μFlow: Leveraging Average Images for Improving Generalisation of Deepfake Faces Detectors

Orazio Pontorno*[1], Mattia Litrico*[1], Luca Guarnera[1], Mario Valerio Giuffrida[2], and Sebastiano Battiato[1]

[1] University of Catania, Catania, Italy
[2] University of Nottingham, Nottingham, UK
{orazio.pontorno,mattia.litrico}@phd.unict.it,
{luca.guarnera,sebastiano.battiato}@unict.it,
valerio.giuffrida@nottingham.ac.uk

Abstract. Current generative models, including GANs and diffusion models, have reached an outstanding level of photorealism, posing significant risks to privacy and security. To ensure real-world applicability, deepfake detectors must generalise effectively to unseen generators. However, most existing approaches rely on supervised training with both real and fake images, which limits their generalisation especially across generators categories (e.g. GANs vs DMs). In this work, we introduce ***μFlow***, a one-class deepfake detector trained only on real images without relying on pseudo-deepfakes or synthetic artifacts. Our approach builds on the observation that averaging multiple images amplifies consistent generative traces, producing highly discriminative feature representations. We leverage this property by modelling the distribution of features extracted from averaged images and training a normalizing flow to align the feature space of individual images with this distribution. This alignment yields a likelihood-based criterion that separates real and fake samples while promoting strong generalisation. We evaluate ***μFlow*** on a fully out-of-distribution setting, where both real and fake datasets are unseen during training. Experimental results show that our method significantly outperforms SOTA detectors. Project page: opontorno.github.io/MuFlow.

Keywords: One-Class Deepfake Detection · Average Images · Normalizing Flows

1 Introduction

The generation of deepfakes has reached a worrisome level of photorealism, increasingly raising privacy concerns among the general public. Deepfakes are obtained by either forging a real image [15,30,47,63], or creating them from scratch using generative models [14, 31, 46, 48], such as GANs [16] or diffusion models (DMs) [20]. Given the recent strides in generative models, detecting forgeries has become increasingly challenging using conventional methodologies. Nonetheless,

* These authors equally contributed to this work.

these models introduce distinctive traces that can be detected by deep learning techniques [43, 62]. Most of the current deepfake detectors are trained using a labelled training set comprising both real and fake images [27, 50, 60, 67].

These approaches can easily detect deepfakes from generators seen during training (in-distribution) but struggle to generalise to unseen generators (out-of-distribution).

For this reason, recent approaches have focused on assessing the generalisation capability of deepfake detectors. In fact, recent works focused on improving robustness through searching for universal traces [55, 56, 62], vision-language models (VLMs) [54, 64] or pretrained models [11,40]. However, these approaches are trained on both real and fake images, and tend to overfit to distinctive traces injected by specific generators seen during training [62]. This limits their generalisation performance, as generative pipelines improve much faster than detector datasets can be updated.

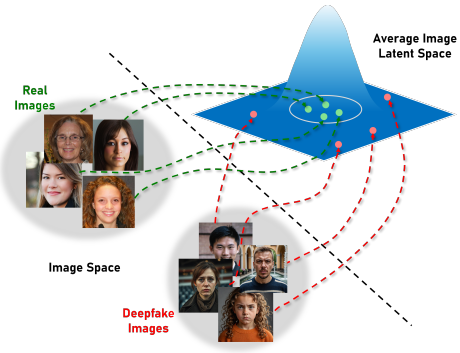

Fig. 1: ***μ*Flow** overview: image features are mapped into a discriminative distribution trained from average image features determined from just real images. In this way, fake images obtain a lower likelihood at test time, as being considered out-of-distribution samples.

To overcome this issue, some approaches proposed to leverage real-only training [33, 37, 52, 66]. Such approaches offer a promising direction to improve the generalisability of deepfake detectors since they are not constrained by specific generators. However, several approaches [33, 37, 66] rely on generating pseudo-deepfakes for learning a discriminative features space [58], limiting their performance to the quality of such pseudo-deepfake. A different approach is OC-FakeDect [28] that maps real samples into a normal Gaussian distribution. However, this approach is characterized by limited discriminative power. Therefore, there is a lack of deepfake detectors able to generalise to unseen generators, without requiring the need of fake samples during training but still maintaining high discriminability.

In this work, to improve the generalisation to unseen generators, we propose ***μFlow***, a novel deepfake face detector trained only on real faces without the need of generating pseudo-deepfakes or artificial artifacts. To detect deepfakes, we train ***μFlow*** to map real samples in a pre-determined distribution and we treat log-likelihoods as *fakeness* scores at inference time (Fig. 1). Previous methods [18, 28] project features into a normal Gaussian distribution. However, as shown in Fig. 2 (a), features extracted by real and fake samples exhibit low inter-class

variability. Therefore, mapping features into a normal distribution results in high likelihoods for both classes, limiting current methods performance.

In contrast, we observe that averaging multiple faces significantly amplifies the distinctive traces injected by generative models [10, 45, 59]. As demonstrated in Fig. 2 (b), the representations extracted from the average images are clearly clustered among real and fake samples, making this feature space highly discriminative and suitable for deepfake detection. However, despite their discriminative power, average images cannot be directly used at inference time, as we need to discriminate between individual real and fake images. To overcome this issue, we propose leveraging the discriminative space derived from the average image by projecting into it the features extracted from single real images. The underlying intuition is that, once aligned with this discriminative space, features from individual images become inherently discriminative.

We benchmark ***μFlow*** on three publicly available face datasets. Specifically, we evaluate its generalisation performance using a fully out-of-distribution evaluation setting. While trained solely on reals images from FFHQ [24], it is tested on both unseen reals from CelebA-HQ [22] and deepfakes on unseen generators of the WILD [4] dataset. Under this severe generalisation setting, our method outperforms the state-of-the-art by a large margin.
The main contributions of our work are:

- We approach deepfake face detection as a one-class classifier problem, training solely on real faces. Unlike previous methods, ***μFlow*** does not require generating pseudo-deepfakes or synthetic artifacts, thus avoiding dependence on their quality and improving generalization.
- Based on the observation that average real image features are highly discriminative, we propose to align single-image features with such a space, enhancing their separability. We also provide an analytical derivation showing that mapping features into this discriminative space and maximizing their likelihood yields a direct criterion for distinguishing real from fake images.
- We validate our method on 19 unseen generators including both GANs and DMs architecture from the WILD dataset. ***μFlow*** outperforms the SOTA by a large margin on several out-of-domain settings.

2 Related Work

Deepfake Detection. The detection of forged contents is an extremely active research field, aimed at analysing and detecting fake text, video, audio, as well as images [26]. Recently approaches have extensively focused on improving the generalisation of detectors, aiming at correctly identifying deepfakes from unseen generators. FreqNet [55] analysed the high-frequencies to find discriminative features. NPR [56] analysed generator-specific traces using up-sampling layers. D^3 [62] proposed to scale up the training set, including multiple generators to better learn such traces. Other approaches leveraged the generalisation capabilities of vision-language models such as CLIP, by prompt learning [54] or

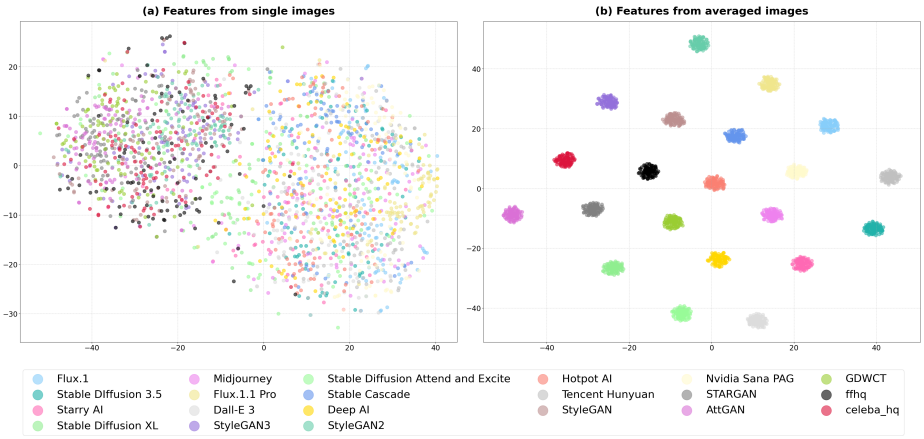

Fig. 2: t-SNE analysis: (a) features extracted from single images; (b) features extracted from the average of images. The space of the average images exhibits a higher inter-class variability, yielding higher discriminative power for deepfake detection.

finetuning [64]. UFD [40] was the first to show that a feature space not explicitly learned for generated image detection improve robustness due to its unbiased decision boundaries. Similarly, [11] trained SVMs with the features from the penultimate layer, achieving good generalisation performance. However, all of these approaches still train with both real and fake samples, inevitably limiting the generalisation to unseen generators. Differently, we do not use any fake image during training, preventing the detector to focus on specific traces.

One-class Training for Deepfake Detectors. To improve the generalisation, some methods proposed to train detectors using only real samples. DiffFake [52] use pairs of real faces from the same person to detect anomalous inconsistencies during inference. SeeABLE [33] perturbs images to create pseudo-deepfakes, aligns them to prototypes, and detects anomalies by their distance from these prototypes. UNTAG [37] generated pseudo artifacts on real images and learns to detect them in a self-supervised manner. CLIP-Flow [66] combines CLIP features with a normalizing flow trained on real images and proxy samples. Similarly, [49] synthesises self-blended images (SBI) from single real faces to reproduce general blending traces, while [68] stitches real images via an inconsistency image generator (I2G) to produce pseudo-fakes. To capture local and semantic anomalies on pseudo-fakes, AUNet [1] learns altered relations among synthetic facial movements, and [39] introduces a localized artifact attention network (LAA-Net) to detect fine-grained blending traces. Although trained only on real images, such methods still rely on synthesised pseudo-deepfakes or artifacts to obtain discriminative features, making their performance dependent on the quality of these generated proxies. OC-FakeDect [28] did not generate artificial artifacts, as it trains a one-class Variational Autoencoder solely on real images and determines *fakeness* scores via reconstruction errors. However, with the advent

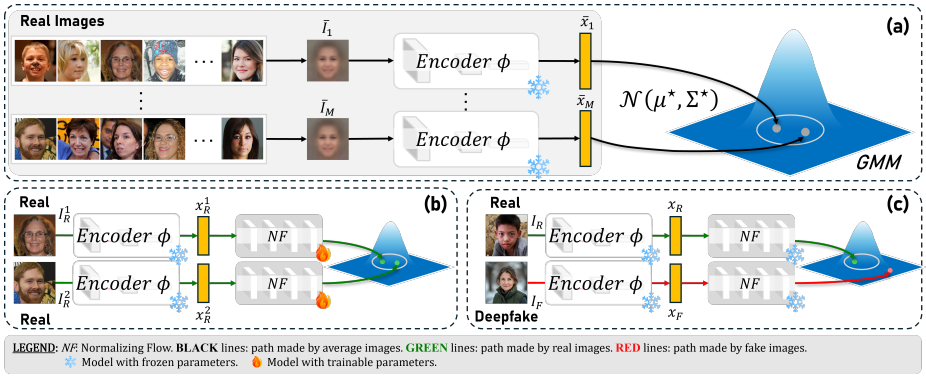

Fig. 3: μFlow graphical overview. **(a) Discriminative Space Learning:** We learn a discriminative latent space extracting features from average real images (Sec. 3.3), using a Gaussian Mixture Model. **(b) FastFlow Training:** We train FastFlow to project representations extracted from real images into the learnt discriminative space (Sec. 3.4). **(c) Inference:** At test time, the representation of a real image is mapped in the region of the latent space with higher likelihood, while fake images are mapped to a low-likelihood regions (Sec. 3.5).

of novel generators, both real and fake images likely have similar features and thus similar reconstruction error, limiting their successful detection. Different from these approaches, ***μFlow*** is trained solely on real faces and it does not require the generation of pseudo-deepfake samples or artifacts. Instead, it learns a discriminative features space by leveraging average images to better highlight traces injected by deepfake generators.

Normalizing Flows. Normalising flow models are designed to estimate complex data distributions using invertible transformations with tractable Jacobian determinants. By mapping data into a latent space, they enable exact likelihood computation [32]. FastFlow [65] and CFLOW-AD [18] proposed to use normalizing flows for anomaly detection. Specifically, they train a normalizing to project image features into the standard normal distribution. During inference, likelihoods of transformed features are then treat as anomaly scores. Different from these methods, we map image features into an ad-hoc distribution where real and deepfakes can be easily discriminated.

3 Proposed Method

The workflow for ***μFlow*** is depicted in Fig. 3. We learn a discriminative latent space from the representation extracted by averaging sampled real faces (Fig. 3a). We train a normalizing flow [65] using features extracted from only real images that are mapped into this discriminative space (Fig. 3b). As we trained with only real data, during inference, fake images obtain a lower likelihood and they are classified as deepfake. (Fig. 3c). In the sections below, we provide a detailed problem formulation and description of our method.

3.1 Problem Formulation

The task of detecting deepfakes can be formalised as follows: let $\mathcal{I} \subset \mathbb{R}^{H \times W \times 3}$ be the image space and consider a dataset $\mathcal{D} = \mathcal{R} \cup \mathcal{F}$, composed of real faces $\mathcal{R} \subset \mathcal{I}$ and fake faces $\mathcal{F} \subset \mathcal{I}$. Fake samples are produced by a set of N generators $\mathcal{G} = \{G_i\}_{i=1}^{N}$: for each i, let $\mathcal{F}^{(i)} = \{f_1^{(i)}, \ldots, f_{m_i}^{(i)}\} \subset \mathcal{I}$ denote the m_i images generated by G_i, so that $\mathcal{F} = \bigcup_{i=1}^{N} \mathcal{F}^{(i)}$. Our goal is to correctly distinguish between real and fake images for any input drawn from $\mathcal{D}$.

Now, let split the N generators into two disjoint sets for training and testing. The training set $\mathcal{D}_{\texttt{train}} = \mathcal{R}_{\texttt{train}} \cup (\bigcup_{i=1}^{k} \mathcal{F}^{(i)})$ includes training real images $\mathcal{R}_{\texttt{train}}$ and deepfakes from $k < N$ generators, while the testing set $\mathcal{D}_{\texttt{test}} = \mathcal{R}_{\texttt{test}} \cup (\bigcup_{i=k+1}^{N} \mathcal{F}^{(i)})$ is composed by testing real images $\mathcal{R}_{\texttt{test}}$ and deepfake from the other $N-k$ generators, with $\mathcal{R} = \mathcal{R}_{\texttt{train}} \cup \mathcal{R}_{\texttt{test}}$ and $\mathcal{R}_{\texttt{train}} \cap \mathcal{R}_{\texttt{test}} = \emptyset$.
Overview. We train a one-class classifier to detect deepfakes, using solely real images, *i.e.,* ***μFlow*** *is not* provided with fake images during training, while still able to detect deepfakes from unseen generators. To this aim, all the generators are unseen to ***μFlow*** and provided *only* at testing time, by setting $k = 0$, $\mathcal{F}_{\texttt{train}} = \emptyset$ and $\mathcal{F}_{\texttt{test}} = (\bigcup_{i=1}^{N} \mathcal{F}^{(i)})$. This means that the training set for ***μFlow*** reduces to $\mathcal{D}_{\texttt{train}} \equiv \mathcal{R}_{\texttt{train}}$, and the testing set is $\mathcal{D}_{\texttt{test}} \equiv \mathcal{R}_{\texttt{test}} \cup \mathcal{F}_{\texttt{test}}$.

3.2 Training a One-class Classifier for Deepfake Detection

Deepfake detection approaches are challenged in generalising to unseen generators [29, 55, 62]. This is mainly because the detector learns to detect traces injected by training generators, which may differ from those injected by testing generators, especially if training and testing generators come from different categories (GANs vs. DMs) [8, 10].

We train a one-class classifier that learns the distribution of real training samples, such that out-of-distribution samples are considered as fake. This allows our model to avoid overfitting on detecting traces injected by a specific generator, enhancing generalisability of our approach. Here, we leverage FastFlow [65], which is a state-of-the-art method for one-class detection.
FastFlow. During training, FastFlow extract features from images and trains a normalizing flow model to map all the extracted features from the raw distribution into a known distribution.

Specifically, during training, an image $I_N \in \mathcal{I}$ is provided to a frozen encoder ϕ, such that $x = \phi(I_N)$ represents the features for the sample I_N. A normalizing flow $f_\theta : X \to Z$, invertible with Jacobian $J_{f_\theta}(x)$, is used to project the image features $x \in p_X(x)$ into the hidden variable $p_Z(z)$ with a bijective invertible mapping, where $p_X(x;\theta) = p_Z\big(f_\theta(x)\big) \left| \det J_{f_\theta}(x) \right|$. The prior density $p_Z = \mathcal{N}(0, I)$ is typically a Gaussian Normal distribution. The normalizing flow is trained by maximizing the expected log-likelihood of features from training samples, *i.e.*, minimizing the negative log-likelihood (NLL):

$$\mathcal{L}_{\text{NLL}}(\theta) = -\,\mathbb{E}_{x \sim X}\big[\log p_X(x)\big] = -\,\mathbb{E}_{x \sim P_X}\big[\log p_Z(z) + \log \left| \det J_{f_\theta}(x) \right|\big]\,, \tag{1}$$

where $z = f_\theta(x) \sim p_Z$ and $x = f_\theta^{-1}(z)$.

At testing time, samples that lie outside of the training distribution will have lower likelihoods and they will be detected and identified as outliers.

3.3 Searching Generator Traces on Average Images

Following the one-class training paradigm, we train FastFlow [65] only with real samples. However, as shown in Fig. 2 (a), features extracted by a pretrained encoder from real and fake samples exhibit low inter-class variability. Consequently, FastFlow trained on such features tends to map both real and fake samples into the standard normal distribution, assigning high likelihoods to both classes and preventing effective separation. State-of-the-art approaches overcome this issue by training the detectors on both real and fake images but losing generalisability on unseen generators.

The current literature in deepfake detection has shown that averaging fake images amplifies the traces injected by generators [9], making them easier to detect. As shown in Fig. 2 (b), features extracted from average real and fake images are clearly separable, showing a strong signal for deepfake detection.

3.4 Projecting Features into the Average Image Feature Space

Although features extracted from average images are highly discriminative, they cannot be directly utilised during testing, as it is typically expected to perform inference with a single image. *How can we leverage the discriminative information encoded in the average images to effectively train a model capable of correctly classifying single images during inference?*

To this end, we propose to align the less discriminative feature space of single images with the highly discriminative feature space from average images. The underlying idea is that, by effectively mapping features from single samples into this discriminative space, these features become inherently discriminative. The next paragraphs provide an analytical derivation supporting this idea. Following this intuition, building upon FastFlow, we train the normalizing flow using only real samples to project their features to the discriminative space from average real images. Different from Fastflow, our method projects the features to an ad-hoc discriminative distribution, rather than the standard normal distribution.

Learning the underlying distribution of average real images. To learn the mapping from *single-image* real features to the discriminative space obtained from average images features, we first need to learn their underlying distribution. To this aim, we generate a set of M average real images $\{\overline{I}_R^j\}_{j=1}^M$ by averaging K images randomly selected from the training real samples $\mathcal{R}_{train}$, as follows:

$$\overline{I}_R^j \;:=\; \frac{1}{K}\sum_{i\in S_j} I_R^i, \qquad \overline{x}_R^j \;:=\; \phi(\overline{I}_R^j), \tag{2}$$

where S_j is a set of randomly selected indices of real training images and $\overline{x}_R^j$ are features extracted by the pretrained encoder $\phi : \mathcal{I} \to X$ on the average image

$\overline{I}_R^j$. We then collect these representations in the set $\mathcal{A}_\mathcal{R} = \{\overline{x}_R^1, \ldots, \overline{x}_R^M\}$ and we use this set to estimate the parameters $(\mu^\star, \Sigma^\star)$ by fitting a Gaussian mixture model (GMM) that encode the underlying distribution of average real features.

Training with only real samples. To leverage the discriminative power of features obtained from average images, we train the normalizing flow to map real features x_R into the discriminative distribution $P_Z^* = \mathcal{N}(\mu^\star, \Sigma^\star)$ rather than a standard Gaussian normal distribution.

We hypothesise that real images are more correctly mapped on P_Z^*, compared to the fake images. This is equivalent to stating that the normalizing flow extracts latent representations from real images having a lower Mahalanobis distance with respect to our target distribution rather than fake images, thus providing a direct criterion to discriminate between real and fake images.

More formally, let $x_R = \phi(I_R) \in X$ be the representation of a real image I_R and $z_R = f_\theta(x_R) \in Z$ be its latent vector extracted by the normalizing Flow $f_\theta : X \to Z$. The squared Mahalanobis distance $d^2_{P_Z^*}(z_R)$ of the latent vector z_R to our target distribution P_Z^* is defined as:

$$d^2_{P_Z^*}(z_R) = (z_R - \mu^\star)^\top \Sigma^{\star -1}(z_R - \mu^\star). \tag{3}$$

For any randomly selected real I_R and fake image I_F, we expect that:

$$d^2_{P_Z^*}(f_\theta(\phi(I_R))) < d^2_{P_Z^*}(f_\theta(\phi(I_F))). \tag{4}$$

Minimising the Mahalanobis distance for real images to the target distribution. To enforce this hypothesis, we train the normalizing flow to maximise the log likelihoods for real image features x_R from $P_Z^\star$, equivalently minimising NLL loss as follows:

$$\mathcal{L}_{\text{NLL}}(\theta) = -\mathbb{E}_{X_R \sim \mathcal{X}_{train}} \left[\log p_X(x_R; \theta)\right], \tag{5}$$

where $p_X(\cdot; \theta)$ is the probability density function (PDF) of the (unknown) distribution of real features x_R. Optimising this objective is equivalent to minimising the Mahalanobis distance of the latent representation from the real images z_R to the target distribution P_Z^*.

Indeed, $p_X(x_R; \theta)$ can be obtained as:

$$p_X(x_R; \theta) \;=\; p_Z^*\big(f_\theta(x_R)\big) \left|\det J_{f_\theta}(x_R)\right|. \tag{6}$$

By using Eq. (6), we estimate the log-likelihood term in Eq. (5) as follows:

$$\log p_X(x_R; \theta) = \log p_Z^*(z_R) + \log \left|\det J_{f_\theta}(x_R)\right|. \tag{7}$$

Given that $p_Z^*(z_R) = \mathcal{N}(z_R; \mu^\star, \Sigma^\star)$, we can expand the term $\log p_Z^*(z_R)$ as:

$$\log p_Z^*(z_R) = \log \left(\frac{\exp\left(-\frac{1}{2}((z_R - \mu^\star)^\top \Sigma^{\star -1}(z_R - \mu^\star)\right)}{\sqrt{(2\pi)^D \det(\Sigma^\star)}} \right), \tag{8}$$

where D is the dimension of the latent z_R. By substituting Eq. (3) in Eq. (8), we have:

$$\begin{aligned}\log p_Z^*(z_R) &= \log\left(\frac{1}{\sqrt{(2\pi)^D \det(\Sigma^\star)}}\right) + \log\left(\exp\left(-\frac{1}{2}d^2_{P_Z^*}(z_R)\right)\right) \\ &= C - \frac{1}{2}d^2_{P_Z^*}(z_R),\end{aligned} \tag{9}$$

where $C = -\frac{1}{2}\left(D\log(2\pi) + \log\det(\Sigma^\star)\right)$ is a constant scalar term since $\Sigma^\star$ is precomputed. Hence, by integrating Eq. (9) in Eq. (7), we obtain:

$$\log p_X(x_R;\theta) = C - \frac{1}{2}d^2_{P_Z^*}(z_R) + \log\left|\det J_{f_\theta}(x_R)\right|. \tag{10}$$

Plugging Eq. (10) into Eq. (5), we find that minimising Eq. (5) is equivalent to:

$$\arg\min_\theta \; -\mathbb{E}_{I_R\sim\mathcal{R}_{train}}\left[C - \frac{1}{2}d^2_{P_Z^*}(f_\theta(\phi(I_R))) + \log\left|\det J_{f_\theta}(x_R)\right|\right], \tag{11}$$

where $z_R = f_\theta(\phi(I_R))$.
Since C is a constant, we can remove it from the minimisation objective:

$$\arg\min_\theta \mathbb{E}_{I_R\sim\mathcal{R}_{train}}\left[\frac{1}{2}d^2_{P_Z^*}(f_\theta(\phi(I_R))) - \log\left|\det J_{f_\theta}(x_R)\right|\right]. \tag{12}$$

The first term of this minimisation objective correspond to the first term of Eq. (4). Consequently, this derivation explicitly shows that optimising Eq. (5) results in minimising $d^2_{P_Z^*}(z_R)$, e.g. the Mahalanobis distance of the real latent representations with respect to the discriminative distribution P_Z^*. While our optimisation does not directly affect the second term in Eq. (4), results in Sec. 4.1 empirically show that minimising $d^2_{P_Z^*}(z_R)$ provides, at test time, a direct criterion to discriminate between real and fake latent representations.

3.5 Overall Framework

Fig. 3 shows the complete pipeline of ***μFlow***. Given a real image I_R, we first extract its feature representation $x_R = \phi(I_R)$ using a pretrained encoder $\phi : \mathcal{I} \to X$. Building on the observation that features from averaged images are highly discriminative, we aim to model their underlying distribution. To this end, we generate a set of averaged real images $\overline{I}_R$, extract their corresponding features $\overline{x}_R = \phi(\overline{I}_R)$, and fit a Gaussian distribution $P_Z^* = \mathcal{N}(\mu^\star, \Sigma^\star)$ that encode the discriminative power of this feature space.

Training. During training, a normalizing Flow $f_\theta : X \to Z$ is trained using only real samples to project single-image features x_R into latent representation $z_R = f_\theta(x_R) \sim P_Z^*$ optimising the following NLL loss:

$$\arg\min_\theta \mathbb{E}_{I_R\sim\mathcal{R}_{train}}\left[\frac{1}{2}d^2_{P_Z^*}(z_R) - \log\left|\det J_{f_\theta}(x_R)\right|\right] \tag{13}$$

Optimising this objective allows the model to learn a discriminative representation without relying on fake or pseudo-fake samples.

We then determine a threshold on collected likelihoods $L(\theta) = \frac{1}{2} d^2_{P^*_Z}(z_R) - \log \left| \det J_{f_\theta}(x_R) \right|$, as follows:

$$\tau = m + \gamma_{1-\alpha}\, \sigma, \tag{14}$$

where $m = \text{mean}(L(\theta))$ and $\sigma = \text{std}(L(\theta))$ are the empirical mean and standard deviation of the training likelihoods, $\gamma_{1-\alpha} = \Phi^{-1}(1-\alpha)$, Φ is the cumulative distribution function of the normal distribution, and $\alpha \in (0, 1)$ is the prescribed significance level. Under the Gaussian assumption that $L(\theta)$, α directly controls the theoretical false positive rate, *i.e.*, $\mathbb{P}(L(\theta) > \tau) = \alpha$.

Inference. We compute the *fakeness* score of a test image I as:

$$s(I) = d^2_{P^*_Z}\big(f_\theta(\phi(I))\big) - \log \left| \det J_{f_\theta}(\phi(I)) \right|, \tag{15}$$

where $\phi(I)$ are features extracted by the encoder ϕ and $f_\theta(\phi(I))$ are latent representations from the normalizing flow.

We consider I as fake if $s(I) > \tau$. A higher score indicates that the features lie far from the learned distribution, as a result of detecting the distinctive traces injected by deepfake generators.

4 Experimental Results

Datasets. To ensure a fair evaluation, we adopted recent high-resolution human faces datasets: the **The Flickr-Faces-HQ (FFHQ)** [24] and **CelebA-HQ** [22] datasets for real images; and a subset of deepfake generators in the **WILD** [4] datasets used during testing. FFHQ dataset contains 70k images of human faces, offering significantly great diversity in age, ethnicity, and background. CelebA-HQ is a large-scale face attributes dataset with consisting of 30k aligned and cropped face images of celebrities. Both are widely used in generative modeling tasks due to its controlled structure and facial consistency across samples. WILD contains 20k high-resolution synthetic human faces, including text-to-image commercial generators and generators that simulate real-world "in-the-wild" conditions, *i.e.*, obtained from both GANs and DMs and generated through both image-to-image and text-to-image approaches. To ensure a balanced representation of generative architecture, we added 3 additional GANs from [17]. Subsequently, we divided all the generators into the following categories:

- **GANs** contains images synthesized by *StyleGAN* [24], *StyleGAN2* [25], and *StyleGAN3* [23] from the WILD dataset, and supplemented by *StarGAN* [7], *GDWCT* [6], and *AttGAN* [19] from [17].
- **Diffusion Models Open-source (DM-OS)** contains images generated by open-source DMs, including *Flux.1* [2], *Stable Diffusion 3.5* [51], *Stable Diffusion XL* [42], *Stable Cascade* [36], *Stable Diffusion Attend and Excite* [5].

– **Diffusion Models Closed-source (DM-CS)** contains images generated by closed-source DMs, including *Dall-E 3* [41], *Midjourney* [38], *Starry AI* [53], *Deep AI* [12], *Hotpot AI* [21], *Nvidia Sana PAG* [61], *Tencent Hunyuan* [35], *Flux.1.1 Pro* [3].

Implementation details. We used the ResNet50 as an encoder pretrained on ImageNet-1K [13] with no fine-tuning. Supplementary material shows stable performance across different encoders. For the normalizing flow (NF), we use the same architecture as in [65]. With the encoder frozen, the NF is trained on a single NVIDIA RTX A6000 with a batch size of 32 within 1000 epochs. We use the AdamW optimizer with an initial learning rate of 10^{-4} and a weight decay of 10^{-5}. We set $\alpha = 0.01$ in Eq. (14). In Sec. 4.2, we show performance of our method with different values of α.
Baselines. We compare ***μFlow*** with an extensive set of baselines including DFX-SN [44], FreqNet [55], NPR [56], ODDN [57], and D^3 [62]. Unlike ***μFlow***, these methods require both real and fake images in training. Consequently, we trained these models by iteratively using one of the three fake categories as fake training dataset, while assessing generalization at testing time on the remaining two. Furthermore, we include recent baselines that are also trained only on real images: AUNet [1], LAA-NET [39], SBI [49], and PCL+I2G [68], OC-FakeDect [28], AFSD [34]. For fair comparisons, we trained all these baselines using their official implementations.

4.1 Results

We evaluate the generalisation capability of ***μFlow*** by training it on real images from FFHQ [24], while testing on the unseen fakes from WILD [4]. Note that, at testing time, we use unseen reals from CelebA-HQ [22] to further assess the ability of ***μFlow*** to generalise across both unseen reals and fake images.
Generalisation results across generator categories. Tab. 1 shows the OOD generalization performance on unseen generators and compares our method with several recent SOTA baselines trained on both real and fake images. We group results based on training dataset used to train SOTA methods: *Real+GANs*, *Real+DM-CS*, and *Real+DM-OS*. Specifically, methods trained with *Real+GANs* use real images together with deepfakes generated by GANs. Methods in *Real+DM-CS* are trained with real images and deepfakes generated using closed-source diffusion model implementations, while *Real+DM-OS* methods use deepfakes produced by open-source diffusion models. Note that for all the configurations, our method is trained *only* with real samples.

In *Real+GANs*, all compared methods underperform into generalising to diffusion-based forgeries (DM-CS and DM-OS). The average ACC across the three OOD test domains is significantly lower than ours. The second best-performing comparison method in this setting, ODDN, achieves an average ACC of 81.8, whereas our method reaches 96.8, yielding a significant improvement of +15 in ACC, respectively. These results show that SOTA methods struggle in generalising to different categories of generators (GANs vs. DMs). In the *Real+DM-CS*

Table 1: Comparisons of ***μFlow*** with state-of-the-art approaches trained on different training sets (Real + GANs, Real + DM-CS, or Real + DM-OS), whereas our method is trained only on real images. ACC: Accuracy. AUC: Area Under the Curve. AP: Average Precision. DM-CS: Diffusion Model-Closed Source. DM-OS: Diffusion Model-Open Source. **Bold**: best performing method; underlined: second-best method. *Mix** refers to detectors trained on fake samples generated with a subset of GANs and DMs, while still being tested on unseen architectures (GANs, DM-CS, and DM-OS).

			Out-of-domain Testing									**Average**		
Trained on	Method	Venue	GANs			DM-CS			DM-OS					
			ACC ↑	AUC ↑	AP ↑	ACC ↑	AUC ↑	AP ↑	ACC ↑	AUC ↑	AP ↑	ACC ↑	AUC ↑	AP ↑
Real + GANs	DFX-SN [44]	MTA25	-	-	-	74.1	71.7	69.9	72.5	69.0	67.0	73.3	70.1	68.0
	FreqNet [55]	AAAI24	-	-	-	76.4	75.5	74.1	77.0	73.2	74.2	76.7	74.3	75.4
	NPR [56]	CVPR24	-	-	-	78.6	79.2	78.9	76.1	77.3	80.1	77.4	78.2	79.7
	ODDN [57]	AAAI25	-	-	-	81.4	83.2	84.8	82.2	82.9	85.3	81.8	83.6	85.0
	D^3 [62]	CVPR25	-	-	-	80.5	82.0	88.8	81.3	83.1	87.0	80.9	82.0	87.9
Real-only	***μFlow* (Ours)**		-	-	-	**96.8**	**96.8**	**95.9**	**96.9**	**96.8**	**96.1**	**96.8**	**96.8**	**96.0**
Real + DM-CS	DFX-SN [44]	MTA25	67.9	67.0	66.1	-	-	-	80.2	84.6	86.0	74.1	75.8	76.0
	FreqNet [55]	AAAI24	68.3	66.0	65.3	-	-	-	86.9	94.0	92.8	77.6	80.0	79.0
	NPR [56]	CVPR24	68.5	68.5	69.1	-	-	-	85.9	93.4	92.7	77.2	81.0	80.9
	ODDN [57]	AAAI25	84.5	79.2	75.9	-	-	-	97.5	**98.5**	91.7	91.0	88.8	83.8
	D^3 [62]	CVPR25	75.1	83.4	85.0	-	-	-	**98.6**	97.9	**99.9**	86.8	90.7	92.5
Real-only	***μFlow* (Ours)**		**90.3**	**90.4**	**93.9**	-	-	-	96.9	96.8	96.1	**93.6**	**93.6**	**95.0**
Real + DM-OS	DFX-SN [44]	MTA25	69.9	70.4	68.6	81.2	80.7	83.0	-	-	-	75.6	75.6	75.8
	FreqNet [55]	AAAI24	71.8	67.6	70.7	87.7	94.7	94.4	-	-	-	79.8	81.2	82.6
	NPR [56]	CVPR24	75.1	68.7	73.5	92.1	97.7	**98.0**	-	-	-	83.6	83.2	85.8
	ODDN [57]	AAAI25	85.2	71.2	70.3	98.4	**99.5**	96.7	-	-	-	91.8	85.3	83.5
	D^3 [62]	CVPR25	71.1	76.6	77.9	**99.7**	98.0	97.1	-	-	-	85.4	87.3	87.5
Real-only	***μFlow* (Ours)**		**90.3**	**90.4**	**93.9**	96.8	96.8	95.9	-	-	-	**93.5**	**93.6**	**94.9**
Real + Mix*	DFX-SN [44]	MTA25	75.2	77.9	77.1	77.0	80.4	82.2	81.2	81.9	82.0	77.8	80.1	80.4
	FreqNet [55]	AAAI24	79.9	86.0	87.5	77.9	84.6	84.6	73.5	73.2	76.8	77.1	81.3	83.0
	NPR [56]	CVPR24	83.0	89.8	89.5	87.4	96.5	90.2	85.4	95.7	95.4	85.3	94.0	91.7
	ODDN [57]	AAAI25	**92.7**	91.5	93.5	90.3	83.3	86.5	91.1	86.3	85.6	91.4	87.0	88.5
	D^3 [62]	CVPR25	90.4	**92.5**	93.7	95.8	94.6	92.6	91.7	96.5	**96.4**	92.6	94.5	94.2
Real-only	***μFlow* (Ours)**		90.3	90.4	**93.9**	**96.8**	**96.8**	**95.9**	**96.9**	**96.8**	96.1	**94.7**	**94.7**	**95.3**
Real-only	OC-FakeDect [28]	CVPR20	48.9	51.1	50.0	62.9	62.3	59.8	60.4	60.9	59.8	57.4	58.1	56.5
	AFSD [34]	PRL	67.3	69.0	69.3	64.7	61.3	67.0	63.0	62.9	70.1	65.0	64.4	68.8
	PCL+I2G [68]	ICCV21	84.1	86.0	92.2	75.0	78.7	69.3	73.6	73.0	68.3	77.6	79.2	76.6
	SBI [49]	CVPR22	84.3	86.8	93.0	75.2	79.4	69.7	74.0	73.2	68.5	77.8	79.8	77.1
	AUNet [1]	CVPR23	84.8	87.0	93.3	75.3	80.0	70.2	74.3	73.5	70.1	78.1	80.2	77.9
	LAA-Net [39]	CVPR24	86.0	88.2	**94.2**	83.2	84.1	81.0	79.9	82.3	83.3	83.0	84.9	86.2
	***μFlow* (Ours)**		**90.3**	**90.4**	93.9	**96.8**	**96.8**	**95.9**	**96.9**	**96.8**	**96.1**	**94.7**	**94.7**	**95.3**

training setting, the performance improves substantially, particularly for ODDN. By training on DM-CS, D^3 increases performance on other DMs (DM-OS), showing its ability to generalise within the same category of generators (DMs). Nonetheless, our real-only approach still outperforms it (+2.6 ACC). Similarly, in the *Real+DM-OS* scenario we observe the strongest performance among the SOTA methods, particularly with D^3 (85.4 ACC). However, our method still outperforms the SOTA (+1.7 ACC).

Generalisation results mixing generator categories. Here, we do not separate deepfakes by generator category. All competing methods are trained on a mixed dataset containing both GANs and DMs images, while testing on unseen architectures from both categories. As shown in Tab. 1 (*Real+Mix** group) competing methods benefit from training from a heterogeneous set of generators.

Table 2: Performance under several content-preserving transformations.

Method	Resize			Horizontal Flip			Gaussian Noise			Salt Pepper			**Average**		
	ACC ↑	AUC ↑	AP ↑	ACC ↑	AUC ↑	AP ↑	ACC ↑	AUC ↑	AP ↑	ACC ↑	AUC ↑	AP ↑	ACC ↑	AUC ↑	AP ↑
DFX-SN [44]	72.1	73.0	73.5	71.8	75.3	74.1	70.9	74.2	72.4	74.9	72.9	76.4	72.4	73.8	74.1
FreqNet [55]	76.0	81.8	81.7	76.3	74.3	79.9	75.0	78.8	76.4	73.9	79.8	77.3	75.3	78.7	78.8
NPR [56]	85.0	92.8	91.1	84.4	87.9	88.0	77.2	80.7	79.9	70.3	74.1	75.2	79.2	83.9	83.5
ODDN [57]	89.4	**94.8**	**95.9**	89.3	90.2	**92.7**	76.9	74.5	74.9	75.2	77.7	76.1	82.7	84.3	84.9
D^3 [62]	89.1	91.7	91.3	90.3	91.8	91.3	88.2	**91.9**	**92.0**	90.0	90.1	**93.8**	89.4	91.4	**92.1**
PCL+I2G [68]	76.0	76.1	78.7	75.9	76.5	79.8	68.8	74.7	77.2	65.5	74.7	76.2	71.5	75.5	78.0
SBI [49]	76.4	76.3	80.1	77.0	77.1	80.5	69.9	75.1	77.9	66.1	75.3	76.9	72.3	75.9	78.8
AUNet [1]	76.6	76.7	80.0	77.4	77.4	80.2	70.3	75.8	78.4	67.0	75.8	77.5	72.8	76.4	79.0
LAA-NET [39]	82.3	83.0	84.7	81.3	82.5	83.4	81.0	82.1	80.9	79.9	78.7	80.1	81.1	81.6	82.3
***µFlow* (Ours)**	**90.6**	91.5	88.4	**92.6**	**92.7**	91.9	**89.7**	89.9	91.7	**90.8**	**91.7**	89.6	**90.9**	**91.5**	90.4

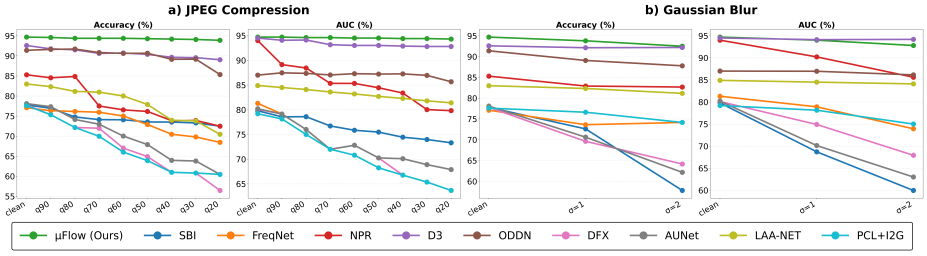

Fig. 4: ACC and AUC under inference-time degradations: (a) JPEG compression with decreasing quality factor; (b) Gaussian blur with increasing standard deviation.

This strengthens our hypothesis that such methods learn specific traces injected by generators and thus benefit of seeing multiple generators during training. D^3 and ODDN perform strongly, reaching ACC of 92.6 and 91.4, respectively. However, despite our method is trained only on real images, ***µFlow*** obtains an average ACC 94.7 ACC, improving the second-best performing approach by +2.1.

Comparisons with real-only training baselines. Finally, we compare ***µFlow*** with prior real-only training approaches, namely PCL+I2G, SBI, AUNet, and LAA-Net. While these methods do not rely on fake samples during training, their generalisation capability remains limited. In particular, the best-performing baseline in this group, LAA-Net, achieves an average ACC of 83.0, whereas our method reaches 94.7, yielding a substantial improvement of +11.7. The gap is especially pronounced on diffusion-based forgeries (DM-CS and DM-OS), where our approach consistently obtains nearly 97% ACC, largely outperforming all competitors. These results demonstrate that simply training on real images is not sufficient to ensure strong OOD generalisation, which is instead achieved by leveraging the discriminative power of average images.

Table 3: Ablation study on ***μFlow*** across target distributions and inference classifiers.

Target distribution	Classifier	α	ACC	AUC	AP
$\mathcal{N}(0, \mathbb{I})$	LOF	-	59.1	59.9	63
	OC-SVM	-	62.1	61.9	66
	τ	0.1	54.8	55.0	60.4
		0.05	57.6	56.9	66.0
		0.01	59.9	62.0	63.2
$\mathcal{N}(\mu^\star, \Sigma^\star)$	LOF	-	<u>94.2</u>	94.1	90.9
	OC-SVM	-	93.9	**95.3**	89.7
	τ	0.1	89.7	88.3	91.3
		0.05	93.9	93.6	<u>95.0</u>
		0.01	**94.7**	<u>94.7</u>	**95.3**

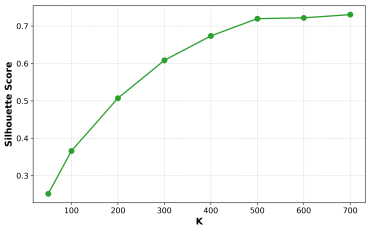

Fig. 5: Silhouette score with different values of K.

4.2 Analysis

Robustness Analysis. Tab. 2 shows the robustness of our approach under content-preserving transformations (resize, horizontal flip, Gaussian noise, salt pepper). Although ***μFlow*** is not trained against these transformations, performance marginally reduces, achieving comparable results with respect to the state-of-the-art. Fig. 4 shows experiments by applying signal-degrading perturbations to test images. Overall, ***μFlow*** is not highly impacted by such perturbations, compared to the state-of-the-art. Moreover, it maintains stable performance with the increasing of perturbation intensity, showing the robustness of our latent representations.

Ablation Studies. In this section, we analyse contributions of each component of our method. Specifically, we compare results using the discriminative distribution $\mathcal{N}(\mu^* \Sigma^*)$ rather than the normal Gaussian distribution. Moreover, we compare the use of the fixed threshold τ wrt. one-class naive machine learning models, such as Local Outlier Factor (LOF) and One-class Support Vector Machine (OC-SVM). Tab. 3 (top part) show that using the normal Gaussian distribution fails to separate the classes. Differently, using the discriminative space $\mathcal{N}(\mu^* \Sigma^*)$ learnt from average images, our method drastically increase performance, validating our hypothesis to leverage the discriminative power of average images. Tab. 3 (bottom part) shows that our method obtains similar performances using both a fixed threshold or naive machine learning models. Finally, we report an ablation study over different values of α. Although the overall performance remains stable across the explored range, we observe a consistent—albeit moderate—improvement for $\alpha = 0.01$, which yields the best trade-off between detection sensitivity and false positive control. Consequently, we adopt $\alpha = 0.01$ in all subsequent experiments.

Evaluating sample size for average images. The training of ***μFlow*** also relies on the choice of the number K of training images to use to determine the average images (Eq. (2)). Intuitively, larger K better suppresses image-specific noise while amplifying generator-specific traces, yielding more discriminative features. To choose the best value of K, we compute the *Silhouette Score* across a wide range of values (Fig. 5). This metric evaluates how much clusters are

separated and thus the discriminative power of features. The score follows a logarithmic growth trend, reaching 0.72 at $K = 500$ and 0.74 at $K = 700$. Although average images are computed only once prior to training, higher values of K imply an increasing computational time. We therefore set $K = 500$ as a trade-off between discriminative power and efficiency.

5 Conclusion & Limitations

In this paper, we introduced ***μFlow***, an out-of-distribution deepfake detector. Unlike the state-of-the-art, we train our model with real images only and we use a modified formulation of normalised flow to take advantage of the discriminative power of average images. Results on real and generated images of faces have shown that ***μFlow*** outperforms the state-of-the-art across the board. Nonetheless, our work presents some limitations, being restricted to detecting face deepfakes. As future work, we plan to extend the method to handle multiple semantic categories, as well as apply it to videos.

Acknowledgements

Orazio Pontorno is a PhD candidate enrolled in the National PhD in Artificial Intelligence, XXXIX cycle, organized by Università Campus Bio-Medico di Roma.
This work was supported by the DEFORM project, funded by the European Union's Horizon Europe Research and Innovation Programme under Grant Agreement No. 101308502.

References

1. Bai, W., Liu, Y., Zhang, Z., Li, B., Hu, W.: Aunet: Learning relations between action units for face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24709–24719 (2023)
2. Black Forest Labs: FLUX.1: Open-weight 12b parameter text-to-image model (official announcement). `https://bfl.ai/blog/24-08-01-bfl` (2024), official launch announcement introducing the FLUX.1 model suite (whitepaper/blog post)
3. Black Forest Labs: FLUX 1.1 [pro]: Advanced text-to-image generation model (2024), `https://blackforestlabs.ai/1-1-pro/`, official documentation page (model description, not an academic paper)
4. Bongini, P., Mandelli, S., Montibeller, A., Casu, M., Pontorno, O., Ragaglia, C.V., Zanchetta, L., Aquilina, M., Wani, T.M., Guarnera, L., Tondi, B., Boato, G., Bestagini, P., Amerini, I., De Natale, F., Battiato, S., Barni, M.: Wild: a new in-the-wild image linkage dataset for synthetic image attribution. In: 2025 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2025). `https://doi.org/10.1109/IJCNN64981.2025.11227289`
5. Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., Cohen-Or, D.: Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. ACM Transactions on Graphics (Proc. SIGGRAPH) **42**(4), 148:1–148:10 (2023). `https://doi.org/10.1145/3592116`

6. Cho, W., Choi, S., Park, D.K., Shin, I., Choo, J.: Image-to-image translation via group-wise deep whitening-and-coloring transformation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10639–10647 (2019)
7. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8789–8797 (2018)
8. Coccomini, D., Caldelli, R., Falchi, F., Gennaro, C.: On the generalization of deep learning models in video deepfake detection. Journal of Imaging **9** (2023). https://doi.org/10.3390/jimaging9050089
9. Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., Verdoliva, L.: On the detection of synthetic images generated by diffusion models. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 1–5 (2022), https://api.semanticscholar.org/CorpusID:253254809
10. Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., Verdoliva, L.: On the detection of synthetic images generated by diffusion models. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
11. Cozzolino, D., Poggi, G., Corvi, R., Nießner, M., Verdoliva, L.: Raising the bar of ai-generated image detection with clip. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 4356–4366 (2023), https://api.semanticscholar.org/CorpusID:265552100
12. DeepAI: Deepai text-to-image generator (2024), https://deepai.org/machine-learning-model/text2img, online text-to-image generation service (documentation page)
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
14. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. NIPS '21, Curran Associates Inc., Red Hook, NY, USA (2021)
15. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. ArXiv **abs/2208.01618** (2022), https://api.semanticscholar.org/CorpusID:251253049
16. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in Neural Information Processing Systems **27** (2014)
17. Guarnera, L., Giudice, O., Battiato, S.: Deepfake detection by analyzing convolutional traces. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 666–667 (2020)
18. Gudovskiy, D., Ishizaka, S., Kozuka, K.: Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1819–1828 (2022). https://doi.org/10.1109/WACV51458.2022.00188
19. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: Attgan: Facial attribute editing by only changing what you want. IEEE transactions on image processing **28**(11), 5464–5478 (2019)

20. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
21. Hotpot AI: Hotpot ai: Ai-powered image and text generation tools (2024), https://hotpot.ai/, official website of Hotpot.ai image generator service
22. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (ICLR) (2018)
23. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Advances in Neural Information Processing Systems 34 (NeurIPS 2021) (2021), https://proceedings.neurips.cc/paper/2021/hash/076ccd93ad68be51f23707988e934906-Abstract.html
24. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
25. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8107–8116 (2020). https://doi.org/10.1109/CVPR42600.2020.00813
26. Kaur, A., Noori Hoshyar, A., Saikrishna, V., Firmin, S., Xia, F.: Deepfake video detection: challenges and opportunities. Artificial Intelligence Review **57**(6), 159 (May 2024). https://doi.org/10.1007/s10462-024-10810-6, https://doi.org/10.1007/s10462-024-10810-6
27. Ke, J., Wang, L.: Df-udetector: An effective method towards robust deepfake detection via feature restoration. Neural Networks **160**, 216–226 (2023)
28. Khalid, H., Woo, S.S.: Oc-fakedect: Classifying deepfakes using one-class variational autoencoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 656–657 (2020)
29. Khan, S.A., Dang-Nguyen, D.T.: Deepfake detection: analyzing model generalization across architectures, datasets, and pre-training paradigms. IEEE Access **12**, 1880–1908 (2023)
30. Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2416–2425 (2022). https://doi.org/10.1109/CVPR52688.2022.00246
31. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
32. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper_files/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf
33. Larue, N., Vu, N.S., Struc, V., Peer, P., Christophides, V.: Seeable: Soft discrepancies and bounded contrastive learning for exposing deepfakes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21011–21021 (2023)
34. Leyva, R., Sanchez, V., Epiphaniou, G., Maple, C.: Data-agnostic face image synthesis detection using bayesian cnns. Pattern Recognition Letters **183**, 64–70 (2024)

35. Li, Z., Zhang, J., Lin, Q., Xiong, J., Long, Y., Deng, X., Zhang, Y., Liu, X., Huang, M., Xiao, Z., Chen, D., He, J., Li, J., Li, W., Zhang, C., Quan, R., Lu, J., Huang, J., Yuan, X., Zheng, X., Li, Y., Zhang, J., Zhang, C., Chen, M., Liu, J., Fang, Z., Wang, W., Xue, J., Tao, Y., Zhu, J., Liu, K., Lin, S., Sun, Y., Li, Y., Wang, D., Chen, M., Hu, Z., Xiao, X., Chen, Y., Liu, Y., Liu, W., Wang, D., Yang, Y., Jiang, J., Lu, Q.: Hunyuan-DiT: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding (2024), arXiv preprint – Tencent Hunyuan team diffusion model
36. Lopez, J.: Introducing stable cascade. Stability AI Official Blog (Feb 2024), https://stability.ai/news/introducing-stable-cascade, official research preview announcement by Stability AI
37. Mejri, N., Ghorbel, E., Aouada, D.: Untag: Learning generic features for unsupervised type-agnostic deepfake detection. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5 (2023). https://doi.org/10.1109/ICASSP49357.2023.10095983
38. Midjourney, Inc.: Midjourney (text-to-image generative service) (2022), https://www.midjourney.com/, proprietary AI image generation model (no public technical paper)
39. Nguyen, D., Mejri, N., Singh, I.P., Kuleshova, P., Astrid, M., Kacem, A., Ghorbel, E., Aouada, D.: Laa-net: Localized artifact attention network for quality-agnostic and generalizable deepfake detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17395–17405 (2024)
40. Ojha, U., Li, Y., Lee, Y.J.: Towards universal fake image detectors that generalize across generative models. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 24480–24489 (2023), https://api.semanticscholar.org/CorpusID:257038440
41. OpenAI: DALL·E 3 Technical Report (2024), https://cdn.openai.com/papers/dall-e-3.pdf, official technical report
42. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. In: International Conference on Learning Representations (ICLR) (2024), https://arxiv.org/abs/2307.01952, accepted to ICLR 2024 – arXiv preprint arXiv:2307.01952
43. Pontorno, O., Guarnera, L., Battiato, S.: On the exploitation of dct-traces in the generative-ai domain. In: 2024 IEEE international conference on image processing (ICIP). pp. 3806–3812. IEEE (2024)
44. Pontorno, O., Guarnera, L., Battiato, S.: Deepfeaturex-sn: Generalization of deepfake detection via contrastive learning. Multimedia Tools and Applications pp. 1–20 (2025)
45. Robertson, D.J., Kramer, R.S.S., Burton, A.M.: Face averages enhance user recognition for smartphone security. PLOS ONE **10**(3), 1–11 (03 2015). https://doi.org/10.1371/journal.pone.0119460, https://doi.org/10.1371/journal.pone.0119460
46. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models . In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10674–10685. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2022). https://doi.org/10.1109/CVPR52688.2022.01042, https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042

47. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation . In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22500–22510. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2023). https://doi.org/10.1109/CVPR52729.2023.02155, https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.02155
48. Saharia, C., Chan, W., Saxena, S., Lit, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Gontijo-Lopes, R., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
49. Shiohara, K., Yamasaki, T.: Detecting deepfakes with self-blended images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18720–18729 (2022)
50. Soudy, A.H., Sayed, O., Tag-Elser, H., Ragab, R., Mohsen, S., Mostafa, T., Abohany, A.A., Slim, S.O.: Deepfake detection using convolutional vision transformers and convolutional neural networks. Neural Computing and Applications **36**(31), 19759–19775 (2024)
51. Stability AI: Introducing stable diffusion 3.5. Stability AI Official Blog (Oct 2023), https://stability.ai/news/introducing-stable-diffusion-3-5, official model release announcement
52. Stamnas, S., Sanchez, V.: Difffake: Exposing deepfakes using differential anomaly detection. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV) Workshops. pp. 695–705 (February 2025)
53. Starry AI: Starry ai (ai art generator) (2023), https://starryai.com/, commercial text-to-image generator service (no academic publication)
54. Tan, C., Tao, R., Liu, H., Gu, G., Wu, B., Zhao, Y., Wei, Y.: C2p-clip: Injecting category common prompt in clip to enhance generalization in deepfake detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 7184–7192 (2025)
55. Tan, C., Zhao, Y., Wei, S., Gu, G., Liu, P., Wei, Y.: Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5052–5060 (2024)
56. Tan, C., Zhao, Y., Wei, S., Gu, G., Liu, P., Wei, Y.: Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28130–28139 (2024)
57. Tao, R., Le, M., Tan, C., Liu, H., Qin, H., Zhao, Y.: Oddn: Addressing unpaired data challenges in open-world deepfake detection on online social networks. The 39th Annual AAAI Conference on Artificial Intelligence (AAAI 2025) (2024)
58. Torralba, A., Oliva, A.: Statistics of natural image categories. Network: computation in neural systems **14**(3), 391 (2003)
59. Torralba, A., Oliva, A.: Statistics of natural image categories. Network: Computation in Neural Systems **14**(3), 391–412 (2003). https://doi.org/10.1088/0954-898X_14_3_302, https://doi.org/10.1088/0954-898X_14_3_302, pMID: 12938764
60. Wang, R., Juefei-Xu, F., Ma, L., Xie, X., Huang, Y., Wang, J., Liu, Y.: Fakespotter: a simple yet robust baseline for spotting ai-synthesized fake faces. In: Proceedings

of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. pp. 3444–3451 (2021)
61. Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., Han, S.: Sana: Efficient high-resolution image synthesis with linear diffusion transformer (2024), https://arxiv.org/abs/2410.10629, arXiv preprint – NVIDIA Research (SANA model)
62. Yang, Y., Qian, Z., Zhu, Y., Russakovsky, O., Wu, Y.: Dˆ 3: Scaling up deepfake detection by learning from discrepancy. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 23850–23859 (2025)
63. Yang, Y., Wang, R., Qian, Z., Zhu, Y., Wu, Y.: Diffusion in diffusion: Cyclic one-way diffusion for text-vision-conditioned generation. ArXiv **abs/2306.08247** (2023), https://api.semanticscholar.org/CorpusID:259165446
64. Yermakov, A., Cech, J., Matas, J.: Unlocking the hidden potential of clip in generalizable deepfake detection. ArXiv **abs/2503.19683** (2025), https://api.semanticscholar.org/CorpusID:277313445
65. Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R., Wu, L.: Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. arXiv preprint arXiv:2111.07677 (2021)
66. Yuan, Z., Wang, K., Quan, W., Yan, D.M., Wu, T.: Clip-flow: A universal discriminator for ai-generated images inspired by anomaly detection. In: Proceedings of the 1st on Deepfake Forensics Workshop: Detection, Attribution, Recognition, and Adversarial Challenges in the Era of AI-Generated Media. pp. 3–11 (2025)
67. Zhang, X., Karaman, S., Chang, S.F.: Detecting and simulating artifacts in gan fake images. In: 2019 IEEE international workshop on information forensics and security (WIFS). pp. 1–6. IEEE (2019)
68. Zhao, T., Xu, X., Xu, M., Ding, H., Xiong, Y., Xia, W.: Learning self-consistency for deepfake detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 15023–15033 (2021)