A Bias-Free Training Paradigm for More General AI-generated Image Detection

Fabrizio Guillaro[1] Giada Zingarini[1] Ben Usman[2] Avneesh Sud[2]
Davide Cozzolino[1] Luisa Verdoliva[1]

[1]University Federico II of Naples [2]Google DeepMind

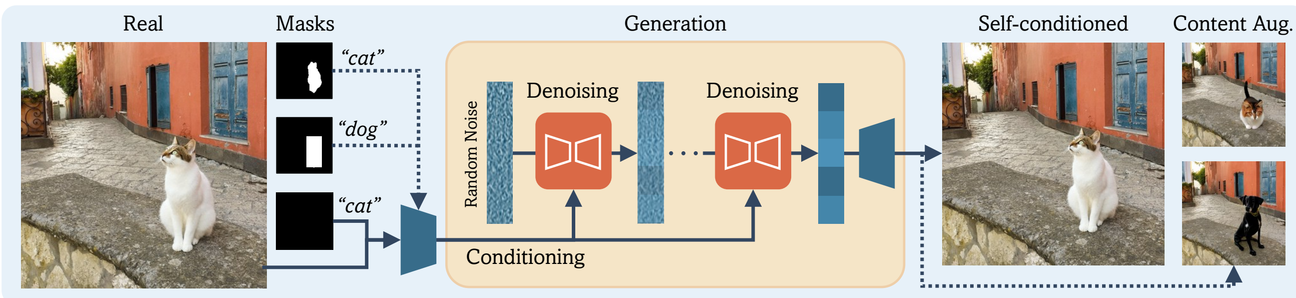


Figure 1. We introduce a new training paradigm for AI-generated image detection. To avoid possible biases, we generate synthetic images from self-conditioned reconstructions of real images and include augmentation in the form of inpainted versions. This allows to avoid semantic biases. As a consequence, we obtain better generalization to unseen models and better calibration than SoTA methods.

Abstract

Successful forensic detectors can produce excellent results in supervised learning benchmarks but struggle to transfer to real-world applications. We believe this limitation is largely due to inadequate training data quality. While most research focuses on developing new algorithms, less attention is given to training data selection, despite evidence that performance can be strongly impacted by spurious correlations such as content, format, or resolution. A well-designed forensic detector should detect generator specific artifacts rather than reflect data biases. To this end, we propose B-Free, a bias-free training paradigm, where fake images are generated from real ones using the conditioning procedure of stable diffusion models. This ensures semantic alignment between real and fake images, allowing any differences to stem solely from the subtle artifacts introduced by AI generation. Through content-based augmentation, we show significant improvements in both generalization and robustness over state-of-the-art detectors and more calibrated results across 27 different generative models, including recent releases, like FLUX and Stable Diffusion 3.5. Our findings emphasize the importance of a careful dataset design, highlighting the need for further research on this topic. Code and data are publicly available at https://grip-unina.github.io/B-Free/.

1. Introduction

The rise of generative AI has revolutionized the creation of synthetic content, enabling easy production of high-quality sophisticated media, even for individuals without deep technical expertise. Thanks to user-friendly interfaces and pretrained models, users can create synthetic content such as text, images, music, and videos through simple inputs or prompts [50]. This accessibility has democratized content creation, enabling professionals in fields like design, marketing, and entertainment to leverage AI for creative purposes. However, this raises concerns about potential misuse, such as the creation of deepfakes, misinformation, and challenges related to intellectual property and content authenticity [4, 17, 24].

Key challenges for current GenAI image detectors include generalization — detecting synthetic generators not present in the training set — and ensuring robustness against image impairments caused by online sharing, such as compression, resizing, and cropping [41]. In this context, large pre-trained vision-language models like CLIP [34] have demonstrated impressive resilience to these distribution shifts [30]. The success of these models in forensic applications suggests that pre-training on large and diverse datasets may be a promising path forward. An important aspect often overlooked in the current literature is the selection of good datasets to train or fine-tune such models that

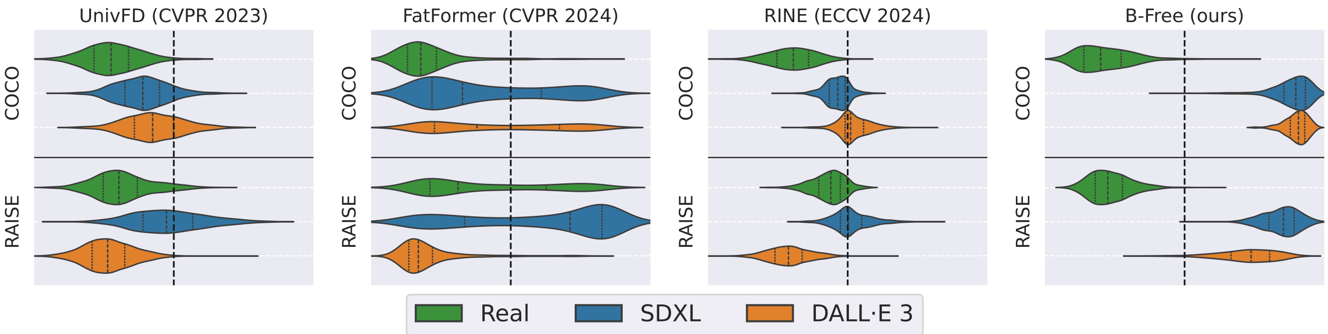


Figure 2. Forensic detectors can exhibit opposite behaviors depending on their training dataset. The four plots show the prediction distributions for three ViT-based detectors, UnivFD [30], FatFormer [26] and RINE [22], and the proposed one. The fake images (SD-XL or DALL-E 3) are generated from images of a single dataset (RAISE on top, COCO on the bottom) and tested only against real images of the same dataset (Synthbuster [2] and the test dataset from [11]). We observe that for the same detector (e.g., RINE) and the same fake-image generator (e.g., DALL-E 3) the score distributions can vary significantly depending on the dataset used, going from real (left of the dotted line) to fake (right of the dotted line) or vice versa. This is likely due to the presence of biases in the training set that heavily impact the detector prediction. Our detector, on the other hand, shows consistent and correct results.

primarily rely on hidden, unknown signatures of generative models [29, 47]. Indeed, it is important to guarantee that the detector decisions are truly based on generation-specific artifacts and not on possible dataset biases [7, 27, 42]. In fact, datasets used during the training and testing phases of forensic classifiers could be affected by different types of polarization.

Format issues have been the Achilles' heel of forensic detectors since at least 2013, when [5] recognized that a dataset for image tampering detection [14] included forged and pristine images compressed with different JPEG quality factors. Therefore, a classifier trained to discrminate tampered and pristine images may instead learn their different processing histories. This issue has been highlighted in [19] with reference to datasets of synthetic and real images. In fact, the former are often created in a lossless format (PNG), while the latter are typically compressed in lossy formats like JPEG. Again a classifier could learn coding inconsistencies instead of forensic clues. Likewise it could learn resampling artifacts, as it was recently shown in [35] - in this case a bias was introduced by resizing all the real images from the LAION dataset to the same resolution, while keeping the fake ones unaltered.

Forensic clues are subtle and often imperceptible to the human eye, making it easy to introduce biases when constructing the training and test sets, as well as the evaluation protocol. Semantic content itself can also represent a source of bias. For this reason, several recent proposals [2, 3, 11] take great care to include pairs of real and fake images characterized by the same prompts when building a training or test dataset. To gain better insights about the above issues, in Fig. 2 we show the performance of three SoTA ViT-based approaches [22, 26, 30] in distinguishing real images from fake images generated by SD-XL and DALL-E 3. For each method we consider two settings: in the first case, real images come from the RAISE dataset [12] and fakes are generated starting from images of the same dataset. The second case uses COCO as source of reals instead of RAISE. We note an inconsistent behavior of SOTA forensic detectors on the same synthetic generator which can be caused by the presence of biases during training. FakeInversion [7] proposes an effective approach towards semantic alignment of training data using reverse image search to find matching reals, but fails to capture real image distribution after 2021.

To mitigate potential dataset biases, in this work we propose a new training paradigm, B-Free, where synthetic images are generated using self-conditioned reconstructions of real images and incorporate augmented, inpainted variations. This approach helps to prevent semantic bias and potential misalignment in coding formats. Furthermore, the model avoids resizing operations that can be particularly harmful by washing out the subtle low-level forensic clues [18]. Overall, we make the following contributions:

- We propose a large curated training dataset of 51k real and 309k fake images. Real images are sourced from COCO, while synthetic images are self-conditioned generations using Stable Diffusion 2.1. This helps the detector to focus on artifacts related to the synthetic generation process avoiding content and coding related biases.
- We show that incorporating proper content-based augmentation leads to better-calibrated results. This ensures that in-lab performance more closely aligns with the expected performance on real-world images shared across social networks.
- We study the effect of different distribution shifts and show that by leveraging a pre-trained large model fine-tuned end-to-end on our dataset, we achieve a SoTA accuracy superior to 90% even on unseen generators.

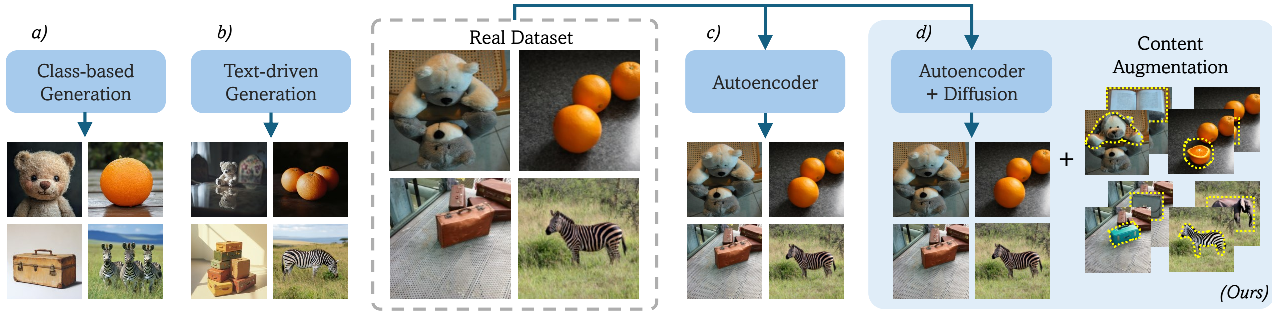


Figure 3. Overview of existing (a, b, c) and proposed (d) strategies for building an aligned training dataset. Some methods try to match synthetic images to the corresponding real images by using class-based generation (a) or text-to-image generation with real images' descriptions (b). In (c) real images are fed to an autoencoder to generate a **reconstructed fake** with the same content. Differently from (c), in our approach a **self-conditioned fake** is generated using diffusion steps (d), and we also add a content augmentation step.

2. Related Work

A well-curated training set is of vital importance for any data-driven method. In recent years this awareness has much grown also in the forensic field and there have been many efforts in this direction following two main lines of work: *i)* forming a reliable dataset by carefully selecting "natural" fakes, or *ii)* creating a fully synthetic dataset by injecting forensic artifacts in real images.

Selecting good natural training data. Wang et al.'s paper [43] was among the first to demonstrate the importance of selecting a suitable training set for gaining generalization to unseen synthetic generators. The selected dataset included images from a single generation architecture (ProGAN) and 20 different real/false categories (Fig. 3.a) and included augmentation in the form of common image post-processing operations, such as blurring and compression. Results clearly show that generalization and robustness strongly benefit from the many different categories included during training as well as from the augmentation procedure. In fact, this dataset has been widely utilized in the literature, where researchers follow a standard protocol assuming the knowledge of one single generative model during training. This scenario describes a typical real world situation where new generative architectures are unknown at test time.

The dataset proposed in [43] was used in [30] to fine-tune a CLIP model with a single learnable linear layer, achieving excellent generalization not only on GAN models but also on Diffusion-based synthetic generators never seen during training. Likewise, it was used in [22] to train a CNN classifier that leverages features extracted from CLIP's intermediate layers to better exploit low-level forensic features. In [37, 40] image captions (either paired to the dataset images or generated from them) were used as additional input for a joint analysis during training. The approach proposed in [26] is trained using only 4 classes out of the 20 categories proposed in [43], as well as other recent methods [38–40].

Alternatively, some methods rely on datasets comprising images from a single diffusion-based generator, such as Latent Diffusion [7, 10, 11], Guided Diffusion [44] or Stable Diffusion [23, 37]. Prior work [7, 11] highlights the importance of aligning both training and test data in terms of semantic content. This choice allowed to better exploit the potential of fixed-pretraining CLIP features by strongly reducing the number of images needed for fine-tuning [11]. In addition, it has the key merit of reducing the dataset content bias, thus allowing for better quality training, and is also adopted in other approaches both during training [1, 3] and at test time to carry out a fairer evaluation [2].

Creating training data by artifact injection. A different line of research is to create simulated fake images by injecting traces of the generative process in real images. A seminal work along this line was done by Zhang et al. [52] for GAN image detection. The idea is to simulate artifacts shared by several generators. These peculiar traces are caused by the up-sampling processes included in the generation pipeline and show up as peaks in the frequency domain. Besides these frequency peaks, synthetic images, both GAN-based and diffusion-based, have been shown to exhibit spectral features that are very different from those of natural images [15, 16]. In fact, real images exhibit much richer spectral content at intermediate frequencies than synthetic ones [9, 46].

For GAN-generated images, producing realistic simulated fakes requires training the generation architecture specifically for this task [20, 52]. In contrast, diffusion-based image generation can leverage a pre-trained autoencoder embedded within the generation pipeline, which projects images into a latent space without the need for additional training [11, 28]. This procedure has been very recently used in a concurrent work [35] to reduce semantic biases during training (Fig. 3.c). Different from [35] we generate synthetic data by also performing the diffusion steps. Later in this work we will show that this choice allows us

Reference	# Real/ # Fake	Real Source	# Models
Synthbuster [2]	1k / 9k	RAISE	9
GenImage [53]	1.3M / 1.3M	ImageNet	8
FakeInversion [7]	44.7k / 44.7k	Internet	13
SynthWildX [11]	500 / 1.5k	X	3
WildRF [6]	1.25k / 1.25k	Reddit, FB, X	unknown

Table 1. This table provides an overview of the datasets used in our evaluation, including the number of real and fake images, the sources of the real data, and the number of generative models used to create the synthetic images.

to exploit even subtler inconsistencies at lower frequencies, enhancing the detector performance (Fig. 3.d).

3. Evaluation Protocol

3.1. Datasets

In our experimental analysis, we want to avoid or at least minimize the influence of any possible afore-mentioned biases. To this end, we carefully select the evaluation datasets as outlined below. Experiments on further datasets are provided in the supplementary material.

To avoid format bias, we use Synthbuster [2], where both real and generated images are saved in raw format. Therefore, a good performance on this dataset cannot come from the exploitation of JPEG artifacts. A complementary strategy to avoid format biases is to reduce the mismatch between real (compressed) and synthetic (uncompressed) images by compressing the latter. To this end, we modified the fake class in GenImage [53] by compressing images at a JPEG quality close to those used for the real class, as suggested in [19]. This modified dataset, referred to as GenImage unbiased, comprises 5k real and 5k fake images, a small fraction of the original dataset.

To avoid content bias, we also evaluate performance on datasets where fakes are generated using automated descriptions of real images. In studies like [2, 3] these descriptions are refined into manually created prompts for text-based generation. As a result, the generated images closely align with the content of the real images, minimizing possible biases due to semantic differences. A more refined dataset in this regard is FakeInversion [7], where real images are retrieved from the web using reverse image search, thus ensuring stylistic and thematic alignment with the fakes.

To allow in-the-wild analysis, we experiment also on datasets of real/fake images collected from the web, such as WildRF [6] and SynthWildX [11]. Both datasets comprise images coming from several popular social networks. Tags were used to find fake images on Reddit, Facebook and X. A short summary of all the datasets used in our evaluation is listed in Table 1.

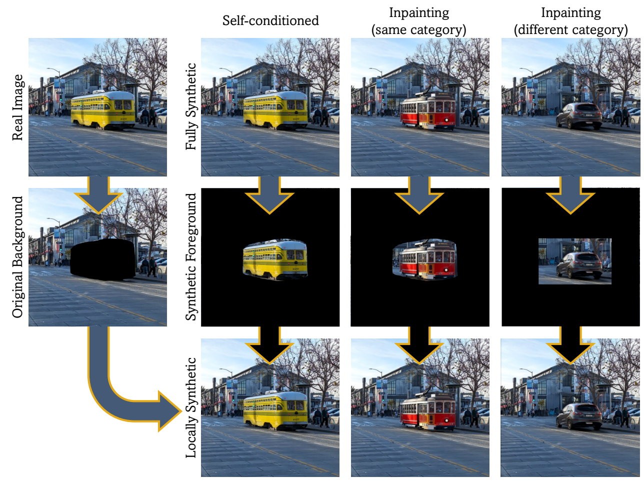


Figure 4. **Content augmentation** process. Starting with a real image, we use its generated variants (first row) and their locally manipulated versions (last row), created by replacing the original background. When inpainting with a different category, we use a bounding box instead of an object mask to allow space for new objects of varying shapes and sizes.

3.2. Metrics

Most work on GenAI image detection measure performance by means of threshold-independent metrics, such as Area Under the Curve (AUC) or average precision (AP). These metrics indicate ideal classification performance, however the optimal separating threshold is not known and, quite often, the balanced accuracy at a fixed threshold (e.g. 0.5) remains low, especially when there are significant differences between training and testing distributions [41]. Some papers address this problem by adjusting the threshold through a calibration procedure, assuming access to a few images from the synthetic generator under evaluation [10, 30, 43]. In a realistic situation the availability of such calibration images cannot be guaranteed.

In this work, to provide a comprehensive assessment of performance, we use both AUC and Accuracy at 0.5, in addition we compute the Expected Calibration Error (ECE) and the Negative Log-Likelihood (NLL). ECE measures the ability of a model to provide prediction probabilities well aligned with the true probabilities. More precisely, we use the Binary ECE, which is the weighted average of the differences between the actual probability and the predicted probability across different bins [32]. Then, we use the balanced Negative Log-Likelihood [33], which evaluates the similarity between the distribution of the model's predictions and the actual data distribution, penalizing both low confidence in the correct class and overconfidence in incorrect ones. More details on these metrics can be found in the supplementary material.

	Method	Training setting	Synthbuster					New Generators		WildRF			AVG
			Midjourney	SDXL	DALL·E 2	DALL·E 3	Firefly	FLUX	SD 3.5	Facebook	Reddit	Twitter	AUC↑/bAcc↑
AUC / bAcc	paired by text	–	96.9 / 56.9	**99.5** / 78.1	78.7 / 50.1	98.8 / 56.6	91.3 / 51.1	94.6 / 51.5	96.6 / 66.9	97.8 / 72.5	84.4 / 67.2	96.1 / 68.1	93.5 / 61.9
	reconstructed	–	**100. / 99.8**	**100. / 100.**	81.1 / 52.2	**99.1** / 75.8	96.7 / 62.1	98.2 / 64.0	**99.7** / 88.8	**98.9 / 97.5**	77.8 / 75.5	94.8 / 91.1	94.6 / 80.7
	reconstructed	inpainted	**100. / 99.9**	**100. / 99.9**	90.1 / 62.0	**99.2** / 79.5	98.1 / 77.2	95.8 / 69.1	**99.4** / 89.1	**98.8** / 95.9	78.7 / 76.1	94.9 / 90.3	95.5 / 83.9
	self-conditioned	–	**99.9** / 97.2	**100. / 99.2**	90.4 / 58.1	98.9 / 76.1	**99.4** / 89.7	95.4 / 59.4	**100. / 98.4**	95.4 / 86.6	75.7 / 66.3	91.4 / 82.7	94.7 / 81.4
	self-conditioned	cutmix/mixup	**99.9** / 94.3	**99.9** / 97.8	93.5 / 53.2	**99.1** / 72.7	**99.8** / 76.4	90.4 / 52.3	**99.8** / 91.0	96.4 / 90.3	80.3 / 74.8	93.8 / 82.9	95.3 / 78.6
	”	inpainted	**100. / 99.4**	**100. / 99.6**	96.7 / 77.8	**99.4** / 92.8	**99.9 / 99.2**	**98.7** / 87.5	**99.9 / 98.4**	**98.9** / 94.4	89.4 / 81.2	97.5 / 92.0	98.0 / 92.2
	”	inpainted+	**99.9** / 98.8	**100. / 99.7**	**99.7 / 95.9**	**99.6** / 96.8	**100. / 99.6**	97.9 / 85.3	**99.3** / 95.1	98.0 / 95.0	**96.0 / 89.8**	**99.4 / 96.5**	**99.0** / 95.2
	”	inpainted++	**100. / 99.6**	**100. / 99.8**	**99.7 / 95.7**	**99.9 / 98.2**	**100. / 99.7**	**99.3 / 92.3**	**99.9 / 98.9**	**99.0** / 95.6	**95.8** / 86.3	**99.7 / 97.4**	**99.3 / 96.4**
	Method	Training setting	Midjourney	SDXL	DALL·E 2	DALL·E 3	Firefly	FLUX	SD 3.5	Facebook	Reddit	Twitter	NLL↓/ECE↓
NLL / ECE	paired by text	–	1.96 / .418	0.72 / .218	3.60 / .496	1.71 / .416	2.95 / .484	2.62 / .473	1.42 / .327	1.00 / .273	1.77 / .324	1.31 / .310	1.91 / .374
	reconstructed	–	**0.00 / .003**	**0.00 / .001**	3.33 / .469	0.83 / .240	1.56 / .368	1.46 / .354	0.38 / .115	0.13 / **.025**	1.20 / .192	0.43 / .082	0.93 / .185
	reconstructed	inpainted	0.01 / .008	0.01 / .008	1.16 / .353	0.40 / .197	0.51 / .222	0.79 / .290	0.23 / .107	**0.12** / .032	0.73 / .153	0.29 / .066	0.42 / .144
	self-conditioned	–	0.08 / .031	0.02 / .008	1.48 / .399	0.59 / .234	0.27 / .114	1.22 / .379	0.04 / .021	0.31 / .084	0.92 / .237	0.40 / .078	0.53 / .158
	self-conditioned	cutmix/mixup	0.14 / .063	0.06 / .028	1.66 / .440	0.67 / .268	0.48 / .238	1.82 / .452	0.22 / .103	0.29 / .076	0.77 / .162	0.48 / .141	0.66 / .197
	”	inpainted	0.02 / .016	0.02 / .017	0.49 / .199	0.18 / .085	0.04 / .025	0.27 / .117	0.05 / .021	0.16 / .070	0.39 / .059	0.19 / .033	0.18 / .064
	”	inpainted+	0.04 / .008	0.01 / .006	0.12 / **.044**	0.10 / .037	**0.02 / .011**	0.40 / .149	0.14 / .044	0.17 / .032	**0.25 / .043**	0.11 / **.028**	0.14 / .040
	”	inpainted++	0.02 / .013	0.01 / .011	**0.10** / .045	**0.06 / .031**	0.02 / .019	**0.19 / .084**	**0.04 / .014**	0.14 / .039	0.28 / .075	**0.09** / .047	**0.10 / .038**

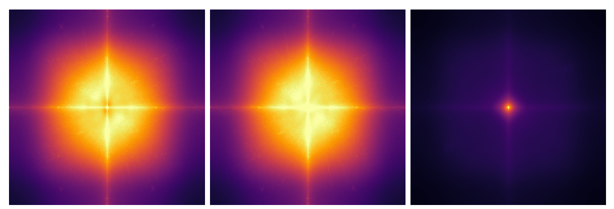
Table 2. Ablation study. We compare several forms of content alignment and content augmentation. Performance are in terms of AUC/Accuracy (top) and ECE/NLL (bottom). Note that all variants share a standard augmentation (blurring + JPEG compression) as proposed in [43]. For content alignment we consider image pairing strategies described in Fig. 3: text-driven generation, reconstruction through autoencoder, and our proposal using self-conditioned images. For the last solution we test several forms of augmentation: a standard cutmix/mixup, and three proposed strategies based on inpainting described in Sec. 4.2. Bold underlines the best performance for each column with a margin of 1%.

4. Proposed Method

To realize and test our bias-free training paradigm we:

- build a dataset consisting of real and generated fake images, where the latter are well aligned with their real counterparts but include the forensic artifacts of the diffusion-based generation process. The dataset is created starting from the images collected from the training set of MS-COCO dataset [25], for a total of 51,517 real images. It is then enriched through several forms of augmentation, including locally inpainted images, and comprises eventually 309,102 generated images.
- use this aligned dataset to fine-tune end-to-end a Vision Transformer (ViT)-based model. Specifically, we adopt a variant of the ViT network proposed in [13] with four registers and use the pretraining based on the self-supervised learning method DINOv2 [31]. During training, we avoid resizing the image and rely on large crops of 504×504 pixels. At inference time, we also extract crops of the same dimension from the image (if the image is larger we average the results of multiple crops).

To ensure fake images semantically match the content of real images, we exploit the conditioning mechanism of Stable Diffusion models that allows us to control the synthesis process through a side input, which can be a class-label, a text or another image. The side input is firstly projected to an intermediate representation by a domain specific encoder, and then feeds the intermediate layers of the autoencoders for denoising in the embedding space. After several denoising steps, a decoder is used to obtain the conditioned synthetic image from embedded vector (See Fig. 1). In our self-conditioned generation, we use the inpainting diffusion model of Stable Diffusion 2.1 [36], that has three side inputs: the reference image, a binary mask of the area to inpaint, and a textual description. Using an empty mask, we induce the diffusion steps to regenerate the input, that is, to generate a new image with exactly the same content as the input image. For the content augmentation process, we use the Stable Diffusion 2.1 inpainting method to replace an object with a new one, chosen from the same category or from a different one. Moreover, as shown in Fig. 4, besides the default inpainting, which regenerates the whole image, we consider also a version where the original background is restored. Note that during training, we balance the real and fake class taking an equal number of images from each.



(a) (b) (c)

Figure 5. Power spectra computed by averaging (2000 images) the differences between: (a) real and reconstructed images, (b) real and self-conditioned images, and (c) reconstructed and self-conditioned images. We can observe that the self-conditioned generation embeds forensic artifacts even at lower frequencies compared to reconstructed images. This means that it is possible to better exploit such inconsistencies to distinguish real from fakes.

In the following, we present our ablation study. To avoid dataset bias we use WildRF and Synthbuster. In addition, we test on 1000 FLUX and 1000 Stable Diffusion 3.5 images, which are some of the latest synthetic generators.

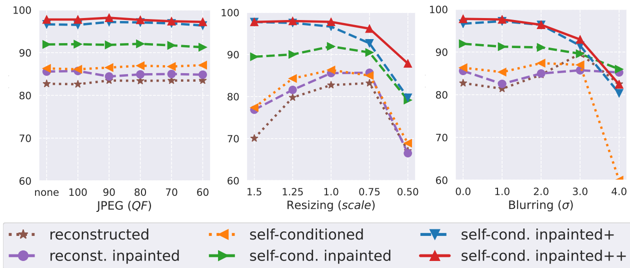


Figure 6. Robustness analysis in terms of balanced Accuracy carried out on nine generators of the Synthbuster dataset.

Architecture	FT	AUC↑	bAcc↑	NLL↓	ECE↓
DINOv2+reg	LP	80.8	68.5	0.58	.141
DINOv2+reg	e2e	**99.0**	**95.2**	**0.14**	**.040**
DINOv2	e2e	**98.4**	91.1	0.24	.077
SigLIP	e2e	95.4	89.9	0.28	.066

Table 3. We compare our solution, DINOv2+reg trained end-to-end, with linear probing (LP) and also consider alternative architectures, basic DINOv2 and SigLIP.

Architecture	Training Set	AUC↑	bAcc↑	NLL↓	ECE↓
CLIP/ViT	ProGAN [43]	54.7	45.2	4.85	.525
	LDM [10]	63.9	48.5	3.39	.487
	Ours	**75.2**	**73.9**	**0.66**	**.225**
RINE	ProGAN [43]	66.1	65.0	5.43	.312
	LDM [10]	83.0	75.7	0.53	.132
	Ours	**89.9**	**83.2**	**0.39**	**.089**

Table 4. Ablation study on the influence of the training data (ProGAN, Latent Diffusion and our dataset) on methods used for AI-generated image detection: CLIP [34] and RINE [22].

4.1. Influence of content alignment

In Tab. 2 we show the performance achieved with different dataset alignment strategies, as described in Fig. 3. Note that all variants are trained with standard augmentations, including blurring and JPEG compression, as proposed in [43]. From the Table, we can observe that there is a large gain in terms of balanced accuracy ($\simeq$20%) when moving from a text-driven generation (first row) to a solution where real and fake images share the semantic content, both reconstructed and self-conditioned. The proposed solution that uses a diffusion pass demonstrates further improvement on average across all the evaluation metrics. This is highlighted in Fig. 5, where we show the power spectra evaluated by averaging the difference between the real and reconstructed images and the real and self-conditioned images. We observe that self-conditioned generation introduces forensic artifacts even at the lowest frequencies, indicating a detector trained on such images can exploit inconsistencies on a broader range of frequencies.

4.2. Effect of content augmentation

We also analyze the effect of different content augmentation strategies (Fig. 4). We consider standard operations like cut-mix [48] and mix-up [51] and compare them with our proposed solutions that include three variants:

- *inpainted*, we replace an object with another from the same category plus the version where the background is substituted with pristine pixels (effectively a local image edit);
- *inpainted+*, we replace an object with another from both the same and a different category plus the corresponding versions where the background is substituted with pristine pixels;
- *inpainted++*, we further add some more standard augmentation operations, such as scaling, cut-out, noise addition, and jittering.

Overall, it is evident from Tab. 2 that augmentation plays a critical role in enhancing model generalization and this can be appreciated especially by looking at balanced accuracy and calibration measures. In fact, adding inpainting to reconstruction increases the accuracy from 80.7 to 83.9, while the joint use of self-conditioning and inpainting grants a significant extra gain, reaching 92.2 or even 96.4 with *inpainted++*. The most significant gains are observed on DALL·E 2, DALL·E 3 and FLUX that, probably, differ the most from Stable 2.1 in terms of architecture and hence require a stronger augmentation strategy to generalize.

In Fig. 6, we analyze the impact of our content augmentation, assessing robustness under various operations: JPEG compression, resizing, and blurring. All three proposed variants of augmentation offer a clear advantage, especially when resizing is applied. The joint use of self-conditioning and inpainting results in the most robust approach.

4.3. Influence of training data and architecture

We conduct additional experiments to gain deeper insights into the impact of the chosen architecture and the proposed training data on the same datasets shown in Tab. 2.

First we compare our adopted model, DINOv2+reg trained end-to-end, with an alternative fine-tuning strategy that involves training only the final linear layer, known as linear probing (LP) that is largely adopted in the literature [11, 30]. From Tab. 3 we can see that this latter solution does not perform well. One possible explanation is that features from last layer capture high-level semantics, while our dataset is built to exploit low-level artifacts that derive from first and intermediate layers [10, 22]. In the same Table we compare DINOv2+reg with the basic DINOv2 [31] architecture and SigLIP [49] and we can observe that DINOv2 with the use of registers achieves the best performance, probably thanks to the fact that it avoids to discard local patch information [13].

bAcc(%)↑/NLL↓	Synthbuster					New Generators		WildRF			AVG
	Midjourney	SDXL	DALL·E 2	DALL·E 3	Firefly	FLUX	SD 3.5	Facebook	Reddit	Twitter	bAcc↑/NLL↓
CNNDetect	49.5 / 8.45	49.8 / 6.90	50.2 / 5.75	49.5 / 12.9	50.3 / 3.66	49.5 / 10.1	50.0 / 5.30	50.0 / 9.84	50.7 / 6.66	50.1 / 8.64	50.0 / 7.83
DMID	**100. / 0.00**	**99.7 / 0.01**	50.1 / 5.99	50.0 / 7.08	51.0 / 1.72	63.7 / 1.27	**99.9 / 0.01**	87.8 / 0.52	74.3 / 1.82	79.1 / 0.85	75.6 / 1.93
LGrad	57.7 / 6.88	58.5 / 6.81	55.6 / 7.10	47.9 / 7.58	47.4 / 7.50	54.9 / 7.11	51.9 / 7.24	66.6 / 3.74	57.8 / 4.72	45.7 / 5.31	54.4 / 6.40
UnivFD	52.4 / 2.35	68.0 / 1.15	83.5 / 0.43	47.3 / 3.94	90.7 / 0.24	48.4 / 3.45	69.3 / 1.07	48.8 / 3.06	59.5 / 1.37	56.0 / 2.01	62.4 / 1.91
DeFake	69.7 / 0.72	76.3 / 0.56	64.0 / 0.92	84.9 / 0.36	72.4 / 0.63	79.2 / 0.46	81.2 / 0.42	66.3 / 0.89	65.9 / 0.82	63.4 / 0.94	72.3 / 0.67
DIRE	49.7 / 15.3	49.9 / 15.3	50.0 / 15.3	50.0 / 15.3	49.9 / 15.3	50.0 / 15.3	50.0 / 15.3	51.9 / 4.98	79.5 / 2.15	56.7 / 4.39	53.7 / 11.9
AntifakePrompt	70.4 / -	84.7 / -	65.5 / -	86.0 / -	70.0 / -	59.6 / -	60.7 / -	69.7 / -	68.9 / -	78.0 / -	71.3 / -
NPR	44.9 / 16.6	50.3 / 16.2	50.2 / 16.2	0.6 / 29.9	0.4 / 47.3	50.3 / 16.2	50.3 / 16.2	50.0 / 32.2	78.3 / 9.39	51.8 / 25.2	42.7 / 22.5
FatFormer	44.4 / 5.22	66.7 / 2.76	54.1 / 3.64	35.9 / 6.90	60.1 / 3.59	39.4 / 6.10	49.1 / 5.06	54.7 / 4.54	69.5 / 2.54	54.8 / 4.40	52.9 / 4.48
FasterThanLies	61.3 / 2.98	71.1 / 1.79	50.8 / 5.15	53.5 / 3.79	55.2 / 4.40	53.8 / 4.10	53.7 / 3.76	46.2 / 3.32	51.0 / 3.99	53.9 / 3.31	55.1 / 3.66
RINE	54.6 / 5.03	71.8 / 1.99	82.2 / 0.77	45.3 / 20.5	91.2 / 0.36	46.7 / 10.1	81.3 / 1.22	52.8 / 6.51	67.7 / 2.46	56.0 / 5.23	65.0 / 5.43
AIDE	57.5 / 0.95	68.4 / 0.70	34.9 / 1.34	33.7 / 1.38	24.8 / 2.00	62.9 / 0.82	63.3 / 0.82	56.9 / 0.94	72.1 / 0.62	57.3 / 1.01	53.2 / 1.06
LaDeDa	50.7 / 24.8	50.7 / 24.8	50.5 / 24.8	41.1 / 25.4	47.4 / 25.6	50.5 / 24.8	50.7 / 24.8	70.3 / 7.19	74.7 / 7.93	59.6 / 9.40	54.6 / 19.9
C2P-CLIP	52.8 / 1.10	77.7 / 0.48	55.6 / 0.99	63.2 / 0.73	59.5 / 0.89	50.1 / 1.30	60.9 / 0.93	54.4 / 0.97	68.4 / 0.67	57.4 / 0.91	60.0 / 0.90
CoDE	76.9 / 0.82	75.2 / 0.81	54.6 / 2.44	73.2 / 0.98	58.6 / 2.00	59.8 / 1.97	67.7 / 1.27	70.0 / 0.97	66.1 / 1.29	70.9 / 1.01	67.3 / 1.36
B-Free (ours)	**99.6** / 0.02	**99.8 / 0.01**	**95.6 / 0.10**	**98.2 / 0.06**	**99.7 / 0.02**	**92.3 / 0.19**	98.9 / 0.04	**95.6 / 0.14**	**86.2 / 0.28**	**97.3 / 0.09**	**96.3 / 0.10**

Table 5. Comparison with SoTA methods in terms of balanced Accuracy and balanced NLL across different generators. Note that AntifakePrompt [8] provides only hard binary labels hence calibration measures cannot be computed. Bold underlines the best performance for each column with a margin of 1%.

Then, in Tab. 4, we consider a CLIP-based model and the architecture of RINE, and vary the training dataset by including two well known datasets largely used in the literature, one based on ProGAN [43] and the other on Latent Diffusion [10]. We note that our training paradigm achieves the best performance over all the metrics with a very large gain (RINE increases the accuracy from 65% to 83.2%).

5. Comparison with the State-of-The-Art

In this Section, we conduct a comparison with SoTA methods on 27 diverse synthetic generation models. To ensure fairness, we include only SoTA methods with publicly available code and/or pre-trained models. The selected methods are listed in Table 6 and are further described in the supplementary material together with additional experiments. For all the experiments now on, *ours* refers to the detector trained using *inpainted++* augmentation.

A first experiment is summarized in Tab. 5 with results given in terms of balanced accuracy and NLL. Most of the methods struggle to achieve a good accuracy, especially on more recent generators. Instead, B-Free obtains a uniformly good performance on all generators, irrespective of the image origin, whether they are saved in raw format or downloaded from social networks, outperforming the second best (see last column) by +20.7% in terms of bAcc. Then, we evaluate again all methods on GenImage (unbiased), FakeInversion [7], and SynthWildX [11]. As these datasets encompass multiple generators, we only report the average performance in Tab. 7. On these additional datasets, most methods provide unsatisfactory results, especially in the most challenging scenario represented by SynthWildX, with images that are shared over the web. The proposed method performs well on all datasets, just a bit worse on FakeInversion. Finally in Fig. 7 we study how AUC compares with balanced accuracy for all the methods over several datasets. We observe that some methods, like NPR and LGrad, present a clear non-uniform behavior, with very good performance on a single dataset and much worse on the others. This seems to suggest that these methods may not be truly detecting forensic artifacts, instead are rather exploiting intrinsic biases within the dataset. Differently, the proposed method presents a uniform performance across all datasets and a small loss between AUC and accuracy.

Ref.	Acronym	Training Real/Fake	Size (K)	Aug.
[43]	CNNDetect	LSUN / ProGAN	360 / 360	✓
[10]	DMID	COCO, LSUN / Latent	180 / 180	✓
[38]	LGrad	LSUN / ProGAN	72 / 72	✓
[30]	UnivFD	LSUN / ProGAN	360 / 360	✓
[37]	DeFake	COCO / SD	20 / 20	
[44]	DIRE	LSUN-Bed / ADM	40 / 40	
[8]	AntifakePrompt	COCO / SD3,SD2-inp	90 / 60	✓
[39]	NPR	LSUN / ProGAN	72 / 72	
[26]	FatFormer	LSUN / ProGAN	72 / 72	
[23]	FasterThanLies	COCO / SD	108 / 542	✓
[22]	RINE	LSUN / ProGAN	72 / 72	✓
[45]	AIDE	ImageNet / SD 1.4	160 / 160	✓
[6]	LaDeDa	LSUN / ProGAN	360 / 360	
[40]	C2P-CLIP	LSUN / ProGAN	72 / 72	✓
[3]	CoDE	LAION / SD1.4, SD2.1, SDXL, DeepF. IF	2.3M / 9.2M	✓
	B-Free (ours)	COCO / SD2.1	51 / 309	✓

Table 6. AI-generated image detection methods used for comparison and whose code is publicly available. We specify source and size of the training dataset, and whether augmentation is applied.

Analysis on content shared on-line. Distinguishing real from synthetic images on social networks may be especially challenging due to the presence of multiple re-posting that impair image quality over time. A recent study conducted in [21] analyzed the detector behavior on different instances of an image shared online, showing that the per-

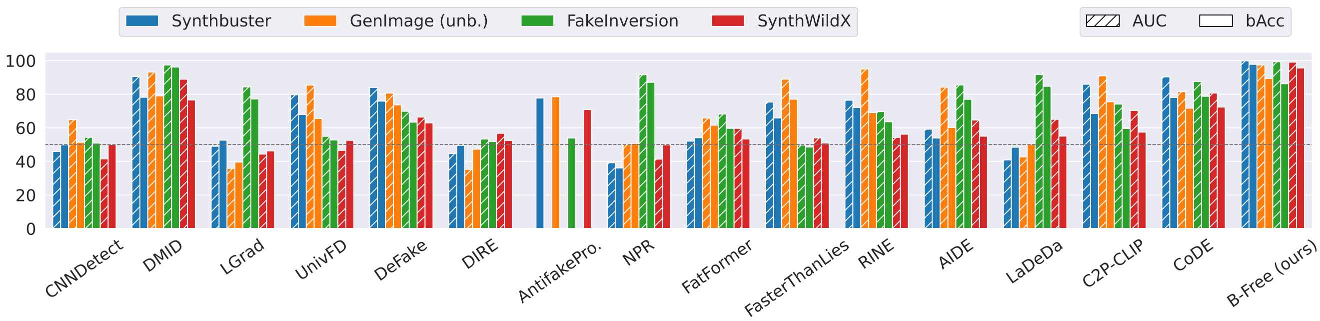


Figure 7. Average performance in term of AUC and bAcc on four datasets: Synthbuster, GenImage, FakeInversion, SynthWildX.

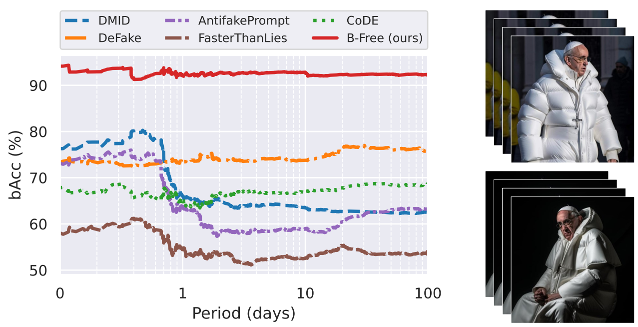


Figure 8. Results of SoTA detectors on real and fake images that went viral on internet, analyzing multiple web-scraped versions of each image. The performance is in terms of balanced accuracy evaluated from the initial online post (Log scale). Only detectors with an accuracy over 50% are shown.

bAcc(%)↑/NLL↓	GenImage	FakeInver.	SynthWildX	AVG
CNNDetect	51.3 / 7.88	50.9 / 7.94	50.0 / 8.08	50.7 / 7.96
DMID	79.0 / 1.66	**96.1 / 0.25**	76.6 / 0.82	83.9 / 0.91
LGrad	39.6 / 7.12	77.2 / 2.27	46.3 / 5.53	54.3 / 4.97
UnivFD	65.5 / 1.31	52.8 / 2.19	52.5 / 2.55	56.9 / 2.02
DeFake	73.7 / 0.74	63.3 / 0.95	62.9 / 0.98	66.6 / 0.89
DIRE	47.3 / 6.54	51.8 / 13.4	52.5 / 4.60	50.5 / 8.19
AntifakePrompt	78.5 / -	53.9 / -	70.8 / -	67.8 / -
NPR	50.7 / 25.3	87.0 / 4.96	49.9 / 28.2	62.6 / 19.5
FatFormer	61.5 / 3.99	59.7 / 3.45	53.3 / 4.75	58.2 / 4.06
FasterThanLies	77.0 / 1.23	48.6 / 3.64	50.9 / 3.40	58.8 / 2.76
RINE	69.1 / 2.57	63.6 / 4.84	56.2 / 6.07	63.0 / 4.49
AIDE	60.2 / 1.01	76.9 / 0.54	55.0 / 1.05	64.0 / 0.86
LaDeDa	50.2 / 29.2	84.7 / 3.03	55.1 / 10.2	63.3 / 14.1
C2P-CLIP	75.5 / 0.57	59.6 / 0.82	57.4 / 0.91	64.2 / 0.76
CoDE	71.7 / 1.43	78.8 / 0.74	72.3 / 0.95	74.2 / 1.04
B-Free (ours)	**89.3 / 0.27**	86.2 / 0.32	**95.6 / 0.14**	**90.4 / 0.24**

Table 7. Comparison with SoTA methods in terms of average performance in terms of balanced accuracy and NLL for three additional datasets: GenImage, FakeInversion and SynthWildX.

formance degrades noticeably in time due to repeated reposting. To better understand the impact of our augmentation strategies on such images, we collected a total of 1400 real/fake images that went viral on the web, including several versions of the same real or fake image.

Fig. 8 illustrates the accuracy, which is evaluated over a 100-day period from the time of initial publication, with times on a logarithmic scale. We compare our proposal with the best performing SoTA methods. We can notice that the performance drops after only one day, after which most competitors are stuck below 70%, with the exception of DeFake that achieves around 75%. Only the proposed method, which comprises more aggressive augmentation, is able to ensure an average accuracy around 92% even after many days from the first on-line post.

6. Limitations

The method proposed in this work is trained using fake images that are self-conditioned reconstructions from Stable Diffusion 2.1 model. If new generators will be deployed in the future that have a completely different synthesis process, then is it very likely that this approach will fail (the principles and ideas shared in this work may still hold). Further, being a data-driven approach it can be adversarially attacked by a malicious user. This is a very relevant issue that we plan to address in our future work.

7. Conclusions

In this paper, we propose a new training paradigm for AI-generated image detection. First of all, we empirically demonstrate the importance of pairing real and fake images by constraining them to have the same semantic content. This helps to better extract common artifacts shared across diverse synthetic generators. Then we find that using aggressive data augmentation, in the form of partial manipulations, further boosts performance both in term of accuracy and of calibration metrics. This is extremely relevant especially when working in realistic scenarios, such as image sharing over social networks. Our findings emphasize that careful dataset curation and proper training strategy can be more impactful compared to developing more complex algorithms. We hope this work will inspire other researchers in the forensic community to pursue a similar direction, fostering advancements in bias-free training strategies.

Acknowledgments. We gratefully acknowledge the support of this research by a Google Gift. In addition, this work has received funding from the European Union under the Horizon Europe vera.ai project, Grant Agreement number 101070093, and was partially supported by SERICS (PE00000014) under the MUR National Recovery and Resilience Plan, funded by the European Union - NextGenerationEU.

References

[1] Roberto Amoroso, Davide Morelli, Marcella Cornia, Lorenzo Baraldi, Alberto Del Bimbo, and Rita Cucchiara. Parents and Children: Distinguishing Multimodal Deep-Fakes from Natural Images. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2024. 3

[2] Quentin Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 2023. 2, 3, 4

[3] Lorenzo Baraldi, Federico Cocchi, Marcella Cornia, Lorenzo Baraldi, Alessandro Nicolosi, and Rita Cucchiara. Contrasting Deepfakes Diffusion via Contrastive Learning and Global-Local Similarities. In *ECCV*, 2024. 2, 3, 4, 7

[4] Clark Barrett, Brad Boyd, Elie Bursztein, Nicholas Carlini, Brad Chen, Jihye Choi, Amrita Roy Chowdhury, Mihai Christodorescu, Anupam Datta, Soheil Feizi, et al. Identifying and Mitigating the Security Risks of Generative AI. *Foundations and Trends® in Privacy and Security*, 6(1):1–52, 2023. 1

[5] Giuseppe Cattaneo and Gianluca Roscigno. A possible pitfall in the experimental analysis of tampering detection algorithms. In *International Conference on Network-Based Information Systems*, 2014. 2

[6] Bar Cavia, Eliahu Horwitz, Tal Reiss, and Yedid Hoshen. Real-Time Deepfake Detection in the Real-World. *arXiv preprint arXiv:2406.09398*, 2024. 4, 7

[7] George Cazenavette, Avneesh Sud, Thomas Leung, and Ben Usman. FakeInversion: Learning to Detect Images from Unseen Text-to-Image Models by Inverting Stable Diffusion. In *CVPR*, pages 10759–10769, 2024. 2, 3, 4, 7

[8] You-Ming Chang, Chen Yeh, Wei-Chen Chiu, and Ning Yu. AntifakePrompt: Prompt-Tuned Vision-Language Models are Fake Image Detectors. *arXiv preprint arXiv:2310.17419*, 2023. 7

[9] Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *CVPR Workshops*, pages 973–982, 2023. 3

[10] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP*, pages 1–5, 2023. 3, 4, 6, 7

[11] Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the Bar of AI-generated Image Detection with CLIP. In *CVPR Workshops*, pages 4356–4366, 2024. 2, 3, 4, 6, 7

[12] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. RAISE: a raw images dataset for digital image forensics. In *ACM MMSys*, page 219–224. Association for Computing Machinery, 2015. 2

[13] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers. In *ICLR*, 2024. 5, 6

[14] Jing Dong, Wei Wang, and Tieniu Tan. CASIA Image Tampering Detection Evaluation Database. In *IEEE ChinaSIP*, 2013. 2

[15] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: CNN based Generative Deep Neural Networks are failing to reproduce spectral distributions. In *CVPR*, pages 7890–7899, 2020. 3

[16] Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images. In *NeurIPS*, pages 3022–3032, 2020. 3

[17] Ziv Epstein, Aaron Hertzmann, et al. Art and the science of generative AI. *Science*, 380(6650):1110–1111, 2023. 1

[18] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In *ICME*, pages 1–6, 2021. 2

[19] Patrick Grommelt, Louis Weiss, Franz-Josef Pfreundt, and Janis Keuper. Fake or JPEG? Revealing Common Biases in Generated Image Detection Datasets. In *ECCV Workshops*, 2024. 2, 4

[20] Yonghyun Jeong, Doyeon Kim, Youngmin Ro, Pyounggeon Kim, and Jongwon Choi. FingerprintNet: Synthesized Fingerprints for Generated Image Detection. In *ECCV*, pages 76–94, 2022. 3

[21] Dimitrios Karageorgiou, Quentin Bammey, Valentin Porcellini, Bertrand Goupil, Denis Teyssou, and Symeon Papadopoulos. Evolution of Detection Performance throughout the Online Lifespan of Synthetic Images. In *ECCV Workshops*, 2024. 7

[22] Christos Koutlis and Symeon Papadopoulos. Leveraging Representations from Intermediate Encoder-blocks for Synthetic Image Detection. In *ECCV*, pages 394–411, 2024. 2, 3, 6, 7

[23] Romeo Lanzino, Federico Fontana, Anxhelo Diko, Marco Raoul Marini, and Luigi Cinque. Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks. In *CVPR Workshops*, pages 3771–3780, 2024. 3, 7

[24] Li Lin, Neeraj Gupta, Yue Zhang, Hainan Ren, Chun-Hao Liu, Feng Ding, Xin Wang, Xin Li, Luisa Verdoliva, and Shu Hu. Detecting multimedia generated by large AI models: A survey. *arXiv preprint arXiv:2204.06125*, 2024. 1

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 5

[26] Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware Adaptive Transformer for Generalizable Synthetic Image Detection. In *CVPR*, pages 10770–10780, 2024. 2, 3, 7

[27] Zhuang Liu and Kaiming He. A Decade's Battle on Dataset Bias: Are We There Yet? In *ICLR*, 2025. 2

[28] Sara Mandelli, Paolo Bestagini, and Stefano Tubaro. When Synthetic Traces Hide Real Content: Analysis of Stable Diffusion Image Laundering. In *WIFS*, pages 1–6, 2024. 3

[29] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do GANs Leave Artificial Fingerprints? In *MIPR*, pages 506–511, 2019. 2

[30] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, pages 24480–24489, 2023. 1, 2, 3, 4, 6, 7

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research Journal*, 2024. 5, 6

[32] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining Well Calibrated Probabilities Using Bayesian Binning. *AAAI*, 29(1), 2015. 4

[33] Joaquin Quiñonero-Candela, Carl Edward Rasmussen, Fabian Sinz, Olivier Bousquet, and Bernhard Schölkopf. Evaluating Predictive Uncertainty Challenge. In *Machine Learning Challenges Workshop*, pages 1–27, 2006. 4

[34] Alec Radford, JongWook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, pages 8748–8763, 2021. 1, 6

[35] Anirudh Sundara Rajan, Utkarsh Ojha, Jedidiah Schloesser, and Yong Jae Lee. On the Effectiveness of Dataset Alignment for Fake Image Detection. *arXiv preprint arXiv:2410.11835*, 2024. 2, 3

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. Stable Diffusion. `https://github.com/Stability-AI/stablediffusion`, 2022. 5

[37] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. DE-FAKE: Detection and Attribution of Fake Images Generated by Text-to-Image Generation Models. In *ACM SIGSAC*, pages 3418–3432, 2023. 3, 7

[38] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on Gradients: Generalized Artifacts Representation for GAN-Generated Images Detection. In *CVPR*, pages 12105–12114, 2023. 3, 7

[39] Chuangchuang Tan, Huan Liu, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the Up-Sampling Operations in CNN-based Generative Network for Generalizable Deepfake Detection. In *CVPR*, 2024. 7

[40] Chuangchuang Tan, Renshuai Tao, Huan Liu, Guanghua Gu, Baoyuan Wu, Yao Zhao, and Yunchao Wei. C2P-CLIP: Injecting Category Common Prompt in CLIP to Enhance Generalization in Deepfake Detection. *arXiv preprint arXiv:2408.09647*, 2024. 3, 7

[41] Diangarti Tariang, Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Synthetic Image Verification in the Era of Generative AI: What Works and What Isn't There Yet. *IEEE Security & Privacy*, 22: 37–49, 2024. 1, 4

[42] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 2

[43] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*, pages 8695–8704, 2020. 3, 4, 5, 6, 7

[44] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. DIRE for diffusion-generated image detection. In *ICCV*, pages 22445–22455, 2023. 3, 7

[45] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A Sanity Check for AI-generated Image Detection. In *ICLR*, 2025. 7

[46] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion Probabilistic Model Made Slim. In *CVPR*, pages 22552–22562, 2023. 3

[47] Ning Yu, Larry S Davis, and Mario Fritz. Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints. In *ICCV*, pages 7556–7566, 2019. 2

[48] 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 *ICCV*, pages 6023–6032, 2019. 6

[49] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11975–11986, 2023. 6

[50] Fangneng Zhan, Yingchen Yu, Rongliang Wu, Jiahui Zhang, Shijian Lu, Lingjie Liu, Adam Kortylewski, Christian Theobalt, and Eric Xing. Multimodal Image Synthesis and Editing: The Generative AI Era. *IEEE TPAMI*, 45(12): 15098–15119, 2021. 1

[51] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *ICLR*, 2018. 6

[52] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and Simulating Artifacts in GAN Fake Images. In *WIFS*, 2019. 3

[53] Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. GenImage: A Million-Scale Benchmark for Detecting AI-Generated Image. *NeurIPS*, 36:77771–77782, 2023. 4