HAL open science

CLIP-Flow: A Universal Discriminator for AI-Generated Images Inspired by Anomaly Detection

Zhipeng Yuan, Kai Wang, Weize Quan, Dong-Ming Yan, Tieru Wu

To cite this version:

Zhipeng Yuan, Kai Wang, Weize Quan, Dong-Ming Yan, Tieru Wu. CLIP-Flow: A Universal Discriminator for AI-Generated Images Inspired by Anomaly Detection. 1st Deepfake Forensics Workshop: Detection, Attribution, Recognition, and Adversarial Challenges in the Era of AI-Generated Media (DFF 2025) at ACM Multimedia 2025, Oct 2025, Dublin, Ireland. pp.1-9, 10.1145/3746265.3759663 . hal-05296631

HAL Id: hal-05296631

https://hal.science/hal-05296631v1

Submitted on 7 Oct 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

CLIP-Flow: A Universal Discriminator for AI-Generated Images Inspired by Anomaly Detection

Zhipeng Yuan, Kai Wang, Weize Quan, Dong-Ming Yan, and Tieru Wu

Abstract

With the rapid advancement of AI generative models, the visual quality of AI-generated images (AIIs) has become increasingly close to natural images, which inevitably raises security concerns. Most AII detectors often employ the conventional image classification pipeline with natural images and AIIs (generated by a generative model), which can result in limited detection performance for AIIs from unseen generative models. To solve this, we proposed a universal AI-generated image detector from the perspective of anomaly detection. Our discriminator does not need to access any AIIs and learn a generalizable representation with unsupervised learning. Specifically, we use the pre-trained CLIP encoder as the feature extractor and design a normalizing flow-like unsupervised model. Instead of AIIs, proxy images, *e.g.*, obtained by applying a spectral modification operation on natural images, are used for training. Our models are trained by minimizing the likelihood of proxy images, optionally combined with maximizing the likelihood of natural images. Extensive experiments demonstrate the effectiveness of our method on AIIs produced by various image generators.

Index Terms

Image forensics, anomaly detection, normalizing flow, CLIP

I. Introduction

Nowadays, we can easily see fake pictures on the Internet, many of which are synthetic images generated by deep learning algorithms. Indeed, with the rapid development and maturity of deep generative models (such as generative adversarial models and diffusion models), people can easily generate the desired high-quality images. Although image generation technology benefits our lives in some aspects, there will inevitably be people who will use this technology maliciously. For example, AI-generated images (AIIs) can be used for personal and political attacks, making it easier for people to believe the content of fake news. Therefore, it is urgent to develop a universal discriminator of AIIs created by various generative models and tools.

By choosing a classifier and using two groups of natural images and AIIs as training samples, we can learn an image discriminator in a supervised manner, which is shown in the top of Fig. 1. This approach performs well in identifying generated images similar to those in the training set, but when tested on other types of generated images, the performance can drop drastically. For example, a classifier trained using natural images and ProGAN-generated images [1] can well identify images generated by ProGAN or generative models with similar structures to ProGAN (such as StyleGAN [2]). However, when faced with images created by other generative models with different structures (such as diffusion models), the classification performance can be very poor. Unfortunately, in the real world, new generative models are constantly emerging, and it is not realistic to update trained detectors at all times to deal with these new models. Our goal in this paper is to achieve high generalization performance for the AII detector, *i.e.*, with high classification accuracy on AIIs created by unknown generative models, by exploring a new approach inspired by anomaly detection.

As mentioned above and observed in the influential paper of [3], the supervised learning approach may have limitations in learning only the very specific artifacts present in the AIIs in the training set, leading to limited generalization capability. This has motivated us to tackle the detector generalization problem by exploiting a different learning mechanism of anomaly detection [4]. In existing anomaly detection methods, training is usually performed only on normal samples and any sample that deviates significantly from the normal samples will be classified as an anomaly. Intuitively, in our AII detection task, normal samples correspond to natural images. Given this idea, we started by considering using only natural images to train anomaly detection methods. Unfortunately, limited generalization performance has been achieved with various anomaly detection algorithms and feature extractors, including the well-known CLIP embedding extractor [5] which has proven its great utility in AII detection as shown in [3]. One possible explanation for this limited generalization is the difficulty in using existing anomaly detection algorithms to capture the highly diverse representations of natural images. This has led us to reconsider the problem formulation and to better adapt the anomaly detection mechanism to our task of AII detection.

We assume that natural images follow an underlying distribution that is different from AIIs. As mentioned previously, it is difficult to identify AIIs by modeling this distribution, which has motivated us to rethink the problem from a new perspective

Z. Yuan and T. Wu are with Jilin University, Jilin, China. (Corresponding author: Tieru Wu.)

K. Wang is with Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, Grenoble, France.

W. Quan and D.-M. Yan are with the MAIS & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and also with School of Artificial Intelligence, the University of Chinese Academy of Sciences, Beijing, China.

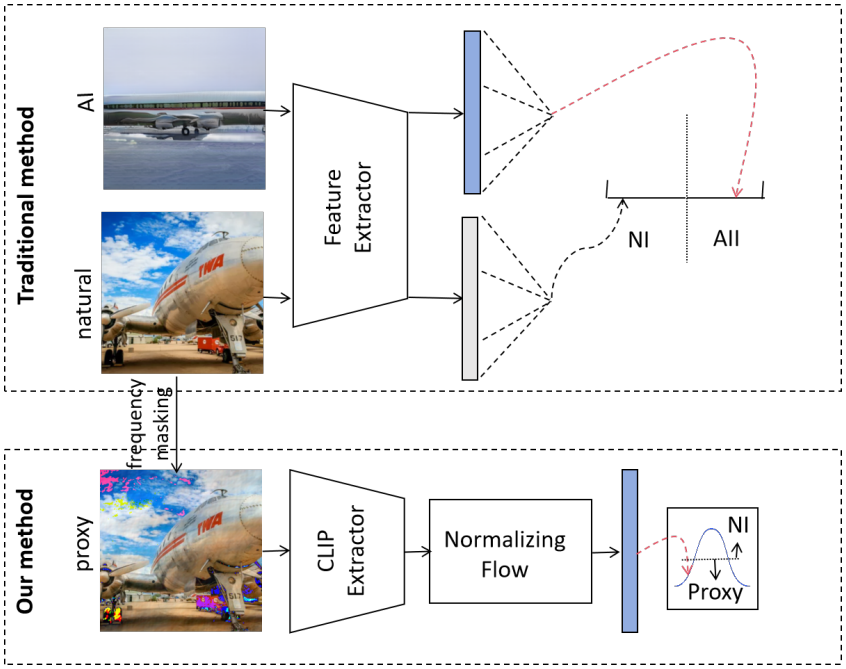

Fig. 1: The previous supervised methods used binary classification of natural image (NI) and AI-generated image (AII), and we use the flow model for likelihood optimization of proxy image (optionally also with NI) from the perspective of anomaly detection.

with a focus on the so-called proxy images whose distribution deviates from that of natural images. Indeed, it is believed that despite the high visual quality, synthetic images in general exhibit deviations in the spatial domain (*e.g.*, different correlations between pixel values) [6] or the frequency domain (*e.g.*, artifacts in certain frequency bands) [7], [8]. Following this idea, we apply spatial or spectral operation (*e.g.*, the random frequency masking [8] which shows the best performance in our study) on natural images to construct a group of processed images, regard them as anomalous samples, extract their CLIP features, and train a detector by minimizing their likelihood based on a normalizing flow-like model, a popular tool for anomaly detection [9]–[11] (bottom of Fig. 1). Well-constructed proxy images are expected to be closer to AIIs than to natural images in the feature space. This "unconventional" anomaly detection framework called *CILP-Flow* shows very good performance on benchmark test sets of natural image and AII classification, notably by not using any AII during training. We further improve the generalization performance by proposing a self-supervised (in a loose sense) framework in which proxy images and their corresponding natural images are used together to train the flow model (still without using any AII during training), with minimization and maximization of the likelihood of the two types of images, respectively.

Our contributions are summarized as follows:

- To the best of our knowledge, we conducted a careful and exploratory study on the usefulness of methods in the field of anomaly detection to solve the problem of AII detection.
- When we only use anomalous frequency-masked proxy images for training, we achieve very competitive AII detection performance on popular benchmark test sets with an mAP of 93.85% and an accuracy of 84.93%.
- Our model has good flexibility and can also be trained in a self-supervised manner: when trained using both natural images and proxy images, the model achieves higher detection performance with an mAP of 95.56% and an accuracy of 87.30%.
- We proposed a novel approach to solving the AII detection problem that is inspired by anomaly detection, and we hope that our work can inspire future efforts in this promising research direction.

II. Related Work

A. AI-Generated Image Detection

Spatial-Based Approaches. This kind of methods applies deep models to learn discrimination features in the pixel space directly. Wang *et al.* [12] explored two simple image augmentation operations (*i.e.*, JPEG compression and Gaussian blur) to improve the CNN-based detector. Chai *et al.* [13] constructed a patch-based AII detector using the truncated part of CNN backbones (*e.g.*, Xception and ResNet50). Natraj *et al.* [14] combined the co-occurrence matrix of the input pixels and CNN to distinguish between natural images and AIIs, while Bai *et al.* [15] identified AI-generated images based on co-occurrence relations in the feature channels of CNN model. Ojha *et al.* [3] applied the ViT encoder of the pre-trained CLIP to detect

fake images and obtained decent generalization performance. With the help of a pre-trained diffusion model, Wang *et al.* [16] extracted the reconstruction error between the input image and its reconstructed version to distinguish between real images and diffusion-generated images.

Frequency-Based Approaches. This type of methods identifies fake images based on revealing traces in the frequency domain. Zhang *et al.* [17] and Frank *et al.* [18] designed classifiers that take the frequency spectrum as input, attempting to capture the artifacts caused by the up-sampler of GAN models. Durall *et al.* [19] and Dzanic *et al.* [20] developed AI-generated image discriminators based on the difference between natural images and AIIs in the high-frequency component. Liu *et al.* [21] utilized the frequency spectrum information of learnable image noise patterns to detect AIIs. Recently, authors of [8] and [22] explored frequency-based modification to improve the generalization of an AII detector with traditional binary cross-entropy loss.

B. Anomaly Detection

Anomaly detection usually requires that only normal samples are used during training, and the model is asked to distinguish whether a sample is normal or anomalous during testing. Typical applications of anomaly detection include industrial defect detection, disease detection, video surveillance, *etc.* [4]. Recent anomaly detection methods for images can be roughly grouped into three main categories: reconstruction-based methods, distillation-based methods, and density-based methods.

Reconstruction-Based Methods. This kind of methods reconstructs normal samples in the training stage, and classifies samples that cannot be well reconstructed in testing as anomalies. GANomaly [23] uses the GAN structure as a reconstruction model. DRAEM [24] uses an autoencoder as a reconstruction model and, for better performance, also trains a discriminator to identify artificially generated anomalous samples using Perlin noise.

Distillation-Based Methods. STFPM [25] attempts to distinguish anomalous samples by training a student model to learn the intermediate layer features extracted by a teacher model for normal samples. After such a distillation procedure, the student model tends to have a higher loss on anomalous samples. RD [26] is another distillation-based method, where a one-class embedding is learned to represent normal data.

Density-Based Methods. Such methods extract features of normal samples and model the distribution of these features. An anomaly score can be derived usually based on the probability of a sample under the learned distribution. Different methods mainly differ in terms of feature extractor and distribution modeling [27], [28]. For example, PaDiM [28] detects anomalies by extracting features of local image patches and then learning a multivariate Gaussian distribution of normal patches during training. FastFlow [10] and CFlow [11] use the normalizing flow, a powerful generative model based on neural network, for the distribution learning. Considering the conceptional and computational simplicity of density-based approaches compared to other methods, we use the normalizing flow model in our work for AI-generated image discrimination.

III. Proposed Method

The proposed method is illustrated in Fig. 2. Unlike previous supervised detectors, our method does not need any AI-generated image during training. Instead, we create and use proxy images, *e.g.*, obtained by applying random frequency masking on natural images. We then use the CLIP model to extract features from the proxy images, adapt them with dimension reduction (DR) and normalization, and turn the distribution of adapted features into a normal distribution using unsupervised normalizing flow. Our model can also be jointly trained with unprocessed natural images (upper part of Fig. 2). During testing, we can calculate an anomaly score based on the flow model to classify images.

A. Feature Extractor

In many computer vision tasks, the ViT encoder of the large-scale pre-trained CLIP model [5] is often used as the feature extraction backbone. Similarly, Ojha *et al.* [3] and Li and Wang [22] observed superior forensic performance in AI-generated image detection with CLIP features compared to other popular image embeddings. We have similar observations in our preliminary experiments when combining different image features with the flow model. Therefore, in order to construct a universal AII discriminator, in our work, we also choose the ViT encoder of CLIP to extract the image feature. Particularly, ViT extracts a high-dimensional feature from an image x, and we reduce its dimension to balance the performance and the computational cost. We then normalize it, and the feature adaptation process is written as follows:

$$z = \frac{DR(\phi(x))}{\|DR(\phi(x))\|_2}, \tag{1}$$

where $\phi(.)$ is the CLIP feature extraction function and $DR(.)$ is the dimension reduction operation with a linear layer.

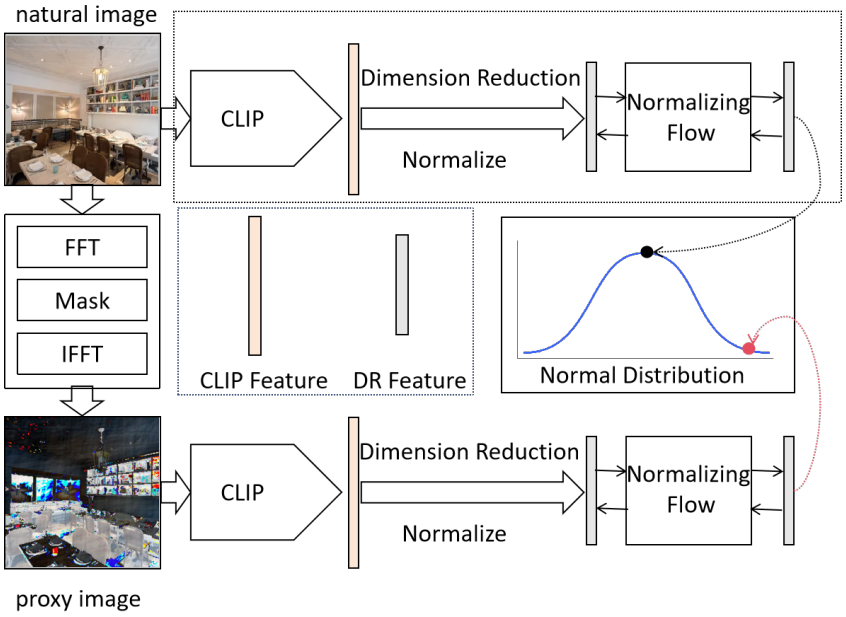

Fig. 2: Framework of CLIP-Flow.

B. Proxy Image Construction

We have attempted different methods to construct proxy images. Comparative experimental results are presented in Section IV, which show that operations in either spatial or spectral domain can work well in our proposed framework and that the frequency masking operation described below achieves the best detection performance.

There are many works that exploit frequency information for AII detection, with the basic idea that there is a telltale difference between natural and generated images in the frequency domain of the image. We also use frequency information, more specifically by applying random frequency masking [8] to process the original image. This is combined with a novel learning framework inspired by anomaly detection to achieve good generalization and flexibility. Let x be a natural image, we first use the Fourier transform to convert it to the frequency domain. Then, we produce a random binary mask and use it to mask a ratio of a certain part (low, mid, high) of the frequency spectrum (please refer to [8] for more details). We convert this masked frequency spectrum back to RGB space with the inverse Fourier transform to obtain a proxy image $x^{'}$. The whole process is as follows:

$$x^{'} = ifft(mask \odot fft(x)), \tag{2}$$

where $fft(.)$ represents Fast Fourier transform, $mask$ is a random binary mask, $\odot$ is the element-wise multiplication, and $ifft(.)$ represents inverse Fast Fourier transform.

As mentioned in Section I, performance remains limited if we train only on natural images x with anomaly detection algorithms, possibly due to the high diversity of natural images and the blindness regarding the anomalies to be detected. Instead, decent performance is achieved if we train a flow model only with proxy images $x^{'}$ by minimizing their likelihood values (bottom branch of Fig. 2). This implies that frequency-masked proxy images are in general closer to AIIs than to natural images in the feature space. Performance can be further improved if we use both natural images x and proxy images $x^{'}$ for training (top and bottom branches of Fig. 2), by maximizing and minimizing the likelihood value, respectively. This can be loosely considered as a self-supervised learning method that demonstrates strong zero-shot performance on the AII detection task. Both learning methods are inspired by anomaly detection and, to the best of our knowledge, are new in the AII detection literature.

C. Normalizing Flow and Training Objective

For distribution learning and sample likelihood estimation, we choose the normalizing flow model [29], [30], which is a popular probabilistic method for generative modeling. It is able to map a simple probability distribution (such as the Gaussian distribution) to a complex distribution via a series of reversible transformations with the trick of change of variable. With the flow model, it is also possible to transform a complex distribution into a simple one, *e.g.*, the Gaussian distribution. In our CLIP-Flow method, we assume that the distribution of the adapted CLIP feature z of natural image follows a (complex)

TABLE I: The performance (in AP, %) of our model and existing methods. In the last group of rows, N means model trained only on natural images with likelihood maximization, P means model trained only on proxy images with likelihood minimization, and N+P means model trained on both types of images above with likelihood maximization and minimization respectively. The symbols are the same in the following tables.

Methods	Variant	Generative Adversarial Networks						Deep Fakes	Low level vision		Perceptual loss		Guided	LDM			Glide			DALL-E	Total
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN		SITD	SAN	CRN	IMLE		200 steps	200 w/ CFG	100 steps	100 27	50 27	100 10		mAP
Anomaly Detection	PaDiM	43.04	40.86	47.06	45.58	39.86	42.13	46.21	48.40	54.34	42.18	39.22	41.04	52.62	59.57	51.56	40.77	38.81	41.03	73.61	46.73
	CFA	57.62	66.60	56.41	71.13	68.63	70.23	41.54	66.38	51.65	77.34	83.02	68.64	54.50	46.34	53.94	69.44	74.36	69.00	40.95	62.51
	GANomaly	48.93	49.39	52.33	44.37	45.80	49.51	52.97	49.74	62.29	37.57	38.88	39.49	50.41	56.47	48.60	39.71	37.01	40.67	56.38	47.39
	DSR	50.35	45.80	50.04	48.46	51.29	45.79	56.47	47.79	52.07	40.44	54.58	47.48	45.86	48.02	45.60	46.87	45.37	49.70	47.36	48.39
	DRAEM	92.73	43.24	59.50	94.27	51.48	97.15	48.54	41.53	51.09	47.00	40.52	41.73	95.28	90.72	95.33	43.59	43.54	42.67	98.35	64.12
	STFPM	53.60	55.16	47.89	45.88	53.32	62.27	51.11	52.85	58.08	44.64	49.31	41.79	51.69	53.11	51.11	43.77	41.05	46.75	60.58	50.73
	RD	47.46	46.54	48.43	40.60	41.96	49.92	49.52	42.62	54.34	34.29	32.21	38.32	47.02	43.06	47.28	34.74	33.45	33.85	50.11	42.93
	DFM	49.00	50.92	47.02	48.86	46.67	50.00	49.69	49.16	53.43	45.71	42.57	47.90	59.60	58.85	59.79	52.64	49.52	52.18	59.25	51.20
	FastFlow	49.08	49.49	44.68	45.81	47.58	49.37	53.00	48.30	55.53	62.88	61.13	48.58	54.58	54.95	55.12	54.95	53.69	55.01	64.40	53.06
	CFlow	48.97	55.10	51.05	50.44	47.34	45.87	48.09	84.65	49.38	45.88	53.92	40.88	75.21	59.88	76.73	50.40	47.72	44.60	56.43	54.33
CNNSpot	Blur+JPEG (0.1)	**100.00**	93.47	84.50	99.54	89.49	98.15	89.02	73.75	59.47	98.24	98.40	73.72	70.62	71.00	70.54	80.65	84.91	82.07	70.59	83.58
	Blur+JPEG (0.5)	**100.00**	96.83	88.24	98.29	98.09	95.44	66.27	86.00	61.20	**98.94**	**99.52**	68.57	66.00	66.68	65.39	73.29	78.02	76.23	65.93	81.52
	ViT:CLIP (B+J 0.5)	99.98	93.32	83.63	88.14	92.81	84.62	67.23	**93.48**	55.21	88.75	96.22	55.74	52.52	54.51	52.20	56.64	61.13	56.64	62.74	73.44
PatchForensics	ResNet50-Layer1	98.86	72.04	68.79	92.96	55.90	92.06	60.18	65.82	52.87	68.74	67.59	70.05	87.84	84.94	88.10	74.54	76.28	75.84	77.07	75.28
	Xception-Block2	80.88	72.84	71.66	85.75	65.99	69.25	76.55	76.19	76.34	74.52	68.52	75.03	87.10	86.72	86.40	85.37	83.73	78.38	75.67	77.73
Co-Occur	-	99.74	80.95	50.61	98.63	53.11	67.99	59.14	68.98	60.42	73.06	87.21	70.20	91.21	89.02	92.39	89.32	88.35	82.79	80.96	78.11
FreqSpec	CycleGAN	55.39	**100.00**	75.08	55.11	66.08	**100.00**	45.18	47.46	57.12	53.61	50.98	57.72	77.72	77.25	76.47	68.58	64.58	61.92	67.77	66.21
DIRE	-	**100.00**	77.12	76.30	92.64	71.08	**100.00**	**95.32**	62.02	63.30	95.77	94.03	83.34	95.19	93.89	95.12	93.78	95.56	95.99	61.27	86.41
UnivFD	LC	**100.00**	99.46	**99.59**	97.24	**99.98**	99.60	82.45	61.32	79.02	96.72	99.00	87.77	99.14	92.15	99.17	94.74	95.34	94.57	97.15	93.38
Freq-mask	CLIP	**100.00**	99.04	96.89	92.66	99.83	98.86	77.17	67.41	75.80	78.83	95.85	81.77	95.45	80.93	96.48	90.66	91.71	90.23	86.73	89.28
OnlyReal	-	99.99	**100.00**	99.48	**99.77**	99.96	**100.00**	94.75	40.65	68.01	89.25	98.96	**89.93**	**99.90**	**99.64**	**99.91**	96.55	97.60	96.36	**99.99**	93.18
Ours	N	87.99	63.86	74.00	57.69	63.39	93.40	66.89	40.37	51.34	37.26	45.58	61.18	42.24	42.90	42.50	34.73	35.75	34.79	54.43	54.23
	P	99.86	98.40	98.38	89.42	99.84	98.13	92.23	84.18	72.09	98.14	98.79	76.29	98.49	94.33	98.46	95.49	97.07	96.48	97.15	93.85
	N+P	99.94	99.87	99.24	97.10	99.92	99.50	87.34	80.06	**79.48**	94.85	96.94	87.78	99.41	96.93	99.44	**99.35**	**99.59**	**99.28**	99.55	**95.56**

distribution as $z \sim p_Z(z)$, the chosen simple distribution is denoted $u \sim p_U(u)$, and the reversible bijective transformation between the two distributions is denoted $g^{-1} : Z \to U$. Then, the log-likelihood of any $z \in Z$ can be estimated as:

$$\begin{aligned} \log p_Z(z, \theta) &= \log p_U(u) + \log |\det(J)| \\ &= \log p_U(g^{-1}(z)) + \log |\det(J)|, \end{aligned} \tag{3}$$

where the matrix $J = \nabla_z g^{-1}(z; \theta)$ is the Jacobian of a bijective invertible flow model parameterized by θ.

We then calculate for z a KL divergence to measure the difference between its distribution and the approximated distribution with the transformation of flow model:

$$\begin{aligned} D_{KL}[p_Z^*(z)||p_Z(z)] &= -\mathbb{E}_{p_Z^*(z)}[\log p_Z(z; \theta) - \log p_Z^*(z)] \\ &= -\mathbb{E}_{p_Z^*(z)}[\log p_Z(z; \theta)] + c_1, \end{aligned} \tag{4}$$

where p_Z^* represents the true distribution of z, p_Z represents the approximated distribution by using the flow model parameterized by θ, and c_1 is a term independent of θ.

Here we propose a *new* flow model which can handle two types of features having "complementary" distributions. More specifically, for feature $z^{'}$ of proxy image, we assume that it follows a distribution related to that of natural image as $z^{'} \sim K/p_Z(z^{'})$ (under the mild condition of $p_Z(z^{'}) \neq 0$, $\forall z^{'}$), where K is a normalization constant to ensure a valid probability density function integrating to 1. This is a reasonable assumption because at certain position in feature space with high probability density of natural image, it certainly leads to low probability density for artificial proxy image at that position, and *vice versa*. Similar to Eq. (4), we have:

$$\begin{aligned} D_{KL}[p_{Z'}^*(z^{'})||\frac{K}{p_Z(z^{'})}] &= -\mathbb{E}_{p_{Z'}^*(z')}[\log(\frac{K}{p_Z(z'; \theta)})] + c_2 \\ &= \mathbb{E}_{p_{Z'}^*(z')}[\log p_Z(z^{'}; \theta)] + c_3, \end{aligned} \tag{5}$$

where $p_{Z'}^*$ represents the true distribution of $z^{'}$, K/p_Z is the approximated distribution by using the flow model parameterized by θ, and c_2 and c_3 are terms independent of θ.

For the flow model to work properly, we need to minimize the KL divergences in Eqs. (4) and (5), which elegantly leads to a maximization of the log-likelihood of natural images and a minimization of the log-likelihood of proxy images, according to these two equations (please note their different signs for the non-constant term). Therefore, we use the likelihood estimation to obtain the following loss function to minimize:

$$\mathcal{L}(\theta) = -\frac{1}{N_o} \sum_{n=1}^{N_o} \log p_Z(z_n; \theta) + \frac{1}{N_p} \sum_{m=1}^{N_p} \log p_Z(z_m^{'}; \theta), \tag{6}$$

where N_o is the number of natural images, N_p is the number of proxy images, and z_n and $z_m^{'}$ represent respectively the feature of the natural and proxy image. The first term on the right side of Eq. (6) means to maximize the likelihood of natural images, while the second term means to minimize the likelihood of proxy images. We have a so-called self-supervised training when we use both terms. In contrast, when we use only one type of images for unsupervised training, one of the two terms vanishes, *e.g.*, the first term vanishes when we train the flow model using only proxy images.

We set the transformed distribution as a multidimensional Gaussian distribution $u \sim \mathcal{N}(0, I)$. At the same time, since the loss is the sum of each dimension, following [11] we divide it by the number of dimensions C to normalize the loss. Finally, the training objective of the flow model is to minimize the following loss function:

$$\begin{aligned} Loss = &- \frac{1}{2N_o} \sum_{n=1}^{N_o} \frac{\|u_n\|_2^2 - 2\log|\det(J)|}{C} \\ &+ \frac{1}{2N_p} \sum_{m=1}^{N_p} \frac{\|u_m^{'}\|_2^2 - 2\log|\det(J)|}{C}, \end{aligned} \tag{7}$$

where $u_n = g^{-1}(z_n;\theta)$ and $u_m^{'} = g^{-1}(z_m^{'};\theta)$ are the transformed features of the flow model corresponding to the natural image and the proxy image, respectively.

D. Anomaly Score

Our goal is to determine whether an image is a natural image or an AII. For this purpose and as shown in Eq. (8) below, we naturally define and use an anomaly score based on the *negative* log-likelihood of the sample. The normalized log-likelihood term in Eq. (8) is exactly the same as that used in the flow model loss function in Eq. (7), where we minimize (or maximize) the log-likelihood of proxy images (or natural images). Therefore, natural images tend to have low values of this score (because of the negative sign before the log-likelihood term in the definition of the anomaly score), while AI-generated images tend to have high values.

$$score = -\frac{\log p_Z(z;\theta)}{C} = \frac{\|u\|_2^2 - 2\log|\det(J)|}{2C}. \tag{8}$$

IV. Experiments

A. Experimental Settings

Datasets. Following the representative methods on AII detection [3], [12], we used natural images from LSUN for training, with a size of 256×256 pixels and 20 categories. Detection models are evaluated on the benchmark (in total 19 datasets) shared by the above two references [3], [12], including six GAN models: ProGAN, CycleGAN, BigGAN, StyleGAN, GauGAN, and StarGAN; DeepFakes; two low-level vision models: SITD and SAN; two perceptual loss-based models: CRN and IMLE; three diffusion models: Guided, LDM (latent diffusion model) with three different generation modes, and Glide with three different variants; DALL-E. These tested samples have diverse content and can be used to comprehensively assess the performance of detection models.

Metrics. Following [3], [12], we use the average precision (AP) and classification accuracy for comparison. AP is a comprehensive reflection of the classification ability of our anomaly scores. In order to find a fixed threshold to calculate the accuracy, we created a validation set. Positive samples are LSUN natural images that are neither in the training set nor in the testing set, and negative samples are frequency-perturbed images by masking the frequency band of (30, 100), generated in a different way than the frequency masking used for the training set.

Implementation Details. Our model is implemented with PyTorch and trained on NVIDIA GeForce RTX GPU. We use Adam optimizer with a learning rate of 10^{-4} and batch size of 128. CLIP: ViT-L/14 is the feature extractor. The bijection mapping in the flow model is implemented with an affine coupling layer [29]. Proxy images are generated by random frequency masking [8] with the masking ratio of 1.0 (for P mode, our method trained only with proxy images) and 0.1 (for N+P mode, our method trained with both natural and proxy images) in the low part of the spectrum. Experimentally, for P mode, it is advantageous to use proxy images with large changes, in order to confidently minimize likelihood values on them. For N+P mode, since it is a self-supervised approach, we can use proxy images with small changes for strict distinction under the setting of a hard discrimination problem. Proxy images constructed with frequency masking achieve the best AII detection performance, and the results obtained with other types of proxy images are presented and compared later in this section. During training, all images are resized to 256×256 with bilinear interpolation and then randomly cropped to 224×224. The normalization uses the same mean and variance as in the CLIP model [5]. The P mode takes 30 epochs and the N+P mode takes 10 epochs. During testing, the central 224×224 crop of the 256×256 resized image is used for evaluation.

TABLE II: The performance (in Accuracy, %) of our model and existing methods.

Methods	Variant	Generative Adversarial Networks						Deep Fakes	Low level vision		Perceptual loss		Guided	LDM			Glide			DALL-E	Total
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN		SITD	SAN	CRN	IMLE		200 steps	200 w/ CFG	100 steps	100 27	50 27	100 10		Avg. acc
CNNSpot	Blur+JPEG (0.1)	99.99	85.20	70.20	85.70	78.95	91.70	53.47	66.67	48.69	86.31	86.26	60.07	54.03	54.96	54.14	60.78	63.80	65.66	55.58	69.58
	Blur+JPEG (0.5)	**100.00**	80.77	58.98	69.24	79.25	80.94	51.06	56.94	47.73	87.58	94.07	51.90	51.33	51.93	51.28	54.43	55.97	54.36	52.26	64.73
	Oracle (B+J 0.5)	**100.00**	90.88	82.40	93.11	93.52	87.27	62.48	**76.67**	57.04	**95.28**	**96.93**	65.20	63.15	62.39	61.50	65.36	69.52	66.18	60.10	76.26
	ViT:CLIP (B+J 0.5)	98.94	78.80	60.62	60.56	66.82	62.31	52.28	65.28	47.97	64.09	79.54	50.66	50.74	51.04	50.76	52.15	53.07	52.06	53.18	60.57
PatchForensics	ResNet50-Layer1	94.38	67.38	64.62	82.26	57.19	80.29	55.32	64.59	51.24	54.29	55.11	65.14	79.09	76.17	79.36	67.06	68.55	68.04	69.44	68.39
	Xception-Block2	75.03	68.97	68.47	79.16	64.23	63.94	75.54	75.14	**75.28**	72.33	55.30	67.41	76.50	76.10	75.77	74.81	73.28	68.52	67.91	71.24
Co-Occur	-	97.70	63.15	53.75	92.50	51.10	54.70	57.10	63.06	55.85	65.65	65.80	60.50	70.70	70.55	71.00	70.25	69.60	69.90	67.55	66.86
FreqSpec	CycleGAN	49.90	**99.90**	50.50	49.90	50.30	99.70	50.10	50.00	48.00	50.60	50.10	50.90	50.40	50.40	50.30	51.70	51.40	50.40	50.00	55.45
DIRE	-	**100.00**	67.50	64.53	83.40	63.03	**99.95**	93.27	56.67	63.47	55.54	55.55	75.35	88.15	89.10	89.50	87.15	90.35	91.40	62.90	77.73
UnivFD	LC	**100.00**	98.50	**94.50**	82.00	**99.50**	97.00	66.60	63.00	57.50	59.50	72.00	70.03	94.19	73.76	94.36	79.07	79.85	78.14	86.78	81.38
OnlyReal	-	99.15	93.50	89.90	**98.05**	80.50	99.60	**85.00**	49.50	71.50	50.10	50.10	**81.10**	**97.95**	**97.40**	**98.10**	88.40	91.90	87.20	**98.10**	84.58
Ours	N	81.75	54.85	58.84	59.46	56.38	50.06	49.94	50.00	52.05	50.00	50.00	53.65	49.05	48.00	48.45	46.97	47.30	46.30	51.60	52.88
	P	95.76	91.74	92.72	75.09	98.26	90.26	83.16	76.11	56.62	92.61	93.34	60.55	92.45	80.40	92.80	82.94	86.75	84.75	87.30	84.93
	N+P	99.22	93.67	90.80	84.42	88.11	96.65	66.77	54.44	68.49	84.79	85.12	78.55	96.80	88.35	96.50	**96.30**	**97.25**	**96.10**	96.55	**87.30**

B. Comparison with Existing Methods

In this subsection, we compare our method with several advanced anomaly detection approaches: PaDiM [28], CFA [31], GANomaly [23], DSR [32], DRAEM [24], STFPM [25], RD [26], DFM [27], FastFlow [10], CFlow [11], and existing AI-generated image detectors: CNNSpot [12], PatchForensics [13], Co-Occur [14], FreqSpec [17], DIRE [16], UnivFD [3], Freq-mask [8] and OnlyReal [22]. The numerical results are reported in Table I. For anomaly detection methods, we train them completely according to the settings and instructions of the original models using only natural images in our training set and test them on the AII detection benchmark test sets. For most AI-generated image detection methods, we use the results as presented in the UnivFD paper [3]. Besides, we also compare with DIRE [16], Freq-mask [8] and OnlyReal [22] (all based on supervised learning algorithm), with the same setting of UnivFD.

From the first part of Table I, we can see that existing anomaly detection methods have limited performance on the task of AI-generated image detection, with mAP remaining quite low around 50% for the vast majority of tested methods (last column of Table I). These results are the outcome of our extensive study on the performance evaluation of recent anomaly detection methods on AII detection, to our knowledge the first time in the literature for such a study. Possible explanations of the limited performance, as mentioned before, are the insufficient ability of existing methods to capture the very diverse content of natural images and the blindness to anomalies to be detected. The DRAEM method [24] shows slightly better performance than other anomaly detection methods with mAP of around 64%. Upon inspection, we hypothesize that this better performance is mainly due to the construction of some artificial anomaly samples in the DRAEM method. This is in part in line with the idea of constructing proxy images in our method, which nevertheless shows much better AII detection performance, as presented in the following.

In the last group of rows of Table I, we show the test results of three modes of our model. N represents our model trained with likelihood maximization on 7,200 natural images, P represents our model trained with likelihood minimization on 7,200 proxy images, and N+P represents our model trained with both the likelihood maximization on natural images and likelihood minimization on proxy images (in total 14,400 images). We can see that when we only maximize the likelihood of natural images, the performance is not ideal, again probably due to the reasons mentioned in the last paragraph for existing anomaly detection methods. On the contrary, when we only minimize the likelihood of proxy images, the performance of our model has been significantly improved. The mAP of our method has reached a high value of 93.85%, while not using any AI-generated image for training. This implies that, despite their simplicity of construction, proxy images have distinctive differences from natural images, making them closer in the feature space to AIIs generated by a variety of generative models, including GANs and diffusion models, than to natural images. Finally, if we jointly maximize the likelihood of natural images with the N+P mode, the mAP of our model can be further improved to 95.56%. Towards the end of Table I, we compare with the very recent method OnlyReal [22], which, similar to our method, does not use AIIs to build a forensic detector. However, it is worth noting that the essential structure of OnlyReal remains a supervised binary model with the conventional cross-entropy loss and consistent with traditional AII detection methods, and that it cannot be trained using only a single class of images. In contrast, our method can be effectively trained in an unsupervised manner using only a single class of proxy images, making it more flexible and more inspiring for future research. Regarding the AII detection performance, we can see from Table I that OnlyReal achieves best performance on many test sets (such as CycleGAN, StarGAN and LDM), while the performance of our method (N+P mode) is very close on these datasets. On datasets where OnlyReal performs poorly (such as SITD and CRN), our model still performs well, so the final average performance mAP of our method is higher than OnlyReal (95.56% vs. 93.18%).

In Fig. 3, we show histograms of the anomaly scores of some test sets. It can be seen that, for ProGAN, StarGAN and LDM_200, with our method trained with P mode, the histograms of natural images and AIIs remain different, but they are

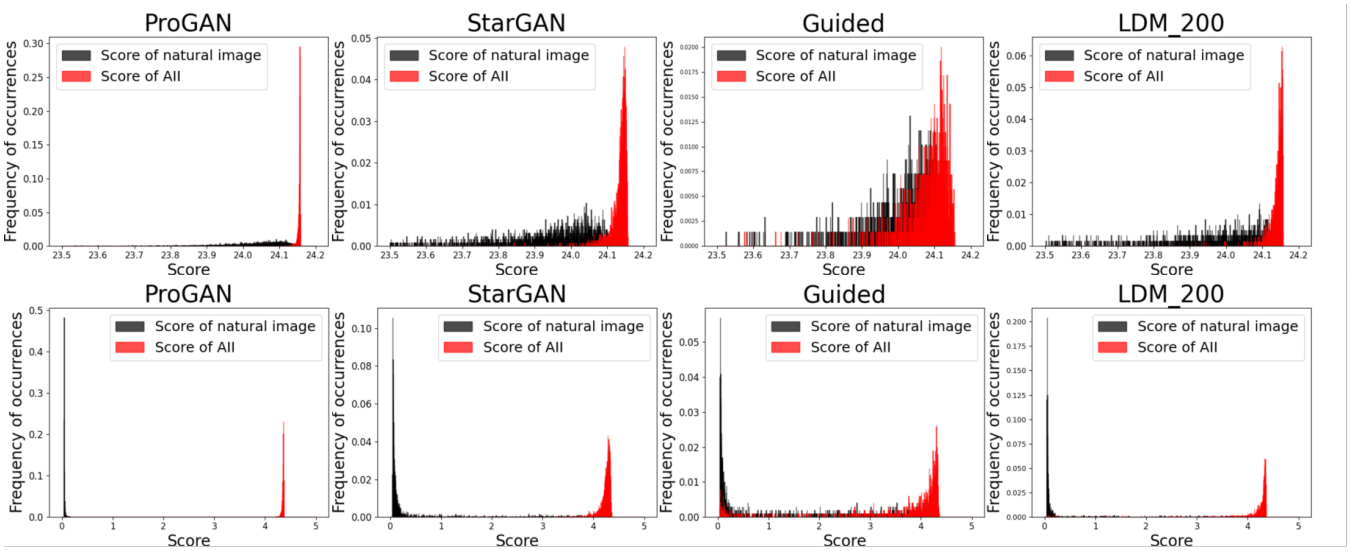

Fig. 3: Histograms of anomaly scores on test sets of ProGAN, StarGAN, Guided, and LDM_200 (from left to right), for the P mode (top row) and the N+P mode (bottom row) of our method. The range of score values is different for the two modes due to different training losses (*cf.* main text). The scores of the proxy images, not shown here for the sake of brevity, are all densely clustered at very high values.

close and have small overlap; when we use our model trained with N+P mode, the two histograms are very well separated with a gap between them. Similar to existing methods, our model has limited performance on Guided; nevertheless, the histogram separability is much improved with N+P mode than with P mode (third column of Fig. 3). The magnitude of the anomaly scores in the two modes is different, as shown in Fig. 3, and this is mainly due to the different training losses used in the two modes. For the P mode, the training strategy is to minimize the log-likelihood (thus maximizing the anomaly score) of the proxy images, while for the N+P mode, we additionally maximize the log-likelihood of the natural images. The more complex training loss in the N+P mode experimentally leads to anomaly scores of lower magnitude, probably due to the need to reach a trade-off between the losses of the two types of images.

Table II presents the test accuracy results of our model and representative existing methods. For our model, we choose a threshold based on a synthetic validation set as mentioned in Section IV-A. The average accuracy reaches 84.93% for N mode of our model, which is 3.55% higher than the reference UnivFD method. When we maximize the likelihood of natural images and minimize the likelihood of proxy images at the same time, *i.e.*, N+P mode, the average accuracy of our model improves further and reaches 87.30%.

TABLE III: The performance (in AP, %) of different types of proxy images. FM stands for frequency masking.

Mode	Type	Generative Adversarial Networks						Deep Fakes	Low level vision		Perceptual loss		Guided	LDM			Glide			DALL-E	Total
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN		SITD	SAN	CRN	IMLE		200 steps	200 w/ CFG	100 steps	100 27	50 27	100 10		mAP
N+P	Smoothing	96.41	68.84	74.09	84.56	88.84	48.36	42.69	67.86	59.21	80.09	85.49	61.61	89.19	78.36	88.48	83.72	85.67	84.02	83.13	76.25
	Sharpening	99.66	96.22	98.59	88.49	97.94	**99.74**	79.05	77.89	67.66	44.84	80.11	79.33	96.99	92.87	96.63	65.93	70.97	69.34	99.56	84.31
	Gaussian noise	99.73	96.35	96.82	88.82	98.88	98.89	69.30	73.18	63.22	90.33	96.83	73.54	95.93	90.42	95.80	70.16	78.13	73.36	99.13	86.78
	Color jitter	99.87	98.73	97.90	**98.15**	97.88	99.57	81.71	70.94	71.05	76.70	96.79	82.18	**99.50**	**98.10**	**99.45**	97.79	98.52	98.17	**99.88**	92.78
	FM (onlyphase)	**99.96**	99.86	**99.31**	96.11	**99.93**	99.73	85.78	79.52	79.31	**95.39**	**97.62**	87.66	99.20	95.98	99.14	98.93	99.32	98.87	99.32	95.31
	FM	99.94	**99.87**	99.24	97.10	99.92	99.50	**87.34**	**80.06**	**79.48**	94.85	96.94	**87.78**	99.41	96.93	99.44	**99.35**	**99.59**	**99.28**	99.55	**95.56**

TABLE IV: The performance (in AP, %) of proxy image generation with the masking of different frequency parts. We randomly set the low, mid, and high part of the image spectrum (as defined in [8]) to 0 with a probability of 0.1.

Mode	Frequency Part	Generative Adversarial Networks						Deep Fakes	Low level vision		Perceptual loss		Guided	LDM			Glide			DALL-E	Total
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN		SITD	SAN	CRN	IMLE		200 steps	200 w/ CFG	100 steps	100 27	50 27	100 10		mAP
N+P	Low	99.94	**99.87**	**99.24**	97.10	**99.92**	**99.50**	87.34	80.06	**79.48**	**94.85**	96.94	**87.78**	**99.41**	96.93	99.44	**99.35**	**99.59**	**99.28**	99.55	**95.56**
	Mid	99.50	98.34	97.38	**98.83**	98.51	96.31	73.16	74.52	57.31	90.51	95.70	75.96	99.39	**98.07**	**99.45**	97.98	98.66	98.25	**99.79**	91.98
	High	**99.96**	99.81	99.12	97.46	99.88	99.30	**88.80**	**80.43**	78.31	94.64	**97.01**	86.36	99.40	97.04	99.41	99.20	99.51	99.12	99.54	95.49

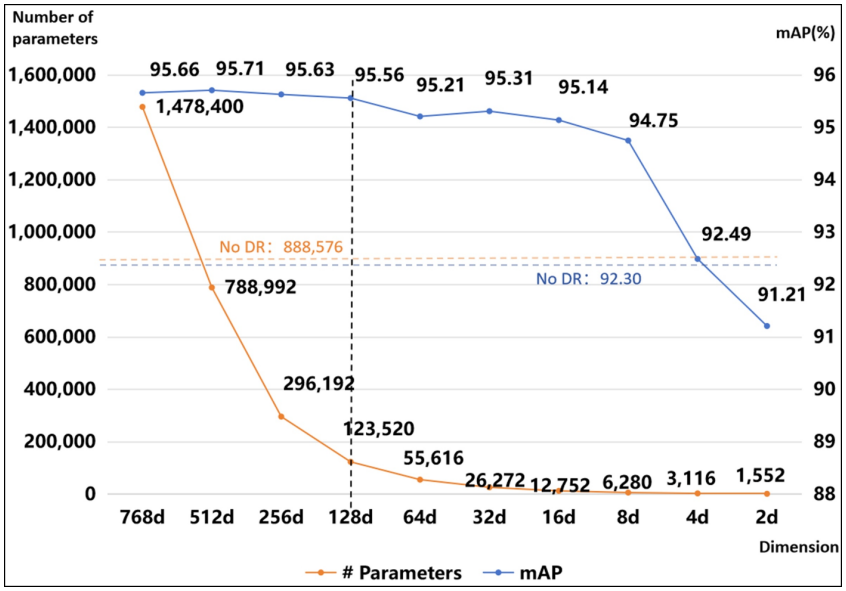

Fig. 4: Results of ablation studies regarding the DR layer.

C. Additional Results and Ablation Studies

Other Types of Proxy Images. As mentioned in Section III-B, we have also attempted to use proxy images constructed by other types of operations, including spatial operations such as smoothing, sharpening, Gaussian noise addition, and color jitter, as well as a variant of frequency masking that randomly masks only the phase while keeping the spectral amplitude unchanged. For the sake of brevity, we show in Table III the results in the N+P mode of our detector. The parameter setting of the new operations is similar to that of frequency masking, *i.e.*, for the N+P mode a relatively small modification is introduced with appropriate parameters. For example, Gaussian noise was added with a moderate standard deviation of 5 (pixel value between 0 and 255) to produce noisy proxy images. It can be seen from Table III that routine spatial operations (smoothing, sharpening, and noise addition) result in limited detection performance, while color jitter leads to much better overall mAP. One possible explanation is that color jitter introduces a plausible color deviation from natural images. We notice that only masking the spectrum phase achieves a high mAP, very close to the frequency masking of both the phase and the amplitude. This observation is interesting and remains understandable because the spectrum phase is proven to carry important structural and perceptual information [33]–[35]. It is also encouraging to see that our method can cope well with both spatially (color jitter) and spectrally (frequency masking) modified proxy images for the training of an effective AII detector, while not mimicking specific artifacts of AIIs. Future research shall be devoted to understanding the effect of different operations and to constructing more effective proxy images.

Spectrum Part for Frequency Masking. In our work, we generate proxy images by randomly masking the low part of the image spectrum. What effect would masking information on the high part and the mid part have? The corresponding results are reported in Table IV. We can see that our method can perform properly in all three cases, with the low and high parts having a higher mAP than the mid part. AIIs of some models such as ProGAN, GauGAN, IMLE, and Glide can be distinguished very well when we mask either of the three parts of the spectrum to generate proxy images. AIIs of some other models such as DeepFakes, SAN, and Guided are much better distinguished when we mask the low or high part than the mid part. In general, the detection performance on each generative model and with each frequency part is different, and masking the low and high parts has better overall performance than masking the mid part, with the low part masking being slightly better than the high part masking. Therefore, we choose to perform random frequency masking in the low part of the spectrum of natural images to produce proxy images.

Dimension Reduction. We conducted ablation experiments to verify the impact of dimension reduction (DR) of the CLIP feature. In the experiments, we simultaneously maximized the likelihood of natural images and minimized the likelihood of proxy images, and kept other parameters unchanged while only varying the output dimension of the DR layer. As shown in the results of Fig. 4, if the DR layer is not used (the two "No DR" horizontal lines), not only the number of parameters will be very large, but also the model performance will decrease. Using the DR layer to reduce the feature dimension simplifies the model and improves the performance. Regarding the choice of the reduced feature dimension, we chose 128 which has a good trade-off between model complexity (its number of parameters) and performance (its mAP).

Feature Normalization. After the CLIP feature dimension is reduced by the linear layer of DR, we also normalize its magnitude to 1. Our intuitions behind this normalization are as follows: 1) The CLIP model was trained by contrastive learning with a cosine similarity loss, so its features are in theory independent of their magnitude; 2) With feature normalization, the

TABLE V: AP (%) on datasets from recent generative models. Our method has strong generalization to unseen generators.

Generator	Ours	UnivFD
DALL·E 3	90.48	61.78
DeepFloyd	97.17	86.47
RealVisXL v4.0	90.70	61.98
Stable Diffusion v1.5	94.93	85.96
Average	**93.32**	**74.05**

input distribution of the flow model is constrained on a unit hyper-sphere, which would be less complicated to learn than an arbitrary distribution and thus reduce the risk of overfitting. Experimentally, the mAP is 93.41% without normalization, while the mAP reaches 95.56% (an increase of 1.15%) with feature normalization.

D. Evaluation on Emerging Generative Models

To further examine generalization, we built new test datasets based on prompt-driven image generation. Specifically, from real images, we used a vision-to-language model to generate prompts, then synthesized corresponding fake images using state-of-the-art generators, including DALLE 3, DeepFloyd, RealVisXL v4.0, and Stable Diffusion v1.5. Each new dataset comprises 1000 real and 1000 generated images. We compare our method with UnivFD [3]. As shown in Table V, our model (P mode, without any fine-tuning, *i.e.*, the same model evaluated in previous subsections) achieves high AP (Average Precision) scores across the new test datasets. These emerging and unseen generators produce high-quality, semantically rich images that pose substantial challenges for AII detectors. Overall, our method demonstrates strong detection performance even on unseen and recently released generative models, suggesting enhanced generalization to real-world deployment settings. At last, we would like to clarify that our goal in this work is to explore AII detection from a new anomaly detection perspective. We do not aim to outperform every concurrent and recent method (mostly based on supervised learning), and we do not claim SOTA performance.

V. Conclusion

Unlike the conventional AII detection pipeline that relies on supervised learning, we propose a universal and generalizable framework inspired by anomaly detection. Our model includes a CLIP-based feature extractor and a normalization flow-like module. Without requiring access to AI-generated images, our model is trained using anomalous frequency-masked images and optionally natural images by optimizing the image likelihood under the flow model. Experimental results demonstrate the strong generalization capability of our CLIP-Flow model on various AII detection benchmark datasets. In particular, our work shows the feasibility of leveraging anomaly detection with unsupervised or self-supervised learning for universal detection of AI-generated images. An interesting by-product of our study is a new flow model that can handle two types of samples of complementary distributions, and this new model could be useful in other data analysis applications. In the future, we would like to explore other proxy image generation approaches, propose new anomaly detection methods that can cope better with the AII detection problem, and extend our framework to other multimedia forensic challenges. Code is available at https://github.com/Yzp1018/CLIP-Flow.

Acknowledgments

This work is partially funded by the French National Research Agency (Grants numbers ANR-23-IAS4-0004-02 and ANR-15-IDEX-02) and the Excellent Youth Program of State Key Laboratory of Multimodal Artificial Intelligence Systems.

References

[1] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018, pp. 1–12.

[2] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[3] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 480–24 489.

[4] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2021.

[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021, pp. 8748–8763.

[6] W. Quan, P. Deng, K. Wang, and D.-M. Yan, "Cgformer: Vit-based network for identifying computer-generated images with token labeling," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 235–250, 2024.

[7] M. Cai, H. Zhang, H. Huang, Q. Geng, Y. Li, and G. Huang, "Frequency domain image translation: More photo-realistic, better identity-preserving," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 930–13 940.

[8] C. T. Doloriel and N.-M. Cheung, "Frequency masking for universal deepfake detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2024, pp. 13 466–13 470.

[9] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[10] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu, "FastFlow: Unsupervised anomaly detection and localization via 2D normalizing flows," *arXiv preprint arXiv:2111.07677*, pp. 1–11, 2021.

[11] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 98–107.

[12] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot...for now," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8692–8701.

[13] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? Understanding properties that generalize," in *European Conference on Computer Vision*, 2020, pp. 103–120.

[14] L. Natraj, T. Manhar Mohammed, S. Chandrasekaran, A. Flenner, A. K. Roy-Chowdhuri, and B. Manjunath, "Detecting GAN generated fake images using co-occurrence matrices," in *Electronic Imaging*, 2019, pp. 532:1–6.

[15] W. Bai, Z. Zhang, B. Li, P. Wang, Y. Li, C. Zhang, and W. Hu, "Robust texture-aware computer-generated image forensic: Benchmark and algorithm," *IEEE Transactions on Image Processing*, vol. 30, pp. 8439–8453, 2021.

[16] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li, "DIRE for diffusion-generated image detection," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 388–22 398.

[17] Z. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in GAN fake images," in *IEEE International Workshop on Information Forensics and Security*, 2019, pp. 1–6.

[18] J. Frank, T. Eisenhofer, L. Schonherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International Conference on Machine Learning*, 2020, pp. 3247–3258.

[19] R. Durall, M. Keuper, and J. Keuper, "Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7887–7896.

[20] T. Dzanic, K. Shah, and F. D. Witherden, "Fourier spectrum discrepancies in deep network generated images," in *Advances in Neural Information Processing Systems*, 2020, pp. 3022–3032.

[21] B. Liu, F. Yang, X. Bi, B. Xiao, W. Li, and X. Gao, "Detecting generated images by real images," in *European Conference Computer Vision*, 2022, pp. 95–110.

[22] J. Li and K. Wang, "Detecting computer-generated images by using only real images," in *International Conference on Machine Vision*, 2024, pp. 1–13.

[23] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," in *Asian Conference on Computer Vision*, 2019, pp. 622–637.

[24] V. Zavrtanik, M. Kristan, and D. Skočaj, "DRM - A discriminatively trained reconstruction embedding for surface anomaly detection," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8330–8339.

[25] G. Wang, S. Han, E. Ding, and D. Huang, "Student-teacher feature pyramid matching for unsupervised anomaly detection," in *British Machine Vision Conference*, 2021, pp. 1–14.

[26] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9737–9746.

[27] N. A. Ahuja, I. Ndiour, T. Kalyanpur, and O. Tickoo, "Probabilistic modeling of deep features for out-of-distribution and adversarial detection," in *Bayesian Deep Learning Workshop*, 2019, pp. 1–5.

[28] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: a patch distribution modeling framework for anomaly detection and localization," in *International Conference on Pattern Recognition*, 2021, pp. 475–489.

[29] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *International Conference on Learning Representations*, 2017, pp. 1–12.

[30] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *Journal of Machine Learning Research*, vol. 22, no. 57, pp. 1–64, 2021.

[31] S. Lee, S. Lee, and B. C. Song, "CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization," *IEEE Access*, vol. 10, pp. 78 446–78 454, 2022.

[32] V. Zavrtanik, M. Kristan, and D. Skočaj, "DSR – A dual subspace re-projection network for surface anomaly detection," in *European Conference on Computer Vision*, 2022, pp. 539–554.

[33] A. V. Oppenheim and J. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 529–541, 1981.

[34] X. S. Ni and X. Huo, "Statistical interpretation of the importance of phase information in signal and image reconstruction," *Statistics & Probability Letters*, vol. 77, no. 4, pp. 447–454, 2007.

[35] E. Gladilin and R. Eils, "On the role of spatial phase and phase correlation in vision, illusion, and cognition," *Frontiers in Computational Neuroscience*, vol. 9, pp. 1–14, 2015.