

Using this formulation, we can use SD3 as classifiers, despite its different underlying architecture. The only difference lies in the weighting function w_t, which for SD3 follows a logit-normal distribution rather than the uniform weighting used in SD1.5/2.0. However, we empirically find that uniform weighting performs better for classification.

2.3. Self-Bench

To isolate image domain effects, we introduce SELF-BENCH, a benchmark for evaluating diffusion classifiers across *in-domain* vs. *out-of-domain* settings. We assume that if a diffusion model can generate certain images, it should also be able to discriminate them. Thus, we define *in-domain* as data self-generated by the model, and *cross-domain* (:=*out-of-domain*) as images generated by another diffusion model.

Table 1. **SELF-BENCH Statistics**: For each task, we show the number of images in `Full` (`F`) and `Correct` (`C`) sets.

Task	Single Obj.		Two Obj.		Colors		Color Attrib.		Position		Counting	
Filter	F	C	F	C	F	C	F	C	F	C	F	C
SD1.5	320	271	396	105	376	219	400	18	400	6	320	98
SD2.0	320	271	396	129	376	263	400	36	400	19	320	111
SD3-m	320	314	396	306	376	314	400	252	400	113	320	230
Total	960	856	1188	540	1128	796	1200	306	1200	138	960	439

Table 2. **Categorization of compositional benchmarks**. For EQBench and Vismin, an official subset is used.

Category	Datasets
Attribute	Aro (Attribute) [27] , SugarCrepe (Attribute) [7] COLA (Multi Object) [20] EQBench (EQ-Kubric Attribute, EQ-SD) [26] MMVP (Color) [25], Vismin (Attribute) awal2024vismin CLEVR (pair binding size & color) [10] CLEVR (recognition color & shape) [10] CLEVR (binding color shape & shape color) [10] Ours - SELF-BENCH (Colors, Color Attribution)
Object	Winoground (Object) [24] , SugarCrepe (Object) [7] Vismin (Object) [1] Ours - SELF-BENCH (Single Object, Two Objects)
Position (Spatial Relation)	WhatsUp (WhatsUp A & B) [11] WhatsUp (COCO-spatial & GQA-spatial one & two) [11] SPEC (Absolute Spatial, Relative Spatial) [18] EQBench (Location) [26] , Vismin (Relation) [1] CLEVR (spatial) [10] MMVP (Spatial, Orientation, Perspective) [25] Ours - SELF-BENCH (Position)
Counting	SPEC (Count) [18] , EQBench (EQ-Kubric Counting) [26] Vismin (Counting) [1] Ours - SELF-BENCH (Counting)

To build SELF-BENCH (Fig. 1), we use prompts from Geneval [4] to generate images. From all the generated images (`Full`), we manually filter out failures (Tab. 1) and retain only the `Correct` ones. For each image, we create discriminative tasks by pairing it with one matching and multiple non-matching prompts from its category.

3. Experiments

3.1. Implementation Details

Benchmarks. Tab. 2 presents the benchmarks used in our study, categorized into *Attribute*, *Object*, *Position*, *Counting*, including our SELF-BENCH dataset, see Sec. 2.3.

Baselines. For SD1.5 and SD2.0 [21], we use the Euler Discrete scheduler [12] and uniformly sample the timesteps. For SD3-m [3], we use Flow Matching Euler Discrete [3] scheduler for flow matching diffusions. We sample 30 timesteps from each model uniformly, following [5]. We use five different versions of CLIP models: RN50x64, ViT-B/32, ViT/L14 [19], ViT/H14, and ViT/G14 [9].

3.2. Scaling evaluation to ten benchmarks

We hypothesize that diffusion models are generally effective at compositional tasks and should outperform CLIP in

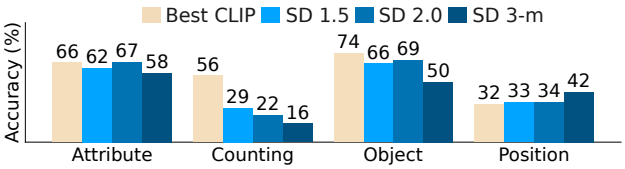

Figure 2. **Compositional generalization**. The bar represents the average classification accuracies across all tasks within each category. Notably, in the Position and Attribute categories, diffusion models outperform CLIP. However, CLIP generally achieves higher overall performance compared to Stable Diffusion models. Additionally, SD3-m does not outperform other Stable Diffusion models in most benchmarks, except in the Position category.

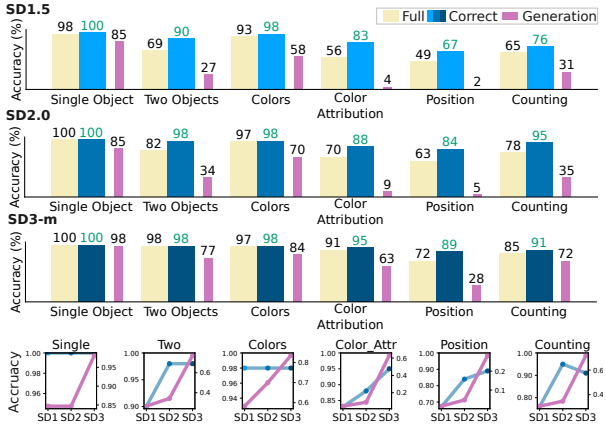

Figure 3. **Self-Bench: In-domain performance**. (Top three plots) Each row represents the classification accuracy of a diffusion classifier from a specific SD model when evaluated on its own generated data. (Bottom) A positive correlation is observed between generative and discriminative performance. Left axis: discrimination; right axis: generation accuracy.

such settings (H1). To test this, we conduct a comprehensive evaluation across ten compositional benchmarks covering various tasks. Each task belongs to one of four categories: *Object*, *Attribute*, *Position*, and *Counting*. In total, we analyze 33 sub-tasks in our main study.

Fig. 2 shows the average performance of diffusion classifiers on the compositional benchmarks. For the *Position* task, SD3-m performs the best compared to other diffusion models and CLIP models. However, in other tasks, CLIP models usually show better performance than diffusion classifiers. Thus, our hypothesis is only partially supported, i.e., trends across datasets or tasks vary. This finding contrasts with previous work [2, 14], which reported more consistent advantages for diffusion classifiers. Among diffusion classifiers, SD3-m is not the best; often, SD1.5 or SD2.0 models show better results.

3.3. Studying domain effects via Self-Bench

To understand why generative models struggle with certain datasets or tasks, we hypothesise that diffusion models understand (through classification) what they generate. (H2). We analyze models' performance on their own generations

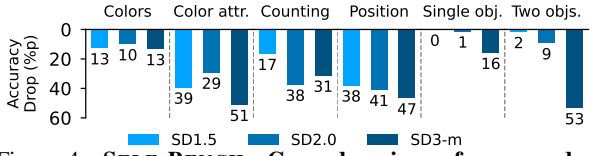

Figure 4. **SELF-BENCH: Cross-domain performance degradation.** The bar represents the average drop rate between *in-domain* and *cross-domain* evaluation, averaged over different cross-domain settings. We observe significant accuracy drops.

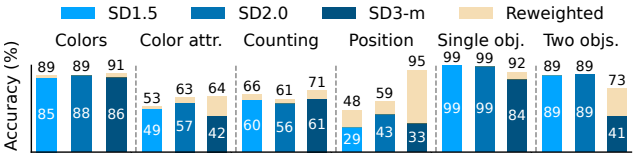

Figure 5. **Timestep reweighting helps address the domain problem in real-world benchmarks.** Low-shot timestep reweighting is effective in real-world benchmarks. Reweighted models consistently outperform the baseline; the gains are most pronounced for the SD3-m model.

("*in-domain*") and on generations of other models ("*cross-domain*") using our SELF-BENCH.[1]

Fig. 3 (top three rows) shows that diffusion classifiers perform well in-domain, but their accuracy drops significantly in cross-domain settings shown in Fig. 4, highlighting the strong influence of domain shifts. The correlation coefficient between generation and *in-domain* discrimination accuracy is 0.77 (Fig. 3 bottom part). This shows that generation accuracy and discrimination accuracy are positively correlated in *in-domain* settings, which is opposite to our observations in Sec. 3.2.

3.4. Studying timestep weighting effects

We hypothesize that the shortcomings of diffusion classifiers arise from how information is processed across timesteps during both generation and classification, leading to a larger domain gap (H3). Here, we examine how low-shot timestep weighting contributes to classification performance.

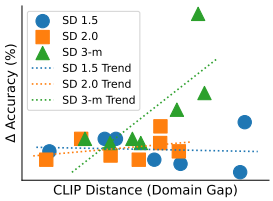

Figure 6. **Timestep Weighting and Domain Gap.** CLIP distances between real-world datasets and SELF-BENCH generations, and corresponding accuracy gains from timestep weighting.

Reweighted SD3 performs the best in real-world benchmarks. We find that timestep reweighting is an effective method to enhance the performance of diffusion classifiers in *cross-domain* scenarios (see Fig. 5, top). Notably, learning timestep weights requires only a small amount of data (5% of the dataset)

[1]We show the average performance of a Single Object and Two Objects, referred to as *Object*, and Colors and Color Attribution, referred to as *Attribute*.

and provides significant advantages that transfer well to low-data settings (see Fig. 5, bottom). Both results show that this approach particularly benefits SD3-m.

Timestep weighting helps mitigate the domain gap. Since timestep weighting significantly boosts SD3-m on real-world tasks, we ask: Does it help mitigate domain gap? To explore this, we compare two image sets per task: (i) the original real-world dataset (used in Fig. 5) and (ii) synthetic images generated using the same prompts. Although both target the same task, they differ in visual domain. Using CLIP (ViT-B/32)[19], we compute the L2 distance between average embeddings from 50 randomly sampled images in each set to estimate domain gap[2]. As shown in Fig. 6, SD3 shows a clear correlation: larger domain gaps correspond to greater gains from timestep weighting. This trend does not hold for SD1 or SD2. We hypothesize that SD1.5 and SD2.0 perform near-optimally with uniform weighting, while SD3-m may suppress certain timesteps due to training on a smaller, more filtered, and human-aligned dataset than LAION-5B [22].

4. Conclusion

This paper investigates diffusion classifiers and their compositional discriminative abilities. In a broad evaluation, we find that they are indeed capable but in limited cases (e.g., Position). We introduce SELF-BENCH, a dataset of diffusion-generated images, revealing domain shift effects, and, lastly, propose a simple task-specific low-shot timestep weighting strategy to mitigate the domain gap between diffusion model's generation and real-world test dataset.

References

[1] Rabiul Awal, Saba Ahmadi, Le Zhang, and Aishwarya Agrawal. Vismin: Visual minimal-change understanding. *arXiv:2407.16772*, 2024. 2

[2] Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero shot classifiers. *NeurIPS*, 2023. 1, 2, 3

[3] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis, 2024. *arxiv:2403.03206*, 2. 1, 2

[4] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *NeurIPS*, 2023. 1, 2

[5] Xuehai He, Weixi Feng, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, William Yang Wang, and Xin Eric Wang. Discffusion: Discriminative diffusion models as few-shot vision and language learners. *arXiv:2305.10722*, 2023. 1, 2

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2

[7] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *NeurIPS*, 2023. 1, 2

[8] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *NeurIPS*, 2023. 1

[9] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 2

[10] 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 *CVPR*, 2017. 2

[11] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's" up" with vision-language models? investigating their struggle with spatial reasoning. *arXiv:2310.19785*, 2023. 2

[12] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022. 2

[13] Benno Krojer, Elinor Poole-Dayan, Vikram Voleti, Christopher Pal, and Siva Reddy. Are diffusion models vision-and-language reasoners? In *NeurIPS*, 2023. 1, 2

[14] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *ICCV*, 2023. 1, 2, 3

[15] Xiao Li, Zekai Zhang, Xiang Li, Siyi Chen, Zhihui Zhu, Peng Wang, and Qing Qu. Understanding diffusion-based representation learning via low-dimensional modeling. In *NeurIPSW*, 2024. 1, 2

[16] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv:2210.02747*, 2022. 2

[17] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 2

[18] Wujian Peng, Sicheng Xie, Zuyao You, Shiyi Lan, and Zuxuan Wu. Synthesize diagnose and optimize: Towards fine-grained vision-language understanding. In *CVPR*, 2024. 2

[19] 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 *ICML*, 2021. 1, 2, 4

[20] Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan Plummer, Ranjay Krishna, and Kate Saenko. Cola: A benchmark for compositional text-to-image retrieval. *NeurIPS*, 2023. 2

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 2022. 4

[23] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2

[24] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, 2022. 1, 2

[25] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, 2024. 1, 2

[26] Tan Wang, Kevin Lin, Linjie Li, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Equivariant similarity for vision-language foundation models. In *ICCV*, 2023. 2

[27] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023. 1, 2

[2] CLIP captures both stylistic and semantic shifts, broadly referred to as "domain"