Identifying Prompted Artist Names from Generated Images

Grace Su

CMU-RI-TR-25-51

June 23, 2025

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Prof. Jun-Yan Zhu, *chair*
Prof. Jean Oh
Sheng-Yu Wang

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

A common and controversial use of text-to-image models is to generate pictures by explicitly naming artists, such as "in the style of Greg Rutkowski". We introduce a benchmark for *prompted-artist recognition*: predicting which artist names were invoked in the prompt from the image alone. The dataset contains 1.95M images covering 110 artists and spans four generalization settings: held-out artists, increasing prompt complexity, multiple-artist prompts, and different text-to-image models. We evaluate feature similarity baselines, contrastive style descriptors, data attribution methods, supervised classifiers, and few-shot prototypical networks. Generalization patterns vary: supervised and few-shot models excel on seen artists and complex prompts, whereas style descriptors transfer better when the artist's style is pronounced; multi-artist prompts remain the most challenging. Our benchmark reveals substantial headroom and provides a public testbed to advance the responsible moderation of text-to-image models. We release the dataset and benchmark to foster further research: https://graceduansu.github.io/IdentifyingPromptedArtists.

Acknowledgments

First, I would like to thank my advisor, Prof. Jun-Yan Zhu, for the support and guidance throughout my two years at CMU. His thoughtful feedback and advice has been instrumental to my growth as a researcher.

I am deeply grateful to my collaborators on this project. I would like to thank Richard Zhang for his insightful feedback and mentorship. I am also thankful to Aaron Hertzmann and Eli Shechtman for the regular project discussions. In addition, I would like to thank Sheng-Yu Wang, who provided essential advice, guidance, and mentorship.

Beyond this project, I am also thankful to Nupur Kumari for mentoring and collaborating with me on my first project at CMU. Thank you to everyone else in the lab—Ruihan Gao, Maxwell Jones, Gaurav Parmar, Kangle Deng, and Ava Pun—for the group discussions, engaging conversations, and supportive research environment. I also appreciate Yifei Liu and Jialu Gao for their collaboration and camaraderie throughout our coursework.

Finally, I would like to thank my family for their constant support and encouragement.

Funding

This work started when Grace Su was an Adobe intern. Grace Su is supported by the NSF Graduate Research Fellowship (Grant No. DGE2140739). This project was partially supported by NSF IIS-2403303, Adobe Research, and the Packard Fellowship.

Contents

1 Introduction **1**

2 Related Work **4**
2.1 Generated Image Detection. 4
2.2 Data Attribution. 4
2.3 Style Similarity. 5
2.4 Generated Image–Text Datasets. 5

3 Prompted Artist Identification Benchmark **6**
3.1 Seen and Held-Out Artists 6
3.2 Varying Prompt Complexity 7
3.3 Different Text-to-Image Models 8
3.4 Multiple Artists per Prompt 9
3.5 Image Similarity Analysis 10

4 Experimental Setup **13**
4.1 Models 13
4.2 Evaluation Framework 14

5 Results **16**
5.1 Single-artist Classification 16
5.2 Multi-artist Classification 18
5.3 Additional Results 21

6 Conclusion **25**

A Supplementary Material **27**
A.1 Dataset Curation Details 27
A.1.1 Artist Curation 27
A.1.2 Prompt Curation and Image Generation 28
A.1.3 Prompting Recent Text-to-Image Models with Artist Names . 30
A.1.4 Dataset Statistics Tables 30
A.2 Training Details 31
A.3 Experiments and Ablations 33
A.3.1 Retrieve from Real Images 33

A.3.2 Retrieve from Prototypes 34
A.3.3 k-NN Evaluation Results 35
A.3.4 Classification Method Experiments 35
A.4 Evaluation Details 37
A.4.1 Bootstrapping Procedure 37
A.4.2 Quantitative Result Tables 39

Bibliography **43**

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

1.1 **Prompted Artist Identification Benchmark**. We introduce the first large-scale benchmark for identifying prompted artist names from generated images. The benchmark covers four axes of generalization that match realistic use cases: **(1) Artists:** we collect artists commonly used in prompts and simulate open-set artist classification by testing on artists not seen during training. **(2) Prompt complexity:** users describe images in many different ways, including short, simple prompts as well as more descriptive, complex prompts. **(3) Text-to-image models:** users can generate images using various text-to-image models, which have different training data and architectures that may affect the generated image's overall style. **(4) Number of artists:** users may include multiple artists in the prompt to mix styles, creating images that are not easily attributable to a single artist. 2

1.2 **Prompted Artist Identification Dataset**. We construct a structured dataset of 1.95M images to benchmark different methods on predicting prompted artist names from generated images. To help disentangle the effect of prompting given an artist name, we query a text-to-image model given the same content prompt (rows), but insert different artist names (columns). Our dataset consists of images generated by SDXL [59], SD1.5 [62], PixArt-Σ [12], and Midjourney [52], with both complex and simple prompts. Each artist's style tends to become less visually prominent when the prompt becomes more complex, especially when the prompt calls for additional styles and adjectives or specifies the content that may not be in the distribution of content in the artist's work (e.g., 2nd row, "a village carved into a red canyon rock wall"). The variance in the visibility of the artist's style across different prompts and models demonstrates the difficulty of the prompted artist identification task. 3

3.1 **Dataset Statistics**. The prompted artist identification benchmark employs a structured dataset of 1.95M images to evaluate vision methods across four axes of generalization. We collect 110 of the most frequently prompted artists, split into 100 seen and 10 held-out artists (1^{st} chart). Next, we collect 1,000 complex prompts and 500 simple prompts in which artist names are inserted (2^{nd} chart). For seen artists, we use a separate set of prompts for testing. For held-out artists, we further divide the test prompts to generate a set of reference images used during inference. Then, we generate single artist-prompted images with SDXL [59], SD1.5 [62], and PixArt-Σ [12], and collect Midjourney images [52] (3^{rd} chart). Finally, we evaluate how well methods generalize to multiple artists in the prompt by generating datasets of SDXL images prompted with 2 artists and 3 artists (4^{th} chart). . . . 7

3.2 **Statistics on number of artists per prompt.** To understand how frequently text-to-image model users prompt for multiple artists, we take a random sample of 10,000 prompts collected from Midjourney users in JourneyDB [78] and visualize the distribution of the number of artists per prompt. While 14.7% of the prompts invoke only one artist name, 4.8% contain 2 artist names, and 2.5% contain 3 artist names. 10

3.3 **Multi-artist prompted images**. In the example images shown, we observe that the effect of adding each artist's name in the prompt diminishes as more artists are added. Each row shows a set of images generated with the same prompt and generation seed, and the number of artists inserted into the prompt increases from left to right. For each prompt, the image changes less with each additional artist, and images generated with complex prompts generally change less than those generated with simple prompts. 12

5.1 **Single-artist prediction results**. We compare the prompted artist classification accuracy of various visual representation methods. We test within our seen artists set (100-way classification, x-axis) and on held-out artists (10-way classification, y-axis). We also test on images generated with different text-to-image models (SDXL, SD1.5, PixArt, and Midjourney), and complex and simple prompts. Although prototypical networks [73], the trained vanilla classifier, and CSD [75] surpass the CLIP [60], DINOv2 [7], and AbC [81] methods, attaining high accuracy across all scenarios remains difficult. We include all numbers as tables in the supplemental material. 17

5.2 **Generalization to unseen generators**. For the best-performing methods, prototypical networks and CSD, we evaluate performance on unseen text-to-image models by progressively increasing the training set of images. Cells below the green diagonals evaluate generalization to unseen domains. For both methods, expanding the training dataset does not enhance performance on unseen text-to-image models; improvements occur only when the training data includes images generated by the specific model being evaluated. 18

5.3 **Multi-artist prediction results**. We evaluate visual representation methods on the multi-artist classification task, where the input image is prompted with multiple artists' names. We test on SDXL-generated images with 2 artists and 3 artists in the prompt, and report ranked mAP@10 on seen artists (x-axis) and held-out artists (y-axis). All methods generally exhibit reduced performance on images generated from complex prompts compared to those generated from simple prompts. As expected, the prototypical network achieves the highest performance across the datasets due to its training on multi-artist prompted images. However, its performance falls short of saturation. We include all numbers as tables in the supplemental material. . . . 19

5.4 **Generalization to increasing numbers of prompted artists**. For prototypical networks, we evaluate performance on unseen numbers of artists in the prompt by progressively increasing the training set of images to include multi-artist prompted images. Cells below the green diagonals evaluate generalization to unseen domains. Increasing the size of the training set does not improve generalization to unseen numbers of artists. While using training images with the same number of artists as the test set enhances classification for seen artists, it does not benefit performance on held-out artists. 20

5.5 **Ablation on number of training prompts**. We plot the prototypical network classification accuracy on seen artists (left) and held-out artists (right) as a function of the number of prompts in the training set. Classification accuracy on seen artists improves steadily with more training prompts, but performance on held-out artists remains unchanged across all dataset sizes. The prototypical network is able to learn a representation of the artist's style that generalizes to unseen prompts with seen artists, but not to prompts with held-out artists. 22

A.1 **Artist statistics.** We plot the frequency of each artist's name in our LAION training set, with artists labeled public domain (blue) or non-public domain (orange), as defined by 95 years post-mortem. Out of our set of 100 seen artists, 45 are non-public domain, and their images make up 38.7% of our real image dataset. 29
A.2 **FLUX and SD3.5 examples.** For FLUX.1-dev [40] and SD3.5 [22], inserting artist names into the prompt frequently has little to no influence on the generated image's style compared to using the same prompt without any artist name. Most of the resulting images are photorealistic, and the generated images do not match the artists' actual styles (bottom row). 31
A.3 **Effect of k on accuracy for k-NN classification**. We perform k-NN evaluation on multiple k values and find that for each retrieval-based method, the performance only changes a small amount as k increases. Thus, the performance ranking of the methods remains unchanged across different k values and the gap between our method and baselines remains the same. 36
A.4 **Bootstrapping procedure convergence**. We plot standard error against the number of bootstrap iterations for the prototypical network's artist classification accuracy on SDXL images and simple prompts. The number of bootstrapping iterations were increased from 50 to 4000 in increments of 50. The standard error converges at around 2000 iterations, where the standard error remains within the range of 90% of the data points. Thus, we use 2000 bootstrap iterations in our main benchmark evaluation. 39

List of Tables

3.1 **Average CLIP image similarities**. (a) For each text-to-image model, we compute the average CLIP similarity between the image generated without an artist name and the image generated with one artist name (first two columns), and the average CLIP similarity between the image generated with an artist name and the real prototype for that artist (last two columns). The artist's influence and generated image's alignment to the artist's real style decreases when the prompt is more complex. Additionally, PixArt images are less aligned to the artist's real style than SDXL and SD1.5 images. (b) Given the same prompt and generation seed, we compute the average CLIP similarity between SDXL images generated with 0, 1, 2, and 3 artist names. The influence of each artist's name in the prompt diminishes as more artist names are added. 12

5.1 **Artist name detection dataset**. To evaluate different methods on the task of detecting whether an image was prompted with an artist name, we construct a dataset of SDXL images prompted with artist names and images prompted with the same content but without the artist names. The training set contains all artist-prompted SDXL images from our benchmark, and the test set is balanced between content-only and artist-prompted images. 22

5.2 **Artist name detection results**. We evaluate the best prompted artist classification methods on the artist name detection task and report accuracy, average precision (AP), and area under the receiver operating characteristic curve (AUCROC). Even though all methods are fine-tuned for the detection task, none achieve perfect accuracy. Performance is consistently higher on simple prompts, and the fully fine-tuned CLIP model under-performs on the test set, indicating susceptibility to overfitting. This is likely because the artist name detection task is inherently harder. 23

5.3 **Accuracy on non-public domain, seen artists**. We evaluate the performance of various methods on the task of detecting the presence of a non-public domain artist name in the prompt. We test on the 100-way seen artists test set, which contains 55 public domain and 45 non-public domain artists. The trained classifier methods, prototypical networks and vanilla classifiers, perform the best, suggesting that training on generated images can improve detection of non-public domain artists, given a closed set of artists. 24

A.1 **Single-artist dataset statistics.** The prompted artist identification dataset includes images generated with SDXL [59], SD1.5 [62], PixArt-Σ [12], and Midjourney [52]. As described in Section 3, we collect 1,000 complex prompts, 500 simple prompts, and use 110 different artist names in the prompts. For each artist and prompt combination, we generate images using 10 different seeds for SDXL complex prompts and 2 different seeds for the other subsets. We split the 110 artists into 100 seen and 10 held-out artists. For seen artists, we use a separate set of prompts for testing. For held-out artists, we additionally split test prompt images for test-time reference. 32

A.2 **Multi-artist dataset statistics.** To evaluate prompted artist identification of images generated with multiple artists in the prompt, we generate datasets of images prompted with 2 artists and 3 artists. The datasets are generated using the same procedure as the single-artist dataset, except we sample 100 artist combinations for each number of artists. 33

A.3 **Real artist image dataset**. We use a subset of LAION-Styles [76] containing 100 seen artists and 10 held-out artists. Then, we benchmark prompted artist identification of various visual representation methods and use this dataset for real image references of each artist's style. . 34

A.4 **Comparison to baselines retrieving from prototypes**. We additionally evaluate average embedding retrieval for baselines CSD [76] and CLIP [60] and compare them to the prototypical network [73]. . 35

A.5 **Classification method experiments**. In the main paper, we benchmark two classification methods: prototypical networks and vanilla classification. Here, we test more classifier method variants and training datasets, reporting accuracy numbers averaged across the SDXL complex and simple prompt evaluation sets. (a) We test **model types**, comparing vanilla classification, supervised contrastive learning, and prototypical networks. As classifiers have a fixed label set, we use nearest neighbors on their feature space when testing on held-out artists. Supervised contrastive learning performs worse than both the prototypical network and vanilla classification, so we do not include it in our benchmark evaluation. (b) We test **training sets** for the prototypical network. Learning from real images is not as effective as learning from generated images. Using both simple and complex prompts outperforms using one or the other, demonstrating they provide complementary information. (c) For the prototypical network, we compare using the **prototype vs. nearest neighbors**. When using the learned prototypical network representation for nearest neighbors, performance slightly degrades. Using the generated images for prototypes, rather than real, also degrades performance, so we use real images as prototypes for our main implementation. 38
A.6 **Classification accuracy on SDXL images**. We compare retrieval-based methods and fine-tuned classifiers on SDXL images from our prompted artist identification dataset. While retrieval-based methods can use either real or generated images as the reference database, we find that retrieval from generated images outperforms retrieval from real images for all methods, across all evaluation sets except for images with complex prompts and held-out artists. Thus, we focus on retrieval from generated images in our main benchmark evaluation. 40
A.7 **Classification accuracy on SD1.5 images**. 40
A.8 **Classification accuracy on PixArt-Σ images**. 41
A.9 **Classification accuracy on Midjourney images**. 41
A.10 **Ranked mAP@10 artist retrieval on SDXL 2-artist prompted images**. 42
A.11 **Ranked mAP@10 artist retrieval on SDXL 3-artist prompted images**. 42

Chapter 1

Introduction

Current text-to-image models allow users to generate images specifying artistic styles, often by direct reference to artists' names, such as 'in the style of a named artist'. Some online artwork sharing platforms prohibit uploading such imagery [1, 24], due to potential harm to artists [3, 26, 51] and the risk of producing derivative works that appear very similar to, or indistinguishable from, the original artist's works. However, without access to the original prompt, it is unclear how to detect such violations.

In this work, we focus on the problem of automatically classifying which artist's name was directly used in a prompt. We refer to this problem as identifying the "prompted artist" from the generated image. To tackle this problem, we introduce a large-scale benchmark of 1.95 million labeled images. Our benchmark is designed to evaluate generalization across four axes that reflect real-world use of text-to-image models for generating images derived from existing artists' styles, as illustrated in Figure 1.1.

(1) Artists. The set of artists that may be included in a prompt is open-ended. Any given model is likely to encounter images generated with artists that were not included in its training set, so we create a set of held-out artists in the benchmark.

(2) Prompt complexity: Users construct prompts in a variety of ways, from short, simple prompts to long, complex prompts that may specify additional styles, reducing the visibility of the artist's style in the resulting image. Thus, we manually design simple prompts (e.g., "picture of the ocean in the style of..."), along with long, complex prompts scraped from databases of real use-cases [78]. We then hold out

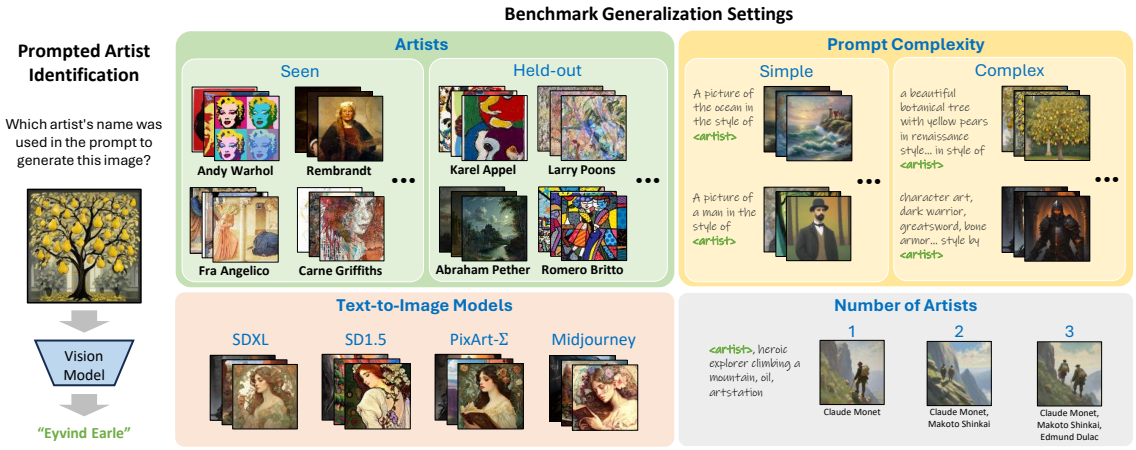

Figure 1.1: **Prompted Artist Identification Benchmark**. We introduce the first large-scale benchmark for identifying prompted artist names from generated images. The benchmark covers four axes of generalization that match realistic use cases: **(1) Artists:** we collect artists commonly used in prompts and simulate open-set artist classification by testing on artists not seen during training. **(2) Prompt complexity:** users describe images in many different ways, including short, simple prompts as well as more descriptive, complex prompts. **(3) Text-to-image models:** users can generate images using various text-to-image models, which have different training data and architectures that may affect the generated image's overall style. **(4) Number of artists:** users may include multiple artists in the prompt to mix styles, creating images that are not easily attributable to a single artist.

a set of test prompts from the training set to evaluate generalization across unseen prompts and content. To reduce the association of a given artist name with specific content, we query the model using the same content prompt, with different artist names inserted into the prompt.

(3) Text-to-image models: Users can choose from a large selection of popular image generators. The complex interaction of the network architecture, learning algorithm, training images, and input prompts leads to varying representations of the same artist's style that may be challenging to identify. We collect training and evaluation set images from commonly-used models: SDXL [59], SD1.5 [62], PixArt-Σ [12], and Midjourney [52].

(4) Number of artists per prompt: Users may include multiple artists in one prompt, creating combined styles that can be challenging to associate with any one artist. To study these cases, we create subsets for images generated with 2 and 3

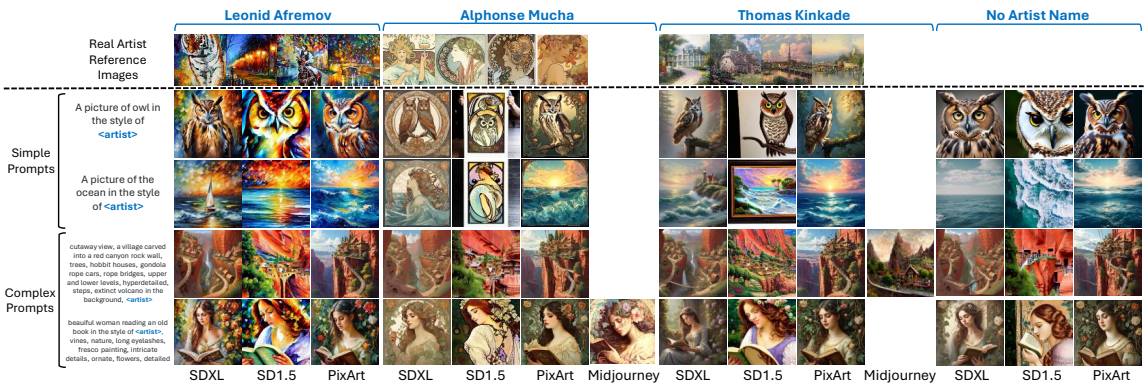

Figure 1.2: **Prompted Artist Identification Dataset**. We construct a structured dataset of 1.95M images to benchmark different methods on predicting prompted artist names from generated images. To help disentangle the effect of prompting given an artist name, we query a text-to-image model given the same content prompt (rows), but insert different artist names (columns). Our dataset consists of images generated by SDXL [59], SD1.5 [62], PixArt-Σ [12], and Midjourney [52], with both complex and simple prompts. Each artist's style tends to become less visually prominent when the prompt becomes more complex, especially when the prompt calls for additional styles and adjectives or specifies the content that may not be in the distribution of content in the artist's work (e.g., 2nd row, "a village carved into a red canyon rock wall"). The variance in the visibility of the artist's style across different prompts and models demonstrates the difficulty of the prompted artist identification task.

artists in our benchmark.

We evaluate a wide range of methods on our benchmark: feature similarity methods, contrastive style descriptors, data attribution methods, supervised classifiers, and few-shot prototypical networks. The degree of generalization varies across different methods. Our benchmark reveals substantial room for improvement across all generalization settings evaluated. Supervised and few-shot models trained on images generated with prompted artists perform better on seen artists and complex prompts. However, style descriptors trained on real artwork generalize better on simple prompts, where the artist's style is more apparent. We find that capturing and recognizing the representation of prompted artists, as learned and expressed by a generative model, is a related, yet distinct problem from style recognition of real artwork. Meanwhile, prompts referencing multiple artists continue to pose the greatest challenge. We release the benchmark and dataset to help advance the responsible moderation of AI-generated content at https://graceduansu.github.io/IdentifyingPromptedArtists.

Chapter 2

Related Work

2.1 Generated Image Detection.

While generated images have become increasingly photorealistic, they continue to leave detectable traces, back with manipulated faces [19, 31, 33, 44, 63, 95], GANs [10, 49, 50, 54, 80, 89, 90], and now with diffusion models [9, 16, 17, 21, 27, 55, 57, 71, 83, 94]. Despite these advances, robustness to in-the-wild perturbations and generalization to future generators are a continuous challenge [9, 80]. Here, we study a distinct aspect: we seek not only to detect if something is generated, but also to study properties that led to its generation, i.e., if an artist's style was targeted for replication.

2.2 Data Attribution.

While most generated images are "novel" and not an exact copy of a training image, all reflect some specific elements of the training set. Data attribution seeks to link a synthesized image to the constituent training images that influenced it [23, 81], assuming the set of training images and training process is known, i.e., a closed world. However, for our artist name classification task, we may not know the training set and learning algorithm. Moreover, existing data attribution algorithms are computationally expensive [15, 23, 25, 29, 36, 37, 82, 93]. In contrast, our benchmark

focuses on finding cases where an artist name's influence is direct and intentional. We show that style descriptors and prototype classifiers outperform data attribution methods on prompted artist identification.

2.3 Style Similarity.

Artistic style classification methods are typically trained on real artwork and extract or learn image features that are similar across images of the same style [32, 48, 65, 66, 68, 77]. Some recent works additionally measure the style similarity between a generated image and the artist's original work to determine whether the generated image replicates the artist's style [8, 38, 47, 53, 76]. However, when images are generated from complex prompts that obscure the artist's style, style similarity methods become less effective than a classifier trained directly on our dataset.

2.4 Generated Image–Text Datasets.

Many large-scale datasets containing paired prompts and generated images have been released, including DiffusionDB [84], JourneyDB [78], and TWIGMA [13], text-to-image model evaluation benchmarks [4, 28, 42, 67, 88], and user preference alignment datasets [14, 35, 79, 86]. Other efforts focus on user-generated art from text-to-image models and provide richer stylistic variety [72, 85, 92]. Yet none of these datasets targets images prompted with explicit artist names. The closest work, by Leotta *et al.* [43], represents an early attempt to tackle the challenging problem of inferring artist names from generated images. They offer a dataset of 8,519 DALL·E 2 [61] images covering five artists. In contrast, we construct a large-scale benchmark of 1.95M images, which spans hundreds of artists, multiple generative models, diverse prompts, and varying numbers of artists, enabling comprehensive evaluation of open-set artist-name recognition.

Chapter 3

Prompted Artist Identification Benchmark

Our goal is to create a benchmark evaluating the generalization of prompted artist identification methods across four relevant axes to cover the various images generated from typical prompts from real-world text-to-image model users. Specifically, the benchmark's dataset includes seen vs. held-out artists (Section 3.1), prompt types of varying complexity (Section 3.2), different text-to-image models (Section 3.3), and different numbers of artists per prompt (Section 3.4). We refer to images generated with prompts invoking one or more artist names as "artist-prompted images."

3.1 Seen and Held-Out Artists

In the real world, users can reference an open-ended and continuously growing set of artists in image generation prompts, increasing the likelihood that a prompted artist identification model will encounter names not seen during training. While the training set may cover the most frequently used artists, the ability to generalize to held-out artists without retraining remains important. To ensure the benchmark includes the most relevant artists for real-world applications, we collect a list of artists most commonly used in text-to-image model prompts, then designate a set of held-out artists to evaluate how well vision models generalize to unseen artists.

Specifically, we manually de-duplicate and filter 110 artists from an initial list

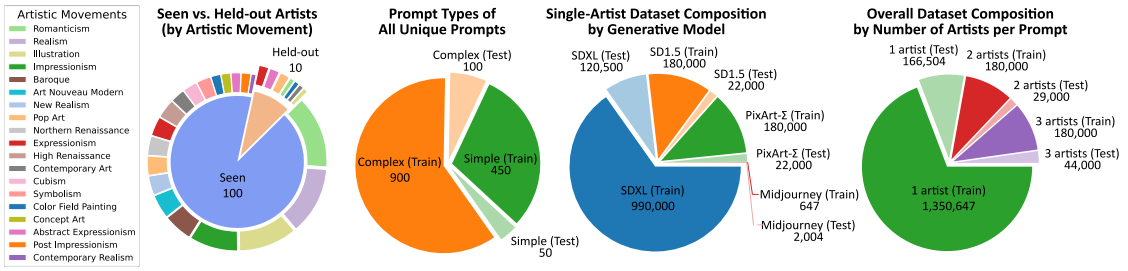

Figure 3.1: **Dataset Statistics**. The prompted artist identification benchmark employs a structured dataset of 1.95M images to evaluate vision methods across four axes of generalization. We collect 110 of the most frequently prompted artists, split into 100 seen and 10 held-out artists (1st chart). Next, we collect 1,000 complex prompts and 500 simple prompts in which artist names are inserted (2nd chart). For seen artists, we use a separate set of prompts for testing. For held-out artists, we further divide the test prompts to generate a set of reference images used during inference. Then, we generate single artist-prompted images with SDXL [59], SD1.5 [62], and PixArt-Σ [12], and collect Midjourney images [52] (3rd chart). Finally, we evaluate how well methods generalize to multiple artists in the prompt by generating datasets of SDXL images prompted with 2 artists and 3 artists (4th chart).

of 400 artists frequently prompted by Stable Diffusion users on the Lexica.art website [69, 76]. For these artists, we then obtain 10k real reference images from LAION-Styles [76], a curated subset of LAION-5B [70] targeted towards artist style attribution. To evaluate how well various methods generalize to different artists, we use 100 artist names for training ("seen") and 10 distinct artists for testing ("held-out") Our real artist image dataset thus includes 9907 images from seen artists and 860 images from held-out artists. We include the artist list, curation details, and artist name frequencies in our LAION subset in the supplement.

3.2 Varying Prompt Complexity

Text-to-image models have the flexibility to generate images with text prompts of varying lengths and complexities. Intuitively, for shorter prompts, the artist's name influences the style of the resulting images more heavily, while longer, more complex prompts may dilute this effect. Hence, we study how classification performance differs in the following two prompt types.

Simple prompts. We generate images with simple prompts in the form of "`A picture of <content> in the style of <artist>`." For diversity, we use 500 different contents sampled from ChatGPT [11]. Figure 1.2 shows examples of the generated images; they typically follow a consistent artist style.

Complex prompts. In a realistic setting, a user often generates images with more complex prompts, such as "`Pisces zodiac sign as a cat alphonse mucha art style realistic 3d render attention to details hyper realistic`." The above prompt is taken from JourneyDB [78], a dataset containing text prompts that real users submitted to Midjourney [52]. We collect 1000 text prompts from JourneyDB, each mentioning one artist's name. Then, we replace the artist's name with one of the 110 artists in our dataset list. For example, the previous text prompt in our dataset would be "`Pisces zodiac sign as a cat <artist> art style realistic 3d render attention to details hyper realistic`." In Figure 1.2, we observe that the artist's style becomes less prominent in the generated image compared to the simple prompts, making classification a much more challenging problem.

Generalize to unseen prompts. To evaluate how different artist classification methods generalize to different prompts, we use 450 simple prompts and 900 complex prompts for training, and hold out 50 simple prompts and 100 complex prompts for testing. We include the list of prompts and processing procedure in the supplement.

3.3 Different Text-to-Image Models

Curating generators and images. Many text-to-image generative models have been developed in recent years with varying degrees of open-source access [30, 42, 45, 46, 91], and more continue to be released as the field progresses. Generated images may come from any of these models, each of which renders prompts and artist names differently, depending on the model's training data and architecture. Thus, our benchmark also evaluates how well prompted artist identification methods generalize to different text-to-image models. To curate a large, structured image dataset labeled with the original generation prompts, we select recent, popular models that are 1) able to generate artistic styles when an artist name is directly prompted for and 2) have open-source weights or have a large published dataset of image-prompt pairs.

We tried generating images prompted with artist names using the recent SD3.5 [22] and Flux [40] open-source models, but found that using artist names often has little to no effect on the generated image style, even when the prompt is simple. This may be because the models' training datasets were filtered to exclude problematic content [2]. While the models may be able to replicate the target style if the user writes prompts describing the style, they do not respond to artist names as well as earlier models like SDXL [59]. We show examples of their generated images in the supplement.

Thus, for each artist and prompt combination, we generate images with 2 seeds using the open-weight models SDXL [59], SD1.5 [62], and PixArt-Σ [12]. For SDXL images generated with complex prompts, we use 10 different generation seeds. To collect images from Midjourney [52], a closed-source model, we filter the JourneyDB dataset [78] for our set of seen and held-out artists. The resulting dataset is summarized by Figure 3.1. Full dataset statistics tables are included in the supplement.

3.4 Multiple Artists per Prompt

Text-to-image model users not only prompt for a single artist's style, but also mix multiple artists' styles by including multiple artist names in the prompt. For instance, in a random sample of 10,000 prompts collected from Midjourney users in JourneyDB [78], we find that 10.8% of the prompts contain multiple artist names, compared to 14.7% containing 1 artist name and 74.5% containing no artist name. The distribution, visualized in Figure 3.2, is long-tailed and heavily skewed, with 4.8% of the prompts containing 2 artist names, 2.5% containing 3 artist names, and 1.3% or less for 4 or more artist names.

Thus, our benchmark primarily contains single-artist prompted images for evaluation. However, we also evaluate how well prompted artist identification methods generalize to multiple artists by generating a dataset of images prompted with 2 artists, and a dataset of images prompted with 3 artists.

Generating images prompted with multiple artists. To curate a dataset of multi-artist prompted images that allow for evaluation of generalization to held-out artists and varying prompt complexity, we slightly modify the procedure described in Section 3.2. We insert multiple artist names into the prompt templates, e.g., "`Pisces`

`zodiac sign as a cat <artist 1> and <artist 2> art style realistic 3d render attention to details hyper realistic`", sample 100 random seen artist combinations and all possible held-out artist combinations for 2 artists and 3 artists, and generate images with SDXL. The multi-artist datasets are summarized by the fourth chart in Figure 3.1.

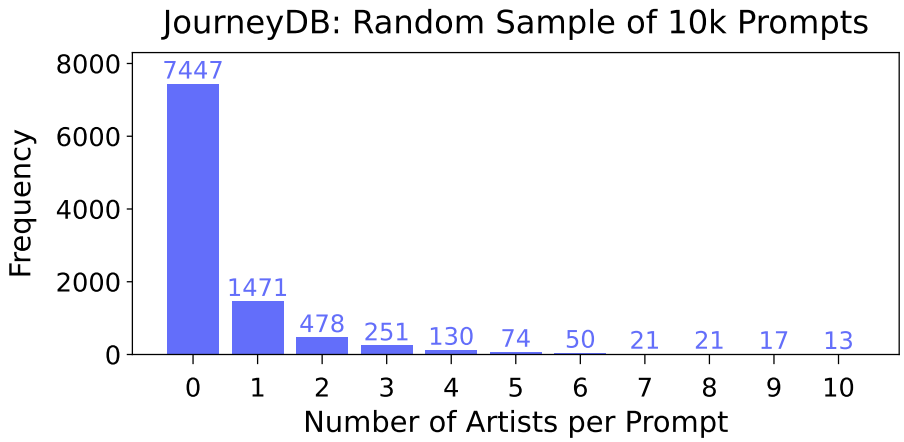

Figure 3.2: **Statistics on number of artists per prompt.** To understand how frequently text-to-image model users prompt for multiple artists, we take a random sample of 10,000 prompts collected from Midjourney users in JourneyDB [78] and visualize the distribution of the number of artists per prompt. While 14.7% of the prompts invoke only one artist name, 4.8% contain 2 artist names, and 2.5% contain 3 artist names.

3.5 Image Similarity Analysis

In Figure 1.2, we qualitatively observe that the generated images become less aligned with the style of the artist's original work when the prompt is more complex. In addition, given the same prompt, the style of the generated image varies depending on which text-to-image model was used. Thus, we conduct a simple quantitative analysis of CLIP [60] image similarity scores and find three aspects that make the prompted artist identification task challenging for generalization: 1) The effect of the artist's name in the prompt on the generated image is diluted when the prompt is more complex. 2) The alignment of the generated images to the artist's average

artwork varies across different text-to-image models. 3) The effect of adding artists into the prompt diminishes as the number of artists increases.

Comparing style alignment of text-to-image models. Different text-to-image models have different training datasets and architectures, which may affect the generated image's overall style. Therefore, we compare the text-to-image models in our benchmark by comparing image similarities across different text-to-image models in Table 3.1a. For each artist-prompted image, we compute the CLIP image similarity between it and the image generated with the same seed and prompt, except that the artist's name is removed. For all models, we observe that the CLIP similarity scores are lower for simple prompts than for complex prompts, indicating that the artist name has a larger influence on the generated image when the prompt is simple. Prompting for an artist's name also has a smaller effect on the generated image when using PixArt, compared to SDXL and SD1.5.

We also compute the similarity between the artist-prompted image embedding and the average embedding of the real artist images, which is the artist's prototype. When the prompt is more complex, the generated images are less aligned with the artist's real style. In addition, PixArt images are less aligned to the artist's real style than SDXL and SD1.5.

Influence of adding artist names into prompts. Since image generation prompts may contain more than one artist name, we study the effect of increasing the number of artist names in the prompt and how difficult it is to identify all the artists. In Table 3.1b, we show that the similarity between each image generated with the same seed and prompt increases as we add more artist names into the prompt, for both complex and simple prompts. We can qualitatively observe this decrease in influence per added artist in Figure 3.3. This indicates that classifying multiple artists in a prompt is more difficult than classifying a single artist, because the effect of each artist name is diluted as more artist names are added to the prompt.

(a) **Text-to-image model comparison**

Model	Content only vs. Artist-prompted		Artist-prompted vs. Real prototype	
	Simple	Complex	Simple	Complex
SDXL	0.571	0.738	0.570	0.476
PixArt	0.632	0.816	0.522	0.441
SD1.5	0.520	0.643	0.590	0.527
Midjourney	–	–	–	0.541

(b) **Increasing number of artists**

Prompt Type	Number of Artists Compared		
	0 vs. 1	1 vs. 2	2 vs. 3
Complex	0.738	0.811	0.866
Simple	0.571	0.711	0.782

Table 3.1: **Average CLIP image similarities**. (a) For each text-to-image model, we compute the average CLIP similarity between the image generated without an artist name and the image generated with one artist name (first two columns), and the average CLIP similarity between the image generated with an artist name and the real prototype for that artist (last two columns). The artist's influence and generated image's alignment to the artist's real style decreases when the prompt is more complex. Additionally, PixArt images are less aligned to the artist's real style than SDXL and SD1.5 images. (b) Given the same prompt and generation seed, we compute the average CLIP similarity between SDXL images generated with 0, 1, 2, and 3 artist names. The influence of each artist's name in the prompt diminishes as more artist names are added.

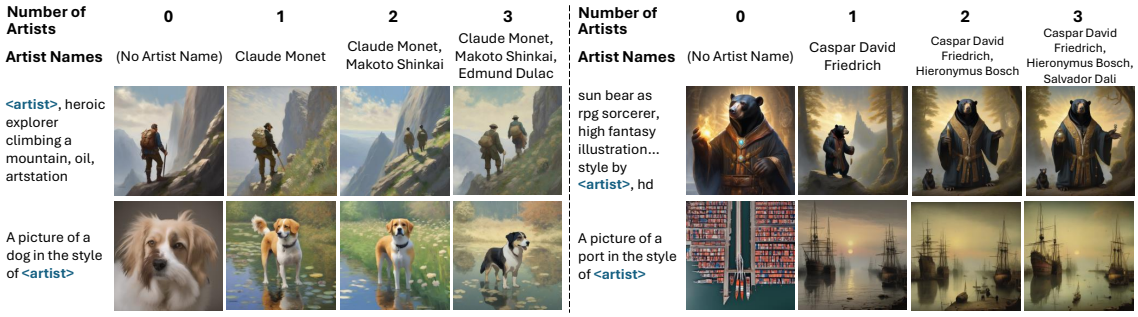

Figure 3.3: **Multi-artist prompted images**. In the example images shown, we observe that the effect of adding each artist's name in the prompt diminishes as more artists are added. Each row shows a set of images generated with the same prompt and generation seed, and the number of artists inserted into the prompt increases from left to right. For each prompt, the image changes less with each additional artist, and images generated with complex prompts generally change less than those generated with simple prompts.

Chapter 4

Experimental Setup

We evaluate a variety of computer vision models with published implementations on our prompted artist identification benchmark and employ evaluation procedures that use the same train and test sets across all models for fair comparison.

4.1 Models

Two approaches to classifying generated images are searching for similar artist reference images and training feed-forward classifiers. While retrieving image features from models trained on large-scale datasets allows for generalization to unseen classes, they require more runtime and storage compared to feed-forward classifiers. Thus, we compare several pre-trained retrieval-based methods and classifiers trained on our prompted artist identification dataset. We include full training details and ablations in the supplement.

- **Contrastive Style Descriptors (CSD) [76]** is a style descriptor that measures style similarity between images. CSD is trained using supervised contrastive learning [34] on ∼500k curated real artist images from LAION-5B [70].
- **DINOv2 [56], CLIP [60]:** commonly-used self-supervised image features trained on internet images.
- **Attribution by Customization (AbC) [81]:** DINO (AbC) and CLIP (AbC) are features for attributing training data in customized text-to-image diffusion

models, fine-tuned from DINO [7] and CLIP [60], respectively. They are trained on images synthesized from SD1.5 models customized on ImageNet [18] and art datasets.

- **Prototypical Network:** We train a classifier based on Prototypical Networks [73], a few-shot learning method, which allows for the prediction of unseen artists. During training, it learns an image encoder that outputs features in the embedding space close to the correct artist's prototype, the averaged pre-trained CLIP features of the artist's real reference images. During inference, the same prototype features can be used to predict the artist label set seen during training. We can also predict artists not seen during training by using the held-out artists' reference images for prototype features.
- **Vanilla Classifier:** We add an MLP with one hidden layer as the classification head to the CLIP image encoder, then train all layers on our dataset.

4.2 Evaluation Framework

Single-artist classification. We evaluate each model's top-1 classification accuracy for seen and held-out artists. For every test set of our prompted artist identification benchmark, each model is given the same set of training/retrieval images for controlled comparison. The training image set is all generated seen artist training images from our SDXL, SD1.5, PixArt, and Midjourney datasets and includes both simple and complex-prompted images. To evaluate retrieval-based methods on held-out artists, we use images generated with held-out prompts for test-time reference. For the vanilla classifier, we only report seen artist classification as the model cannot be applied to classes not seen during training. To estimate the statistical significance of each evaluation, we bootstrap each model's predictions by resampling the evaluation images, with replacement, by artist name, prompt template, and generation seed for 2000 iterations. We plot our bootstrapping procedure's convergence at 2000 iterations in the supplemental material.

Multi-artist classification. To evaluate artist classification on images prompted with multiple artist names, we use a multi-label classification metric: ranked mean average precision for the top 10 unique retrieved labels (mAP@10). We adapt the

prototypical network's training to a multi-label objective by using binary cross-entropy loss with label smoothing. The training dataset includes all single-artist images from SDXL, as well as all 2-artist and 3-artist training images. To evaluate retrieval-based methods, we only use single-artist SDXL images as the reference database.

Chapter 5

Results

We analyze prompted artist classification performance of various visual representation methods on all generalization settings of our benchmark.

5.1 Single-artist Classification

We present the quantitative results of all methods on the prompted artist classification task in Figure 5.1. While all methods surpass random chance, none exceed 91% accuracy, underscoring the inherent difficulty of the task. Performance consistently declines with increasing prompt complexity and when evaluating on PixArt-generated images, compared to SDXL and SD1.5-generated images. This suggests that *the complexity of the prompt and the text-to-image model used for image generation substantially influence classification difficulty.*

Across all generalization tasks, CSD, prototypical networks, and vanilla classifiers are the best-performing methods, followed by CLIP, the AbC models, and finally DINOv2. This indicates that *style descriptors and trained classifiers are more effective than off-the-shelf general-purpose visual representations and data attribution methods* for prompted artist identification.

Prototypical networks outperform CSD on most evaluation datasets and can learn a representation of the artist's style from generated images that is more robust to complex prompts than CSD, which is trained on real artworks. This indicates that *complex prompts generate images that are less aligned to the artist's real style*

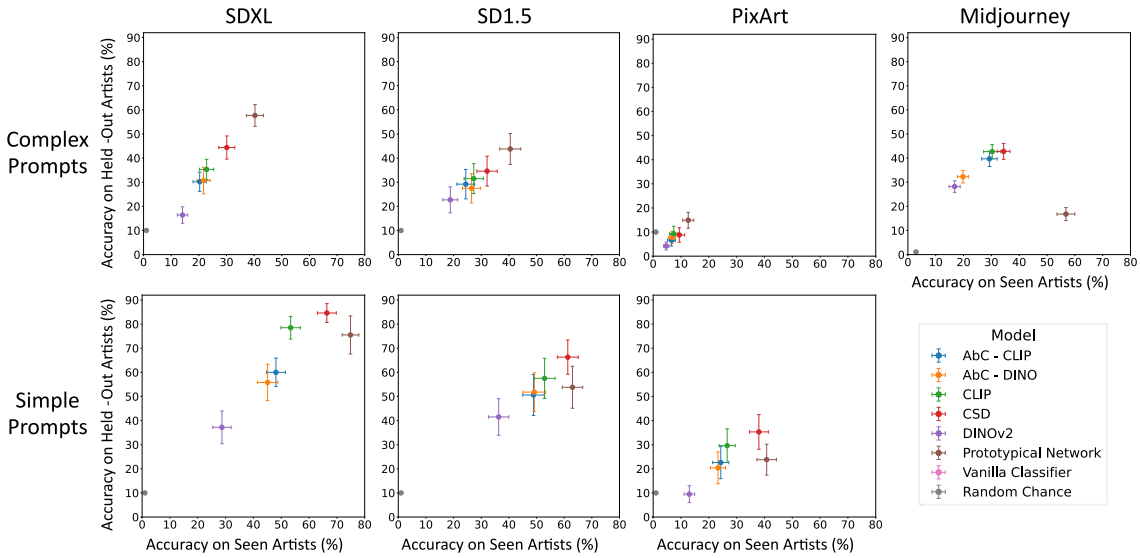

Figure 5.1: **Single-artist prediction results**. We compare the prompted artist classification accuracy of various visual representation methods. We test within our seen artists set (100-way classification, x-axis) and on held-out artists (10-way classification, y-axis). We also test on images generated with different text-to-image models (SDXL, SD1.5, PixArt, and Midjourney), and complex and simple prompts. Although prototypical networks [73], the trained vanilla classifier, and CSD [75] surpass the CLIP [60], DINOv2 [7], and AbC [81] methods, attaining high accuracy across all scenarios remains difficult. We include all numbers as tables in the supplemental material.

compared to simple prompts. However, CSD generalizes better to held-out artists and simple prompted images, and Midjourney images with complex prompts and held-out artists.

Generalization to unseen text-to-image models. We evaluate the generalization of the best-performing methods, prototypical networks, and CSD, to unseen text-to-image models by progressively increasing the training set of images [21] to include images from additional generators (SDXL→SD1.5→PixArt→Midjourney). Results are shown in Figure 5.2. For both methods, increasing the training dataset does not improve performance on unseen text-to-image models—performance only improves on each text-to-image model when the training set includes images from that generator. This suggests that *the models have not learned style representations that generalize across text-to-image models.* Notably, performance on PixArt images does

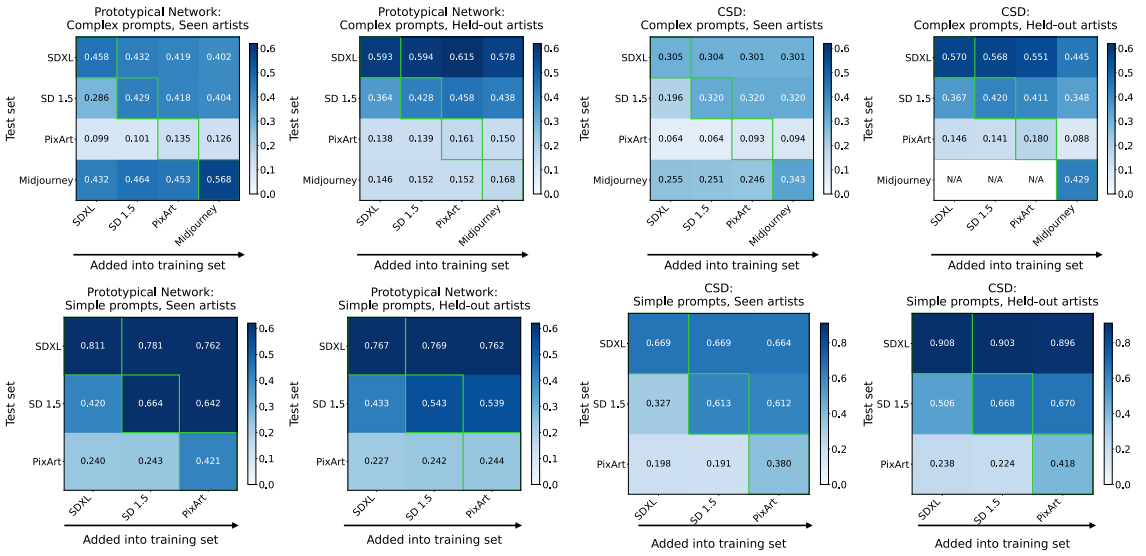

Figure 5.2: **Generalization to unseen generators**. For the best-performing methods, prototypical networks and CSD, we evaluate performance on unseen text-to-image models by progressively increasing the training set of images. Cells below the green diagonals evaluate generalization to unseen domains. For both methods, expanding the training dataset does not enhance performance on unseen text-to-image models; improvements occur only when the training data includes images generated by the specific model being evaluated.

not improve much, even after including PixArt images in the training set, supporting our observation in Section 3.5 that PixArt images are less aligned to the artist's real style compared to SDXL or SD1.5.

5.2 Multi-artist Classification

We further evaluate all methods on the multi-artist classification task, where the input image is prompted with multiple artists' names, and show results in Figure 5.3. *Across methods, performance tends to decline on images produced using complex prompts compared to those produced with simple prompts.* The prototypical network is the best performing method on the multi-artist datasets, as expected, because it is the only method trained on multi-artist prompted images, but performance remains far from saturation. The vanilla classifier is the second-best performing, even though it is not trained on multi-artist prompted images, followed by CSD. The remaining

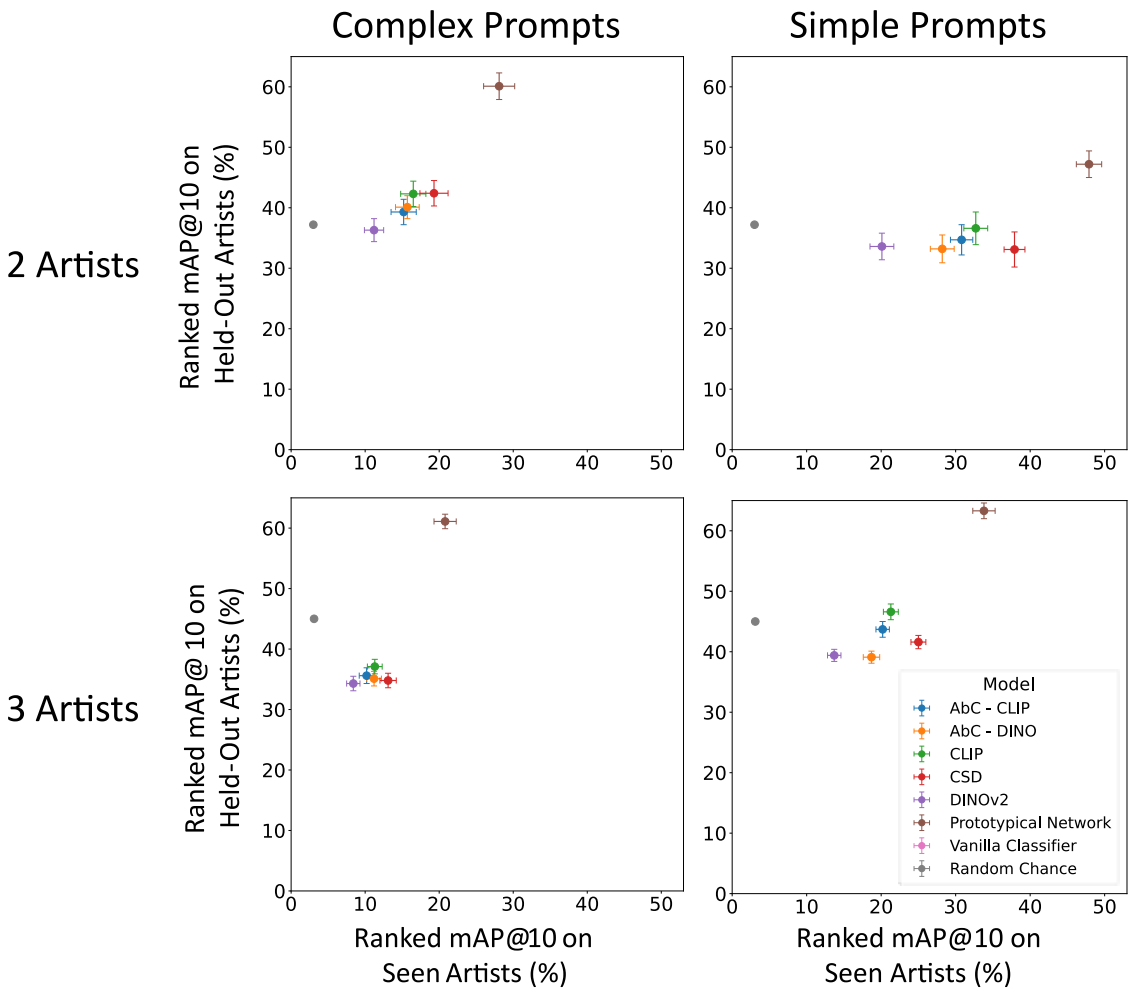

Figure 5.3: **Multi-artist prediction results**. We evaluate visual representation methods on the multi-artist classification task, where the input image is prompted with multiple artists' names. We test on SDXL-generated images with 2 artists and 3 artists in the prompt, and report ranked mAP@10 on seen artists (x-axis) and held-out artists (y-axis). All methods generally exhibit reduced performance on images generated from complex prompts compared to those generated from simple prompts. As expected, the prototypical network achieves the highest performance across the datasets due to its training on multi-artist prompted images. However, its performance falls short of saturation. We include all numbers as tables in the supplemental material.

retrieval-based methods perform similarly to CSD.

Generalization to increasing numbers of prompted artists. We also examine

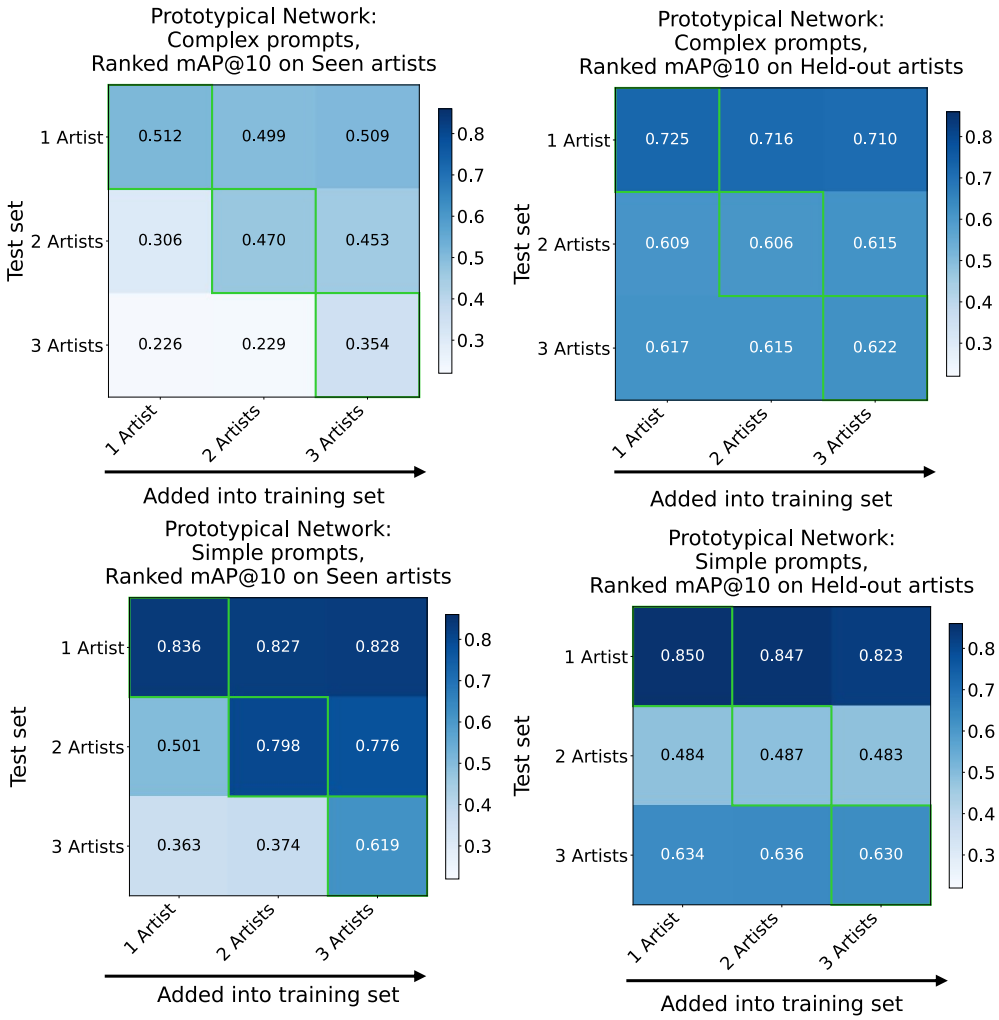

Figure 5.4: **Generalization to increasing numbers of prompted artists**. For prototypical networks, we evaluate performance on unseen numbers of artists in the prompt by progressively increasing the training set of images to include multi-artist prompted images. Cells below the green diagonals evaluate generalization to unseen domains. Increasing the size of the training set does not improve generalization to unseen numbers of artists. While using training images with the same number of artists as the test set enhances classification for seen artists, it does not benefit performance on held-out artists.

the impact of adding multi-artist prompted images to the training set of the prototypical network in Figure 5.4. Expanding the training set does not lead to better generalization to unseen artist counts. Training with images containing the same

number of artists as the test set improves seen artist classification but not held-out artist classification. This indicates that *the prototypical network is not able to learn a representation of the artist's style that generalizes to unseen numbers of artists and held-out artists*, even when trained on multi-artist prompted images. The image similarity analysis in Section 3.5 shows that the images generated with multiple artists are more similar to each other than images generated with a single artist, which may be the reason for this lack of generalization.

5.3 Additional Results

Training dataset ablation. To assess the effect of training dataset size, we vary the number of training prompts for the prototypical network from 150 to 1350 while keeping the ratio of simple to complex prompts fixed at 1:2. As shown in Figure 5.5, classification accuracy on seen artists consistently improves with more training prompts. In contrast, performance on held-out artists remains largely unchanged. This indicates that *while training the prototypical network on artist-prompted images improves the model's representations of styles from seen artists, it does not improve generalization to unseen artists.*

Artist name detection. Predicting whether an image was prompted with an artist name at all is another related task that is useful for generated content moderation applications. Thus, we evaluate several top-performing methods on this binary classification task. We construct a dataset from our artist-prompted SDXL subset, augment it with prompts that do not contain artist names, and balance the test set as shown in Table 5.1.

We evaluate linear probe classifiers on the CLIP, CSD, and prototypical network representations, and a fully fine-tuned CLIP model. Results in Table 5.2 show that while all methods exceed random chance, none achieve perfect accuracy even though they are fine-tuned for the detection task. Performance is consistently higher on simple prompts, aligning with our observation in Table 3.5 that *simple prompts yield greater feature differences between content-only and artist-prompted images—making them easier to distinguish than complex prompts.* Notably, the fully fine-tuned CLIP model under-performs on the test set, indicating susceptibility to overfitting. This is likely because the task is inherently harder with more intra-class variability, as the

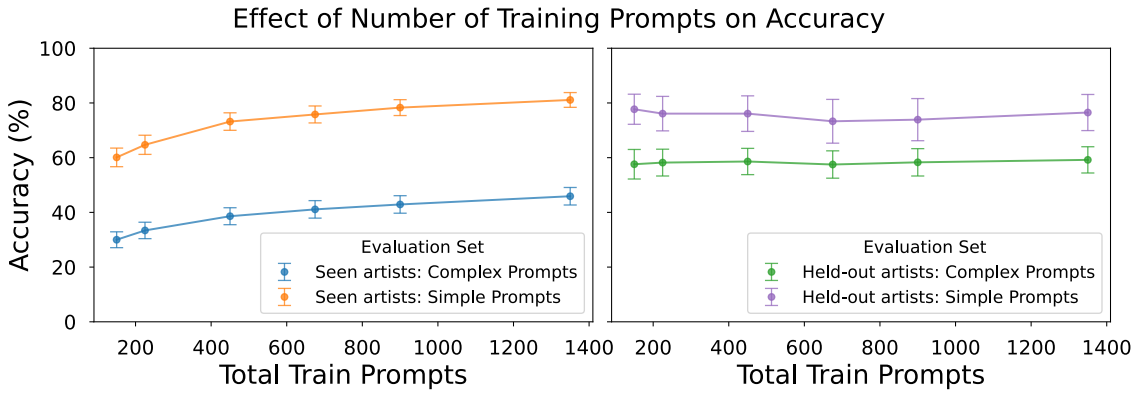

Figure 5.5: **Ablation on number of training prompts**. We plot the prototypical network classification accuracy on seen artists (left) and held-out artists (right) as a function of the number of prompts in the training set. Classification accuracy on seen artists improves steadily with more training prompts, but performance on held-out artists remains unchanged across all dataset sizes. The prototypical network is able to learn a representation of the artist's style that generalizes to unseen prompts with seen artists, but not to prompts with held-out artists.

model must learn to distinguish between images that are prompted with an artist's name and images that are not, and within each class, there is low visual similarity.

		Train		Test	
		Content only	Artist-prompted	Content only	Artist-prompted
Complex Prompts	# Seen artists	–	100	–	5
	# Held-out artists	–	0	–	5
	# Prompts	900	900	100	100
	# Seeds	10	10	10	1
Simple Prompts	# Seen artists	–	100	–	5
	# Held-out artists	–	0	–	5
	# Prompts	450	450	50	10
	# Seeds	2	2	2	1
	# Images	9,900	990,000	1,100	1,100

Table 5.1: **Artist name detection dataset**. To evaluate different methods on the task of detecting whether an image was prompted with an artist name, we construct a dataset of SDXL images prompted with artist names and images prompted with the same content but without the artist names. The training set contains all artist-prompted SDXL images from our benchmark, and the test set is balanced between content-only and artist-prompted images.

Detecting non-public domain artists. An important downstream application of

Method	Complex			Simple		
	Accuracy	AP	AUCROC	Accuracy	AP	AUCROC
Chance	50.0	–	–	50.0	–	–
CLIP - linear probe	61.6	79.9	79.5	82.5	97.7	97.9
CSD - linear probe	70.2	81.3	79.6	**92.5**	**98.2**	**98.3**
Proto. Net. - linear probe	**70.7**	**83.6**	**80.8**	91.0	97.3	97.0
Full CLIP fine-tune	67.3	78.6	79.3	84.5	96.4	97.1

Table 5.2: **Artist name detection results**. We evaluate the best prompted artist classification methods on the artist name detection task and report accuracy, average precision (AP), and area under the receiver operating characteristic curve (AUCROC). Even though all methods are fine-tuned for the detection task, none achieve perfect accuracy. Performance is consistently higher on simple prompts, and the fully fine-tuned CLIP model under-performs on the test set, indicating susceptibility to overfitting. This is likely because the artist name detection task is inherently harder.

prompted artist identification is detecting improper usage, such as the generation of derivative works based on copyright-protected artists. We evaluate the performance of various methods on the task of detecting non-public domain artists' names by grouping the SDXL-prompted artist identification dataset into two meta-labels: public domain or non-public domain, as defined by 95 years post-mortem. We test on the 100-way seen artists test set, which contains 55 public domain and 45 non-public domain artists, and report performance in Table 5.3. All methods perform better than random chance, but the trained classifier methods—prototypical networks and vanilla classifiers—perform the best. This indicates that *given a closed set of artists, training methods on artist-prompted images can improve the detection of non-public domain artists.*

Method	Complex			Simple		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Chance	45.0	–	–	45.0	–	–
CLIP	64.3 ± 1.6	63.7 ± 5.8	69.4 ± 2.8	81.0 ± 2.0	79.7 ± 4.2	84.0 ± 2.8
CSD	68.3 ± 1.7	67.0 ± 5.6	74.1 ± 2.7	86.5 ± 1.8	85.3 ± 3.4	88.9 ± 2.4
Proto. Net.	74.3 ± 2.0	70.3 ± 5.4	**85.2 ± 2.3**	**91.2 ± 1.6**	**90.3 ± 2.8**	92.8 ± 2.4
Classifier	**74.4 ± 1.8**	**71.9 ± 5.3**	81.2 ± 2.6	91.1 ± 1.5	89.9 ± 2.8	**92.9 ± 2.1**

Table 5.3: **Accuracy on non-public domain, seen artists**. We evaluate the performance of various methods on the task of detecting the presence of a non-public domain artist name in the prompt. We test on the 100-way seen artists test set, which contains 55 public domain and 45 non-public domain artists. The trained classifier methods, prototypical networks and vanilla classifiers, perform the best, suggesting that training on generated images can improve detection of non-public domain artists, given a closed set of artists.

Chapter 6

Conclusion

In this work, we explored the task of identifying prompted artist names in a generated image and demonstrated that it poses significant generalization challenges for current visual representation methods. We built the first large-scale benchmark and dataset suited for this task, comprised of 1.95 million images and prompted with 110 artist names. The benchmark is designed to probe four dimensions of generalization: held-out artists, varying prompt complexity, different text-to-image models, and prompts referencing multiple artists. Using this benchmark, we evaluated a diverse set of vision models. In summary, our findings are:

- Supervised classifiers and few-shot prototypical networks [73] trained on artist-prompted images excel on seen artists and complex prompts.
- Contrastive style descriptors [76] transfer better to images where the artist's style is more visually apparent.
- Learning style recognition from original artworks does not fully capture how style is represented in generative models.
- Prompts invoking multiple artist names remain the most challenging.
- There remains substantial room for improvement in generalization across all settings.

These findings highlight the need for vision models with improved generalization ability on prompted artist identification. We release our benchmark and dataset to facilitate the development of generalizable vision methods for the responsible

moderation of AI-generated content.

Limitations. We focused our benchmark on four text-to-image models as a starting point, but we acknowledge that there are several other popular closed-source models that could be evaluated and our Midjourney [52] evaluation was limited in size. We also did not evaluate on images generated with personalization techniques [39, 64, 87], another common way text-to-image users replicate artist styles. Finally, we did not evaluate the performance of multi-modal large language models, as many lack domain-specific knowledge about artists [6].

Appendix A

Supplementary Material

We include additional dataset curation details (Section A.1), training details for the classifier methods (Section A.2), relevant method experiments and ablations (Section A.3), and additional evaluation details (Section A.4). All referenced `.txt` files are available in the released dataset at https://huggingface.co/datasets/cmu-gil/PromptedArtistIdentificationDataset/tree/main/metadata.

A.1 Dataset Curation Details

A.1.1 Artist Curation

As mentioned in Section 3 of the main text, to collect a list of artists commonly used in text-to-image models, we manually de-duplicate and filter our set of seen artists from an initial list of 400 artists frequently prompted by Stable Diffusion users on the `Lexica.art` website [69, 76]. To ensure our set of 10 held-out artists are unseen by CSD [76] during its training, we filter LAION-5B [70] for artists from the leaked list of 16,000 artists [5] used to train Midjourney [52], that have at least four images having an aesthetic score greater than 4.

We then cross-reference our filtered artists with the captions in CSD's dataset to ensure our held-out artists' names do not appear in them. We include our final list of 100 seen artists in `seen_artists.txt`, list of 10 held-out artists in `held_out_artists.txt`, and list of 55 public domain artists in `public_domain_artists.txt`.

Artist images. We collect real reference images for each artist from a filtered version of LAION-5B [70] that only includes safe and legal images [41]. Then, we follow the same curation procedure as LAION-Styles [76], filtering for images with predicted aesthetics scores of 6 or higher, image captions containing the final dataset's 3840 style tags, and images with SSCD [58] similarity less than 0.8.

Multi-artist images. To generate images with prompts containing 2 or 3 artist names, we insert all possible held-out artist combinations and 100 random seen artist combinations into each prompt. We include the 100 random combinations of 2 seen artists in `seen_artist_2combos_100samples.txt`, the 100 random combinations of 3 seen artists in `seen_artist_3combos_100samples.txt`, all combinations of 2 held-out artists in `held_out_artist_all_2combos.txt`, and all combinations of 3 held-out artists in `held_out_artist_all_3combos.txt`.

A.1.2 Prompt Curation and Image Generation

Simple prompts. We generate images with simple prompts in the form of "`A picture of <content> in the style of <artist>`." For diversity, we use 100 subjects sampled from ChatGPT [11]. The prompt we give to ChatGPT is "`Provide a comma-separated list of 100 subjects that are commonly featured in paintings`." We manually inspect and remove subjects that may be too out-of-distribution for most artists' works or abstract concepts, such as "fairies" or "politics", and then prompt ChatGPT again to generate subjects until we curate 100 subjects in total.

Complex prompts. To curate a set of complex prompts from real text-to-image model users, we filter from the JourneyDB [78] train dataset, which contains ∼4M prompt-image pairs. We first obtain a list of prompts that only mention one artist, rather than multiple artists, by instructing the large language model Llama 3 8B Instruct [20]. Specifically, for each JourneyDB prompt, we instruct Llama 3 with the following: "`Output only the names that are artists that are mentioned in the following text as a JSON list. Do not say any extra words:`." Using a large language model for this task ensures we obtain more prompts that we would find with hard string matching to a closed set of artists. Then, we remove prompts where Llama 3 returns more than one artist name. We also remove prompts that are

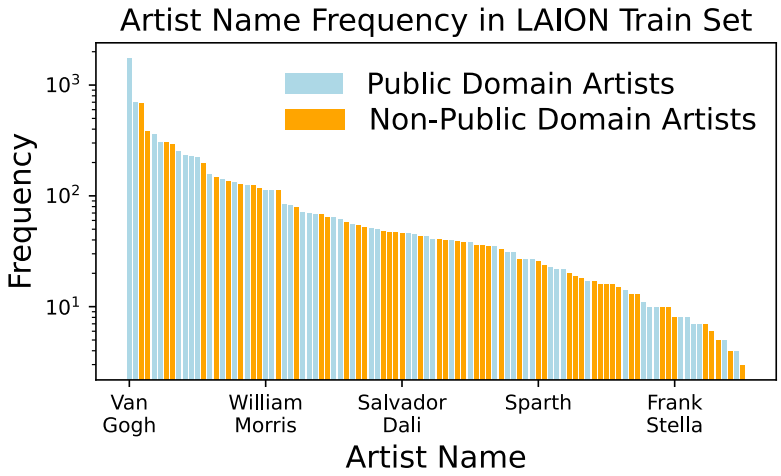

Figure A.1: **Artist statistics.** We plot the frequency of each artist's name in our LAION training set, with artists labeled public domain (blue) or non-public domain (orange), as defined by 95 years post-mortem. Out of our set of 100 seen artists, 45 are non-public domain, and their images make up 38.7% of our real image dataset.

less than 5 words or have more than 77 CLIP [60] text tokens to ensure SDXL [59] is able to condition images on the full prompt. Finally, we sample 1,000 prompts for our dataset and remove near-duplicate prompts with manual inspection. We include our final lists of prompts with the train and test splits. Complex prompts are in `complex_prompts_train.txt` and `complex_prompts_test.txt`, and simple prompts are in `simple_prompts_train.txt` and `simple_prompts_test.txt`. For evaluation on held-out artists, we use images generated with the first 5 prompts in `complex_prompts_test.txt` as test-time reference images for complex prompts, and image generated with the first 5 prompts in `simple_prompts_test.txt` as test-time reference images for simple prompts.

Image generation. For each artist and prompt combination from our curated lists, we generate artist-prompted images using the open-source text-to-image models SDXL [59], SD1.5 [62], and PixArt-Σ [12]. For each model, we use its default generation parameters: SD1.5 with the resolution of 512×512, guidance scale of 7.5, and 50 inference steps; PixArt-Σ with the resolution of 1024×1024, guidance scale of 4.5, and 50 inference steps; and SDXL with the resolution of 1024×1024, guidance scale of 7.5, and 50 inference steps.

Distribution of real artist images. Figure A.1 shows how frequently each artist's

name appears in the captions of our LAION dataset of 100 artists. We compare the frequencies of public-domain artists and non-public-domain artists, using the guideline that an artist's maximum possible U.S. copyright term is 95 years post-mortem [74]. We find that 38.7% of the images come from 45 non-public domain artists. As seen in Figure A.1, public and non-public domain artists have similar frequency distributions. We use this dataset as real image references for artists' styles for the prototypical network.

A.1.3 Prompting Recent Text-to-Image Models with Artist Names

We tried prompting other recent text-to-image models with artist names, specifically Stable Diffusion 3.5 Large (SD3.5) [22], and FLUX.1-dev [40], but found that they using artist names often has little to no effect on the generated image style, even when the prompt is simple. For instance, in Figure A.2, we generate images using simple prompts and training set artists from our dataset. Default parameters were used for each model: SD3.5 with 50 inference steps, guidance scale of 7.0, and FLUX.1-dev with 50 inference steps, guidance scale of 3.5. For each prompt, the models generate artist-prompted images that are nearly the same photorealistic style as the images prompted with no artist names. Furthermore, the artist-prompted images are not aligned with the artists' actual styles. Since these models often fail to stylize images when prompted with artist names, we do not include these models in our main evaluations.

A.1.4 Dataset Statistics Tables

Following the dataset curation procedure, we create a structured dataset of 1.95M images in order to evaluate vision methods across different axes of generalization. The single-artist datasets are summarized in Table A.1. For each artist and prompt combination, we generate images with 2 seeds using the open-weight models SDXL [59], SD1.5 [62], and PixArt-Σ [12]. For SDXL images generated with complex prompts, we use 10 different generation seeds. To collect images from Midjourney [52], a closed-source model, we filter the JourneyDB dataset [78] for our set of seen and

Figure A.2: **FLUX and SD3.5 examples.** For FLUX.1-dev [40] and SD3.5 [22], inserting artist names into the prompt frequently has little to no influence on the generated image's style compared to using the same prompt without any artist name. Most of the resulting images are photorealistic, and the generated images do not match the artists' actual styles (bottom row).

held-out artists. The multi-artist datasets are summarized in Table A.2. We sample 100 random seen artist combinations and all possible held-out artist combinations for 2 artists and 3 artists, and generate images with SDXL.

A.2 Training Details

Prototypical Network. We train the prototypical network [73] for prompted artist identification by fine-tuning the pre-trained CLIP [60] ViT-L/14 image encoder end-to-end on our prompted artist identification dataset. We apply blur and JPEG training data augmentation by using the same recommended setting as Wang *et al*. [80]: the image is randomly blurred and JPEG-ed, each with 10% probability. Random resized crop and horizontal flip are also applied. We sweep learning rates, $1e-7$, $1e-6$, and $1e-5$, and number of training epochs, 1 through 5. We choose the best-performing configuration: fine-tune for 1 epoch with a learning rate of $1e-6$, temperature τ of 0.07, and a batch size of 128 per GPU, distributed across 4 GPUs. This leads to an effective batch size of 512. The compute environment is 4 A6000 GPUs with 40GB of vRAM each and mixed precision (fp16) training.

(a) **SDXL**

	Complex Prompts				Simple Prompts				Total Unique
	Seen artists		Held-out artists		Seen artists		Held-out artists		
	Train	Test	Test-Time Ref.	Test	Train	Test	Test-Time Ref.	Test	
# Artists	100	100	10	10	100	100	10	10	110
# Prompts	900	100	5	90	450	50	5	45	1,500
# Seeds	10	10	10	10	2	2	2	2	10
# Images	900,000	100,000	500	9,000	90,000	10,000	100	900	1,110,500

(b) **SD1.5, PixArt**

	Complex Prompts				Simple Prompts				Total Unique
	Seen artists		Held-out artists		Seen artists		Held-out artists		
	Train	Test	Test-Time Ref.	Test	Train	Test	Test-Time Ref.	Test	
# Artists	100	100	10	10	100	100	10	10	110
# Prompts	450	50	5	45	450	50	5	45	1,000
# Seeds	2	2	2	2	2	2	2	2	2
# Images	90,000	10,000	100	900	90,000	10,000	100	900	202,000

(c) **Midjourney**

	Complex Prompts				Total Unique
	Seen artists		Held-out artists		
	Train	Test	Test-Time Ref.	Test	
# Artists	35	35	95	95	130
# Prompts	647	658	649	697	2,651
# Images	647	658	649	697	2,651

Table A.1: **Single-artist dataset statistics.** The prompted artist identification dataset includes images generated with SDXL [59], SD1.5 [62], PixArt-Σ [12], and Midjourney [52]. As described in Section 3, we collect 1,000 complex prompts, 500 simple prompts, and use 110 different artist names in the prompts. For each artist and prompt combination, we generate images using 10 different seeds for SDXL complex prompts and 2 different seeds for the other subsets. We split the 110 artists into 100 seen and 10 held-out artists. For seen artists, we use a separate set of prompts for testing. For held-out artists, we additionally split test prompt images for test-time reference.

Vanilla Classifier. We add an MLP with one hidden layer as the classification head to the pre-trained CLIP [60] ViT-L/14 image encoder, then train all layers on our dataset. The training settings are consistent with the prototypical network: fine-tune for 1 epoch with a learning rate of $1e-6$, effective batch size of 512, and fp16 mixed precision training.

(a) **2 Artists**

	Complex Prompts				Simple Prompts				Total Unique
	Seen artists		Held-out artists		Seen artists		Held-out artists		
	Train	Test	Test-Time Ref.	Test	Train	Test	Test-Time Ref.	Test	
# Artists	100	100	10	10	100	100	10	10	110
# Prompts	450	50	5	45	450	50	5	45	1,000
# Seeds	2	2	2	2	2	2	2	2	2
# 2-artist combinations	100	100	45	45	100	100	45	45	145
# Images	90,000	10,000	450	4,050	90,000	10,000	450	4,050	209,000

(b) **3 Artists**

	Complex Prompts				Simple Prompts				Total Unique
	Seen artists		Held-out artists		Seen artists		Held-out artists		
	Train	Test	Test-Time Ref.	Test	Train	Test	Test-Time Ref.	Test	
# Artists	100	100	10	10	100	100	10	10	110
# Prompts	450	50	5	45	450	50	5	45	1,000
# Seeds	2	2	2	2	2	2	2	2	2
# 3-artist combinations	100	100	120	120	100	100	120	120	220
# Images	90,000	10,000	1,200	10,800	90,000	10,000	1,200	10,800	224,000

Table A.2: **Multi-artist dataset statistics.** To evaluate prompted artist identification of images generated with multiple artists in the prompt, we generate datasets of images prompted with 2 artists and 3 artists. The datasets are generated using the same procedure as the single-artist dataset, except we sample 100 artist combinations for each number of artists.

A.3 Experiments and Ablations

We analyze and validate the design choices for each of the visual representation methods we evaluate on our benchmark in the main paper. All experiments were conducted with SDXL images from our prompted artist identification dataset. All complex prompts, and a subset of 100 simple prompts (90 train, 10 test), were included in the dataset.

A.3.1 Retrieve from Real Images

Retrieval-based methods can use either real artist images or generated images as the reference database to classify artist-prompted images. We compare both settings on our benchmark and find that for each method, retrieval from generated images outperforms retrieval from real images on most evaluation sets, and the final performance ranking of the methods remains largely unchanged. We use real artist images collected

from LAION-Styles A.3, the same set of images used to compute the prototypes for the prototypical network. For retrieval from generated images, we use all of the generated seen artist training images from our SDXL, SD1.5, PixArt, and Midjourney datasets, including both simple and complex-prompted images. In Table A.6, we report results on SDXL images and observe that retrieval from real images leads to lower performance than retrieval from generated images for all methods, across all evaluation sets except for images with complex prompts and held-out artists. Thus, we focus on retrieval from generated images in our main benchmark evaluation.

	Seen artists	Held-out artists	Total
# Artists	100	10	110
# Images	9,907	860	10,767

Table A.3: **Real artist image dataset**. We use a subset of LAION-Styles [76] containing 100 seen artists and 10 held-out artists. Then, we benchmark prompted artist identification of various visual representation methods and use this dataset for real image references of each artist's style.

A.3.2 Retrieve from Prototypes

In the main benchmark evaluation, we evaluate retrieval-based methods that use individual reference image embeddings for retrieval. However, the prototypical network learns classification via prototype alignment, where we set an artist's prototype to be the averaged feature of the real reference images of the artist's work.

For a fair comparison, we test retrieval-based methods in a similar setting by retrieving from a set of averaged embeddings of real images, computing one average embedding per artist (same as the prototype). The difference is that the methods are *not* fine-tuned to align with the prototypes. We evaluate this average embedding retrieval baseline for CSD [76] and CLIP [60] and compare the results to the prototypical network in Table A.4. We find that the prototypical network outperforms them in most categories.

Method	Reference/ Prototype	Evaluation Set			
		Seen artists (100-way)		Held-out (10-way)	
		Complex	Simple	Complex	Simple
CLIP	Artist Avg.	15.1	38.3	44.9	79.0
CSD	Artist Avg.	19.7	48.1	56.8	**92.0**
Proto. Net.	Artist Avg.	**43.2**	**87.0**	**62.7**	87.0

Table A.4: **Comparison to baselines retrieving from prototypes**. We additionally evaluate average embedding retrieval for baselines CSD [76] and CLIP [60] and compare them to the prototypical network [73].

A.3.3 k-NN Evaluation Results

In our main evaluations of retrieval-based methods on our benchmark, we use k-NN classification with $k = 1$, as we find that increasing k does not help.

We test k-NN classification with majority voting, where the final predicted class label is the majority class label within the set of the top k nearest neighbors, and ties are broken by the majority class label with the closest retrieved embedding to the query embedding. We evaluate multiple k values for k-NN majority voting classification to investigate the effect of k on artist classification accuracy in Figure A.3.

First, we find that the performance ranking of the methods remains unchanged across different k values. Also, for each retrieval-based method, the performance only changes a small amount as k increases. The exception to these trends is the Held-out (Simple Prompts) test case, where the performance saturates at a very small k.

A.3.4 Classification Method Experiments

We analyze and validate the design choices for the classification methods we evaluate on our benchmark in the main paper. Table A.5 shows results on SDXL images from our prompted artist identification dataset, averaged across the complex and simple prompt evaluation sets.

Comparing classifier types. In Table A.5a, we train and compare different model types: vanilla classification, supervised contrastive learning [34], and prototypical networks [73]. While the classifier outperforms prototypical networks on seen artist classification, by default, it cannot classify held-out artists because of the fixed label

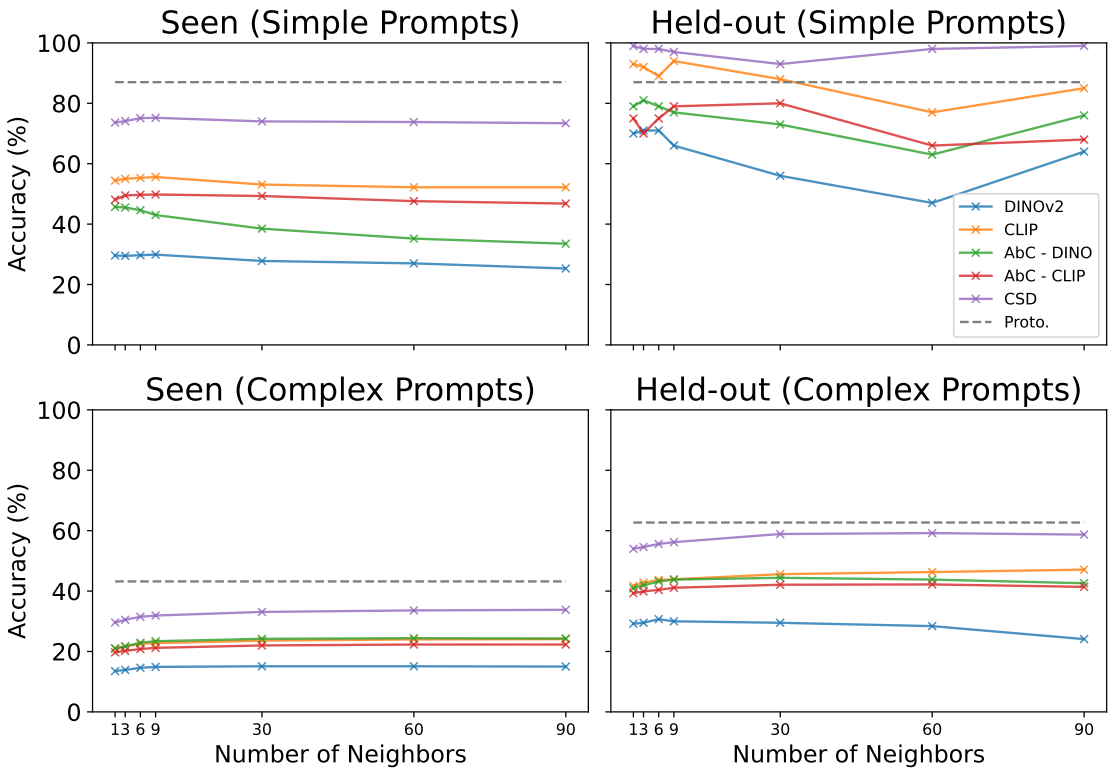

Figure A.3: **Effect of k on accuracy for k-NN classification**. We perform k-NN evaluation on multiple k values and find that for each retrieval-based method, the performance only changes a small amount as k increases. Thus, the performance ranking of the methods remains unchanged across different k values and the gap between our method and baselines remains the same.

set. To test generalization, we remove the final linear layer of the classifier to extract features for retrieval. We find that retrieving from real or generated images performs worse than the prototypical network in held-out artist classification. This indicates training with prototypes of reference images helps with generalization.

Next, we try metric learning by running Supervised Contrastive Learning [34] on the generated images, with all images from the same artist using the same label. We find that this does not achieve strong performance and focus on prototypical networks and vanilla classification for benchmark evaluation.

Nearest neighbors and different sources for prototypes. Prototypical networks can also be used as a feature extractor to run retrieval, which performs on par with using the prototype for classification. We report results Table A.5c. In addition,

using generated images as prototypes leads to worse performance, so we use real images as prototypes in benchmark evaluations. We hypothesize that using a different modality (e.g., real artist images) as prototypes encourages the encoder to extract artist-specific features more, helping overall generalization.

Which training data helps? We vary the training set of the prototypical network and report performance in Table A.5b. First, instead of training to classify artists from generated images, we train a prototypical network using real artist images as input. Training a model without synthesized images drastically hurts performance. Since good classification performance requires extracting artist-related cues even when images are generated from complex prompts, training an encoder from generated images can help learn such cues.

We further confirm this hypothesis by training a prototypical network using only generated images with easy prompts. While this variant performs better than the one trained only with real images, the performance remains worse than the ones trained on generated images with complex prompts. On the other hand, training only with complex prompts yields similar performance as training on the dataset with simple and complex prompts combined.

A.4 Evaluation Details

A.4.1 Bootstrapping Procedure

To estimate the statistical significance of each evaluation on our benchmark, we use a bootstrapping procedure. We bootstrap each model's predictions by resampling the evaluation images, with replacement, by artist name, prompt template, and generation seed for 2000 iterations. We selected 2000 iterations by plotting the convergence of the standard error, as shown in Figure A.4. Given the prototypical network's artist classification predictions on SDXL images and simple prompts, we plot the standard error of the classification accuracy against the number of bootstrap iterations. The number of bootstrapping iterations was increased from 50 to 4000 in increments of 50. The standard error remains within the range of 90% of the data points at around 2000 iterations, indicating that the standard error converges at this point. Thus, we use 2000 bootstrap iterations in our main benchmark evaluation.

(a) **Model type**

Model		Evaluation set	
Training	Testing	Seen Artists (100-way)	Held-out (10-way)
Vanilla Classifier	Classifier	**53.4**	–
	NN - Real	24.4	45.0
	NN - Gen.	51.3	58.0
Supervised Contrastive Learning [34]	Classifier	49.3	–
	NN - Real	24.4	21.6
	NN - Gen.	47.5	29.6
Proto. Net. [73]	Proto. - Real	43.9	**63.0**

(b) **Training set**

Model	Evaluation set	
	Seen artists (100-way)	Held-out (10-way)
Real	16.5	49.7
Simple only	21.1	55.2
Complex only	43.6	62.5
Simple + Complex	**43.9**	**63.0**

(c) **Proto. vs. Nearest Neighbors**

	Reference		Evaluation set	
Method	Real	Gen.	Seen artists (100-way)	Held-out (10-way)
NN - Real	✓		26.7	54.3
NN - Gen.		✓	**43.9**	62.3
Proto. - Gen.		✓	23.0	61.2
Proto. - Real	✓		**43.9**	**63.0**

Table A.5: **Classification method experiments**. In the main paper, we benchmark two classification methods: prototypical networks and vanilla classification. Here, we test more classifier method variants and training datasets, reporting accuracy numbers averaged across the SDXL complex and simple prompt evaluation sets. (a) We test **model types**, comparing vanilla classification, supervised contrastive learning, and prototypical networks. As classifiers have a fixed label set, we use nearest neighbors on their feature space when testing on held-out artists. Supervised contrastive learning performs worse than both the prototypical network and vanilla classification, so we do not include it in our benchmark evaluation. (b) We test **training sets** for the prototypical network. Learning from real images is not as effective as learning from generated images. Using both simple and complex prompts outperforms using one or the other, demonstrating they provide complementary information. (c) For the prototypical network, we compare using the **prototype vs. nearest neighbors**. When using the learned prototypical network representation for nearest neighbors, performance slightly degrades. Using the generated images for prototypes, rather than real, also degrades performance, so we use real images as prototypes for our main implementation.

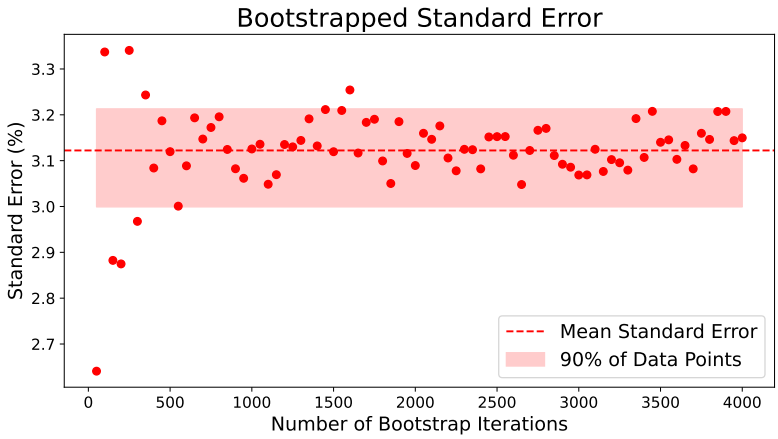

Figure A.4: **Bootstrapping procedure convergence**. We plot standard error against the number of bootstrap iterations for the prototypical network's artist classification accuracy on SDXL images and simple prompts. The number of bootstrapping iterations were increased from 50 to 4000 in increments of 50. The standard error converges at around 2000 iterations, where the standard error remains within the range of 90% of the data points. Thus, we use 2000 bootstrap iterations in our main benchmark evaluation.

A.4.2 Quantitative Result Tables

We include the quantitative results from our main benchmark evaluation in Section 5 as tables below. The single-artist evaluation results include SDXL images in Table A.6, SD1.5 images in Table A.7, PixArt-Σ images in Table A.8, and Midjourney images in Table A.9. The multi-artist evaluation results include 2-artist prompted images in Table A.10 and 3-artist prompted images in Table A.11.

Family	Method	Architecture	Reference/ Prototype images: Real	Reference/ Prototype images: Gen.	Evaluation set: Seen artists (100-way): Complex	Evaluation set: Seen artists (100-way): Simple	Evaluation set: Held-out (10-way): Complex	Evaluation set: Held-out (10-way): Simple
Chance	–	–	–	–	1.0	1.0	10.0	10.0
Retrieval-based methods	AbC - DINO	ViT-B/16	✓		7.2 ± 1.0	13.9 ± 2.0	33.9 ± 6.4	47.7 ± 9.4
				✓	21.7 ± 2.4	45.1 ± 3.7	30.7 ± 5.5	55.8 ± 7.5
	AbC - CLIP	ViT-B/16	✓		10.5 ± 1.5	20.4 ± 2.7	35.4 ± 6.3	52.3 ± 8.4
				✓	20.3 ± 2.4	48.1 ± 3.4	30.2 ± 4.0	60.0 ± 5.9
	DINOv2	ViT-L/14	✓		6.0 ± 1.0	11.0 ± 1.9	29.6 ± 5.5	44.6 ± 8.2
				✓	14.1 ± 1.9	28.7 ± 3.3	16.4 ± 3.4	37.2 ± 6.8
	CLIP	ViT-L/14	✓		12.4 ± 1.7	24.9 ± 2.9	41.1 ± 6.8	62.7 ± 8.3
				✓	22.8 ± 2.6	53.4 ± 3.5	35.3 ± 4.2	78.5 ± 4.7
	CSD	ViT-L/14	✓		15.3 ± 2.0	31.9 ± 3.3	47.4 ± 7.0	74.3 ± 6.6
				✓	30.1 ± 2.9	66.4 ± 3.4	44.4 ± 4.8	**84.6 ± 3.9**
Fine-tuned classifiers	Prototypical Network	ViT-L/14	✓		40.3 ± 3.1	74.9 ± 3.0	**57.7 ± 4.5**	75.5 ± 7.9
	Vanilla Classifier	ViT-L/14		✓	**40.7 ± 3.2**	**76.4 ± 3.0**	–	–

Table A.6: **Classification accuracy on SDXL images**. We compare retrieval-based methods and fine-tuned classifiers on SDXL images from our prompted artist identification dataset. While retrieval-based methods can use either real or generated images as the reference database, we find that retrieval from generated images outperforms retrieval from real images for all methods, across all evaluation sets except for images with complex prompts and held-out artists. Thus, we focus on retrieval from generated images in our main benchmark evaluation.

Family	Method	Evaluation set: Seen artists (100-way): Complex	Evaluation set: Seen artists (100-way): Simple	Evaluation set: Held-out (10-way): Complex	Evaluation set: Held-out (10-way): Simple
Chance	–	1.0	1.0	10.0	10.0
Retrieval-based methods	AbC - DINO	26.5 ± 3.3	49.2 ± 4.1	27.5 ± 6.1	51.8 ± 8.0
	AbC - CLIP	24.4 ± 3.2	48.9 ± 3.9	29.2 ± 6.1	50.6 ± 8.5
	DINOv2	18.8 ± 2.7	36.3 ± 3.7	22.7 ± 5.4	41.5 ± 7.6
	CLIP	27.3 ± 3.5	52.9 ± 3.8	31.5 ± 6.2	57.5 ± 8.3
	CSD	32.1 ± 3.7	61.3 ± 3.8	34.6 ± 6.2	**66.3 ± 7.1**
Fine-tuned classifiers	Prototypical Network	**40.5 ± 3.8**	63.0 ± 3.7	**43.8 ± 6.4**	53.8 ± 8.7
	Vanilla Classifier	40.2 ± 3.8	**65.2 ± 3.7**	–	–

Table A.7: **Classification accuracy on SD1.5 images**.

		Evaluation set			
		Seen artists (100-way)		Held-out (10-way)	
Family	Method	Complex	Simple	Complex	Simple
Chance	–	1.0	1.0	10.0	10.0
Retrieval-based methods	AbC - DINO	6.8 ± 1.3	23.3 ± 2.7	7.8 ± 2.5	20.4 ± 6.6
	AbC - CLIP	6.6 ± 1.4	24.3 ± 2.9	6.6 ± 2.4	22.6 ± 6.7
	DINOv2	4.7 ± 1.0	13.0 ± 1.9	4.2 ± 1.6	9.5 ± 3.5
	CLIP	7.4 ± 1.5	26.6 ± 2.9	9.3 ± 3.1	29.6 ± 7.0
	CSD	9.4 ± 1.9	38.0 ± 3.4	8.8 ± 3.0	**35.3 ± 7.2**
Fine-tuned classifiers	Prototypical Network	12.6 ± 2.0	40.8 ± 3.5	**14.9 ± 3.3**	23.8 ± 6.4
	Vanilla Classifier	**12.9 ± 2.0**	**41.9 ± 3.5**	–	–

Table A.8: **Classification accuracy on PixArt-Σ images**.

		Evaluation set	
		Seen artists (34-way)	Held-out (96-way)
Family	Method	Complex	Complex
Chance	–	2.9	1.0
Retrieval-based methods	AbC - DINO	19.8 ± 2.0	32.3 ± 2.6
	AbC - CLIP	29.3 ± 2.8	39.7 ± 3.2
	DINOv2	16.8 ± 2.0	28.2 ± 2.4
	CLIP	30.3 ± 3.1	42.7 ± 2.9
	CSD	34.4 ± 2.3	**42.8 ± 3.3**
Fine-tuned classifiers	Prototypical Network	**56.8 ± 3.2**	16.8 ± 2.7
	Vanilla Classifier	57.6 ± 2.7	–

Table A.9: **Classification accuracy on Midjourney images**.

		Evaluation set			
		Seen artists (100-way)		Held-out (10-way)	
Family	Method	Complex	Simple	Complex	Simple
Chance	–	3.0	3.0	37.2	37.2
Retrieval-based methods	AbC - DINO	15.7 ± 1.6	28.2 ± 1.6	40.1 ± 1.9	33.2 ± 2.3
	AbC - CLIP	15.2 ± 1.7	30.8 ± 1.5	39.3 ± 2.1	34.7 ± 2.5
	DINOv2	11.2 ± 1.3	20.1 ± 1.6	36.3 ± 1.9	33.6 ± 2.2
	CLIP	16.5 ± 1.7	32.7 ± 1.6	42.3 ± 2.1	36.6 ± 2.7
	CSD	19.3 ± 1.9	37.9 ± 1.4	42.4 ± 2.1	33.1 ± 2.9
Fine-tuned classifiers	Prototypical Network	**45.1 ± 3.2**	**77.6 ± 2.1**	**61.5 ± 2.3**	**48.3 ± 2.4**
	Vanilla Classifier	28.5 ± 2.1	49.3 ± 1.8	–	–

Table A.10: **Ranked mAP@10 artist retrieval on SDXL 2-artist prompted images**.

		Evaluation set			
		Seen artists (100-way)		Held-out (10-way)	
Family	Method	Complex	Simple	Complex	Simple
Chance	–	3.1	3.1	45.0	45.0
Retrieval-based methods	AbC - DINO	11.2 ± 1.0	20.4 ± 1.0	35.1 ± 1.2	39.1 ± 1.0
	AbC - CLIP	10.2 ± 1.0	20.2 ± 0.9	35.6 ± 1.3	43.6 ± 1.3
	DINOv2	8.4 ± 0.9	13.7 ± 0.9	34.3 ± 1.3	39.3 ± 1.1
	CLIP	11.3 ± 1.1	21.3 ± 1.0	37.1 ± 1.2	46.6 ± 1.3
	CSD	13.2 ± 1.1	25.0 ± 1.0	34.8 ± 1.2	41.6 ± 1.1
Fine-tuned classifiers	Prototypical Network	**35.3 ± 2.7**	**61.9 ± 2.1**	**62.2 ± 1.3**	**63.0 ± 1.4**
	Vanilla Classifier	20.7 ± 1.6	34.2 ± 1.4	–	–

Table A.11: **Ranked mAP@10 artist retrieval on SDXL 3-artist prompted images**.

Bibliography

[1] Adobe. Updated artist name guidelines, 2023. URL https://helpx.adobe.com/stock/contributor/help/updated-artist-name-guidelines.html. Accessed: 2024-11-13. 1

[2] Stability AI. Stable safety, 2025. URL https://stability.ai/safety. Accessed on April 16, 2025. 3.3

[3] Sarah Andersen. The alt-right manipulated my comic. then a.i. claimed it. *The New York Times*, December 2022. URL https://www.nytimes.com/2022/12/31/opinion/sarah-andersen-how-algorithim-took-my-work.html. 1

[4] Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20041–20053, 2023. 2.4

[5] Theo Belci. Leaked: the names of more than 16,000 non-consenting artists allegedly used to train midjourney's ai, 1 2024. URL https://www.theartnewspaper.com/2024/01/04/leaked-names-of-16000-artists-used-to-train-midjourney-ai. Accessed: November 20, 2024. A.1.1

[6] Yi Bin, Wenhao Shi, Yujuan Ding, Zhiqiang Hu, Zheng Wang, Yang Yang, See-Kiong Ng, and Heng Tao Shen. Gallerygpt: Analyzing paintings with large multimodal models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7734–7743, 2024. 6

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. (document), 4.1, 5.1

[8] Stephen Casper, Daniel Filan, Gabriel Nicholas, Miles Brundage, and Matthew L. Schwartz. Measuring the success of diffusion models at imitating human artists. In *Workshop on Generative AI and Law (GenLaw), International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2307.04028.

2.3

[9] George Cazenavette, Avneesh Sud, Thomas Leung, and Ben Usman. FakeInversion: Learning to detect images from unseen models by inverting stable diffusion. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2.1

[10] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European Conference on Computer Vision (ECCV)*, 2020. 2.1

[11] ChatGPT. https://chat.openai.com/chat, 2024. 3.2, A.1.2

[12] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision (ECCV)*, 2024. (document), 1, 1.2, 3.1, 3.3, A.1.2, A.1.4, A.1

[13] Yiqun Chen and James Y Zou. Twigma: A dataset of ai-generated images with metadata from twitter. *Conference on Neural Information Processing Systems (NeurIPS)*, 36:37748–37760, 2023. 2.4

[14] Zijie Chen, Lichao Zhang, Fangsheng Weng, Lili Pan, and Zhenzhong Lan. Tailored visions: Enhancing text-to-image generation with personalized prompt rewriting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7727–7736, 2024. 2.4

[15] Sang Keun Choe et al. What is your data worth to gpt? llm-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*, 2024. 2.2

[16] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023. 2.1

[17] Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the bar of ai-generated image detection with clip. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2.1

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009. 4.1

[19] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020. 2.1

[20] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad

Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. A.1.2

[21] David C Epstein, Ishan Jain, Oliver Wang, and Richard Zhang. Online detection of ai-generated images. In *IEEE International Conference on Computer Vision (ICCV) Workshop*, 2023. 2.1, 5.1

[22] 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. In *Forty-first international conference on machine learning*, 2024. (document), 3.3, A.1.3, A.2

[23] Kristian Georgiev, Joshua Vendrow, Hadi Salman, Sung Min Park, and Aleksander Madry. The journey, not the destination: How data guides diffusion models. *arXiv preprint arXiv:2312.06205*, 2023. 2.2

[24] Getty Images. Ai generation faqs, 2024. URL https://www.gettyimages.com/ai/generation/faqs. Accessed: 2024-11-13. 1

[25] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023. 2.2

[26] Melissa Heikkilä. This artist is dominating ai-generated art. and he's not happy about it. *MIT Technology Review*, September 2022. URL https://www.technologyreview.com/2022/09/16/1059598/this-artist-is-dominating-ai-generated-art-and-hes-not-happy-about-it/. 1

[27] Yan Hong, Jianfu Zhang, Jianming Feng, Haoxing Chen, Jun Lan, Huijia Zhu, and Weiqiang Wang. Wildfake: A large-scale challenging dataset for ai-generated images detection. *arXiv preprint arXiv:2402.11843*, 2024. 2.1

[28] Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2.4

[29] Masaru Isonuma and Ivan Titov. Unlearning traces the influential training data of language models. *arXiv preprint arXiv:2401.15241*, 2024. 2.2

[30] Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhu Chen. Genai arena: An open evaluation platform for generative models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 3.3

[31] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detec-

tion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2.1

[32] Sergey Karayev, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller. Recognizing image style. *arXiv preprint arXiv:1311.3715*, 2013. 2.3

[33] Hasam Khalid, Shahroz Tariq, Minha Kim, and Simon S Woo. Fakeavceleb: A novel audio-video multimodal deepfake dataset. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2.1

[34] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 4.1, A.3.4, A.5a

[35] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Conference on Neural Information Processing Systems (NeurIPS)*, 36:36652–36663, 2023. 2.4

[36] Myeongseob Ko, Feiyang Kang, Weiyan Shi, Ming Jin, Zhou Yu, and Ruoxi Jia. The mirrored influence hypothesis: Efficient data influence estimation by harnessing forward passes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2.2

[37] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017. 2.2

[38] Anand Kumar, Jiteng Mu, and Nuno Vasconcelos. Introstyle: Training-free introspective style attribution using diffusion features. *arXiv preprint arXiv:2412.14432*, 2024. 2.3

[39] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

[40] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. (document), 3.3, A.1.3, A.2

[41] LAION.ai. Safety review for laion 5b, Dec 2023. URL https://laion.ai/notes/laion-maintenance/. Accessed on November 21, 2024. A.1.1

[42] Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *Conference on Neural Information Processing Systems (NeurIPS)*, 36:69981–70011, 2023. 2.4, 3.3

[43] Roberto Leotta, Oliver Giudice, Luca Guarnera, and Sebastiano Battiato. Not with my name! inferring artists' names of input strings employed by diffusion models. In *International Conference on Image Analysis and Processing*, pages 364–375. Springer, 2023. 2.4

[44] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2.1

[45] Zhikai Li, Xuewen Liu, Dongrong Joe Fu, Jianquan Li, Qingyi Gu, Kurt Keutzer, and Zhen Dong. K-sort arena: Efficient and reliable benchmarking for generative models via k-wise human preferences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3.3

[46] Andreas Liesenfeld and Mark Dingemanse. Rethinking open source generative ai: open-washing and the eu ai act. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 1774–1787, 2024. 3.3

[47] Chenxi Liu, Towaki Takikawa, and Alec Jacobson. A lora is worth a thousand pictures. *arXiv preprint arXiv:2412.12048*, 2024. 2.3

[48] Hui Mao, Ming Cheung, and James She. Deepart: Learning joint representations of visual arts. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1183–1191, 2017. 2.3

[49] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva. Detection of gan-generated fake images over social networks. In *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018. 2.1

[50] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2019. 2.1

[51] Louise Matsakis. Artists rage against machines that mimic their work. *Wired*, August 2023. URL `https://www.wired.com/story/artists-rage-against-machines-that-mimic-their-work/`. 1

[52] Midjourney. `https://midjourney.com`, 2024. (document), 1, 1.2, 3.1, 3.2, 3.3, 6, A.1.1, A.1.4, A.1

[53] Mazda Moayeri, Samyadeep Basu, Sriram Balasubramanian, Priyatham Kattakinda, Atoosa Chegini, Robert Brauneis, and Soheil Feizi. Rethinking copyright infringements in the era of text-to-image generative models. In *International Conference on Learning Representations (ICLR)*, 2025. 2.3

[54] Lakshmanan Nataraj, Tajuddin Manhar Mohammed, BS Manjunath, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H Bappy, and Amit K Roy-Chowdhury. Detecting gan generated fake images using co-occurrence matrices. In *Electronic*

Imaging, 2019. 2.1

[55] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2.1

[56] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 4.1

[57] Jeongsoo Park and Andrew Owens. Community forensics: Using thousands of generators to train fake image detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2.1

[58] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14532–14542, 2022. A.1.1

[59] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024. (document), 1, 1.2, 3.1, 3.3, A.1.2, A.1.4, A.1

[60] 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 (ICML)*, 2021. (document), 3.5, 4.1, 5.1, A.1.2, A.2, A.3.2, A.4

[61] 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. 2.4

[62] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. (document), 1, 1.2, 3.1, 3.3, A.1.2, A.1.4, A.1

[63] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2.1

[64] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models

for subject-driven generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510, 2023. 6

[65] Dan Ruta, Saeid Motiian, Baldo Faieta, Zhe Lin, Hailin Jin, Alex Filipkowski, Andrew Gilbert, and John Collomosse. Aladin: All layer adaptive instance normalization for fine-grained style similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2.3

[66] Matthia Sabatelli, Mike Kestemont, Walter Daelemans, and Pierre Geurts. Deep transfer learning for art classification problems. In *Proceedings Of The European conference on computer vision (ECCV) workshops*, 2018. 2.3

[67] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Conference on Neural Information Processing Systems (NeurIPS)*, 35:36479–36494, 2022. 2.4

[68] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*, 2015. 2.3

[69] Gustavo Santana. https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts, 2024. 3.1, A.1.1

[70] 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. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 3.1, 4.1, A.1.1

[71] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023. 2.1

[72] Ravidu Suien Rammuni Silva, Ahmad Lotfi, Isibor Kennedy Ihianle, Golnaz Shahtahmassebi, and Jordan J Bird. Artbrain: An explainable end-to-end toolkit for classification and attribution of ai-generated art and style. *arXiv preprint arXiv:2412.01512*, 2024. 2.4

[73] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. (document), 4.1, 5.1, 6, A.2, A.4, A.3.4, A.5a

[74] Artists Rights Society. Artists rights 101. *Artists Rights Society*, 2024. URL https://arsny.com/artists-rights-101. A.1.2

[75] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. (document), 5.1

[76] Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Investigating style similarity in diffusion models. In *European Conference on Computer Vision (ECCV)*, 2024. doi: 10.1007/978-3-031-72848-8_9. URL https://doi.org/10.1007/978-3-031-72848-8_9. (document), 2.3, 3.1, 4.1, 6, A.1.1, A.3, A.3.2, A.4

[77] Gjorgji Strezoski and Marcel Worring. Omniart: a large-scale artistic benchmark. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(4):1–21, 2018. 2.3

[78] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. (document), 1, 2.4, 3.2, 3.3, 3.4, 3.2, A.1.2, A.1.4

[79] Kailas Vodrahalli and James Zou. Artwhisperer: A dataset for characterizing human-ai interactions in artistic creations. In *International Conference on Machine Learning*, pages 49627–49654. PMLR, 2024. 2.4

[80] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. doi: 10.1109/cvpr42600.2020.00872. URL http://dx.doi.org/10.1109/cvpr42600.2020.00872. 2.1, A.2

[81] Sheng-Yu Wang, Alexei A Efros, Jun-Yan Zhu, and Richard Zhang. Evaluating data attribution for text-to-image models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. (document), 2.2, 4.1, 5.1

[82] Sheng-Yu Wang, Aaron Hertzmann, Alexei A Efros, Jun-Yan Zhu, and Richard Zhang. Data attribution for text-to-image models by unlearning synthesized images. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 2.2

[83] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 2.1

[84] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. In *The Annual Meeting of the Association for*

Computational Linguistics (ACL), 2023. 2.4

[85] Yankun Wu, Yuta Nakashima, and Noa Garcia. Not only generative art: Stable diffusion for content-style disentanglement in art analysis. In *Proceedings of the 2023 ACM International conference on multimedia retrieval*, pages 199–208, 2023. 2.4

[86] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Conference on Neural Information Processing Systems (NeurIPS)*, 36:15903–15935, 2023. 2.4

[87] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 6

[88] 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. *Transactions on Machine Learning Research*, 2022. 2.4

[89] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2.1

[90] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2019. 2.1

[91] Rui Zhao, Hangjie Yuan, Yujie Wei, Shiwei Zhang, Yuchao Gu, Lingmin Ran, Xiang Wang, Jay Zhangjie Wu, David Junhao Zhang, Yingya Zhang, et al. Evolvedirector: Approaching advanced text-to-image generation with large vision-language models. *Conference on Neural Information Processing Systems (NeurIPS)*, 37, 2024. 3.3

[92] Matthew Zheng, Enis Simsar, Hidir Yesiltepe, Federico Tombari, Joel Simon, and Pinar Yanardag. Stylebreeder: Exploring and democratizing artistic styles through text-to-image models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 2.4

[93] Xiaosen Zheng, Tianyu Pang, Chao Du, Jing Jiang, and Min Lin. Intriguing properties of data attribution on diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024. 2.2

[94] Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 2.1

[95] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 2.1