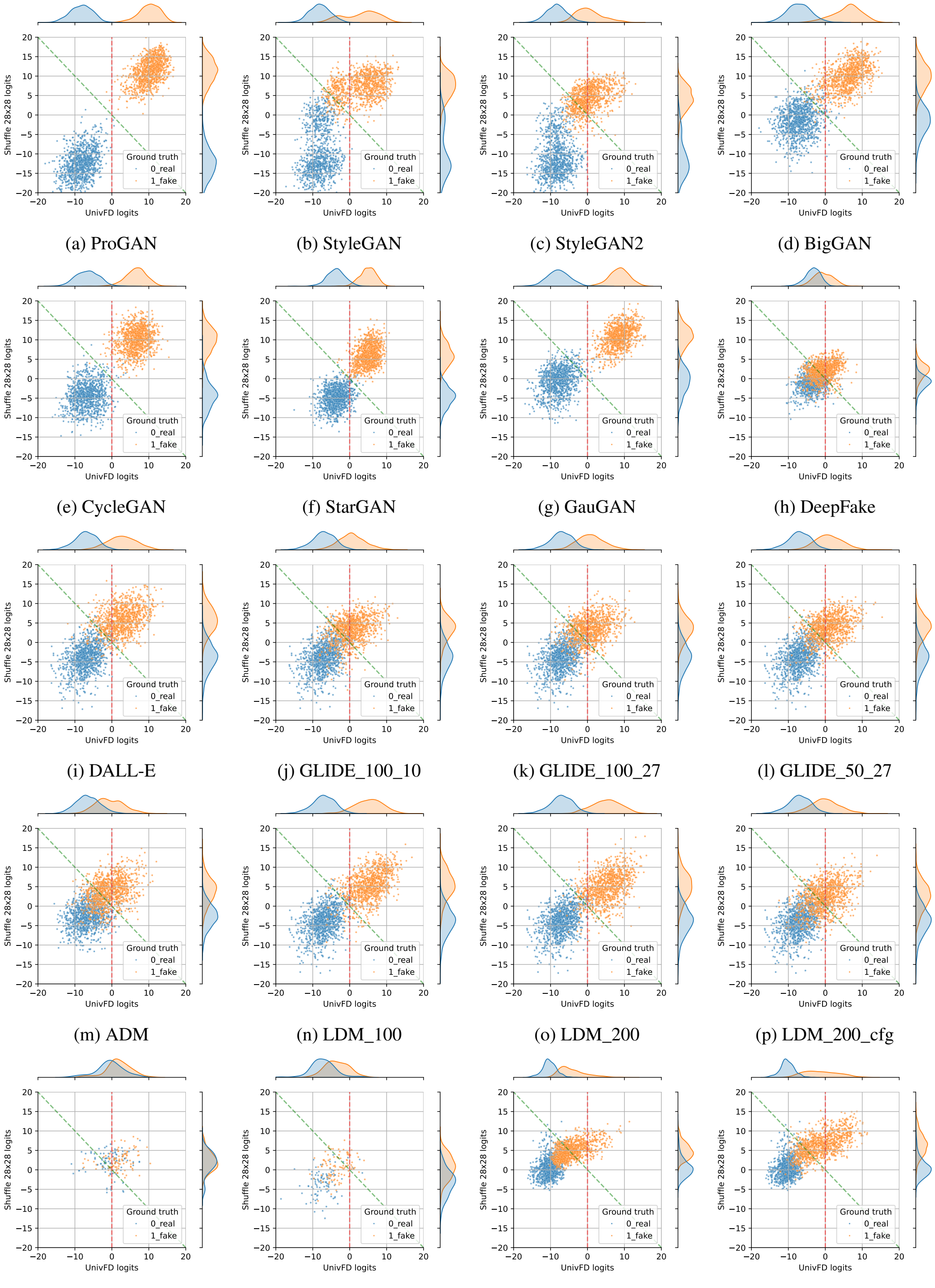
A. Additional results on scatter plots

Additional results to Sec. 4.2 are presented in Fig. 10.



(a) ProGAN

(b) StyleGAN

(c) StyleGAN2

(d) BigGAN

(e) CycleGAN

(f) StarGAN

(g) GauGAN

(h) DeepFake

(i) DALL-E

(j) GLIDE_100_10

(k) GLIDE_100_27

(l) GLIDE_50_27

(m) ADM

(n) LDM_100

(o) LDM_200

(p) LDM_200_cfg

(q) SITD

(r) SAN

(s) CRN

(t) IMLE

Figure 10. Scatter plots of per-sample scores. X-axis is UnivFD logits, and Y-axis is the logit from PatchShuffle with patch size 28. The decision boundary of UnivFD (red) and SFLD (green) are shown.

B. Datasets

B.1. Train dataset

To establish a baseline for comparison, we adopt the most common setting for training the detection model, namely the train set from ForenSynths [49]. The train set consists of real images and ProGAN [18]-generated images. It involves 20 different object class categories, each containing 18K real images from the different LSUN [51] datasets and 18K synthetic images generated by ProGAN.

B.2. Test dataset

We evaluate the performance of SFLD on (1) conventional benchmarks, (2) TwinSynths which we proposed, (3) low-level vision and perceptual loss benchmarks. In this section, we provide a detailed description of the configurations for the conventional benchmarks and low-level vision and perceptual loss benchmarks.

Conventional benchmark This is from ForenSynths [49] and Ojha *et al*. [32], including 16 different subsets of generated images, synthesized by seven GAN-based generative models, eight diffusion-based generative models and one deepfake model. The subset of GAN-based fake images are from ForenSynths [49], including ProGAN [18], StyleGAN [19], StyleGAN2 [20], BigGAN [2], CycleGAN [55], StarGAN [7], and GauGAN [34]. The subset of diffusion-based fake images are from Ojha *et al*. [32], including DALL-E [10], three different variants of Glide [31], ADM(guided-diffusion) [12], and three different variants of LDM [38]. Deepfake set is from FaceForensices++ [40] which is included in ForenSynths [49]. The real images corresponding to the fake images described above were directly taken from the same datasets. Those are sampled from LSUN [51], ImageNet [41], CycleGAN [55], CelebA [24], COCO [22], and FaceForensics++ [40].

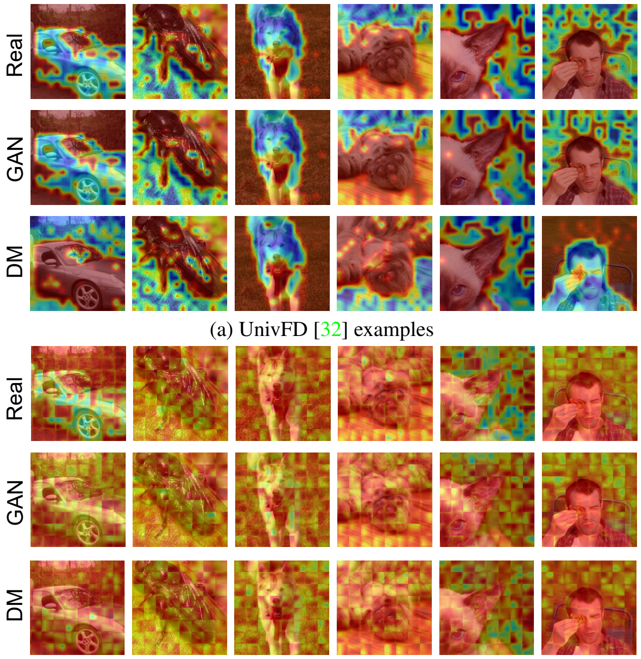
Low-level vision and perceptual loss benchmarks Low-level vision benchmark consists of SITD [5] and SAN [9]. These are image processing models that approximate long exposures in low light conditions from short exposures in raw camera input or process super-resolution on low-resolution images. Perceptual benchmark consists of CRN [6] and IMLE [21]. These models color the semantic segmentation map into a realistic image while directly optimizing a perceptual loss. These benchmarks are from ForenSynths [49].

C. Qualitative analysis on TwinSynths dataset

We show the GradCAM visualization of UnivFD [32] and Patch-shuffle 28×28 using the TwinSynths dataset in Fig. 11. Similar to Sec. 4.4, UnivFD is shown to focus on the class-dependent salient region, whereas our method focuses on the entire image region. Moreover, we observed that for TwinSynths dataset, UnivFD does respond identically to real/fake images which indicate its inability to capture subtle fake image fingerprints, whereas our method shows the response to such a difference.

Benchmark	SFLD (224+24)		SFLD (224+56)		SFLD	
	center	full image	center	full image	center	full image
main benchmark	98.04	98.03	98.37	98.39	98.40	98.43
CRN	94.41	96.62	94.17	97.24	91.97	95.79
IMLE	97.55	98.65	98.12	99.23	96.92	98.64
SITD	59.36	64.82	67.71	76.66	60.38	71.90

Table 6. mAP results of the various sizes of test images, comparing two different patch selecting methods. *Center* denotes that the images have been center-cropped to 224×224, while *full image* means that random patches from the full image have been combined to reconstruct a 224×224 image.



(a) UnivFD [32] examples

(b) PatchShuffle(patch size 28) examples

Figure 11. Class activation maps (CAM) for UnivFD [32] and the patch-shuffled detector (ours) in TwinSynths dataset. Each row shows examples from TwinSynths-real, TwinSynths-GAN, TwinSynths-DM sets. GradCAM [15, 43] was used to obtain the heatmaps.

D. Effect of selecting patches from the whole image

Fig. 12 illustrates the concept of patch extraction of SFLD mentioned in Sec. 2.1. Unlike many alternative detection methodologies, SFLD extracts patches from any position within the input image at the test time. This approach enhances the detector's receptive field and improves perfor-

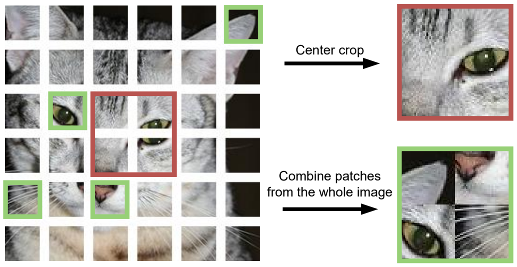


Figure 12. Illustration of the test input processing strategy. In typical methods, a test image is center-cropped before being passed to the detector. Our patch shuffling strategy allows us to select patches from the entire image region, effectively increasing its receptive field.

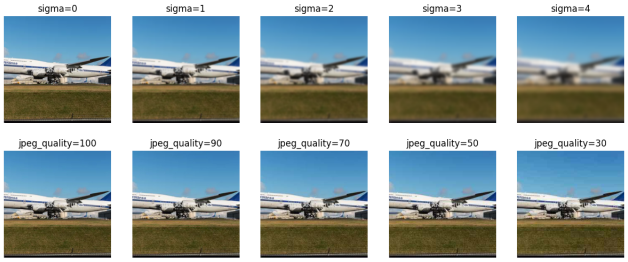


Figure 13. Examples of two image degradation

mance for images that have higher resolution than 224×224. In Tab. 6, we compare results on benchmarks that have high-resolution images. We consider different SFLD ensemble options and the location of the selected patch. The main benchmark consists mostly of 256×256 images, which have little margin with a 224×224 center crop. Meanwhile, the CRN and IMLE benchmarks have 512×256 images, and the SITD benchmark includes images much larger up to 2,848×4,256 or 4,032×6,030.

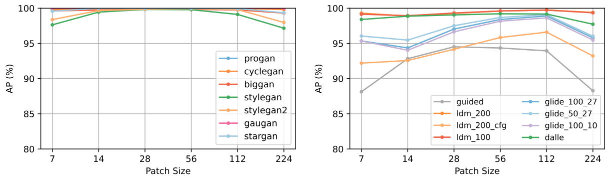
We observed that the discrepancy between the two methodologies was minimal when the test image was small. However, as the image size increased, the performance of the method that solely focused on the center of an image became increasingly constrained.

E. Image degradation examples

Fig. 13 shows examples of image gradations. According to our definition of high- and low-level features, we can consider that the gaussian blur attacks both high- and low-level features in the image, and the JPEG compression attacks on low-level features in the image.

F. Robustness against image degradation

Since image degradation was not considered during training, it may be useful to examine the changes in output distribution (as shown in Fig. 16 in supplementary material) to analyze the model's operational tendencies in detail. Fig. 16 reveals distinctions between the high-level feature model (UnivFD Fig. 16b), low-level feature model (NPR Fig. 16c), and integrated model. The distributions of SFLD and UnivFD remain distinguishable, despite a slight decline in discrimination performance. However, NPR aligns real and generated images into the same distribution. This behavior arises from the operational mechanism of each model. NPR primarily focuses on low-level features, resulting in a catastrophic failure to maintain robustness against JPEG compression. UnivFD demonstrates relative robustness due to its emphasis on high-level features through CLIP visual encoders; however, there is a slight performance penalty because the visual encoder does not completely disregard low-level features. In contrast, SFLD exhibits robustness against JPEG compression by integrating both high- and low-level features through ensemble/fusion, allowing each to compensate for the information lost in the other.



(a) GAN-based generators. (b) Diffusion-based generators.

Figure 14. Results of the ensemble models of UnivFD and the patch-shuffled model with each patch size. For 224, it is the same as UnivFD.

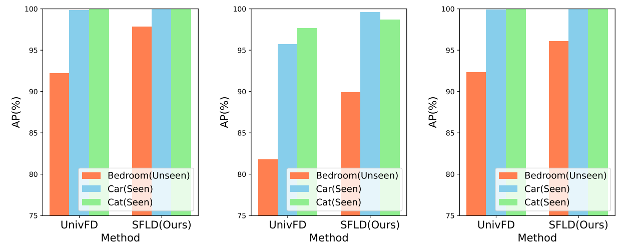
G. Effect of patch sizes

To supplement Fig. 9a in the main text, we checked the AP for each generator, rather than the average AP on the conventional benchmark. Fig. 14 illustrates that SFLD consistently maintains high performance as long as the patch size is not smaller than the patch size of the image encoder backbone. This is because when the shuffling patch size s_N is smaller than the ViT's patch size, the input tokens are affected by patch-shuffling to get an unnatural image patch, resulting in the encoder not properly embedding the visual feature.

H. Ablation on the pre-trained image encoder

The pre-trained image encoder is employed to learn the features of the "real" class. According to [32], directly fine-tuning the encoder makes the detector overfit to a specific generator used in training. This results in low generalization to unseen generators. Therefore, we utilized the frozen CLIP:ViT-L/14 model following UnivFD.

Tab. 7 show that our patch shuffling and ensembling



(a) CLIP-ViT (b) DINOv2-ViT (c) OpenCLIP-ViT

Figure 15. Class-wise detection results for StyleGAN-{*bedroom, car, cat*} class categories reported in AP. *bedroom* class is a novel class that is not in the training set.

strategy improves the performance regardless of the pre-trained backbone. All models are trained only with real and generated images from ProGAN and tested on the various unseen generated images in conventional benchmark. For ImageNet-ViT, we used ViT-B/16 model, following UnivFD paper [32]. Since its encoders have patch size of 16, we utilized 16 and 32 for patch sizes instead of 28 and 56. Moreover, note that simply employing different pre-training datasets or strategies – ImageNet, DINOv2, OpenCLIP – does not address the content bias problem. (see Fig. 15)

I. In-the-wild applications of SFLD

We applied our SFLD to in-the-wild AI-generated image detection, especially to a deepfake detection benchmark. We have already demonstrated performance on a FaceForensics++ [39] subset, which is a deepfake detection benchmark created using face manipulation software [11]. Here, we have added Tab. 8 with experiments using Generated Faces in the Wild [1] datasets. SFLD shows state-of-the-art performance in detecting real-world deepfakes.

J. Pseudocode of SFLD

See Algorithm 1.

K. Related works

AI-generated image detection on specific image generation models Research on distinguishing between synthetic and real images using deep learning models has increased with the development of image generation models.

Early works were focused on finding the fingerprints in images generated with GANs, which were targeted at high-performing image generation models. Two major approaches were the use of statistics from the image domain [28, 30] and the training of CNN-based classifiers. In particular, in the case of using CNNs, there are two main approaches: focusing on the image domain [29, 47, 52] or the frequency domain [14, 27, 48]. Specifically, GAN-generated images have been found to exhibit sharp periodic artifacts in this frequency domain, leading to a variety of applications [8, 14, 37].

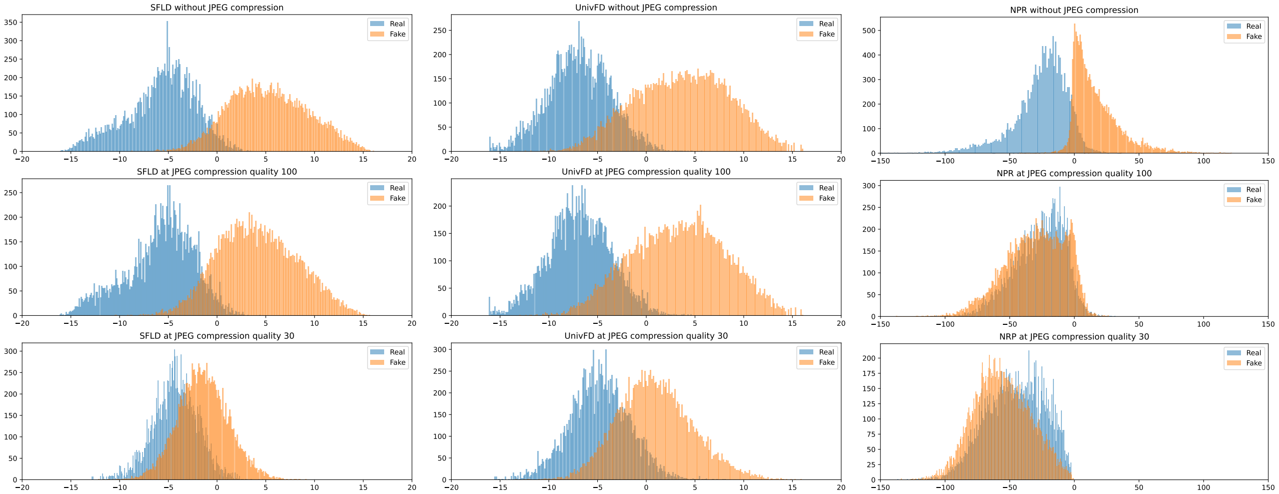
Algorithm 1 PyTorch-style pseudocode of SFLD

```
"""
Args:
    image: A test image instance
    n_views: Number of views for random patch shuffle
         averaging. Defaults to 10.
    visual_encoder: A CLIP-pretrained ViT-L/14 visual
         encoder.
Returns:
    output: a real/fake score normalized to [0,1] range.
"""

# prediction from 224x224 unshuffled view
feature = visual_encoder(image)
output_224 = classifier_univfd(feature)

# prediction from 56x56 random shuffled views
output_56 = []
for _ in range(n_views):
    image_shuffled = patch_shuffle(image, size=56)
    feature = visual_encoder(image_shuffled)
    output = classifier_56(feature)
    output_56.append(output)
output_56 = mean(output_56)

# prediction from 28x28 random shuffled views
output_28 = []
for _ in range(n_views):
    image_shuffled = patch_shuffle(image, size=28)
    feature = visual_encoder(image_shuffled)
    output = classifier_28(feature)
    output_28.append(output)
output_28 = mean(output_28)

# ensemble the logit scores
output = mean([output_224, output_56, output_28])
output = output.sigmoid()
```

Recently, generative models took a big leap forward with the advent of diffusion models, which called for fake image detection methods that are able to respond to diffusion models. However, some studies show that existing models trained to detect conventional GANs often fail in images from diffusion models. For example, periodic artifacts that were clearly visible in GAN were rarely found in diffusion models [8, 37]. In response, new detection methods optimized for diffusion models have emerged, for example, approaches that use diffusion models to reconstruct test images and evaluate them based on how well they are reconstructed [26, 50, 53].

Generalization of AI-generated image detection Recently, the community has shifted its focus towards general AI-generated image detectors that are not specific to GAN or diffusion. In particular, the development of commercially deployed generated models that do not reveal the model structure has increased the demand for such a universal detector.

Apart from existing attempts to learn a specialized feature extractor that simply classifies real/fake in a binary manner, Ojha *et al.* [32] used the features extracted from



(a) The changes of SFLD output distribution (b) The changes of UnivFD output distribution (c) The changes of NPR output distribution

Figure 16. The changes of model output distribution against JPEG compression

Pre-training / Patch sizes	ImageNet-ViT		Pre-training / Patch sizes	DINOv2-ViT [33]		OpenCLIP-ViT [16]		CLIP-ViT	
	Acc.	AP		Acc.	AP	Acc.	AP	Acc.	AP
224 (UnivFD [32])	62.45	69.30	224 (UnivFD [32])	81.89	91.75	86.49	96.90	85.89	96.29
224+16	63.88	72.23	224+28	82.88	93.42	86.50	97.59	91.94	98.03
224+32	63.34	71.36	224+56	82.44	93.04	86.87	97.70	92.05	98.39
224+32+16 (ours)	63.70	72.18	224+56+28 (ours)	82.26	93.26	86.19	97.49	93.30	98.43

Table 7. Detection accuracy and AP on a conventional benchmark of the proposed patch shuffling and ensembling (SFLD) strategy across various pre-trained encoders. For the ImageNet encoder, ViT-B/16 is used. For the other encoders, ViT-L/14 is used.

Method	GFW [1]	
	Acc.	mAP
NPR [46]	53.30	47.63
UnivFD [32]	70.07	85.55
SFLD(224+56)	77.80	86.70
SFLD	77.28	86.70

Table 8. Performance on the in-the-wild deepfake detection benchmark.

a strong vision-language pre-trained encoder that is not trained on a particular AI-generated image. Zhu *et al*. [56] combined anomaly detection methods to increase the discrepancy between real and fake image features.

Furthermore, several studies have concentrated on analyzing pixel-level traces on images inevitably left by the image generators. Tan *et al*. [46] exploited the artifacts that arise from up-sampling operations, based on the fact that most popular generator architectures include up-sampling operations. Chai *et al*. [4] tried to restrict the receptive field to emphasize local texture artifacts.

We design a simple yet powerful general AI-generated image detector that utilizes the feature space of the large pre-trained Vision Language Model. We apply image reformation to capture not only global semantic artifacts but local texture artifacts from the input images, ensuring detection performance and generalizability on unseen generators.