Detecting Image Attribution for Text-to-Image Diffusion Models in RGB and Beyond

Katherine Xu[1], Lingzhi Zhang[2], and Jianbo Shi[1]

[1] University of Pennsylvania
[2] Adobe Inc.

https://github.com/k8xu/ImageAttribution

Abstract. Modern text-to-image (T2I) diffusion models can generate images with remarkable realism and creativity. These advancements have sparked research in fake image detection and attribution, yet prior studies have not fully explored the practical and scientific dimensions of this task. In addition to attributing images to 12 state-of-the-art T2I generators, we provide extensive analyses on what inference stage hyperparameters and image modifications are discernible. Our experiments reveal that initialization seeds are highly detectable, along with other subtle variations in the image generation process to some extent. We further investigate what visual traces are leveraged in image attribution by perturbing high-frequency details and employing mid-level representations of image style and structure. Notably, altering high-frequency information causes only slight reductions in accuracy, and training an attributor on style representations outperforms training on RGB images. Our analyses underscore that fake images are detectable and attributable at various levels of visual granularity than previously explored.

Keywords: Generative Models · Image Attribution · Image Forensics

1 Introduction

In recent years, the emergence of advanced text-to-image (T2I) diffusion models [7, 45, 52, 54, 56, 57, 60, 62] has markedly transformed the landscape of image generation. These advancements enable the creation of highly realistic and imaginative visual content directly from textual descriptions, heralding new possibilities for creative expression and practical applications. However, this progress also introduces significant challenges in differentiating real images from AI-generated images and accurately identifying their origins. Addressing these challenges is vital for copyright enforcement, digital forensics, and maintaining the integrity of visual content across digital platforms.

Previous studies [5,8,11,39,67,71,80] have primarily focused on differentiating AI-generated images from real ones, with some research extending to the attribution of images to their source generators, notably in GAN variants [10, 29, 47, 75] and diffusion models [17, 32, 65]. Yet, these investigations have largely been conducted using generative models that may not reflect the latest advancements in the field. Moreover, these studies have not fully explored the broader, practical, and scientific dimensions of these tasks, which we aim to further examine.

As a first step, our work unifies "real vs. fake" classification and image attribution into a single task by simply treating real images as an additional category. We expand the analysis to include a comprehensive range of state-of-the-art T2I diffusion models, as of March 2024. This includes Stable Diffusion (SD) 1.5 [60], SD 2.0 [60], SDXL [54], SDXL Turbo [62], Latent Consistency Model (LCM) [45], Stable Cascade [52], Kandinsky 2.1 [57], DALL-E 2 [56], DALL-E 3 [7], along with Midjourney versions 5.2 and 6 [1]. To encompass a wide range of visual concepts, we utilize 5,000 captions from MS-COCO [42] for natural scenes and employ GPT-4 [4] to generate another 5,000 creative and surreal prompts. For each prompt, we generate multiple images from each model, amassing nearly half a million AI-generated image dataset to train our image attributor, where the details are discussed in Sec. 3. Regarding performance, our top-performing attributor reaches an accuracy exceeding 90%, significantly surpassing the baseline random chance of merely 7.69%, as detailed in Sec. 4.1.

Moving beyond previous research that focused on attributing images to their originating generators, our study probes further into whether nuanced changes in hyperparameters during the inference phase of the same T2I diffusion model can be identified. We examine hyperparameters including model checkpoints at different training iterations, scheduler types, the number of sampling steps, and initialization seeds. A significant finding from our experiments is the ability to distinguish between initialization seeds with 98%+ accuracy, employing ten unique seeds for image generation within a consistent generator framework. While the accuracy in identifying other hyperparameters doesn't reach the exceptional levels observed with initialization seeds, they all notably exceed random chance. This suggests that even subtle variations in the generation process can indeed be discerned to some extent. More details are discussed in Sec. 4.2.

In the workflow involving AI-generated images, users often enhance these images further by importing them into additional software or models for regional editing via SDXL Inpainting [54] or Photoshop Generative Fill (Ps GenFill) [2], or employing tools like Magnific AI [3] for texture enhancement at higher resolutions. This raises an essential question: Can we still trace these post-edited images back to their original generators, and to what extent is this feasible? In Sec. 4.3, we mimic user-driven regional editing using SDXL Inpainting and Ps GenFill, alongside utilizing Magnific AI on a selected group of test images. Our discussion thoroughly examines and provides insights into how these post-editing interventions impact the image attribution performance.

Lastly, while prior research has demonstrated notable success in differentiating "real vs. fake" images and accurately attributing them to their origins, the exact nature of the detectable traces recognized by classifiers and their locations within the images remain elusive. In Sec. 5, we delve deeper into this scientific question by introducing perturbations in the high-frequency domain and converting images into various mid-level representations, such as depth maps and Canny edges, to assess their impact on image attribution accuracy. This strategy aims to unearth detectable traces across different levels of visual granularity, enriching our understanding of how classifiers recognize and attribute images.

Notably, our investigations reveal that training the image attributor using style representations—specifically, the Gram matrix—enhances accuracy beyond what is achievable with attributors trained on original RGB images. Furthermore, introducing perturbations to high-frequency signals within images results in only minor performance decreases in the attributors. When these models are trained on mid-level representations, they maintain commendable accuracy levels that significantly surpass random chance. This observation suggests that detectable traces extend beyond just the high-frequency domain, encompassing mid-level aspects of texture, structure, and potentially the layout of images.

Overall, our key contributions are as follows:

- Developed an extensive dataset of nearly half a million AI-generated images from cutting-edge T2I models with a variety of natural and surreal prompts.
- Achieved over 90% accuracy in training an image attributor across 12 contemporary T2I generators and real images, significantly outperforming random chance for the 13-way classification task.
- Pioneered the exploration of detectability regarding minor hyperparameter modifications during the inference stage of T2I diffusion models.
- Innovatively replicated user editing workflows on AI-generated images using various tools, thoroughly evaluating their effect on attribution accuracy.
- Introduced a novel approach for analyzing detectable traces within images through high-frequency perturbations and conversion to diverse mid-level representations, yielding significant insights.

2 Related Work

Classifying Fake vs. Real Images. The rise of sophisticated image generators facilitates creating highly realistic images with diverse artistic styles, which has spurred research aimed at detecting synthetic images from real images. Wang *et al*. [71] introduced a CNN model that identifies images generated by GANs [30] and low-level vision models [12,19]. They showed that training diversity is crucial for fake image detectors to achieve good generalization. Additionally, Yu *et al*. [75] discovered that different GAN architectures, training sets, and initialization seeds lead to distinct fingerprints in the generated images.

Various approaches detect synthetic images using visible [48] and invisible artifacts that can lie in the spatial or frequency domain [31]. Spatial domain methods often estimate these digital fingerprints using deep learning methods [8, 67] or by averaging their noise residuals [46]. These detection methods may use local image patches [11], combine local and global image features [39], or use gradients extracted by a pretrained CNN [69]. Moreover, style and texture information have been utilized for fake image detection [5, 43, 80]. Amoroso *et al*. revealed that real and fake images are more easily separable using style features rather than semantics [5], and Zhong *et al*. [80] found it more challenging for generative models to synthesize rich texture regions. Another line of work suggests that GAN-generated images can be detected by studying artifacts in the frequency domain [6, 8, 16, 20, 22, 23, 25, 46, 59, 70].

Recently, there is a trend towards identifying images generated by diffusion models [9, 24, 34, 65, 72, 78, 81]. Wang *et al*. [72] discovered that features of

diffusion-generated images are more easily reconstructed by pretrained diffusion models than real images, and Cozzolino *et al*. [18] observed that images from diffusion models have spectral peaks that distinguish them from real images. Furthermore, the idea of learning classifiers that leverage both visual and language features to supplement low-level features has gained interest [18,50,73]. Ojha *et al*. [50] found that using the feature space of CLIP [55] improves generalization ability for detecting fake images from GANs and diffusion models.

Detecting Fake Image Attribution. In addition to recognizing synthetic images, some approaches strive to identify the source of generated images. Yu *et al*. [75] discovered that different GAN architectures, training sets, and initialization seeds can lead to fingerprint features for attribution. RepMix [10] traces GAN images to their generators while being invariant to semantic content and image perturbations. Girish *et al*. [29] and Marra *et al*. [47] developed an algorithm for online detection and attribution of GAN images. Recent work has also explored fake image detection and attribution from diffusion models [17] and T2I generation models [65]. Guarnera *et al*. [32] proposed a hierarchical approach to categorize images into real or fake, GAN or diffusion-generated, and the specific generator. Guo *et al*. [33] takes a similar approach, but they also determine whether the image was entirely synthesized or only partially edited.

Analyzing Images Generated by Diffusion Models. There have been works studying scene knowledge within pretrained diffusion models [14,21,77], as well as methods for examining the geometry of diffusion-generated images [61]. Du *et al*. [21] discovered that generative models contain rich information about scene intrinsics, and they train a low-rank adapter [36] to produce surface normals, shading, albedo, and depth. Sarkar *et al*. [61] revealed that synthetic images can be differentiated from real ones by analyzing their geometric properties.

In contrast, our work extends beyond prior research by delving deeper into the specific inference stage hyperparameters, image modifications, and levels of visual granularity that are discernible by an image attributor.

3 Dataset Generation

In this work, our objective is to detect and comprehend image attributions for contemporary text-to-image (T2I) models, while also investigating the extent to which traces can be detected across different generators and within the nuanced variations of inference stage controls. To achieve this, we first generate images using a variety of modern T2I models, employing a wide range of text prompts to ensure diversity. Subsequently, we maintain a consistent generator while adjusting inference time hyperparameters, which include the number of inference steps, scheduler types, model checkpoints, and random seeds.

3.1 Images from Diverse Generators and Prompts

As of March 2024, we have employed the following state-of-the-art, open-source T2I models for image generation: SD 1.5 [60], SD 2.0 [60], SDXL [54], SDXL Turbo [62], Latent Consistency Model (LCM) [45], Stable Cascade [52], Kandinsky 2 [57], DALL-E 2 [56], DALL-E 3 [7], along with Midjourney versions 5.2 and 6 [1]. To generate images, we use the OpenAI API for DALL-E 2 and 3,

an automation bot for Midjourney 5.2 and 6, and the Hugging Face diffusers GitHub repository [53] for the remaining models.

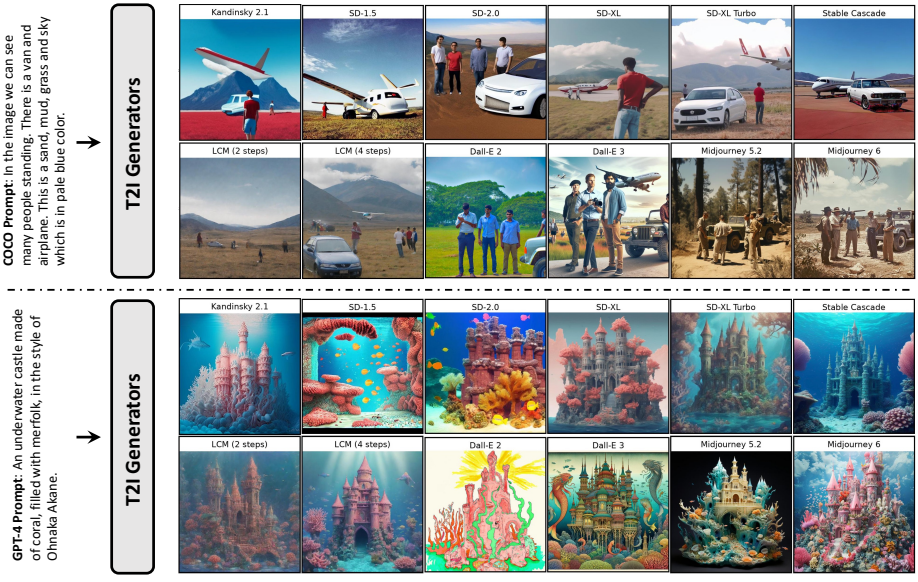

Fig. 1: A depiction of images generated for our dataset, showcasing two distinct types of prompts: MS-COCO [42] derived captions are displayed at the top, while creative prompts generated by GPT-4 are featured at the bottom. For both categories, images were produced using 12 different T2I generators.

To gather a broad spectrum of text prompts, we include both descriptions of natural scenes and imaginative, surreal prompts. This diversity is achieved by leveraging around 5,000 captions from the MS-COCO dataset [42], complemented by approximately 5,000 prompts generated by GPT-4 [4]. The GPT-4 generated prompts stem from a wide-ranging collection of popular user prompts found online, details of which are provided in the supplemental materials. Utilizing this comprehensive set of prompts, we generate images across all the text-to-image (T2I) models referenced, as depicted in Fig. 1. For each prompt, we generated one image for DALL-E 2 and 3 (due to cost considerations), four images for Midjourney, and five images for the other models, culminating in a dataset exceeding 450K generated images. It's important to note that not all images were used during training; the specifics are in the supplemental materials.

3.2 Images from Varying Hyperparameters at Inference Stage

In this research, we expand our focus beyond simply identifying the source generators based on their architectures, to a deeper analysis of the critical yet subtle choices made during the inference stage that have a profound effect on the generated outputs. Initially, we investigate the possibility of identifying specific model checkpoints within the same architecture, specifically Stable Diffusion (SD) [60], based on different training iterations. To facilitate this, we generated images using five versions of SD from 1.1 to 1.5. Despite sharing a common architecture, each version was trained for a distinct number of iterations. Next, we delve

into the impact of using different schedulers or samplers [35, 40, 68, 79] during the inference phase for the same generator. We question whether the generated images can reveal which scheduler was employed. Furthermore, drawing inspiration from studies indicating that the use of different seeds in GAN-generated images can be detected [76], we seek to apply this concept to diffusion models to determine if the choice of seed is detectable in the resulting images. Finally, we conduct experiments with diffusion steps ranging from 5 to 50 in increments of 5 to investigate whether the number of sampling steps employed can leave detectable traces in the images. Selected samples of images generated under different hyperparameter adjustments are presented in Fig. 2.

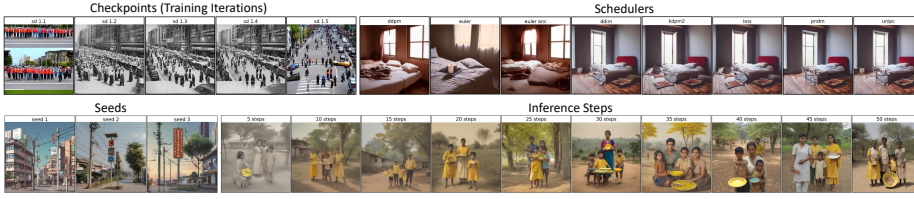

Fig. 2: An illustration showcasing the diversity in generated images influenced by varying hyperparameters: different model checkpoints (within the same architecture), diverse scheduling algorithms, varied initialization seeds, and a range of inference steps.

4 Detecting Image Attribution in RGB

In this section, we benchmark the performance of image attribution across 12 modern text-to-image generators, and we examine the impact of various architectures, training sizes, and cross-domain influences on task performance. We then delve into the detectability of traces for various hyperparameter adjustments during the inference stage. Finally, inspired by typical user workflows, we investigate whether AI-generated images can still be attributed to their original generators after being modified by distinct software or models.

4.1 Training Image Attributors

Problem Setup and Model Performance. Prior research has demonstrated deep networks' ability to distinguish AI-generated images from real ones [11, 18, 39, 46, 48, 50, 71] and to identify their sources [10, 65, 75] effectively. Our study builds on this foundation by merging the tasks of discerning "AI-generated vs. Real Images" and attributing images to their sources into a singular framework. This is achieved by including real images in our dataset and treating them as an additional 'generator', enabling a more detailed analysis of AI-generated content. Concerning the architecture of the image attributor, which functions as an image classifier, previous studies [18, 50] have demonstrated that a straightforward linear probe or nearest neighbor search, when applied to a large pretrained model like CLIP [55], can effectively differentiate AI-generated images from real ones. Inspired by these findings, we employ three network architectures to tackle the attribution task across 12 modern mainstream text-to-image (T2I) generators—such as SDXL Turbo [62], DALL-E 3 [7], and Midjourney 6—plus a real image dataset. These architectures include an EfficientFormer [41] trained from scratch,

a CLIP [55] backbone connected with a linear probe and MLP, and DINOv2 [51] with a similar configuration. We also analyze the impact of incorporating text prompts as additional inputs similar to Sha *et al.* [65], which we found to provide slight yet consistent improvements across all architectures, as shown in Tab. 1.

	E.F. (scratch)	CLIP+LP	CLIP+MLP	DINOv2+LP	DINOv2+MLP
w/o text	90.03%	70.15%	73.09%	67.68%	71.33%
w/ text	90.96%	71.44%	74.19%	69.44%	73.08%

Table 1: Quantitative evaluation (13-way classification accuracy) of various architectures for image attribution learning performed across 12 generators and a corresponding set of real images, with each category containing an equal number of images. The probability of randomly guessing the correct source is $\frac{1}{13}$, corresponding to **7.69%** accuracy. In this context, "E.F." refers to EfficientFormer. The first and second rows in the results table indicate classifiers trained without and with text prompts, respectively.

Analyzing Classifier Performance Across Generators. To provide a more granular view of our analysis, we delve into the performance specifics of each classifier, illustrating a detailed accuracy breakdown through a radar graph and a corresponding confusion matrix, as depicted in Fig. 3. Our findings reveal a noticeable challenge in differentiating between generators from the same family, with notable pairs including "SD 1.5 vs. SD 2.0," "Midjourney 5.2 vs. Midjourney 6," and "LCM (2 steps) vs. LCM (4 steps)." While Midjourney's architecture remains undisclosed to the public, it is reasonable to infer that versions 5.2 and 6 likely share a similar underlying architecture from our analysis. Interestingly, DALL-E 3 presents more confusion when compared to Midjourney versions 5.2 / 6, rather than with DALL-E 2. We attribute this to the significant architectural differences: DALL-E 2 incorporates pixel diffusion in its decoder stage, whereas DALL-E 3 employs multi-stage latent diffusion alongside a distinct one-step VAE decoder, similar to [60], leading to divergent generative characteristics. Finally, we demonstrate that the accuracy of the attributor consistently improves with an increase in the number of training images, as shown on the right side of Fig. 3. However, due to budget constraints, fully exploring the dataset expansion up to the saturation point is deferred to future research endeavors.

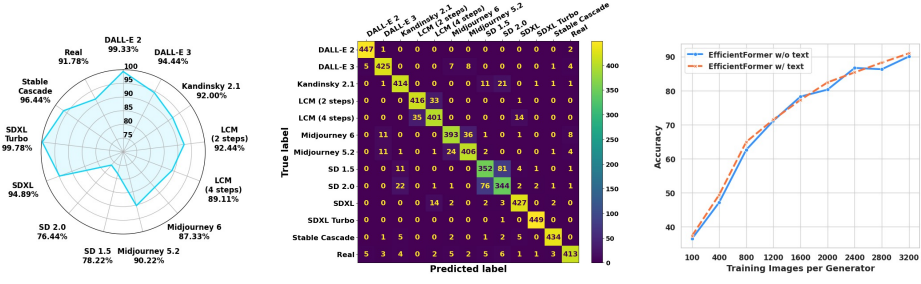

Fig. 3: Left/Middle: Accuracy and confusion matrix of EfficientFormer trained with text prompts, which achieved the highest accuracy in Table 1. **Right:** Accuracy of EfficientFormer as we vary the number of training images.

Cross-domain Generalization. As highlighted in Sec. 3.1, user prompts vary significantly, with some describing natural scenes and others depicting creative or surreal concepts. This diversity led us to examine how a classifier, trained on images generated using MS-COCO captions, would perform when applied to images created from GPT-4's inventive prompts, and conversely. The results, presented in Tab. 2, show a noticeable decline in performance when the classifier is trained and tested across these differing domains. Since we keep the same set of generators and only change the style of prompts, this outcome underscores that learning image attribution uses the visible content in the generated images.

	Train on MS-COCO	Train on GPT-4	Train on Both
Test on MS-COCO	89.04%	71.07%	85.78%
Test on GPT-4	69.24%	79.35%	81.06%

Table 2: Cross-domain generalization accuracy in image attributors. The amount of training and testing data was kept consistent across trials, and an equal number of images was sourced from MS-COCO and GPT-4 prompts for the 'Train on Both' trial.

4.2 Analyzing the Detectability of Hyperparameter Variations

T2I generators often have several adjustable hyperparameters at the inference stage that affect the generated image quality. A natural question that arises is whether images produced using different hyperparameters are distinguishable. To investigate this, we target four hyperparameter choices for Stable Diffusion [60]: model checkpoint, scheduler type, number of sampling steps, and initialization seed. Specifically, we compared Stable Diffusion checkpoints 1.1 to 1.5, each of which are trained using a different number of iterations on the LAION dataset [63]. We then examined the detectability of images generated using eight schedulers: DDIM [68], DDPM [35], Euler [40], Euler with ancestral sampling [40], KDPM 2 [40], LMS [40], PNDM [40], and UniPC [79]. Additionally, we generated images using both SD 2.0 and SDXL for ten different sampling steps ranging from 5 to 50, and ten different seeds ranging from 1 to 10. For each hyperparameter choice, we train a separate EfficientFormer [41] to classify the generated images, and the results are illustrated in Tab. 3 and Fig. 4. As shown in Tab. 3, all six classifiers can detect the hyperparameter choice better than random chance. Interestingly, the initialization seed achieves nearly 100% accuracy, which aligns with prior work by Yu *et al.* [75] that found different seeds lead to attributable GAN fingerprints. Moreover, when looking at the confusion matrix for different sampling steps using SDXL in Fig. 4, we see that images generated using fewer steps are more detectable than those generated using more steps, likely because fewer steps noticeably degrades the generation quality.

4.3 Assessing Detectability of Post-Editing Enhancements

A common workflow for utilizing AI-generated images involves users identifying unwanted artifacts or distracting areas within these images. They often import these images into additional models or software for further editing and refinement, such as SDXL Inpainting [54] or Photoshop Generative Fill (Ps Gen-

	Checkpoints	Schedulers	Sampling Steps	Seeds
Random Chance	20%	12.5%	10%/10%	10%/10%
Accuracy	30.21%	20.18%	25.96%/56.64%	98.80%/99.94%

Table 3: Comparison of accuracy for detecting hyperparameter values based on generated images. For the 'Sampling Steps' and 'Seeds' trials, we trained and evaluated on images from SD 2.0 and SDXL, and the accuracies are formatted as *SD 2.0 / SDXL*. Notably, the 'Seeds' trial attains near perfect performance.

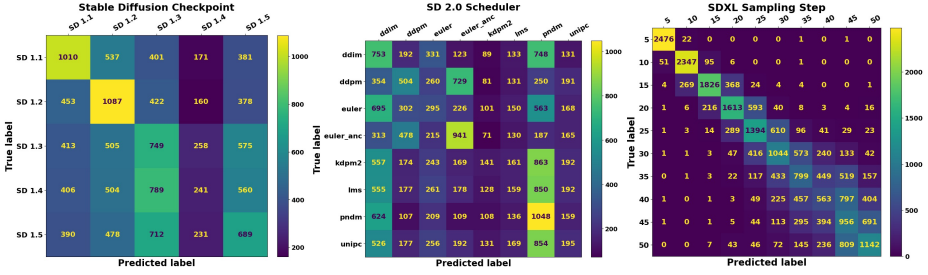

Fig. 4: Confusion matrices for different hyperparameter adjustments, including the Stable Diffusion version, scheduler type, and number of inference steps. We observe that images generated with fewer SDXL sampling steps are more detectable, likely due to visible degradation in the image quality.

Fill) [2], to enhance local regions. Many text-to-image applications are constrained to relatively low resolutions, typically around 1K, or produce images with smooth/blurry texture. Consequently, some professionals opt to upscale or refine the details of these generated images using advanced tools, such as Magnific AI [3]. This practice leads to a pertinent question: Is it possible to still detect the original source generator after the images have undergone further modifications using a variety of software or other AI models? For instance, an image initially created by Midjourney 6 [1] could subsequently be edited with SDXL Inpainting, Photoshop GenFill, or Magnific AI, as illustrated in Fig. 5.

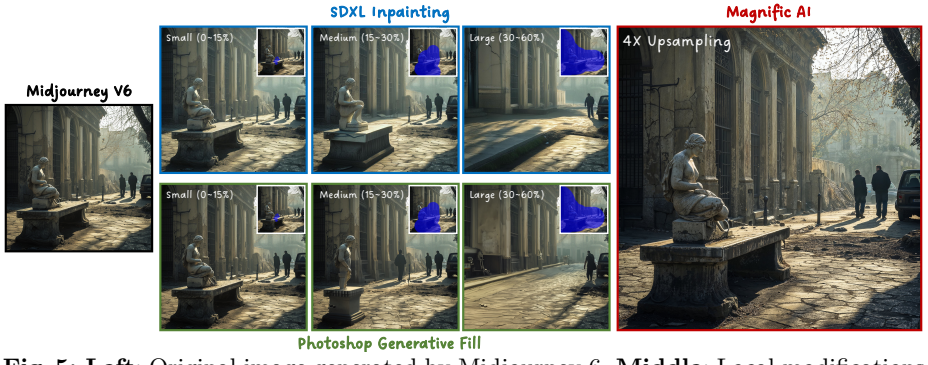

Fig. 5: Left: Original image generated by Midjourney 6. **Middle**: Local modifications utilizing SDXL inpainting and Photoshop Generative Fill across three masks with small, medium, and large holes. **Right**: The image upscaled 4X by Magnific AI.

To simulate typical user edits, we generated free-form masks across three size categories—small (0 to 15%), medium (15 to 30%), and large (30 to 60%)—re-

	SDXL Inpainting			Ps Generative Fill			Magnific AI
Edit Region Ratio	0 ∼ 15%	15 ∼ 30%	30 ∼ 60%	0 ∼ 15%	15 ∼ 30%	30 ∼ 60%	100%
Random Chance	7.69%	7.69%	7.69%	7.69%	7.69%	7.69%	33.33%
Original Image	90.96%	90.96%	90.96%	90.96%	90.96%	90.96%	93.33%
Post-Editing	64.96%	61.56%	55.62%	88.21%	85.44%	71.91%	70.00%

Table 4: Comparison of post-editing detection accuracy across different AI models. We use the best performing image attributor in Table 1 for evaluation, which is EfficientFormer trained with text prompts. Accuracy declines at a similar rate after modifying the image using SDXL Inpainting [54] and Photoshop (Ps) Generative Fill [2].

flecting the common range of edits applied to images. These masks were applied to the entire test set for pixel regeneration using SDXL Inpainting [54] and Ps GenFill [2]. We then assessed the best performing image attributor in Tab. 1, EfficientFormer trained with text prompts, on these post-edited images. According to Tab. 4, we observed a monotonic decrease in accuracy with respect to the modified area of the images. Notably, SDXL Inpainting resulted in greater accuracy loss compared to Ps GenFill for the same images and masks. We hypothesize this disparity arises because the SDXL Inpainting model closely relates to the SDXL text-to-image (T2I) model included in our training generator pool, potentially skewing edited images towards an SDXL-like appearance, whereas Ps GenFill does not closely resemble any generator in our training set. This observation is validated in the corresponding confusion matrix, which we have shared in the supplemental materials. For texture enhancements via Magnific AI [3], budget constraints limited our examination to 10 examples from each of the three generators: DALL-E 3, Midjourney 6, and SDXL Turbo. This limitation set a basic random chance of classification at 33.33%. This analysis, reflected in the last column of Tab. 4, shows approximately 23% degradation, despite editing all pixels in the images. Despite the noted performance reductions, the accuracy for all post-edited images remains significantly above random chance, establishing a strong baseline for the task of post-editing image attribution.

5 Detecting Image Attribution Beyond RGB

Previous studies have demonstrated that training a standard deep network can effectively distinguish between real and generated images, as well as correctly attribute generated images to their original generators. In Sec. 4.1, we observed that a lightweight transformer achieves high accuracy for these tasks, mirroring these findings. These prior studies have suggested that the attributor may leverage middle-to-high frequency information to differentiate images. However, it remains unclear what exactly constitutes this "middle-to-high frequency information" and to what extent the network can still identify detectable traces in the images as we incrementally remove visual details. Therefore, this section presents an extensive empirical study on the impact of progressively eliminating visual information at various levels of granularity on image attribution performance.

High-Frequency Perturbation. Prior research [6,8,16,20,22,23,25,46,59,70] has identified that generators leave unique fingerprints in the high-frequency domain, allowing attributors to learn these high-frequency details and achieve high performance. As an initial step, we investigate the effects of introducing high-frequency perturbations to images on the attributor's performance, which

aims to enforce the classifier learn beyond high-frequency details. For simplicity, we train a separate EfficientFormer [41] on each set of perturbed images. Figure 6 illustrates our observations under four types of perturbation: Gaussian blur, bilateral filtering, adding Gaussian noise, and SDEdit [49]. We note that these perturbations result in a modest decrease in classification accuracy. Specifically for SDEdit, the high-frequency traits of SDXL are embedded into every image, regardless of their source generators, by undergoing processing via the encoder, diffusion UNet, and decoder of SDXL [54]. Remarkably, this process led to only a minor reduction in accuracy, suggesting a robustness in the attributor's ability to identify generator-specific fingerprints despite high-frequency modifications.

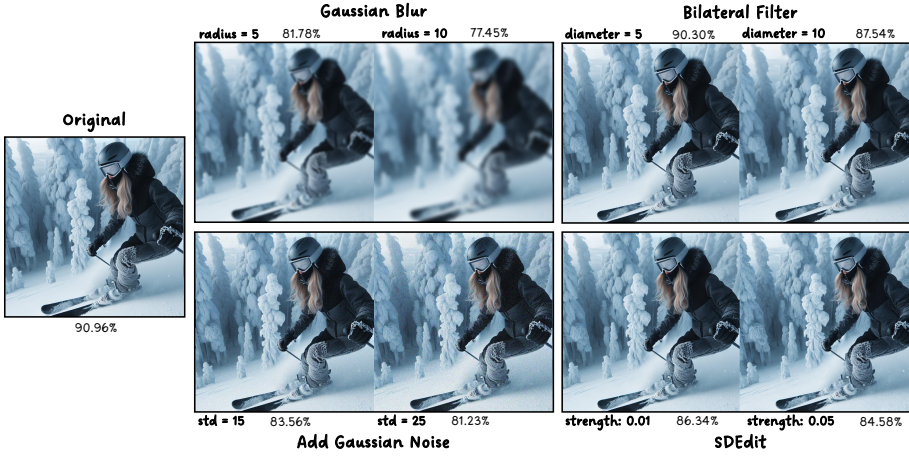

Fig. 6: We showcase a generated image before and after perturbing its high-frequency details via Gaussian blurring, bilateral filtering, adding Gaussian noise, and SDEdit [49]. We trained EfficientFormer on images after each high-frequency perturbation and observed a mild decline in the respective test accuracy, as indicated beside the images.

Middle-Level Representation. High-frequency perturbations result in only minor performance degradation, which suggests that the detectable traces left by different generators might also reside within the mid-frequency domain. To delve deeper into the presence of these detectable traces, we convert the images into various mid-level representations. These include 'Albedo,' [21] 'Shading,' [21] 'Canny Edge,' 'Depth Map,' [74] 'Surface Normal,' [21] and 'Perspective Fields,' [37] utilizing readily available models for the transformations. This approach aims to uncover the extent to which these mid-level frequencies carry generator-specific information that can be leveraged for attribution. We proceed by training a distinct EfficientFormer [41] for each mid-level representation, and we show their classification accuracies in Fig. 7 and confusion matrices in Fig. 8. Notably, although the overall accuracy for the attributors trained on Canny Edge, Depth Map, and Perspective Field images is not high in Fig. 7, they demonstrate remarkable performance at discerning real images from fake images in Fig. 8. This finding aligns with previous work by Sarkar *et al.* [61] suggesting that generative models often fail to generate accurate geometry.

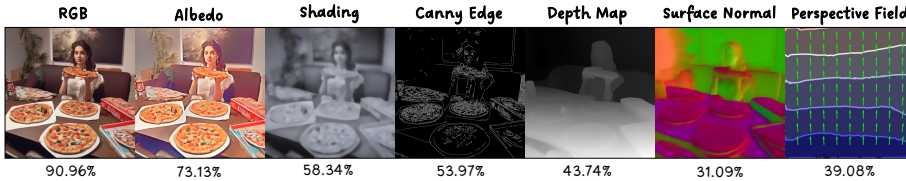

Fig. 7: We present an RGB image and its mid-level representations. We trained EfficientFormer on each mid-level representation and include the corresponding test accuracy under each image. Please keep in mind that the random chance is $\frac{1}{13}$ or **7.69%**.

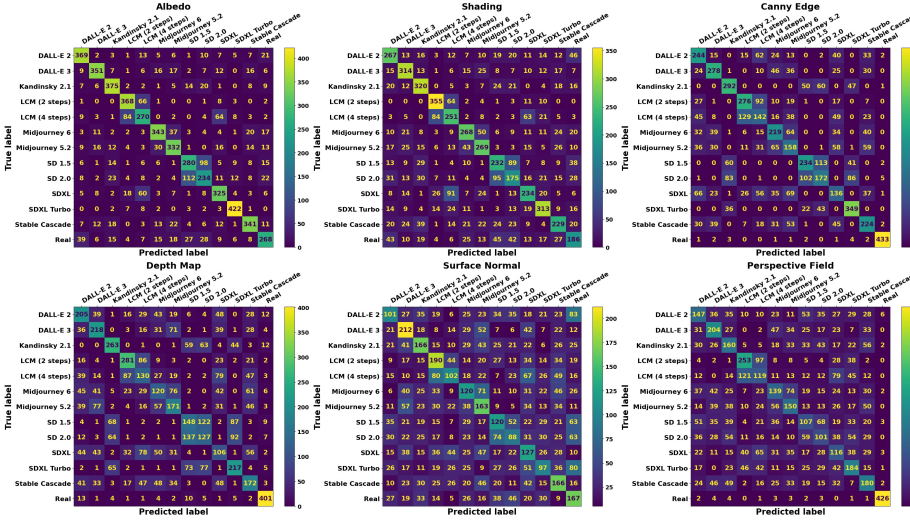

Fig. 8: Confusion matrices for image attributors trained on mid-level representations. Remarkably, attributors trained on "Canny Edge," "Depth Map," and "Perspective Field" images are significantly better at detecting real images than synthetic images.

Image Style Representation. Given the perceptible differences in styles or tones among image generators, it's common to observe distinct characteristics in their outputs. For instance, Midjourney [1] often produces images with a 'cinematic' quality, while DALL-E [7,56] sometimes tends to create images with overly smooth textures and cartoonish appearances, as shown in Fig. 1. This observation leads to a pertinent question: if we train an attributor solely on the stylistic representations of images, can we still identify the source generators?

To capture the style representation of images, we adhere to the methodology established in prior style transfer literature [27,38], employing a pretrained VGG network [66] to extract features across multiple layers. Subsequently, we calculate the Gram matrix [26] for each layer of the network. If we denote the feature at a specific layer as $F \in \mathbb{R}^{H \times W \times N}$, then the Gram matrix is the cosine similarity between each channel in the feature representation, yielding a matrix of dimensions $G \in \mathbb{R}^{N \times N}$. This process aims to distill the stylistic essence of images, providing a unique fingerprint for each generator's output. Specifically, we reshape and concatenate the Gram matrices extracted from multiple layers, and then train EfficientFormer [41] using these aggregated feature vectors.

Remarkably, the image attributor achieves an accuracy of **92.80%** when trained on style representations, surpassing the performance of the attributor trained on original RGB images by **1.84%**. The superior accuracy achieved by this style-based image attributor highlights the critical role of stylistic features, such as texture and color patterns, in distinguishing generators more effectively than traditional RGB data. This suggests that the unique signatures of image generators might be more intricately tied to their style rather than the direct visual content. This insight not only advances our understanding of image attribution techniques but also emphasizes the potential of leveraging stylistic elements for more nuanced AI recognition and analysis tasks.

Furthermore, given the exceptional performance of training on style features for image attribution, we seek to understand what insights we can extract from raw values in the Gram matrices without any model training. To achieve this, we average the Gram matrix from a single layer of VGG across 450 images per generator, and we visualize the density distribution of its values in Fig. 9. We observe that LCM (2 steps) and LCM (4 steps) have similar image style distributions, as does SDXL and SDXL Turbo. Additionally, since we use real images from the MS-COCO dataset [42], the generators with distributions closer to that of real images in Fig. 9 likely generate more natural image styles.

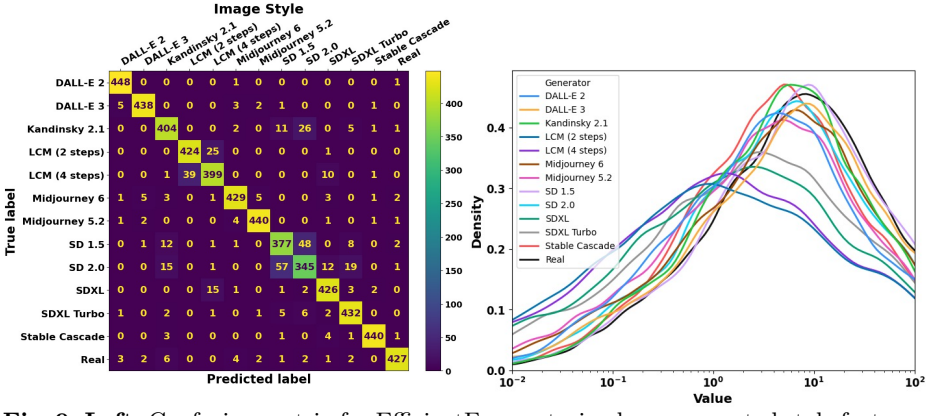

Fig. 9: Left: Confusion matrix for EfficientFormer trained on aggregated style features obtained from Gram matrices. Compared to EfficientFormer trained on original RGB images in Figure 3, we observe that training on image style reduces misclassification between generators of the same family, such as "Midjourney 5.2 vs. Midjourney 6." **Right:** Density distribution of values in the averaged Gram matrix (log-scaled) using 450 images per generator. We include real images as a distinct 'generator'. Image style is moderately distinguishable across generators by analyzing Gram matrices alone.

Image Composition Pattern. Beyond stylistic differences, we hypothesize that various generators might create images with unique composition patterns or layouts from the same text prompt. For instance, given identical prompts, some generators may depict humans in portrait-style photos, while others may place humans further from the camera, treating them as elements within the larger scene. These variations could stem from each generator's learning with its distinctively 'curated' training data distribution and proprietary prompt aug-

mentation techniques, features that are often integral to commercial models like DALL-E [7, 56] and Midjourney [1]. To test our hypothesis, we analyze 100 images generated from the same prompt for each generator. We employ Grounded SAM [58] to compute segmentation masks, serving as a proxy for layout representation. For instance, as depicted in Figure 10, by averaging the segmentation masks for 'person' and 'corgi' across 100 images from each generator, created from the prompt 'a couple, a daughter, and a corgi walking,' we visualize the distribution of image composition. This reveals unique layout patterns among the generators, supporting our hypothesis.

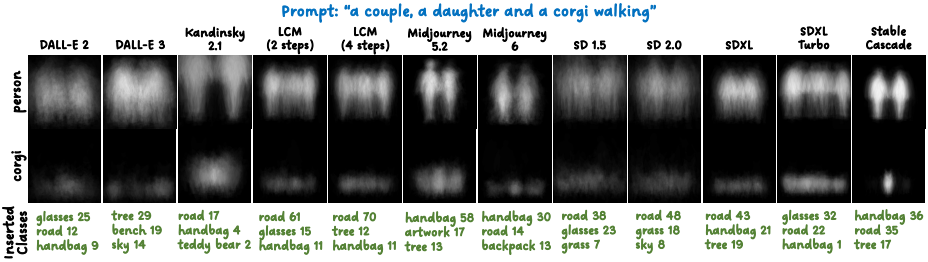

Fig. 10: Image composition analysis across generators for a single prompt. We visualize the averaged segmentation masks for the 'person' and 'corgi' classes. Some generators, such as Stable Cascade, tend to produce objects at specific locations. We also list the top three inserted classes and the number of images (out of 100) with these classes.

Given the noticeable variations in the layout of generated images for a specific prompt, we further investigate whether a classifier can learn to attribute images based solely on their composition. To this end, we segment 111 semantic classes using Grounded SAM [58] and subsequently train EfficientFormer [41] on the segmentation maps with their input prompts by concatenating their respective embeddings. This approach enables the classifier to achieve an accuracy of 17.66%, despite relying on such a coarse representation. Remarkably, this accuracy is more than twice that expected by random chance (7.69%), suggesting that distinct patterns in layout generation do indeed exist across these generators.

6 Conclusion

In this study, we present in-depth analyses on the detection and attribution of images generated by contemporary text-to-image (T2I) diffusion models. Through rigorous testing, our image attributors, trained to recognize outputs from 12 different T2I diffusion models along with a category for real images, reached an impressive accuracy of over 90%, significantly surpassing random chance. Our investigation into the role of text prompts, the challenge of distinguishing generators within the same family, and the ability to generalize across domains provides comprehensive insights. Pioneeringly, we delved into the detectability of hyperparameter adjustments at inference time and assessed the effects of post-editing on attribution accuracy. Going beyond mere RGB analysis, we introduce a new framework for identifying detectable traces across various levels of visual detail, offering profound insights into the underlying mechanics of image attribution. These analyses provide fresh perspectives on image forensics aimed at alleviating the threat of synthetic images on copyright protection and digital forgery.

References

1. https://www.midjourney.com/ 2, 4, 9, 12, 14, 20, 23, 24
2. https://www.adobe.com/products/photoshop/generative-fill.html 2, 9, 10, 22
3. https://magnific.ai/ 2, 9, 10, 22
4. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 2, 5, 20
5. Amoroso, R., Morelli, D., Cornia, M., Baraldi, L., Del Bimbo, A., Cucchiara, R.: Parents and children: Distinguishing multimodal deepfakes from natural images. arXiv preprint arXiv:2304.00500 (2023) 1, 3
6. Bammey, Q.: Synthbuster: Towards detection of diffusion model generated images. IEEE Open Journal of Signal Processing **PP**, 1–9 (01 2023). https://doi.org/10.1109/OJSP.2023.3337714 3, 10
7. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al.: Improving image generation with better captions. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf **2**(3), 8 (2023) 1, 2, 4, 6, 12, 14, 20
8. Bi, X., Liu, B., Yang, F., Xiao, B., Li, W., Huang, G., Cosman, P.C.: Detecting generated images by real images only. arXiv preprint arXiv:2311.00962 (2023) 1, 3, 10
9. Bird, J.J., Lotfi, A.: Cifake: Image classification and explainable identification of ai-generated synthetic images. IEEE Access (2024) 3
10. Bui, T., Yu, N., Collomosse, J.: Repmix: Representation mixing for robust attribution of synthesized images. In: European Conference on Computer Vision. pp. 146–163. Springer (2022) 1, 4, 6
11. Chai, L., Bau, D., Lim, S.N., Isola, P.: What makes fake images detectable? understanding properties that generalize. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16. pp. 103–120. Springer (2020) 1, 3, 6
12. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3291–3300 (2018) 3
13. Chen, J., Yao, J., Niu, L.: A single simple patch is all you need for ai-generated image detection. arXiv preprint arXiv:2402.01123 (2024) 24
14. Chen, Y., Viégas, F., Wattenberg, M.: Beyond surface statistics: Scene representations in a latent diffusion model. arXiv preprint arXiv:2306.05720 (2023) 4
15. Contributors, M.: Openmmlab's pre-training toolbox and benchmark. https://github.com/open-mmlab/mmpretrain (2023) 22
16. Corvi, R., Cozzolino, D., Poggi, G., Nagano, K., Verdoliva, L.: Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 973–982 (2023) 3, 10
17. Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., Verdoliva, L.: On the detection of synthetic images generated by diffusion models. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023) 1, 4
18. Cozzolino, D., Poggi, G., Corvi, R., Nießner, M., Verdoliva, L.: Raising the bar of ai-generated image detection with clip. arXiv preprint arXiv:2312.00195 (2023) 4, 6

19. Dai, T., Cai, J., Zhang, Y., Xia, S.T., Zhang, L.: Second-order attention network for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11065–11074 (2019) 3
20. Dong, C., Kumar, A., Liu, E.: Think twice before detecting gan-generated fake images from their spectral domain imprints. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7855–7864 (2022). https://doi.org/10.1109/CVPR52688.2022.00771 3, 10
21. Du, X., Kolkin, N., Shakhnarovich, G., Bhattad, A.: Generative models: What do they know? do they know things? let's find out! arXiv preprint arXiv:2311.17137 (2023) 4, 11
22. Durall, R., Keuper, M., Keuper, J.: Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7890–7899 (2020) 3, 10
23. Dzanic, T., Shah, K., Witherden, F.: Fourier spectrum discrepancies in deep network generated images. Advances in neural information processing systems **33**, 3022–3032 (2020) 3, 10
24. Epstein, D.C., Jain, I., Wang, O., Zhang, R.: Online detection of ai-generated images. In: ICCV DeepFake Analysis and Detection Workshop (2023) 3
25. Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. In: International conference on machine learning. pp. 3247–3258. PMLR (2020) 3, 10
26. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. Advances in neural information processing systems **28** (2015) 12
27. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2414–2423 (2016). https://doi.org/10.1109/CVPR.2016.265 12
28. Gildenblat, J., contributors: Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam (2021) 25, 30
29. Girish, S., Suri, S., Rambhatla, S.S., Shrivastava, A.: Towards discovery and attribution of open-world gan generated images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14094–14103 (2021) 1, 4
30. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014) 3
31. Gragnaniello, D., Cozzolino, D., Marra, F., Poggi, G., Verdoliva, L.: Are gan generated images easy to detect? a critical analysis of the state-of-the-art. In: 2021 IEEE international conference on multimedia and expo (ICME). pp. 1–6. IEEE (2021) 3
32. Guarnera, L., Giudice, O., Battiato, S.: Level up the deepfake detection: a method to effectively discriminate images generated by gan architectures and diffusion models. arXiv preprint arXiv:2303.00608 (2023) 1, 4
33. Guo, X., Liu, X., Ren, Z., Grosz, S., Masi, I., Liu, X.: Hierarchical fine-grained image forgery detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3155–3165 (2023) 4
34. Ha, A.Y.J., Passananti, J., Bhaskar, R., Shan, S., Southen, R., Zheng, H., Zhao, B.Y.: Organic or diffused: Can we distinguish human art from ai-generated images? arXiv preprint arXiv:2402.03214 (2024) 3
35. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 6, 8

36. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021) 4
37. Jin, L., Zhang, J., Hold-Geoffroy, Y., Wang, O., Matzen, K., Sticha, M., Fouhey, D.F.: Perspective fields for single image camera calibration. CVPR (2023) 11, 22
38. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 694–711. Springer (2016) 12
39. Ju, Y., Jia, S., Ke, L., Xue, H., Nagano, K., Lyu, S.: Fusing global and local features for generalized ai-synthesized image detection. In: 2022 IEEE International Conference on Image Processing (ICIP). pp. 3465–3469. IEEE (2022) 1, 3, 6
40. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022) 6, 8
41. Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., Ren, J.: Efficientformer: Vision transformers at mobilenet speed. Advances in Neural Information Processing Systems **35**, 12934–12949 (2022) 6, 8, 11, 12, 14, 22, 24, 25
42. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014) 2, 5, 13, 21, 30
43. Liu, Z., Qi, X., Torr, P.H.: Global texture enhancement for fake face detection in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8060–8069 (2020) 3
44. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 22
45. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023) 1, 2, 4, 20
46. Marra, F., Gragnaniello, D., Verdoliva, L., Poggi, G.: Do gans leave artificial fingerprints? In: 2019 IEEE conference on multimedia information processing and retrieval (MIPR). pp. 506–511. IEEE (2019) 3, 6, 10
47. Marra, F., Saltori, C., Boato, G., Verdoliva, L.: Incremental learning for the detection and classification of gan-generated images. In: 2019 IEEE international workshop on information forensics and security (WIFS). pp. 1–6. IEEE (2019) 1, 4
48. Matern, F., Riess, C., Stamminger, M.: Exploiting visual artifacts to expose deepfakes and face manipulations. In: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). pp. 83–92 (2019). https://doi.org/10.1109/WACVW.2019.00020 3, 6
49. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021) 11
50. Ojha, U., Li, Y., Lee, Y.J.: Towards universal fake image detectors that generalize across generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24480–24489 (2023) 4, 6
51. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust

visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) 7, 22, 23, 24
52. Pernias, P., Rampas, D., Aubreville, M.: Wuerstchen: Efficient pretraining of text-to-image models. arXiv preprint arXiv:2306.00637 (2023) 1, 2, 4, 20, 23, 24
53. von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Wolf, T.: Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers (2022) 5
54. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023) 1, 2, 4, 8, 10, 11, 20, 22
55. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 4, 6, 7, 22, 23, 24
56. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022) 1, 2, 4, 12, 14, 20
57. Razzhigaev, A., Shakhmatov, A., Maltseva, A., Arkhipkin, V., Pavlov, I., Ryabov, I., Kuts, A., Panchenko, A., Kuznetsov, A., Dimitrov, D.: Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion. arXiv preprint arXiv:2310.03502 (2023) 1, 2, 4, 20, 23, 24
58. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., et al.: Grounded sam: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159 (2024) 14
59. Ricker, J., Damm, S., Holz, T., Fischer, A.: Towards the detection of diffusion model deepfakes. arXiv preprint arXiv:2210.14571 (2022) 3, 10
60. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 1, 2, 4, 5, 7, 8, 20
61. Sarkar, A., Mai, H., Mahapatra, A., Lazebnik, S., Forsyth, D.A., Bhattad, A.: Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. arXiv preprint arXiv:2311.17138 (2023) 4, 11
62. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. arXiv preprint arXiv:2311.17042 (2023) 1, 2, 4, 6, 20, 30
63. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022) 8
64. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision **128**(2), 336–359 (Oct 2019). https://doi.org/10.1007/s11263-019-01228-7, http://dx.doi.org/10.1007/s11263-019-01228-7 25, 30
65. Sha, Z., Li, Z., Yu, N., Zhang, Y.: 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. pp. 3418–3432 (2023) 1, 3, 4, 6, 7
66. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 12

67. Sinitsa, S., Fried, O.: Deep image fingerprint: Accurate and low budget synthetic image detector. arXiv preprint arXiv:2303.10762 (2023) 1, 3
68. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 6, 8
69. Tan, C., Zhao, Y., Wei, S., Gu, G., Wei, Y.: Learning on gradients: Generalized artifacts representation for gan-generated images detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12105–12114 (2023) 3
70. Tian, C., Luo, Z., Shi, G., Li, S.: Frequency-aware attentional feature fusion for deepfake detection. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5 (2023). https://doi.org/10.1109/ICASSP49357.2023.10094654 3, 10
71. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8695–8704 (2020) 1, 3, 6
72. Wang, Z., Bao, J., Zhou, W., Wang, W., Hu, H., Chen, H., Li, H.: Dire for diffusion-generated image detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22445–22455 (2023) 3
73. Wu, H., Zhou, J., Zhang, S.: Generalizable synthetic image detection via language-guided contrastive learning. arXiv preprint arXiv:2305.13800 (2023) 4
74. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: CVPR (2024) 11
75. Yu, N., Davis, L.S., Fritz, M.: Attributing fake images to gans: Learning and analyzing gan fingerprints. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7556–7566 (2019) 1, 3, 4, 6, 8
76. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In: Proceedings of the IEEE/CVF International conference on computer vision. pp. 14448–14457 (2021) 6
77. Zhan, G., Zheng, C., Xie, W., Zisserman, A.: What does stable diffusion know about the 3d scene? arXiv preprint arXiv:2310.06836 (2023) 4
78. Zhang, Y., Xu, X.: Diffusion noise feature: Accurate and fast generated image detection. arXiv preprint arXiv:2312.02625 (2023) 3
79. Zhao, W., Bai, L., Rao, Y., Zhou, J., Lu, J.: Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. Advances in Neural Information Processing Systems **36** (2024) 6, 8
80. Zhong, N., Xu, Y., Qian, Z., Zhang, X.: Rich and poor texture contrast: A simple yet effective approach for ai-generated image detection. arXiv preprint arXiv:2311.12397 (2023) 1, 3, 24
81. Zhu, M., Chen, H., Huang, M., Li, W., Hu, H., Hu, J., Wang, Y.: Gendet: Towards good generalizations for ai-generated image detection. arXiv preprint arXiv:2312.08880 (2023) 3

A Human Performance

In computer vision and machine learning, human performance is typically seen as the benchmark for AI models. However, in the case of image attribution, the scenario reverses—AI significantly outperforms humans. This is highlighted by an experiment conducted by one of our co-authors, who has extensive experience with AI-generated images. Tasked with attributing 650 images to their correct source generators, the co-author achieved only a **37.23%** accuracy rate. This figure, while better than the 7.69% random chance level, is markedly inferior to the accuracy of our top AI classifier, which has 90%+ accuracy. This outcome underlines the exceptional challenge of image attribution, where even well-informed individuals struggle. It showcases the necessity of AI in assisting humans with tasks that are beyond their natural proficiency, emphasizing AI's potential to enhance human performance in specialized domains.

From the perspective of the human evaluator, differentiating between certain AI image generators and others can be nuanced yet discernible. The Latent Consistency Models (LCM) [45], at 2 and 4 steps, are notable for their occasional oversmooth artifacts, a result of undersampling, making them easier to identify compared to other models. DALL-E 3 [7] is distinguished by its tendency to produce surreal, cartoonish images, though these often exhibit repetitive patterns. DALL-E 2 [56], on the other hand, is characterized by a unique 'sharp' visual artifact, likely a consequence of its pixel diffusion process in the decoder, setting it apart from other models. Midjourney versions 5.2 and 6 [1] typically deliver the highest quality images, sometimes with a distinctive cinematic style.

Real images, however, are generally more straightforward to identify. One can often look at the detailed object regions—like hands and text—where AI-generated images tend to falter. The naturalistic photo style of real images also serves as a key differentiation factor from AI-generated content. Other generators, such as SD 1.5 [60], SD 2.0 [60], SDXL [54], SDXL Turbo [62], Kandinsky 2.1 [57], and Stable Cascade [52], present a greater challenge for human evaluators to distinguish due to the subtlety of their differences.

B Data and Implementation Details

GPT-4 Generated Prompts. Building upon Section 3 of our main paper, this section delves into the methodology behind generating creative and surreal prompts using GPT-4 [4]. As illustrated in Fig. 11, our process begins with the formulation of system-level instructions directing GPT-4 to act as an assistant for writing text prompts. We then supply a specific context and a collection of several hundred exemplary prompts. This setup enables GPT-4 to synthesize and generate new, innovative prompts based on the provided examples and context.

Image Generation. We employed 12 T2I diffusion models to generate RGB images without watermarks, and the generated image sizes are as follows:

- **512 × 512:** Kandinsky 2.1, SD 1.1, SD 1.2, SD 1.3, SD 1.4, SD 1.5, SD 2.0, SDXL Turbo
- **1024 × 1024:** DALL-E 2, DALL-E 3, LCM (2 steps), LCM (4 steps), Midjourney 5.2, Midjourney 6, SDXL, Stable Cascade

System: You are an AI assistant to help generate image text prompts.

User: I am trying to generate thousands of novel prompts for AI to generate images using tools, such as Midjourney, Stable Diffusion, and DALL-E. You can use your prior knowledge abour popular prompts, and I am also providing you a few hundred examplar prompts in below:

Cat looking at it and 'on top of the world, cyberpunk poster, in the style of Erik Jones, social media portraiture, sparth, vibrant color fields, graffiti and street art, highly detailed figures, hyper-detailed
……

Could you learn from these prompts and generate 30 more prompts that are sufficiently diverse and novel covering various content and styles.

GPT 4

Response:
A surreal depiction of a city skyline turning into a wave, in the style of Salvador Dali
A cybernetic eagle soaring over a futuristic cityscape, rendered in the style of Moebius
An old oak tree sheltering a boy, a dog and a horse, reflecting Johanna Basford's style
A steampunk table setting, complete with cogs and gears, in a dingy Victorian room, Max Ernst style
The Milky Way stretched across a desert landscape, in vivid neon colors, reflecting the style of Brandon Woelfel
An armored dinosaur in a dramatic rainforest battle, inspired by classic paleoart
An underwater castle made of coral, filled with merfolk, in the style of Ohnaka Akane
A miniaturized New York City, with tiny taxis and pedestrians, mimicking Tatsuya Tanaka's miniature calendar
A fiery phoenix rising from a cityscape, reflecting the style of Simon Stålenhag's retro futuristic scenes
A regal king standing heroically on a dragon's head, detailed in the style of Yoshitaka Amano
A romantic scene of two people kissing under cherry blossom trees, digital oil painting style of Leonid Afremov
A dark space station inhabited with multi-species aliens living harmoniously, vibrant color tones and intricate details, H.R Giger's surrealism style
A demon and an angel playing chess on the edge of a cliff during a storm, in the style of Francisco Goya
An 8k ultra-detailed mural of graffiti covering an entire city block, reflecting the style of Banksy
A hyper-detailed print of a wise owl reading a spellbook, in the style of Brian Froud's fantasies
……

Fig. 11: An illustration of how we use the GPT-4 API to massively generate thousands of creative and surreal prompts.

We also use 5000 real images from the MS-COCO [42] 2017 validation set.

More Visualizations of Hyperparameter Variations. As an extension of Fig. 2 in the main paper, we show more generations by hyperparameter variations in Fig. 12.

Fig. 12: More examples showcasing the diversity in generated images influenced by varying hyperparameters: different model checkpoints within the same architecture, diverse scheduling algorithms, varied initialization seeds, and a range of inference steps.

Training Data. For Sec. 4.1 and 5, we view image attribution as a 13-way classification task with 12 text-to-image diffusion models and 1 set of real images. An exception is the cross-domain generalization study, where we exclude real

images as a 13^{th} class because there are no real images for the GPT-4 generated prompts. We use 3200 training, 450 validation, and 450 testing images per class.

For Sec. 4.2, we analyze four hyperparameters: Stable Diffusion checkpoint, scheduler type, number of sampling steps, and initialization seed. When training classifiers for SD checkpoints, schedulers, and sampling steps, we use 20000 training, 2500 validation, and 2500 testing images per class. For seeds, we use 3200 training, 450 validation, and 450 testing images per class.

For Sec. 4.3, we run inference using the EfficientFormer [41] trained with text prompts from Sec. 4.1. For SDXL Inpainting [54] and Photoshop Generative Fill [2], we use 450 images from each of the 13 classes. For Magnific AI [3], we use 10 images from each of DALL-E 3, Midjourney 6, and SDXL Turbo.

Data Augmentation. During training, we first resize each image to have a shorter edge of size 224 using bicubic interpolation, then center crop the image to size 224×224, and finally randomly flip the image horizontally with probability 0.5. During validation and testing, we only resize and center crop the images.

Image Attributors. We selected three network architectures for the image attribution task, and we use the code implementation from MMPretrain [15]. Our primary architecture is EfficientFormer-L3 [41] trained from scratch because it is a lightweight transformer. Moreover, we employ a pretrained, frozen transformer backbone attached to a linear probe (LP) or multilayer perceptron (MLP). The backbone is either CLIP ViT-B/16 [55] or DINOv2 ViT-L/14 [51], and the MLP consists of three linear layers with sigmoid activation and hidden dimension 256. For the linear probe and MLP classifier heads, there are 768 channels in the input feature map for CLIP+LP and CLIP+MLP, and 1024 channels for DINOv2+LP and DINOv2+MLP.

To train image attributors with text prompts, we compute text embeddings using a pretrained CLIP [55] text encoder. Then, we concatenate image embeddings from the backbone with text embeddings as input to the classifier head.

For all image attributors, we set a batch size of 128 and train for 2000 epochs. We use the checkpoint with the best validation accuracy. Additionally, we utilize the AdamW optimizer [44] with learning rate 0.0002 and weight decay 0.05. The learning rate scheduler has a linear warm-up period of 20 epochs, followed by a cosine annealing schedule with a minimum learning rate of 0.00001.

Perspective Fields. We use the code implementation from [37]. Each input to the attributor trained on Perspective Fields has a size of $640 \times 640 \times 3$. The first 640×640 channel contains latitude values, and the next two 640×640 channels contain gravity values. We adapt the code from [37] to visualize the Perspective Field on a black image in Fig. 5 of the main paper.

C Additional Experiments

C.1 Color Analysis

In addition to studying image style and image composition pattern, we examine whether different generators produce images with distinct color schemes. We use 100 images generated from a set of fixed prompts for our analysis. In Fig. 13, we visualize the density distribution of pixel values in each RGB color channel. We

discover that Kandinsky 2.1 [57], Midjourney 5.2 [1], and Stable Cascade [52] often generate images with a wider range of pixel intensity values. In Fig. 14, we observe that these three generators often create images with glow and shadow effects, which can lead to higher and lower intensities.

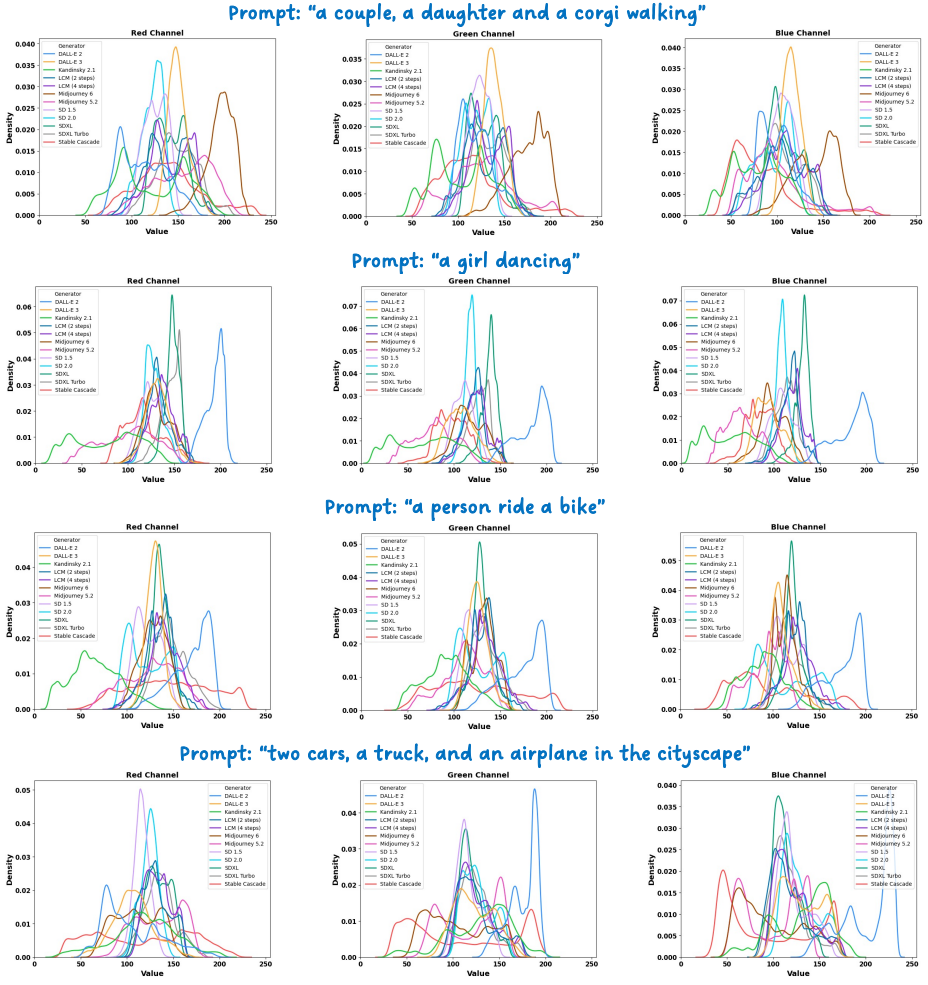

Fig. 13: Density distribution of pixel values in RGB color channels after averaging 100 images for each prompt and generator. Kandinsky 2.1 [57], Midjourney 5.2 [1], and Stable Cascade [52] tend to create images covering a wider range of pixel intensities.

C.2 Frozen vs. Fine-tuned CLIP/DINOv2 Backbone

In Sec. 4.1 of the main paper, we evaluated the accuracy of a frozen CLIP [55] backbone connected with a linear probe and MLP, and a frozen DINOv2 [51] backbone with a similar configuration. In this section, we compare using a frozen and fine-tuned backbone for the CLIP and DINOv2 linear probes. Table 5 indicates that a CLIP backbone provides marginally better performance than

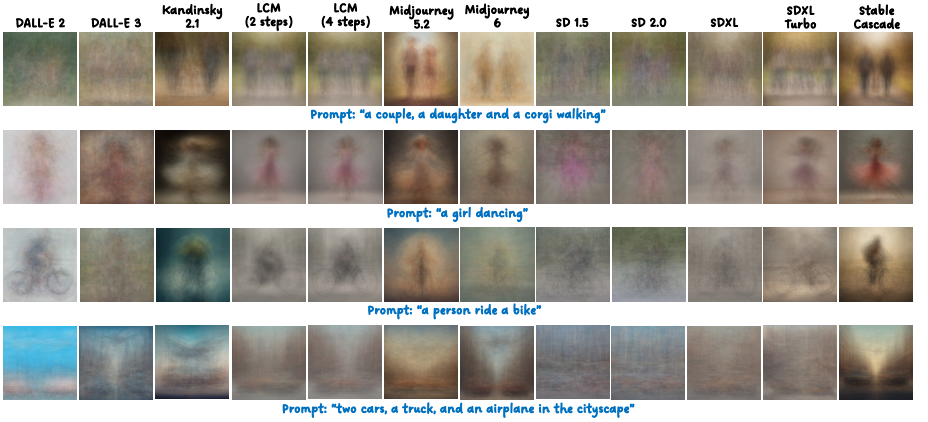

Fig. 14: Visualization of 100 images averaged together for each prompt and generator. Consistent with our observations in Fig. 13, we see that Kandinsky 2.1 [57], Midjourney 5.2 [1], and Stable Cascade [52] often produce images with glow and shadow effects.

a DINOv2 backbone when the backbone is frozen. However, the reverse holds true when the backbone is fine-tuned.

	CLIP + LP		DINOv2 + LP	
Backbone	Frozen	Fine-tuned	Frozen	Fine-tuned
Accuracy	70.15%	95.31%	67.68%	96.67%
Precision	69.95%	95.51%	67.36%	96.71%
Recall	70.15%	95.32%	67.68%	96.67%
F1	70.00%	95.34%	67.45%	96.67%

Table 5: Quantitative comparison of using a frozen or fine-tuned backbone to train CLIP [55] and DINOv2 [51] linear probes. CLIP achieves higher accuracy than DINOv2 when the backbone is frozen, but the opposite is true when the backbone is fine-tuned.

C.3 Image Resolutions

The default EfficientFormer [41] takes inputs of size 224×224. We examine the performance of using five additional image resolutions between 128×128 and 1024×1024 for the image attribution task. As illustrated on the left side of Fig. 15, accuracy tends to increase as image resolution increases.

C.4 Cropped Image Patches

Our previous experiments use most, if not all, image pixels for the image attribution task. We also explore the opposite: how few pixels are necessary to achieve good performance? Inspired by [13,80], we crop a single patch of each image and then train EfficientFormer [41] on these patches instead of the full-sized images. Specifically, we first resize each original image to have a shorter edge of size 512, then center crop the image to create a patch of size $k \times k$, and finally resize the patch to 224×224. We utilized $k = [2, 4, 8, 16, 32, 64, 128, 256]$ and resized images using bicubic interpolation.

On the right side of Fig. 15, we see that accuracy increases with image patch size. Remarkably, even training an image attributor on 2×2 patches can lead to 22.29% accuracy, which is well above the random chance accuracy of 7.69%.

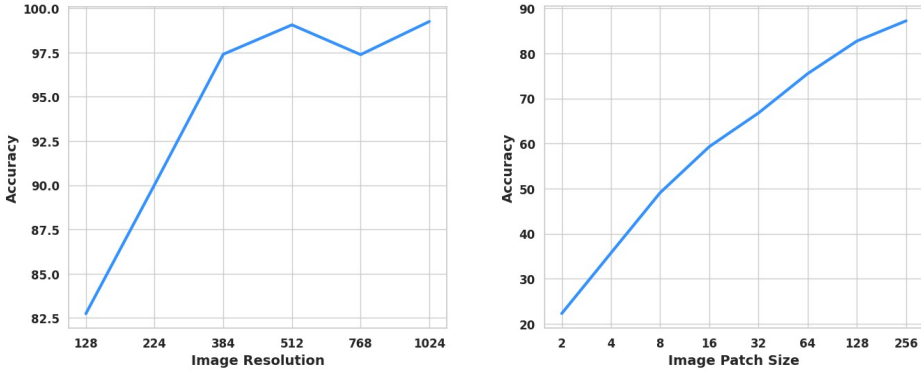

Fig. 15: Left: Accuracy of our EfficientFormer [41] image attributor across six image resolutions on the 13-way classification task. In general, accuracy increases as image resolution increases. **Right:** Accuracy of EfficientFormer across eight image patch sizes. Interestingly, using 2×2 image patches can achieve 22.29% accuracy, whereas the probability of randomly guessing the correct generator is $\frac{1}{13}$, corresponding to 7.69%.

D Elaboration on Results in the Main Paper

In this section, we expand upon the results from the experiments performed in the main paper. Figure 16 and Table 6 showcase the confusion matrices and evaluation metrics for the image attributors in Sec. 4.1. Furthermore, Figure 17 and Table 7 present the confusion matrices and evaluation metrics for the cross-domain generalization study in Sec. 4.1. Additionally, Figure 18 illustrates the confusion matrices for post-editing enhancements in Sec. 4.3. Lastly, Figure 19 visualizes the averaged segmentation masks across generators for two additional prompts, which is an extension of our image composition analysis in Sec. 5.

	E.F. (scratch)	CLIP+LP	CLIP+MLP	DINOv2+LP	DINOv2+MLP
Accuracy	90.03/90.96	70.15/71.44	73.09/74.19	67.68/69.44	71.33/73.08
Precision	90.07/90.98	69.95/71.30	73.13/74.12	67.36/69.09	71.20/72.91
Recall	90.03/90.96	70.15/71.44	73.09/74.19	67.68/69.44	71.33/73.08
F1	90.04/90.96	70.00/71.25	73.07/74.12	67.45/69.17	71.23/72.93

Table 6: Additional quantitative evaluation of image attributors for 13-way classification, consisting of 12 generators and a set of real images. The values (percentages) represent training each attributor *Without / With* text prompts.

E Grad-CAM Visualizations

Figure 20 showcases the Grad-CAM [28, 64] heatmaps for image attributors trained on various image types, including the original RGB images, images after high-frequency perturbations, and mid-level representations. We observe that the image attributors trained on RGB images and images after high-frequency perturbations tend to pay attention to smooth image regions, such as the sky or ground. Nonetheless, even though the attributors focus on varied image regions, it remains difficult to explain how they make their decisions for each image.

	Train on MS-COCO	Train on GPT-4	Train on Both
Accuracy	89.04/69.24	71.07/79.35	85.78/81.06
Precision	89.07/70.38	71.81/79.29	85.88/80.87
Recall	89.04/69.24	71.07/79.35	85.78/81.06
F1	88.99/68.44	71.06/79.21	85.78/80.86

Table 7: Cross-domain generalization in image attributors. The amount of training and testing data was kept consistent across trials, and an equal number of images was sourced from MS-COCO and GPT-4 prompts for the 'Train on Both' trial. The values (percentages) represent testing on images from *MS-COCO / GPT-4* prompts.

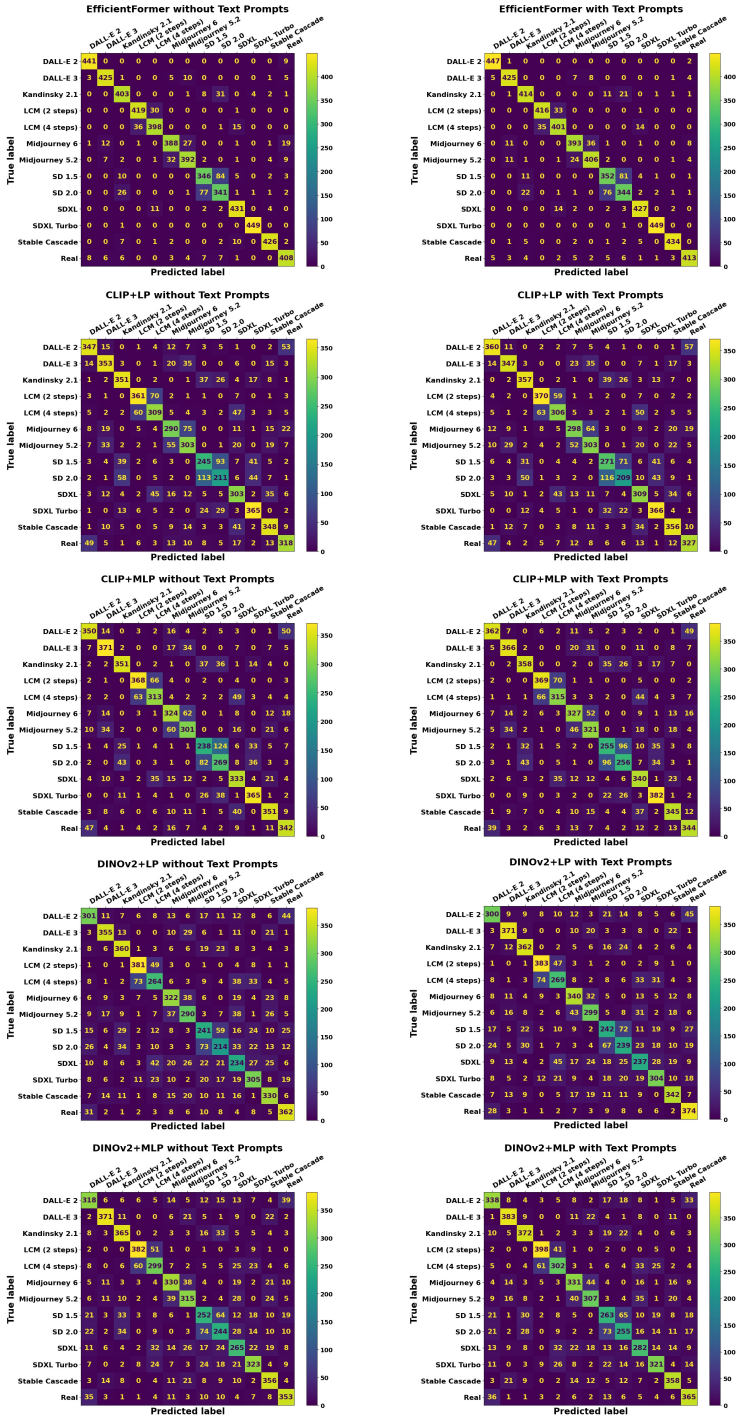

Fig. 16: Confusion matrices for image attributors in Sec. 4.1. It's important to note that the backbone for the CLIP and DINOv2 models are frozen.

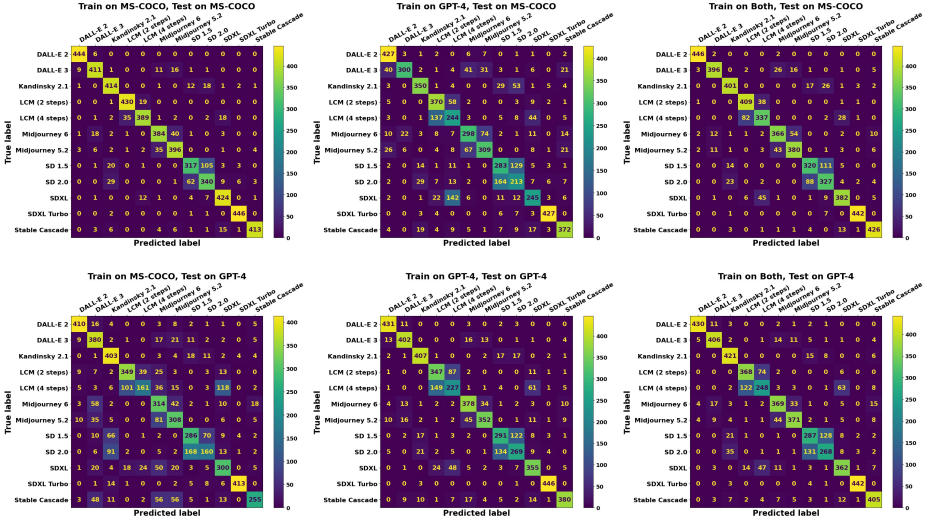

Fig. 17: Confusion matrices for cross-domain generalization in Sec. 4.1.

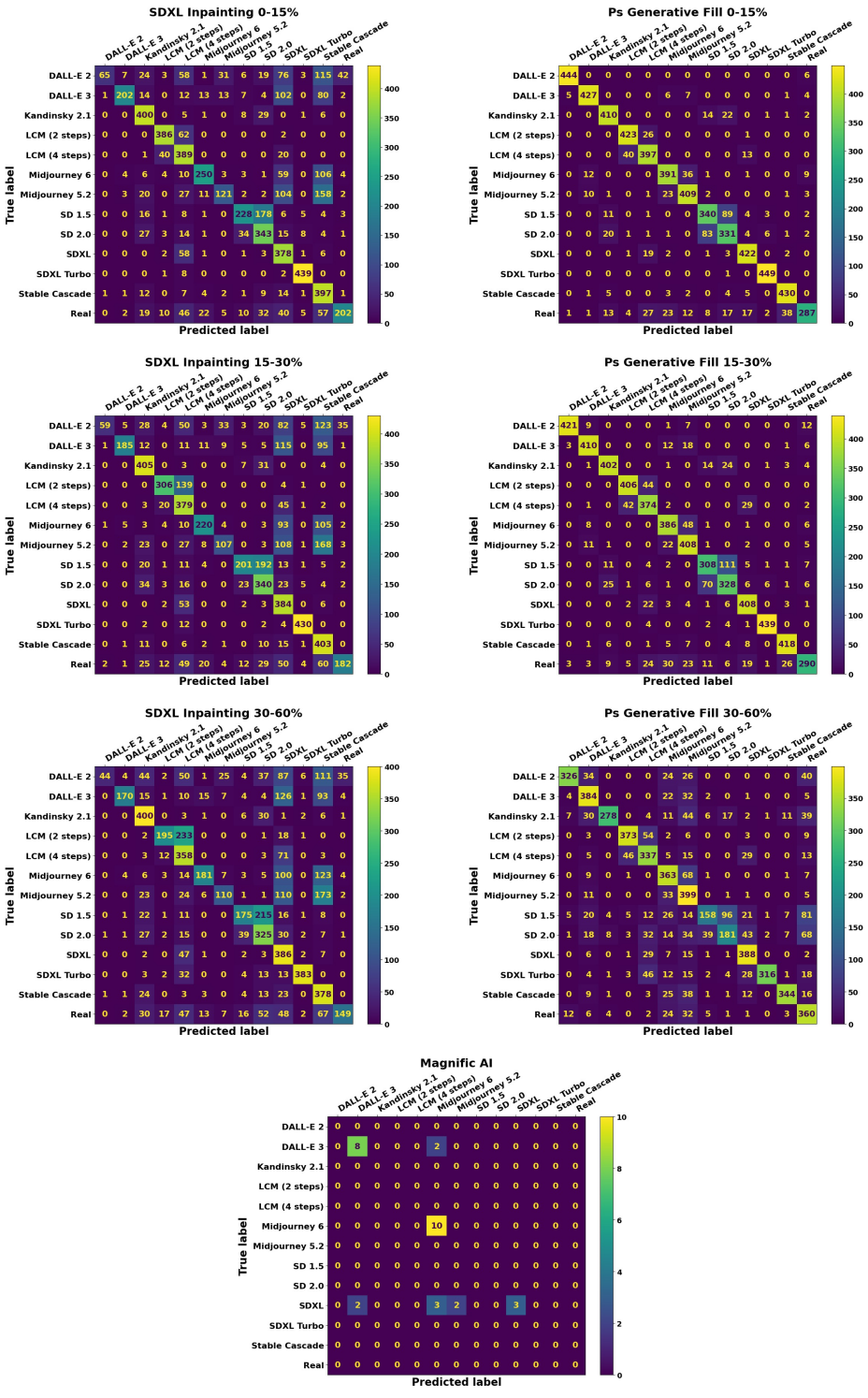

Fig. 18: Confusion matrices for evaluating on post-edited images in Sec. 4.3.

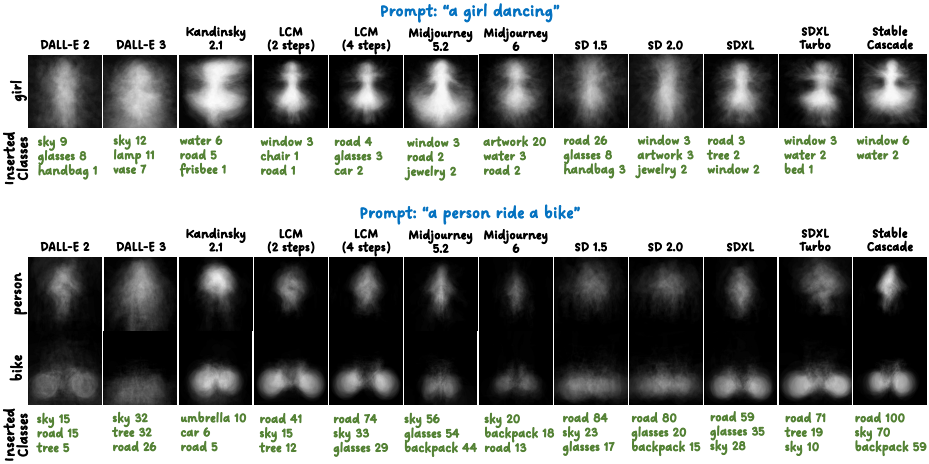

Fig. 19: Additional image composition analyses across generators. We show the averaged segmentation masks for each semantic class indicated on the left side. We also list the top three inserted classes and the number of images (out of 100) with these classes.

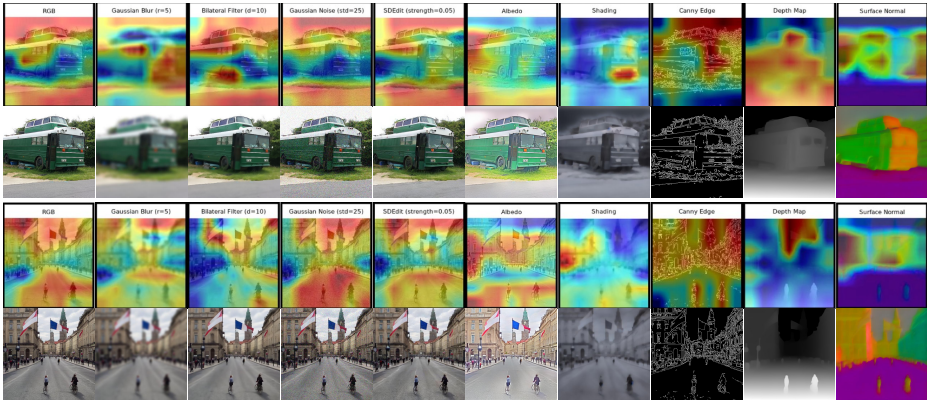

Fig. 20: Grad-CAM [28,64] visualizations for image attributors trained on each image type, where each column represents a distinct attributor. The first and third rows illustrate the Grad-CAM heatmaps overlaid on the input images. The second and fourth rows show the input images without Grad-CAM. The first example on the top is based on a real image from MS-COCO [42], while the second example on the bottom is based on a fake image generated by SDXL Turbo [62]. We notice that the attributors trained on RGB images and images after high-frequency perturbations often focus on relatively smooth image regions, such as the sky or ground.