Online Detection of AI-Generated Images

David C. Epstein   Ishan Jain   Oliver Wang   Richard Zhang
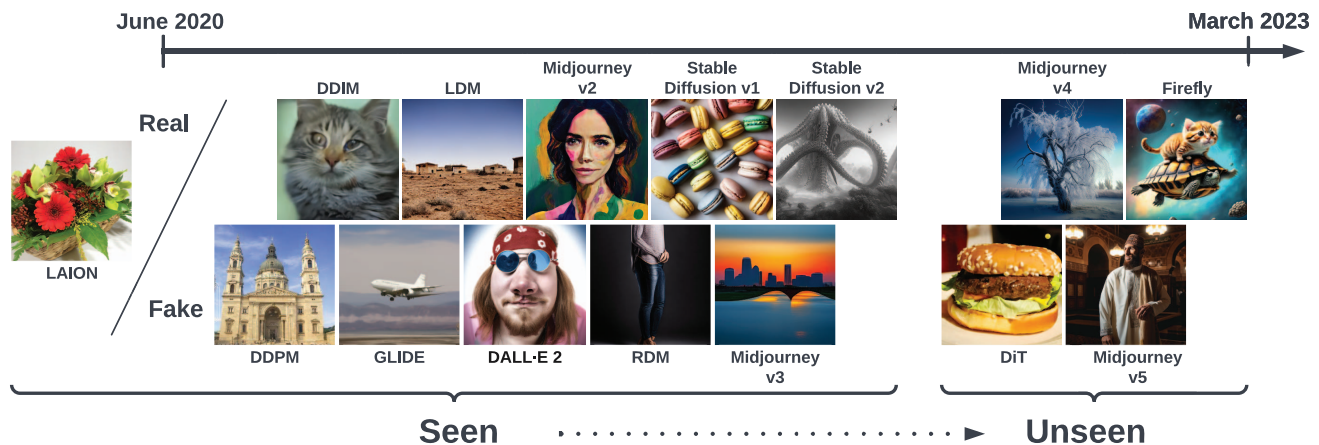
Adobe Inc.

Figure 1: New AI-generated image models are released regularly. We train classifiers that distinguish real images from generated ones in an *online* setting, whereby we add models to training in a simulated release order. We evaluate on *unseen* model releases to see how well detectors might generalize into the future.

Abstract

With advancements in AI-generated images coming on a continuous basis, it is increasingly difficult to distinguish traditionally-sourced images (e.g., photos, artwork) from AI-generated ones. Previous detection methods study the generalization from a single generator to another in isolation. However, in reality, new generators are released on a streaming basis. We study generalization in this setting, training on N models and testing on the next $(N + k)$, following the historical release dates of well-known generation methods. Furthermore, images increasingly consist of both real and generated components, for example through image inpainting. Thus, we extend this approach to pixel prediction, demonstrating strong performance using automatically-generated inpainted data. In addition, for settings where commercial models are not publicly available for automatic data generation, we evaluate if pixel detectors can be trained solely on whole synthetic images.

1. Introduction

Since the advent of photography, image manipulation has been a source of misinformation and propaganda. In the digital age, tools to modify and distribute images have grown in effectiveness and scope. The emergence of AI-generated images has amplified these risks, democratizing the capacity to create realistic synthetic images indistinguishable from authentic images by human observers. Developing mechanisms to detect AI-generated images is critical to help address these concerns.

Recent advances in image generation have increased the quality of synthetic outputs [37, 41, 60, 38], with new models and model variants released on a regular basis. This fast pace of development highlights *future generalization* as a necessary component for an effective generative image detector. Previous methods [57, 12] studying synthetic image detection and generalization suggest that different methods contain related, identifiable artifacts, showing that training a detector on *one* generator generalizes to *another*.

The true setting that we live in is online, with new methods coming out at a regular cadence. While recent meth-

ods are driven by diffusion [48, 22, 49], many contain a fusion of elements, such as autoregressive components [60], GAN-based upsampling or decoding [38, 24, 20], or perceptual losses [38, 62]. In addition, the amount of publicly available information for methods varies by model, from full open-source models with code and weights available to closed-source industry-based models, with only images available. Can the *sum total* of available information on today's generators be used to detect tomorrow's models?

To answer this, we study detection and generalization emulating an *online* setting. We collect a dataset of 14 well-known generative models and simulate a real-world learning setting by training incremental CNN detectors, preserving the historical order by model release date. We evaluate detector performance on images from both previously seen and unseen generative models, giving insight into the real-world performance of a live detector system.

In line with previous works, we find that current generative models continue to have exploitable cues that can be reliably detected when seen in the training set. A single classifier can even detect multiple generators simultaneously. Encouragingly, performance on *unseen* models also increases, as the seen model history grows. This suggests that online training methodologies can exploit a collection of historical synthetic images to generalize to future unseen generative AI models.

In addition, as generative models start serving as the backbone of image *editing* pipelines, images are increasingly a mixture of captured (real) and generated pixels. One common method for generating such composites is by using generative models to "inpaint", or generate the pixels within the confines of a mask, replacing the original pixels behind it. We extend our approach from the detection of wholly generated images to pixel-level predictions to detect such images and study generalization across Stable Diffusion [55, 7] versions and Adobe Firefly [1].

We first investigate if synthetically generated inpainted images can be used to train a pixel-level detector. Next, we recognize that a challenge in pixel-level detection is the variation in available training data (e.g., ground truth masks), especially with closed models where we cannot synthetically generate inpainted images. For such situations, we show that one can leverage whole images to create pixel-wise predictions by using CutMix augmentation [61].

In summary our contributions are:

- We study detection of AI-generated images in an *online* setting, utilizing 570,221 images from 14 generative methods.

- We study pixel-level prediction for inpainting applications, showing that CutMix augmentation improves performance in the absence of pixel-wise training data.

2. Related work

Generative modeling. Generative models aim to model the distribution of data, given a set of samples. Early attempts, utilizing deep networks, include restricted Boltzmann machines (RBMs) [47] and deep Boltzmann machines (DBMs) [42]. Recent approaches include variational autoencoders (VAEs) [28], autoregressive models [53, 52], normalizing flows [16], generative adversarial networks (GANs) [20], and diffusion models [48, 22, 49]. The ProGAN/StyleGAN family [25, 26] has demonstrate photorealistic results, mostly focused on single-class generation. Such methods prompted the exploration of forensics techniques to detect synthetic imagery. Recently, diffusion models have shown breakthrough results on arbitrary text-to-image generation, with multiple methods, such as Stable Diffusion (based on LDMs) [38], DALL·E 2 [37], Imagen [41], released in quick succession. With the capability to now seemingly generate "everything and anything", the ability to distinguish real from synthetic has become increasingly challenging.

Detecting generated images. Media forensics on synthetic images from traditional tools has a long history [18, 54], for example leveraging signals such as resampling artifacts [35], JPEG quantization [5], and shadows [27], and detecting operations such as image splicing [63, 23] or Photoshop warps [56]. As deep generative methods have democratized synthesis of arbitrary image content, recent work has explored the ability of deep discriminative methods to detect such content [57, 12], primarily in the context of GAN-based techniques [25, 26, 10].

A key question is how well detectors generalize to methods not seen during training. Wang et al. [57] find that a simple classifier, trained on one GAN model, can generalize to others, especially when using aggressive augmentations. Chai et al. [12] demonstrate that even small patches contain sufficient cues for detection. Other features, for example, focusing on frequency cues [19, 29], using co-occurrence matrices [31], or even pretrained CLIP [36] features, and techniques such as augmentation with representation mixing [11] are also effective. A common failure case when generalizing to a new generator is that while average precision is high, accuracy is low, demonstrating effective separation between real and fake classes but poor calibration. Ojha et al. [33] demonstrate that a simple nearest neighbors classifier improve accuracy, though at the cost of inference time. We build on the general observation that even baseline classifiers can generalize across generators, by studying and characterizing their behavior in an online setting.

Do images from recent diffusion methods still contain detectable cues? Recent work [15, 14, 45] show that although GAN-based detectors do not generalize to diffusion methods [15], diffusion models are detectable and exhibit

Generation architecture	Method/Dataset	Training set	Method availability		Ordering		Dataset size		
			Paper	Open-src	#	Date	Train	Val	Test
Real images	LAION-400M [44]	–	–	–	–	–	179,900	22,479	22,490
Diffusion U-net	Denoising Diffusion Prob. Model (DDPM) [22]	LSUN [59]	✓	✓	1	Jun 20	6,271	784	785
	Denoising Diffusion Implicit Model (DDIM) [49]	LSUN [59]	✓	✓	2	May 21	8,000	1,000	1,000
	GLIDE [32]	Private	✓	✓	3	Dec 21	7,442	929	931
	DALL·E 2 [37]	Private	✓	✗	5	Apr 22	2,000	954	2,000
Diffusion +Decoder	Latent Diffusion (LDM) [38]	LAION-400M [44]	✓	✓	4	Dec 21	8,172	1,021	1,022
	Retrieval-Augmented Diffusion (RDM) [9]	LAION-400M [44]	✓	✓	7	Jul 22	8,528	1,066	1,066
	Stable Diffusion v1.1-v1.4 [38, 55]	LAION-2B [43]	✓	✓	8	Aug 22	34,508	3,807	3,838
	Stable Diffusion 2.0(-v), 2.1(-v) [38, 7]	LAION-5B [43]	✓	✓	10	Nov 22	35,997	4,000	4,000
Diffusion U-Vit	Diffusion w/ Transformers (DiT) [34]	ImageNet [40]	✓	✓	11	Dec 22	3,199	400	401
Unknown (product release)	Midjourney v2 [3]	Unknown	✗	✗	6	Jul 22	42,875	5,358	5,359
	Midjourney v3 [3]	Unknown	✓	✓	9	Nov 22	70,035	8,754	8,755
	Midjourney v4 [3]	Unknown	✗	✗	12	Feb 23	100,000	10,000	76,122
	Midjourney v5 [3]	Unknown	✗	✗	13	Mar 23	63,310	7,914	7,918
	Adobe Firefly [1]	Unknown	✗	✗	14	Mar 23	15,525	2,070	3,105

Table 1: **Online dataset.** We gather recent generative models. We show the methods, grouped by their image generation method. For our training experiments, we order them chronologically, recreating an online training scenario from recent history. As many of the generated methods are general text-to-image generators, we use a subset of LAION [44] to represent real images. Note that for some methods (Dall·E 2, Midjourney, Firefly), the models are not open-sourced and only accessible through web interfaces, with no official architectural details available for some (Midjourney, Firefly).

some generalizability to each other. We take these studies further by training a detector on 14 methods in an online fashion, simulating their release dates, and releasing an accompanying dataset of 570k images.

While these works detect *whole* images, local prediction also offers important use-cases. For example, forensics methods have targeted edits from traditional tools, such as Photoshop warping [56] and image splicing [63, 23]. Chai et al. [12] show that patch-based classifiers can generate heatmaps for regions that contain more detectable cues. In this work, we investigate if inpainted regions can be localized. We show that even without direct access to inpainted examples, by using CutMix augmentation [61], whole images can be leveraged to produce pixel predictions.

3. Online detection of AI-generated images

3.1. Dataset

To begin, we collect generated images from 14 models, released between June 2020 and March 2023, shown in Table 1. These models reflect the rapid pace of advancement in realistic synthetic image generation, including academic papers [22, 49, 32, 38, 34] as well as company releases [37, 3, 1, 55, 7]. As the architecture of the generative model plays a large role in what features might be detectable, we group approaches by model architecture.

This dataset is useful to evaluate the generalization of detectors to unseen generative models. We use these images and the corresponding release dates of the source models to simulate an online learning setting. For publicly available models, we use the model release time from the repository. For products that can only be queried with an API (Midjourney [3] and Firefly [1]), as the release date and details exact

deployed version is not known, we use the date we query the model. In total, our dataset is composed of 570,221 images (405,862 train, 48,057 val, 116,302 test).

Diffusion with U-Nets First, we collect pixel diffusion methods, starting with DDPM [22], DDIM [49] and GLIDE [32]. These methods are accompanied by papers and source code releases (a smaller model version, in the case of GLIDE). All of them use a U-Net [39] with a diffusion-based objective [48] as their architecture. DDPM and DDIM train unconditional models on smaller datasets, while GLIDE performs text-to-image generation.

Additionally, we include DALL·E 2 [37], a high-profile model from OpenAI. During data collection time, the model was only available through web interface, which makes collecting a large-scale dataset challenging. Instead, we scrapped the DALL·E 2 Reddit, keeping images of 1024×1024 (to filter out extraneous content, such as memes). Because the web interface generates a watermark that would be easily identifiable by a classifier, we crop out the bottom 16 pixels of the image.

Diffusion with other architectures. The next largest family of methods are latent diffusion models (LDMs), first introduced by Rombach et al. [38] and subsequently used with retrieval-based augmentation [9]. These methods use a U-Net to perform diffusion in a latent domain, and then decode the latent signal with a decoder, trained as part of a variational autoencoder [28], in combination with a GAN [20] and LPIPS perceptual loss [62]. LDMs were subsequently popularized by the release of Stable Diffusion [55] and Stable Diffusion 2 [7], a scaled-up version of Latent Diffusion trained on large scale web data, containing multiple subversions. Additionally, several methods

propose to change the diffusion U-Nets with transformers based on ViT [17], which have been shown to have advantages in discriminative tasks [34, 8].

Product releases. With the success of image generation models, companies have released proprietary products. Such models make for interesting test cases, as one can speculate that they contain common elements from publicly available models, but such details are not publically disclosed. We sample images from Midjourney [3] and Adobe Firefly [1]. We obtain Midjourney images by scraping the Discord API. As the model may be changing under the API and we do not know the true underlying model subversion, we date the models based on our scraping date. We query Adobe Firefly images (without the watermarking used in their web interface).

Prompt sourcing. For GLIDE, LDM, RDM, and Firefly (all sets), and Stable Diffusion (train+val), we use prompts from DiffusionDB [58]. For the Stable Diffusion test set, we use prompts from various web sources [6, 4, 50, 2].We sample unique prompts, in order to not have overlap between train, validation, and test sets.

3.2. Training details

We progressively train a binary classifier with a cross-entropy loss to distinguish between naturally sourced "real" images and those generated by AI. We follow best practices from Wang et al. [57], which show that a simple classifier can generalize across generators. We use a common CNN architecture, ResNet-50 [21], pre-trained on ImageNet [40] as the backbone for the online detector model. The training sequence follows Table 1 and Figure 1, simulating the real-world release dates of generative models. Each detector training step in the sequence continues from the previous model weights and incorporates all historical images seen to date. Release dates are determined by paper publication date, service launch announcement, or public release of model weights as relevant for the generative source.

Non-generated images sourced from LAION-400M are included at the start of training and remain a fixture throughout learning. During each training stage, we use a class-balanced random sampler to balance the distribution of generated and non-generated images over an epoch. Wang et al. [56] find that augmentations improve generalization. As such, we use a random 256×256 crop randomly applying Gaussian blur (with probability $p = .01$), grayscale ($p = .05$), and invisible watermarks [46] ($p = .2$), a common feature of generative services. For evaluation, we center crop to 256×256 with no augmentation.

3.3. Results

In Figure 2, we show the results of our online detector. The x-axis of the two heatmaps is the online model's training progress through time. Every column is a step in the learning progression, with a classifier trained on all models up to that cell, which are ordered by release date. For example, the "LDM" column includes images from DDPM, DDIM, GLIDE, and LDM. The y-axis represents the test set, with model release date ordered from bottom to top. In this way, cells in the upper left triangle show *generalization* performance of the online detector model on models, not yet included in the training.

Metrics. Figure 2a shows **accuracy** on the synthetic images. Figure 2b shows the **area under the curve (AuC)** between real and fake, sweeping over different thresholds, plotting the true positive and false positive rate, and taking the integral. The AuC metric shows how well the distributions are separated and is not sensitive to the classifier threshold. A high score indicates that the top most suspicious images can be reliably surfaced from a collection of images, useful in triaging scenarios. However, a common failure case of detectors on unseen distributions of synthetic images is classifying them as all real (low accuracy), as observed by Ojha et al [33], even when AuC is high. Accuracy on real images typically remains near 100%. Thus, we show both AuC, which measures if the classifier has the right set of features, and synthetic accuracy, which measures if the classifier also has the right calibration.

Does *directly* training on a generator work? Before investigating out-of-distribution generalization, we first perform a sanity check that a detector can be directly trained on a given generator. Observe the diagonal in Figure 2, outlined in green, which indicates the performance on a given generator directly after adding it to the training set. Indeed, even a straightforward CNN-based classifier can differentiate from generated images and real images fairly well. Recall that while we are evaluating in-distribution, these are on a separate held-out set of images. AuC is at 100% across all models, indicating perfect separation. Accuracy is near perfect, with worst case being 96% accuracy. This test serves as a sanity check and follows conclusions from previous work [57], that generators continue to have detectable cues, despite leaps in generative quality.

Can a detector reliably detect *multiple seen* generators simultaneously? After adding in other datasets, performance on seen generators (bottom-right) remains at roughly the same accuracy level. The final column is trained on all our generators and produces near-perfect performance across the datasets. This indicates that a single detector can learn the different artifacts from generators, without running out of capacity.

Can a detector generalize to *unseen* generators? Next, we investigate generalization ability and observe the upper-left of the matrix, testing on not-yet-released models. If classifiers do not generalize at all, we would see 0% accuracy and chance 0.5 AuC in the upper triangle. Previous

(a) Accuracy

	DDPM	DDIM	GLIDE	LDM	Dalle2	Midjv2	RDM	SDv1	Midjv3	SDv2	DiT	MidJv4	Midjv5	Firefly
Firefly	0	0	0.03	0.3	0.07	0.12	0.15	0.1	0.41	0.37	0.49	0.27	0.35	0.99
Midjv5	0	0	0.02	0.14	0.08	0.27	0.33	0.38	0.96	0.99	0.99	0.99	0.99	1
MidJv4	0	0	0.01	0.44	0.14	0.38	0.56	0.55	1	1	1	1	1	1
DiT	0.03	0.03	0.1	0.26	0.13	0.12	0.25	0.22	0.71	0.91	1	1	1	1
SDv2	0	0	0.02	0.32	0.22	0.36	0.5	0.64	0.98	1	1	1	1	1
Midjv3	0.02	0.06	0.22	0.82	0.76	0.86	0.89	0.89	1	1	1	1	1	1
SDv1	0	0	0.02	0.87	0.9	0.97	1	1	1	1	1	1	1	1
RDM	0	0.01	0.09	0.49	0.3	0.59	1	1	1	1	1	1	1	1
Midjv2	0.02	0.05	0.3	0.83	0.8	1	1	1	1	1	1	1	1	1
Dalle2	0	0.02	0.09	0.11	0.96	0.97	0.96	0.96	0.96	0.96	0.96	0.95	0.95	0.96
LDM	0.02	0.04	0.14	0.99	0.99	1	1	1	1	1	1	1	1	1
GLIDE	0.1	0.15	0.98	0.99	0.98	0.99	0.99	0.99	1	0.99	0.99	0.99	0.99	0.99
DDIM	0.96	0.99	0.98	0.97	0.98	0.98	0.99	0.99	0.99	0.98	0.99	0.98	0.98	0.99
DDPM	0.96	0.99	0.98	0.97	0.98	0.98	0.99	0.99	0.99	0.98	0.99	0.98	0.98	0.99

(b) Area under Curve (AuC)

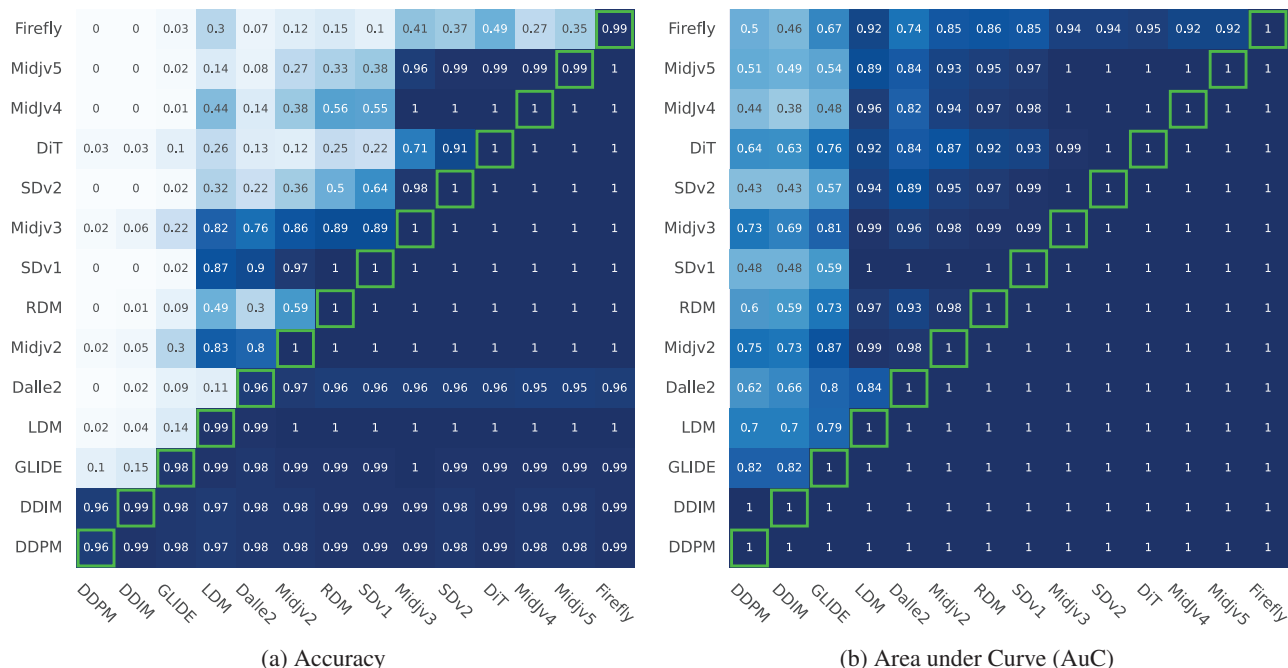|	DDPM	DDIM	GLIDE	LDM	Dalle2	Midjv2	RDM	SDv1	Midjv3	SDv2	DiT	MidJv4	Midjv5	Firefly
Firefly	0.5	0.46	0.67	0.92	0.74	0.85	0.86	0.85	0.94	0.94	0.95	0.92	0.92	1
Midjv5	0.51	0.49	0.54	0.89	0.84	0.93	0.95	0.97	1	1	1	1	1	1
MidJv4	0.44	0.38	0.48	0.96	0.82	0.94	0.97	0.98	1	1	1	1	1	1
DiT	0.64	0.63	0.76	0.92	0.84	0.87	0.92	0.93	0.99	1	1	1	1	1
SDv2	0.43	0.43	0.57	0.94	0.89	0.95	0.97	0.99	1	1	1	1	1	1
Midjv3	0.73	0.69	0.81	0.99	0.96	0.98	0.99	0.99	1	1	1	1	1	1
SDv1	0.48	0.48	0.59	1	1	1	1	1	1	1	1	1	1	1
RDM	0.6	0.59	0.73	0.97	0.93	0.98	1	1	1	1	1	1	1	1
Midjv2	0.75	0.73	0.87	0.99	0.98	1	1	1	1	1	1	1	1	1
Dalle2	0.62	0.66	0.8	0.84	1	1	1	1	1	1	1	1	1	1
LDM	0.7	0.7	0.79	1	1	1	1	1	1	1	1	1	1	1
GLIDE	0.82	0.82	1	1	1	1	1	1	1	1	1	1	1	1
DDIM	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DDPM	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 2: **Online detector performance**. We order generative methods by their release date and train a real vs. fake classifier, in a progressive fashion (x-axis). We show (a) accuracy on synthetic images and (b) area under the curve (AuC) between real and fake images, across different test sets (y-axis). When testing on a generator that is in the training data, performance jumps to near 100% in both metrics, as seen on the green diagonal. After adding other generative sources, performance remains near 100% (bottom triangle). Even before being directly trained on (upper left triangle), accuracy and AuC on generative sources not yet seen by the detector increases, as the historical training set grows (for example, see SDv2 row). AuC considers both generative and non-generative images and is less sensitive to prediction thresholds, suggesting the online detector model rapidly learns features useful for differentiating AI generated images from non-generative images across a wide variety of models, although requiring some additional calibration (threshold selection) to improve accuracy.

work [57, 12] observe that classifiers do generalize, and indeed, we see values higher than chance, even before they are directly trained on, however performance degrades over time, indicating that online training will still be required as new models are released.

Looking at the values one row above the diagonal indicates performance of training on N models and testing on the $(N + 1)^{th}$. Many of the methods – DDIM, Midjourney (v2 to v5), RDM, Stable Diffusion (v1 and v2), DiT – have AuC at 0.98 or above, indicating that the classifier contains the right features early on in training, after DALL·E 2 is added, the 5th model.

The accuracy scores are less consistent. For example, GLIDE achieves 15% accuracy while having perfect AuC, indicating that additional calibration is needed. The last model, Adobe Firefly, has relatively low accuracy, even when training on all previous models, indicating significant differences to previous models.

Which generators have high impact on one another? It is also interesting to note that related models are correlated. For example, adding LDMs [38] has outsized influence on

a number of models, as indicated by the step up in performance in the AuC (Fig 2b). This includes direct descendants of RDMs [9], Stable Diffusion versions [55, 7], as well as Midjourney, an unknown model.

Viewing the accuracy (Fig 2a), Midjourney v3 has large influence over subsequent versions of v4 and v5, indicating a large version change compared to Midjourney v2. As specific details of the model are not known, this sheds some light into the underlying model changes.

4. Detection of Generative Inpainting

Outputs from generative models are increasingly used in editing pipelines, where final images are a composite of both AI-generated pixels and traditionally sourced images. One common way these images are generated is with "inpainting", where a masked region of an image is seamlessly filled with generated content. To evaluate how well we can detect these types of edits, we create a dataset from Adobe Firefly and Stable Diffusion's inpainting models. As ground truth masks may be difficult to acquire in real-world settings, e.g., with closed models, we examine how one can
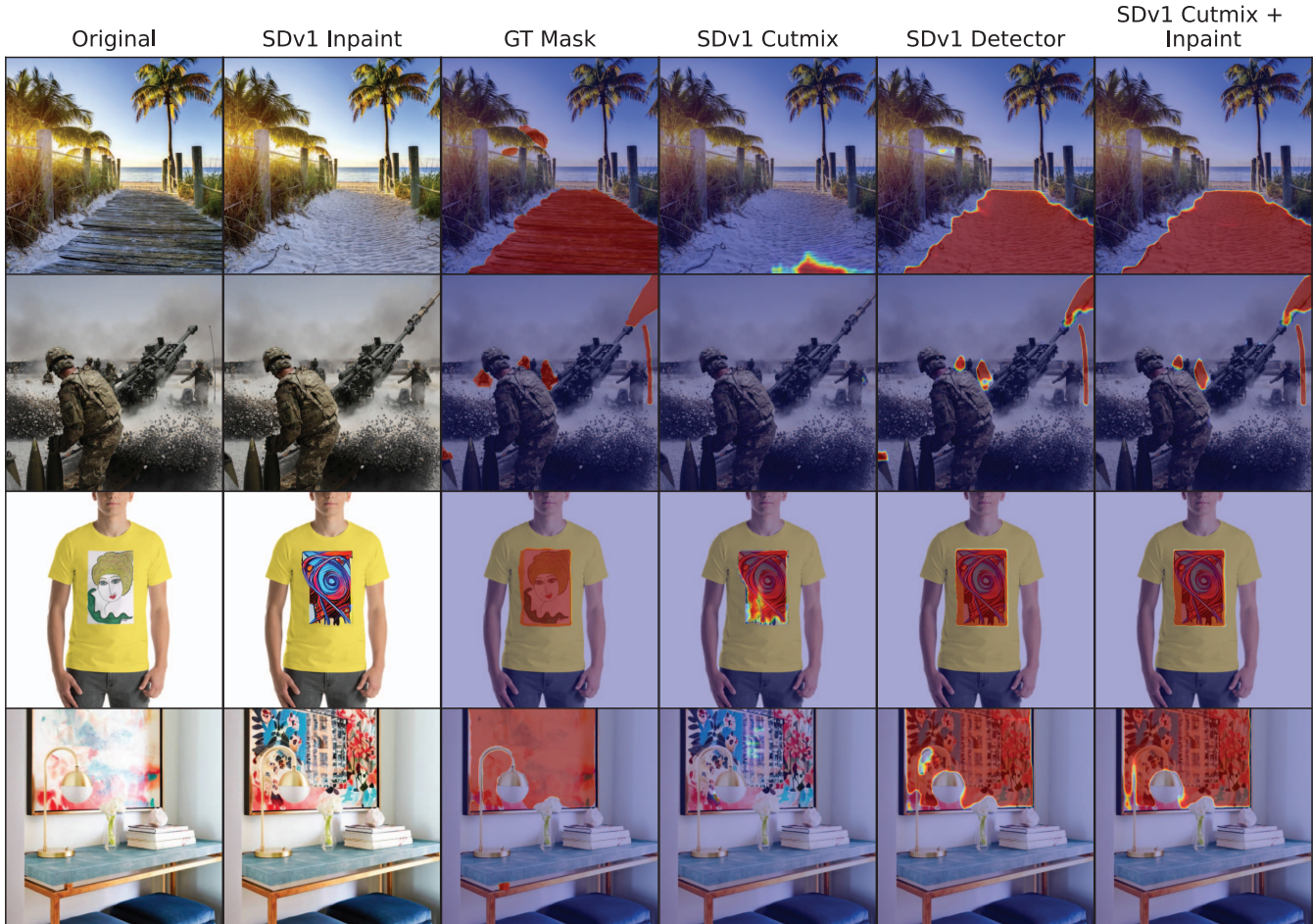
Figure 3: **Inpainting detection and generalization (SDv1).** Here we highlight the successful detection of SDv1 inpainted regions on LAION images. We also compare pixel detectors trained by various methods. While a SDv1 CutMix detector does provide some predictive power, inpainted example images are required in training for high accuracy.

leverage whole images to get pixel-wise labels, using Cut-Mix [61] augmentation. We also investigate generalization by the inclusion of images from multiple inpainting models.

4.1. Dataset

We create three inpainting datasets, using Stable Diffusion (v1 and v2) [55, 7] and Adobe Firefly [1]. Table 2 summarizes model release dates, input source, masked pixel distribution, and dataset size. We sample input images and corresponding prompts from the LAION-400M Dataset[44]. We resize images to 512 pixels on the short side and center crop and generate a mask. We create masks covering 15 to 35% of each image, with random overlapping strokes and shapes. In order to preserve the fidelity of the non-masked region and isolate the generated pixels from the original, we copy the original image back into the non-masked region. Importantly, we do not observe this approach to cause a visible seam.

For whole image sets, we use LAION images from Table

Inpaint Model	Release Date	Input	Mask %	*Train*	*Val*	*Test*
SDv1	Oct 22	LAION	15-35	9375	1250	1875
SDv2	Nov 22	LAION	15-35	9375	1250	1875
Firefly	Mar 23	LAION	15-35	9091	1206	1813

Table 2: **Inpainting dataset.** We investigate pixel-wise generative detection by creating a dataset of inpainting results, using Stable Diffusion and Adobe Firefly.

2 and sample the desired generative models (in equal size) from the online dataset previously described in Table 1.

4.2. Training details

Similarly to whole image detection, we study generalization using a standard Fully Convolutional Network (FCN) [30] with a ResNet-50 [21] architecture, utilizing weighted Dice loss [51] to address data imbalance. At training, we test schemes with the assumption that an inpainting API is or is not available. When not available, we test whether *whole* images can be leveraged for pixel-wise la-
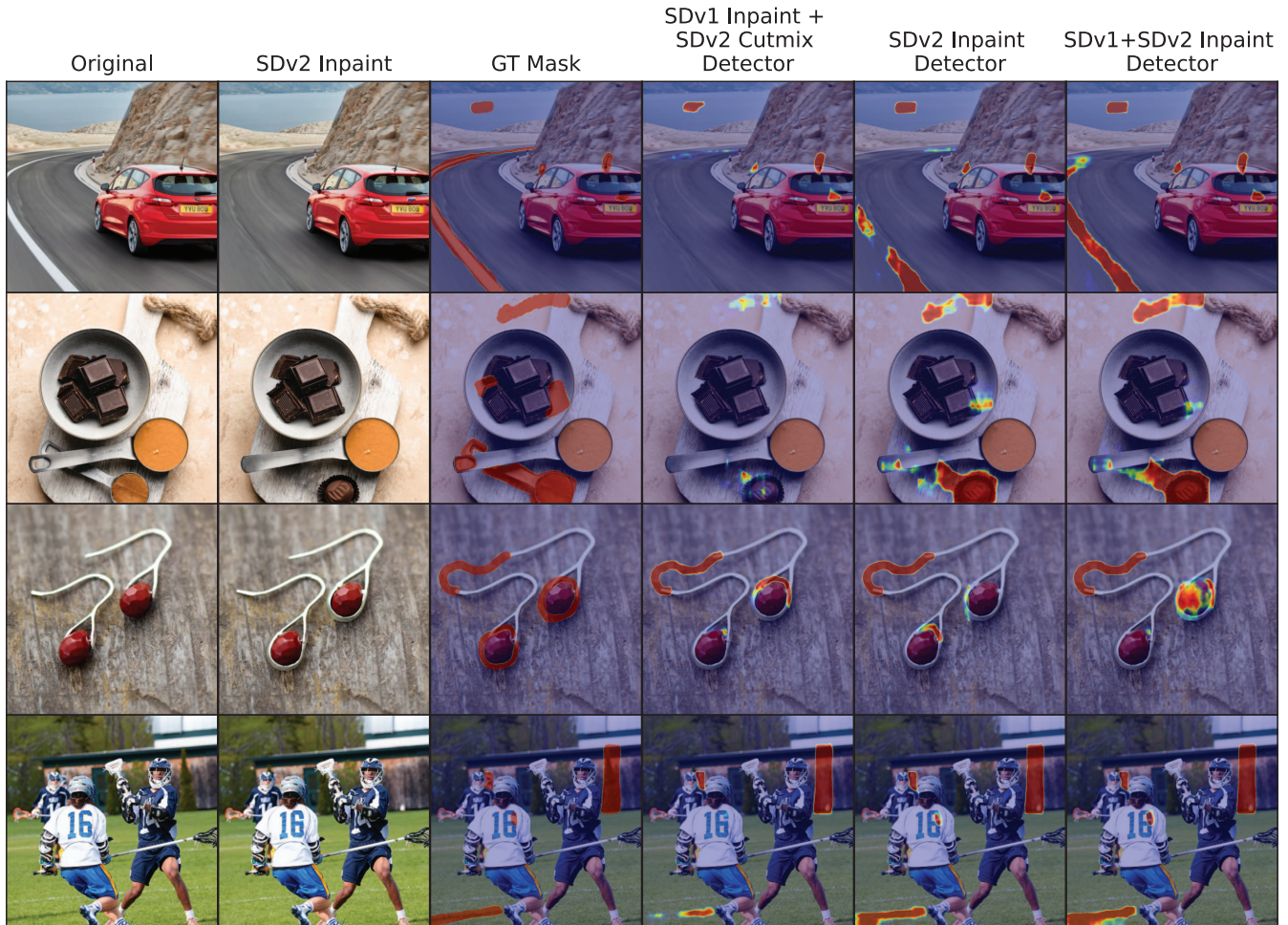
Figure 4: **Inpainting detection and generalization (SDv2).** We can clearly see the progression in accuracy on SDv2 inpainted LAION images as higher quality data and an additional model source is introduced in pixel detector training.
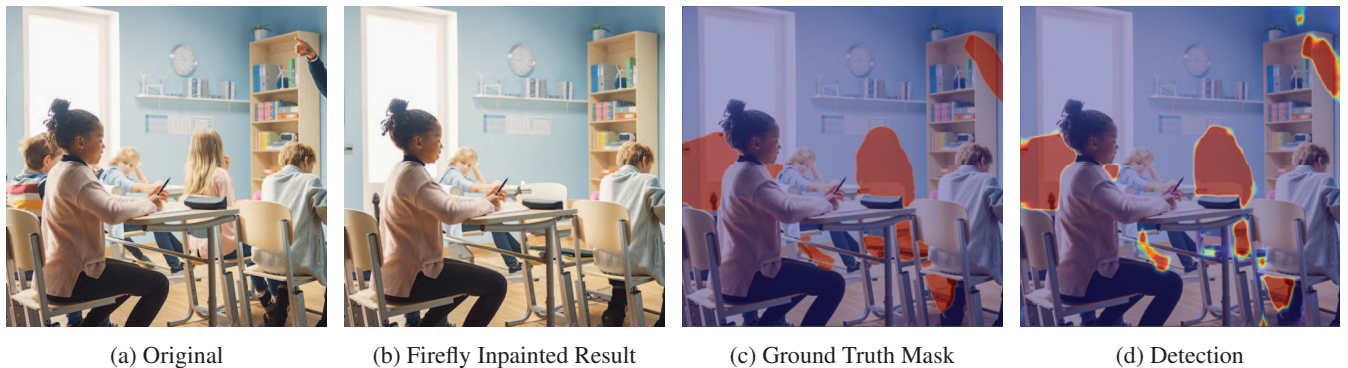

(a) Original   (b) Firefly Inpainted Result   (c) Ground Truth Mask   (d) Detection

Figure 5: **Generative AI pixel detection test.** A detector model trained on inpainted images successfully captures the generated pixels from Adobe Firefly's inpainting model.

bels. We test CutMix blending [61], a naive approach of simply cutting and pasting blocks of one image into another [61]. When an inpainting API is available, we test with synthetically-generated inpainted images with random masks, described in the section above. We also test if adding

whole images with CutMix can help in this situation.

4.3. Results

Can we create a pixel detector? Table 3 shows results on training and detecting SDv1. When only using whole

Training Data	SDv1 Inpainting		Whole image	
	accuracy	f1	accuracy	f1
SDv1 whole image	0.7755	0.1807	0.9979	0.9968
SDv1 cutmix	0.8359	0.4882	0.9960	0.9941
SDv1 inpaint	0.9902	0.9795	0.9918	0.9878
SDv1 inpaint + cutmix	**0.9920**	**0.9832**	**0.9996**	**0.9994**

Table 3: **SDv1 inpainting detection.** Compared to using whole images only, we observe improvement when training with CutMix augmentation, and another jump when trained on inpainted images, with the best model using both. Additionally, we see these methods retain accuracy on whole image detection in a test set comprising both LAION and pure SDv1 images.

Training Data	accuracy	precision	recall	f1
SDv1 inpaint	0.9288	0.9776	0.7173	0.8275
SDv1 inpaint + SDv2 cutmix	0.9600	0.9781	0.8509	0.9101
SDv2 inpaint	0.9872	0.9766	0.9693	0.9730
SDv1 inpaint + SDv2 inpaint	**0.9892**	**0.9811**	**0.9733**	**0.9772**

Table 4: **SDv2 inpainting generalization.** A pixel detector trained on SDv1 inpainting images generalizes well to SDv2 inpainted images (1st row). When SDv2 whole images are added in training via CutMix, performance rises (2nd row), approaching the performance of a detector trained directly on SDv2 inpainted images (3rd row).

Training Data	accuracy	precision	recall	f1
Firefly cutmix	0.8250	0.9376	0.2870	0.4395
SDv1 + SDv2 inpaint + Firefly cutmix	0.9511	0.9749	0.8163	0.8886
Firefly inpaint	0.9811	0.9734	0.9469	0.9600
SDv1 + SDv2 + Firefly inpaint	**0.9891**	**0.9805**	**0.9740**	**0.9772**

Table 5: **Firefly inpaint generalization.** We see an increase in pixel accuracy when SDv1 and SDv2-inpainted images are added to Firefly CutMix detector training (1st to 2nd row), with performance relatively close to that achieved by directly training on Firefly inpainted images (3rd row).

images, we show that using CutMix improves performance ($77.6\% \rightarrow 83.6\%$ accuracy). In Figure 3, we see that Cut-Mix can catch some inpainted regions, though far from perfect. When an inpainting API is available, we show that synthetically generated samples can greatly improve accuracy (99.0%), with small improvements adding CutMix (99.2%). In Figure 5, we show a qualitative example a Firefly inpainting detector, trained on synthetically-generated Firefly samples.

Can a pixel detector be used in conjunction with a whole image detector? Note that performance on detecting whole images is nearly perfect >99.0% across all cases, indicating the model is not simply relying on boundary artifacts.

Does a pixel detector generalize across models? In Table 4, we measure performance on SDv2-generated in-

painted images on detector methods, and if SDv1 can be leveraged to improve performance. We follow a progression from highly limited training data to most abundant. We see in the most limited case, where the detector model only has access to images from the previous version of SDv1, a high-performance accuracy of 92.9%, showing generalization at least in the case of closely related model sources. Leveraging SDv2, even with just whole images using CutMix, improves performance (96%). This is approaching the performance of a stand-alone detector model trained directly on SDv2 inpainted images (98.7%). Using the previous version of SDv1 further improves performance (98.9%). In Figure 4, we show qualitative examples of this progressive improvement.

While SDv1→SDv2 studies generalization across closely related models, in Table 5, we study generalization across unrelated models, (SDv1+SDv2) →Firefly. We see that leveraging inpainted SDv1+SDv2 images improves performance, both compared to training only with Firefly whole images using CutMix (82.5→95.1) and with inpainted Firefly images available (98.1→98.9). Our results indicate that generated images contain sufficient cues even at a local level that can be detected. Furthermore, incorporating previously seen models can improve accuracy.

5. Conclusions and Limitations

We conduct experiments to investigate how well classifiers can detect AI-generated images in a simulated online framework. We see that classifiers do generalize to unseen models, although when there are major architectural changes, performance drops substantially. These experiments suggest that a vigilant classifier, regularly retrained on new generators, has the opportunity to detect future, unreleased models, as long as they are architecturally similar.

While our dataset highlights some key advancements in generative methods, it does not comprise of all models in this time period. Methods such as Imagen [41], Parti [60], Muse [13], and GigaGAN [24] offer high-quality generations with alternative architectures but do not have public APIs or source code released. Exploring detection for such methods is an interesting direction if the models become available. In addition, we study online generalization with simple architectures (ResNet [21] and FCN [30]), following best training practices [57]. Further improving generalization in an online setting is an important future area of work.

References

[1] Adobe firefly. https://firefly.adobe.com/. 2, 3, 4, 6

[2] Dall·e 2 image database. https://dalle2.gallery. 4

[3] Midjourney. https://www.midjourney.com/. 3, 4

[4] Prompt hero. http://prompthero.com/. 4

[5] Shruti Agarwal and Hany Farid. Photo forensics from jpeg dimples. In *2017 IEEE workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2017. 2

[6] Krea AI. Open prompts. https://github.com/krea-ai/open-prompts. 4

[7] Stability AI. Stable diffusion version 2. https://github.com/Stability-AI/stablediffusion. 2, 3, 5, 6

[8] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models, 2023. 4

[9] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems*, 35:15309–15324, 2022. 3, 5

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2

[11] Tu Bui, Ning Yu, and John Collomosse. Repmix: Representation mixing for robust attribution of synthesized images, 2022. 2

[12] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. 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*, pages 103–120. Springer, 2020. 1, 2, 3, 5

[13] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 8

[14] Davide Alessandro Coccomini, Andrea Esuli, Fabrizio Falchi, Claudio Gennaro, and Giuseppe Amato. Detecting images generated by diffusers, 2023. 2

[15] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models, 2022. 2

[16] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 2

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[18] Hany Farid. Image forgery detection. *IEEE Signal processing magazine*, 26(2):16–25, 2009. 2

[19] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020. 2

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2, 3

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 6, 8

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3

[23] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 101–117, 2018. 2, 3

[24] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023. 2, 8

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[27] Eric Kee, James F O'Brien, and Hany Farid. Exposing photo manipulation with inconsistent shadows. *ACM Transactions on Graphics (ToG)*, 32(3):1–12, 2013. 2

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 3

[29] Bo Liu, Fan Yang, Xiuli Bi, Bin Xiao, Weisheng Li, and Xinbo Gao. Detecting generated images by real images. In *European Conference on Computer Vision*, pages 95–110. Springer, 2022. 2

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 6, 8

[31] Lakshmanan Nataraj, Tajuddin Manhar Mohammed, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H Bappy, Amit K Roy-Chowdhury, and BS Manjunath. Detecting gan generated fake images using co-occurrence matrices. *arXiv preprint arXiv:1903.06836*, 2019. 2

[32] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 3

[33] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023. 2, 4

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 3, 4

[35] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on signal processing*, 53(2):758–767, 2005. 2

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 2, 3

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 3, 5

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 3, 4

[41] 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. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1, 2, 8

[42] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455. PMLR, 2009. 2

[43] 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. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 3

[44] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 3, 6

[45] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models, 2023. 2

[46] ShieldMnt. invisible-watermark. `https://github.com/ShieldMnt/invisible-watermark`. 4

[47] Paul Smolensky et al. Information processing in dynamical systems: Foundations of harmony theory. 1986. 2

[48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2, 3

[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 3

[50] Succinctly. Midjourney prompts. `https://huggingface.co/datasets/succinctly/midjourney-prompts`. 4

[51] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer International Publishing, 2017. 6

[52] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016. 2

[53] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 2

[54] Luisa Verdoliva. Media forensics and deepfakes: an overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020. 2

[55] Computer Vision and Learning research group at Ludwig Maximilian University of Munich. Stable diffusion. `https://github.com/CompVis/stable-diffusion`. 2, 3, 5, 6

[56] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A Efros. Detecting photoshopped faces by scripting photoshop. In *ICCV*, 2019. 2, 3, 4

[57] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020. 1, 2, 4, 5, 8

[58] 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, 2023. 4

[59] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 3

[60] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yin-

fei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 1, 2, 8

[61] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 2, 3, 6, 7

[62] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2, 3

[63] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1053–1061, 2018. 2, 3