DeepfakeGenome: Toward Next-Generation Deepfake Attribution

Shunli Wang[1]*, **Xinyu Zhou**[1,2]*, **Yandan Zhao**[1], **Wenbin Liang**[2,3], **Taiping Yao**[1]
Ke-Yue Zhang[1], **Zhiyuan Yan**[4], **Shouhong Ding**[1]†, **Lizhuang Ma**[2]†
[1] Tencent Youtu Lab [2] Shanghai Jiao Tong University
[3] East China Normal University [4] Peking University
{shunliwang,yandanzhao,ericshding}@tencent.com, xinyuzhourain@sjtu.edu.cn

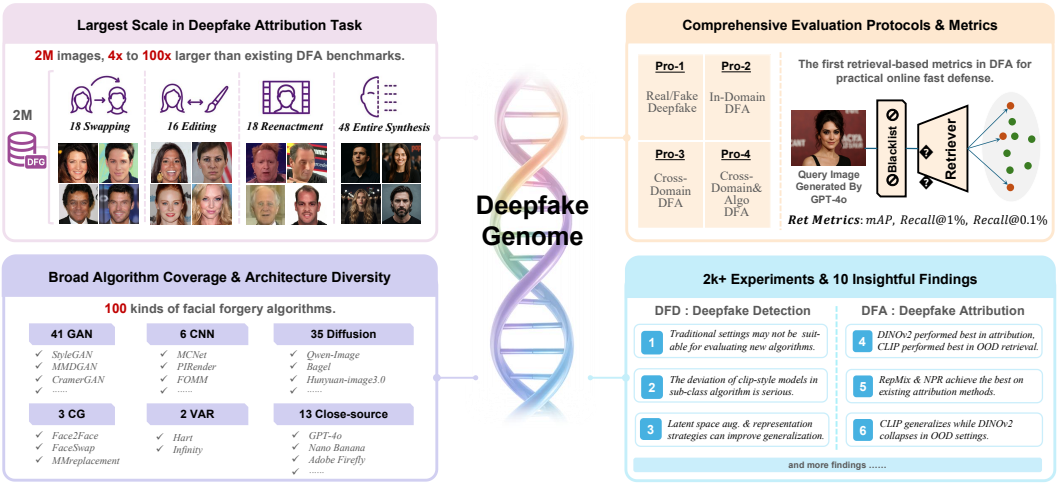


Figure 1: Overview of the proposed DeepfakeGenome benchmark. The DFG contains **100** distinct facial forgery algorithms and **2M** images. We design **4** practical evaluation protocols in Deepfake detection and attribution tasks. Over **2,000** experiments are conducted, yielding **10** insightful findings.

Abstract

In recent years, AIGC technologies are capable of generating hyper-realistic forged facial images, posing severe threats to facial security safety. To tackle these challenges, early Deepfake research primarily centered on real-fake classification. Recently, growing efforts have been devoted to the Deepfake Attribution (DFA) of generated content. Nevertheless, existing research on Deepfake detection and attribution faces saturated performance in binary classification, limited diversity in datasets and algorithms, and imperfect evaluation protocols, which severely impede practical application. To address these limitations, we propose a comprehensive deepfake detection and attribution benchmark named **DeepfakeGenome** (DFG). It contains **100** facial forgery algorithms and **2M** images in total, achieving **4×** to **100×** larger than prior DFA benchmarks. We further design **4** protocols for practical evaluation, including a novel retrieval-based attribution paradigm. Unlike previous open-set evaluation metrics, the proposed retrieval metrics are more aligned with the real-world active defense situation of blacklist registration mechanisms. Based on these elaborate designs, we investigate the performance ceiling of deepfake attribution task. Over **2k** experimental evaluations are conducted, and **10** insightful

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Track on Evaluations and Datasets.

findings are derived. We believe this work can provide new insights into the DFA research field. The Code and Benchmark of DFG are avaliable on `https://huggingface.co/spaces/DeepfakeGenome/DFG_Page`.

1 Introduction

With the rapid advancement of deep learning, AIGC techniques are capable of generating high-fidelity content within an extremely short period [1, 2]. These generative technologies have demonstrated remarkable value in artistic and media creation. Nevertheless, they also pose severe risks to facial security and identity authentication community. Traditional facial Deepfake detection research [3, 4] focused on identifying forged contents including face swapping, facial reenactment, and facial manipulation. With the rapid development of forgery techniques, researchers have moved beyond the binary detection task, and further focus on tracing the source of forged data to enable targeted defense. To address this challenge, Deepfake Attribution (DFA) techniques have emerged. Numerous prior studies [5–12] have dedicated efforts to exploring model-level source attribution for forgery images. By investigating existing research on Deepfake detection and DFA tasks, we identify three key challenges prevailing in this field.

- **Performance saturation of the binary classification on DFD benchmarks**. Current ACC and AUC metrics in Deepfake detection tasks have approached saturation, indicating that existing evaluation results are likely unobjective and incomplete.
- **Low diversity and limited scale of DFA benchmark**. Current DFA benchmarks typically include fewer than 20 algorithms, resulting in simplified, toy-level models. Consequently, there remains a substantial gap between existing methods and practical deployment.
- **Limitations of DFA assessment metrics**. Currently, DFA tasks usually adopt reject class classification to implement Out-of-Distribution (OOD) detection. However, in the real-world attack and defense, the update iteration of forgery algorithm is extremely fast. Traditional DFA metrics cannot meet this evaluation requirement.

An intuitive solution is to construct a more comprehensive and large-scale benchmark. We conduct thorough integration and deduplication on existing Deepfake and DFA datasets [13, 4, 14, 15, 5, 16–19], and incorporate a series of recently released open-source and closed-source image forgery models [1, 20–24, 2, 25]. Ultimately, we construct a benchmark covering **100** facial forgery algorithms. As illustrated in Tab. 1 and Fig. 3, the proposed DFG benchmark comprehensively outperforms previous DFA datasets in terms of both data scale and algorithm diversity. In terms of forgery categories, DFG covers four mainstream manipulation types: *Face Swapping*, *Face Reenactment*, *Face Editing*, and *Entire Face Synthesis*. In terms of algorithm architectures, DFG covers *GAN*, *Diffusion*, *CNN*, *CG*, and *VAR* frameworks, as well as *Closed-source* models. The total number of images reaches **2M**. As shown in Tab. 1, the scale of the DFG dataset is **4**$\times$ to **100**$\times$ larger than that of existing DFA datasets. Sufficient data scale provides a more reliable benchmark for the exploration of DFA algorithms.

Considering the practical deployment of DFA models in real-world scenarios, we design **4 Protocols** in DFG benchmark to comprehensively assess model performance across diverse tasks. It covers real-fake classification evaluation, deepfake attribution assessment, and retrieval-based attribution assessment under different dataset partitioning strategies. Notably, we firstly adopt retrieval-based metrics in clustering tasks [26] for DFA performance assessment. This paradigm is well-suited for real-world deployment. When confronted with emerging OOD attacks, efficient defense and interception can be realized by incorporating a small number of fake samples into the flexible blacklist.

Based on the sufficient data scale and effective evaluation protocols, we conduct extensive experiments and summarize **10** highly insightful findings. After expanding the data volume and algorithm richness, we were surprised to find that some previous consensus in Deepfake detection and DFA research was broken. For example, the SOTA myth of fine-tuning CLIP-ViTs [27] does not continue, and the improvement brought by contrastive learning is less significant than expected. We thoroughly investigate the above phenomena via qualitative and quantitative evaluations with detailed analysis.

2 Related Work

Deepfake Image Detection. Traditional Deepfake detection studies [28–34] primarily focus on improving the generalization performance of models against unseen forgery attacks. Deepfake methods can be broadly categorized into two paradigms: *Learning Fake Artifacts* and *Learning Real Distributions*. The former paradigm includes frequency-based methods [28, 29], blending-based methods [30, 31], and reconstruction-based methods [32]. These methods attempt to model and learn inherent forgery patterns to acquire cross-attack generalization capability. In contrast, the latter paradigm focuses on learning from real samples, which reformulates Deepfake detection as an anomaly detection task [33, 34]. However, with the continuous scaling of model sizes, existing benchmarks including FF++ [13], CDF-v2 [35], and DFDC [36] have gradually reached performance saturation. Therefore, the deepfake task needs more comprehensive evaluation benchmarks to achieve objective and holistic assessments of detection models.

Synthetic Image Detection. Unlike Deepfake detection [9, 37, 38, 3] that only focuses on facial manipulation scenarios, general AIGC detection extends the research scope of synthetic image detection to more diverse and generalized visual contents. These works can be divided into two categories based on whether adopting pre-trained foundational models [27, 39, 40]. Some traditional methods [9, 37] identify synthetic images by modeling forgery artifacts. For instance, NPR [9] achieves superior generalization performance by modeling artifacts introduced by up-sampling operations. DDA [37] leverages VAE [41] reconstruction to achieve strict semantic alignment between real and fake samples, thereby eliminating inherent biases during the learning process. In comparison, other methods [38, 3] exploit the pre-trained knowledge in CLIP-ViT [27] and perform diverse fine-tuning strategies to detect fake images.

Deepfake Attribution. Different from naive real/fake classification, DeepFake Attribution (DFA) task identifies the specific generation algorithm of fake images, providing more comprehensive analytical cues for forgery defense strategies. Early studies [42, 43, 11, 44] primarily focus on exploring inherent fingerprint characteristics of GAN models. Recent research [45, 46, 29] has further extended deepfake attribution to open-world scenarios. Sun *et al.* [45] introduced OWDFA benchmark, which aims to perform attribution on semi-supervised setting. Yang *et al.* [46] propose POSE framework to attribute images to known models and identify those from unknown ones. Li *et al.* [29] integrated high-pass filters and CLIP-ViT [27] to capture fine-grained forgery features to perform framework attribution based on reference sets. However, these methods are trained and evaluated on limited data scale and few algorithms. There remains a substantial gap between these models and practical applications.

3 DeepfakeGenome Benchmark

Brief introduction of DFG Benchmark. We systematically integrate and deduplicate existing deepfake and DFA datasets (FF++ [13], DF40 [4], ForgeryNet [14], DFFD [15], DNA-Det [5], OSMA [16], Wild-20 [17], DiFF [18], DiffusionFace [19]), and sorted out 92 deepfake algorithms. After that, we synthesized 8 kinds of recently accessible Text-to-Image algorithms: Qwen-Image [1], BAGEL [20], HunyuanImage-2.1/3.0 [21, 22], Infinity [23], Hart [24], Nanao Banana [2] and GPT-4o [25]. As shown in **Tab. 9**, the DeepfakeGenome benchmark includes **100** kinds of face forgery algorithms. The total number of frames of the DFG has reached **2M**. Tab. 1 and Fig. 3 compare the DFG benchmark with other DFA benchmarks. It surpasses the existing DFA datasets in the diversity of algorithms, the volume of frames and the novelty of forgery algorithms.

Fig. 2 demonstrates the distribution of 100 algorithms on two taxonomy. In terms of the forgery types, DFG contains 4 forgery types: **18** *Face Swapping* (FS), **18** *Face Reenactment* (FR), **48** *Entire Face Synthesis* (EFS), and **16** *Face Editing* (FE) algorithms. In algorithm architectures, DFG contains **6** classic *Convolutional Neural Network* (CNN), **3** *Computer Graphics* (CG), **41** *Generative Adversarial Network* (GAN), **35** *Diffusion*, **13** *Close-source*, and **2** *Visual Autoregressive* (VAR) algorithms.

Data Partitioning Protocols. To fully exploit the abundant data resources contained in DFG, we designed 4 data partitioning protocols to conduct experiments. Fig. 4 distinguishes 4 protocols with different colors. Tab. 10-13 in the supplementary material lists the specific algorithms and data domains in each partition in detail. The overall design criteria are: data in FF++ [13] domain is for training, and data in other domains is for performance evaluation. In **Protocol-1**, we follow the

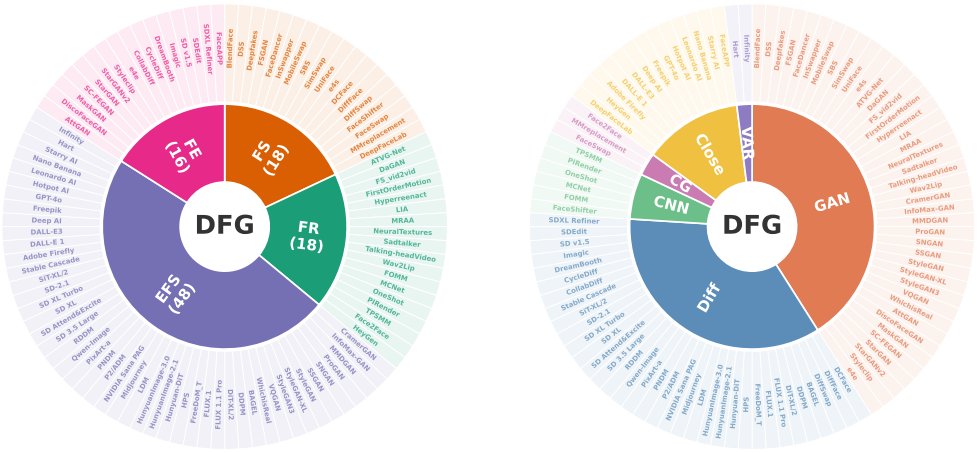


Figure 2: Classify 100 deepfake algorithms in terms of forgery types and algorithm architectures.

Table 1: A brief summary of the existing DFA datasets and the DeepfakeGenome dataset.

DFA Dataset	Venue	Data Domain	#Alg.	#Fms
GAN Finger. [42]	ICCV 2019	Nature&Face	4	110k
DFA [12]	ArXiv 2021	Face	3	18k
OWGAN [47]	ICCV 2021	Nature&Face	11	–
Attribution88 [11]	ECCV 2022	Face	7	105k
DNA-Det [48]	AAAI 2022	Nature&Face	4	509k
OSMA [49]	CVPR 2023	Nature&Face	17	281k
OWDFA [50]	ICCV 2023	Face	20	59k
DEFL [51]	MM 2024	Nature&Face	12	26k
OWDFA++ [52]	IJCV 2024	Face	25	338k
OSS-DFA [53]	ArXiv 2025	Face	20	–
WILD-20 [17]	arXiv 2025	Face	20	20k
BMRL [54]	ArXiv 2026	Face	28	135k
DFG (Ours)	2026	Face	100	2M

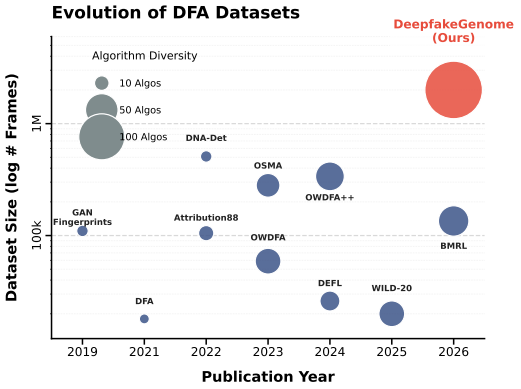


Figure 3: Volume comparison between the DeepfakeGenome and previous DFA datasets.

traditional deepfake settings [3]: training on vanilla FF++ [13] dataset (DeepFakes [55], Face2Face [56], FaceSwap [57] and NeuralTextures [58]), evaluating ACC and AUC on 59 new algorithms of DFG (note that all algorithms in DF40 [4] are removed because the performance has been explored in DF40 [4] paper). In **Protocol-2** & **Protocol-3**, 36 algorithms generated in FF++ domain are adopted as the training set, 36 and 34 algorithms in FF++ and CDF domain are adopted as the testing set, respectively. In this way, *Same Domain, Same Algorithm* and *Cross Domain, Same Algorithm* settings can be constructed. In **Protocol-4**, we adopt 36 algorithms in FF++ domain as training set, and conduct retrieval test for 65 algorithms out of FF++ domain and 36 training algorithms. The *Cross Domain, Cross Algorithm* setting is most suitable for real-world application scenarios.

Performance Metrics. In **Protocol-1**, we adopt ACC and AUC in the classic deepfake task [4, 3] as performance metrics; In **Protocol-2** and **Protocol-3**, since the forgery algorithms in the training and testing sets are identical, the variation lies solely in the data domain. First, we adopt *Accuracy* (*ACC*) and *mean Average Precision* (*mAP*) from multi-class classification tasks as our primary attribution performance metrics. Furthermore, to rigorously assess the quality of the feature representations extracted by the model, we introduce *Ret-mAP* and *Ret-Recall* from information retrieval tasks as additional evaluation metrics. Specifically, suppose there are M forgery algorithms in total, and K_j samples belong to the j-th class. In the retrieval setting, *Ret-mAP* measures the model's capability to rank relevant samples highly in the retrieved list:

$$AP_j = \frac{1}{K_j} \sum_{i=1}^{K_j} \frac{i}{Rank_j(i)}, \quad mAP = \frac{1}{M} \sum_{j=1}^{M} AP_j, \tag{1}$$

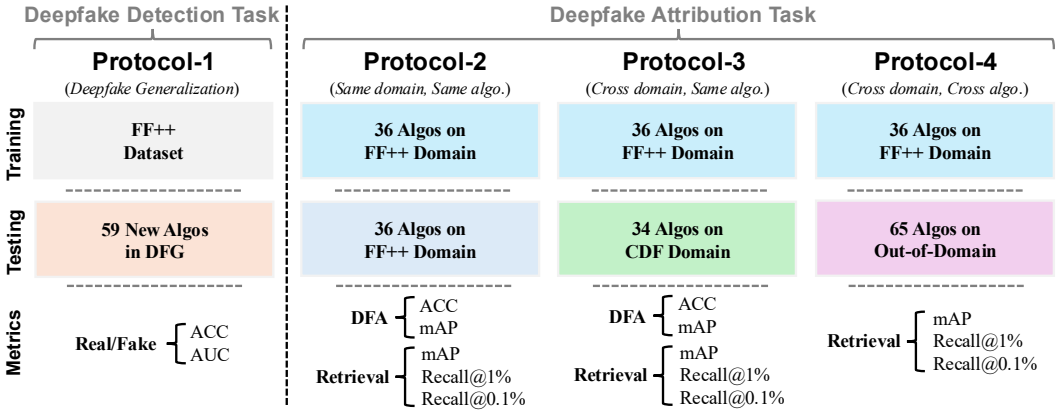


Figure 4: Four data partitioning protocols of the DFG Benchmark.

where $Rank_j(i)$ denotes the rank position of the i-th relevant sample of the j-th class after retrieval and sorting. To calculate the *Ret-Recall*, we first randomly sample 10 query instances from a specific class. We then extract their features and compute the average to obtain a representative query feature vector, denoted as f_{query}. Subsequently, we establish a strict threshold τ_α based on *False Rejection Rate* (*FRR*), denoted as α (*e.g.*, $\alpha = 1\%$ or 0.1%). Under this threshold, we measure the proportion of relevant negative samples (DeepFakes) that are successfully recalled. The formula for $Recall@\alpha$ is defined as:

$$Recall@\alpha = \frac{K_{recall}}{K_j}, \quad K_{recall} = \sum_{x \in \mathcal{S}_j} \mathbb{I}\big(sim(f_{query}, f_x) \geq \tau_\alpha\big), \tag{2}$$

where $\mathcal{S}_j$ represents the set of samples in the j-th class (with $|\mathcal{S}_j| = K_j$), f_x is the feature of sample x, $sim(\cdot,\cdot)$ denotes the similarity metric (*e.g.*, cosine similarity), and $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition is met and 0 otherwise. In **Protocol-4**, we adopt *Ret-mAP*, *Recall@1%* and *Recall@0.1%* to measure the dynamic interception performance against unseen forgery algorithms. Note that these metrics are particularly well-suited for **Blacklist Interception** scenarios in real-world applications, enabling a rapid and robust response to novel attacks.

4 Experiments & Findings

4.1 Experiment Details

During training, we use the Adam optimizer. The learning rate is set to 2×10^{-5} for fine-tuning CLIP-ViT [27] models and to 2×10^{-4} for all other models. The batch size is 32 for both training and testing, and all models are trained for 20 epochs. All CLIP-ViT [27] models are initialized with the default CLIP pre-trained weights released by OpenAI. For the ViT [59], CLIP [27], and DINO [39] model families, input images are resized to 224×224 pixels. For CNN-based models, images are resized to 256×256 pixels. When data augmentation is enabled, the pipeline includes `HorizontalFlipping` with probability 0.5; `RandomBrightnessContrast` with limits of 0.2 and probability 0.5; `HueSaturationValue` with probability 0.3; and `ImageCompression` with quality sampled from $[40, 100]$ and probability 0.1. `GaussianNoise`, `MotionBlur`, `CLAHE`, `ChannelShuffle`, and `Cutout` are each applied with probability 0.1, while `RandomGamma` and `GlassBlur` are each applied with probability 0.3. All experiments are conducted using an Intel Xeon Platinum 8255C CPU @ 2.50 GHz and four NVIDIA V100 GPUs with 32GB.

4.2 Deepfake Detection Results (Pro-1)

Finding-1. **Traditional Deepfake training settings may not be suitable for evaluating emerging algorithms**. Currently, a general consensus [3] in deepfake detection task is that fine-tuning the CLIP-ViT [27] model can achieve the best generalization performance. However, as shown in Tab. 2&3, the generalization results on 59 algorithms does not support this consensus. Specifically, in terms of ACC, the average performance (0.6563) of Effort [3] is indeed higher than CLIP-ViT-FFT [27]

Table 2: ACC & AUC on 59 algorithms in terms of algorithm types. The **highest** and second performance of each row are highlighted.

Types	Basic Backbone		Finetuned CLIP		Reconstruction-based		Frequency-domain			Latent Space		Representation			Aug Adv.
	Xception [60]	Eff-B4 [61]	CLIP-ViT-FFT [27]	Effort [3]	RECCE [62]	SBI [30]	F3Net [63]	SRM [64]	SPSL [65]	ProDet [66]	LSDA [67]	UCF [68]	IID [69]	CORE [70]	sladd [71]
FR (3 Algos)	0.6854	0.7144	0.7323	**0.8252**	0.6976	0.6937	0.7073	0.6644	0.6157	0.6801	0.7189	0.7157	0.6767	0.7101	0.7033
FS (6 Algos)	0.6416	0.6637	0.7752	**0.8184**	0.6194	0.6098	0.6088	0.6121	0.6465	0.6167	0.6128	0.6488	0.6087	0.6393	0.6057
FE (11 Algos)	0.5613	0.5939	0.6315	**0.7586**	0.5762	0.5897	0.5908	0.5846	0.5643	0.5603	0.5908	0.5908	0.6080	0.6032	0.6051
EFS (39 Algos)	0.5923	0.5486	0.5664	0.5895	0.6052	0.6123	0.5682	0.5821	0.5445	0.5021	**0.6841**	0.6841	0.6487	0.6398	0.5962
Avg.	0.5962	0.5772	0.6082	0.6563	0.6060	0.4709	0.6127	0.5898	0.5621	0.5337	**0.6612**	0.5881	0.6385	0.6365	0.6043
FR (3 Algos)	0.7680	0.8475	0.8622	**0.9343**	0.7722	0.7813	0.7959	0.7565	0.7666	0.8306	0.8027	0.8027	0.7511	0.7930	0.8015
FS (6 Algos)	0.7056	0.7249	0.8610	**0.9017**	0.6958	0.6997	0.7403	0.6728	0.7175	0.6614	0.6856	0.6856	0.6554	0.7155	0.6533
FE (11 Algos)	0.6493	0.7026	0.7370	**0.8420**	0.6647	0.6988	0.7037	0.6434	0.6330	0.5926	0.6545	0.6545	0.6632	0.7282	0.6436
EFS (39 Algos)	0.6316	0.7400	0.5983	0.5737	0.6642	0.6830	0.5978	0.6448	0.5535	0.5139	**0.7511**	0.7511	0.7028	0.7081	0.6092
Avg.	0.6494	0.6119	0.6643	0.6754	0.6730	0.4645	0.6926	0.6531	0.5959	0.5597	**0.7291**	0.6421	0.6931	0.7169	0.6299

Table 3: ACC & AUC on 59 algorithms in terms of algorithm architectures.

Architecture	Basic Backbone		Finetuned CLIP		Reconstruction-based		Frequency-domain			Latent Space		Representation			Aug Adv.
	Xception [60]	Eff-B4 [61]	CLIP-ViT-FFT [27]	Effort [3]	RECCE [62]	SBI [30]	F3Net [63]	SRM [64]	SPSL [65]	ProDet [66]	LSDA [67]	UCF [68]	IID [69]	CORE [70]	sladd [71]
CG (1 Algo)	0.5252	0.5430	**0.5837**	0.5824	0.4954	0.4823	0.5072	0.5006	0.5359	0.4819	0.4863	0.5064	0.4751	0.5206	0.4821
GAN (16 Algos)	0.5975	0.6630	0.7352	**0.8477**	0.6157	0.6135	0.6430	0.5806	0.5894	0.5608	0.6395	0.6607	0.6122	0.6366	0.5923
Diffusion (29 Algos)	0.5874	0.5396	0.5710	0.6141	0.5926	0.4309	0.5865	0.5932	0.5665	0.5556	0.6440	0.5568	**0.6479**	0.6339	0.6114
Close (11 Algos)	0.6161	0.5508	0.5393	0.5153	0.6277	0.3876	0.6433	0.5983	0.5169	0.4464	**0.7383**	0.5757	0.6573	0.6509	0.6146
VAR (2 Algos)	0.6413	0.5981	0.5216	0.5491	0.6579	0.6336	0.6336	0.6109	0.5436	0.5050	**0.7475**	0.5704	0.6900	0.6526	0.6002
Avg.	0.5962	0.5772	0.6082	0.6563	0.6060	0.4709	0.6127	0.5898	0.5621	0.5337	**0.6612**	0.5881	0.6385	0.6365	0.6043
CG (1 Algo)	0.5622	0.5732	0.6284	**0.6336**	0.5038	0.4873	0.5467	0.5026	0.5384	0.4746	0.4908	0.5493	0.4624	0.5562	0.4672
GAN (16 Algos)	0.6571	0.7299	0.8437	**0.9574**	0.6995	0.6559	0.7477	0.6453	0.6960	0.6004	0.6956	0.7556	0.6567	0.7473	0.6045
Diffusion (29 Algos)	0.6404	0.5708	0.6229	0.6194	0.6638	0.4231	0.6658	0.6501	0.5715	0.5767	**0.7168**	0.6196	0.7038	0.7098	0.6404
Close (11 Algos)	0.6666	0.5508	0.5418	0.4410	0.6676	0.3224	0.7016	0.6798	0.5281	0.4692	**0.8147**	0.5521	0.7249	0.7116	0.6518
VAR (2 Algos)	0.6661	0.6168	0.5199	0.5418	0.7096	0.3033	0.6646	0.6871	0.5486	0.5276	**0.8221**	0.6018	0.7684	0.6864	0.6411
Avg.	0.6494	0.6119	0.6643	0.6754	0.6730	0.4645	0.6926	0.6531	0.5959	0.5597	**0.7291**	0.6421	0.6931	0.7169	0.6299

(0.6082), but is slightly lower than LSDA [67] (0.6612). In terms of AUC, the overall performance of clip-style models [27, 3] (0.6643, 0.6754) are worse than LSDA [67], IID [69] and CORE [70] (0.7291, 0.6931, 0.7169). We argue that the performance degradation of the clip-style model stems not from poor generalization of CLIP-ViT, but from the over-simplified distribution of the FF++ [13] training dataset. Such a limited dataset causes overfitting when trained with the highly complex CLIP-ViT [27] backbone. This also indicates that the existing deepfake evaluation paradigm may have already fallen behind novel forgery algorithms, and more benchmark explorations are very necessary.

Finding-2. **The deviation of clip-style models in sub-class algorithm is serious**. Although the clip-style models [27, 3] achieve SOTA AUC performance on FR, FS and FE, it can only achieve 0.5983 and 0.5737 in EFS category respectively. Far behind LSDA [67], IID [69] and CORE [70] (0.7511, 0.7028, 0.7081). One possible explanation is *Excessive Semantic Deviation*: these models are fine-tuned from a high-quality real semantic space to produce offset, thus separating the real image from the easy forgery images in FF++ Datasets. However, when encountering more realistic EFS images generated by diffusion architectures, the biased decision preference that deviates from real semantics

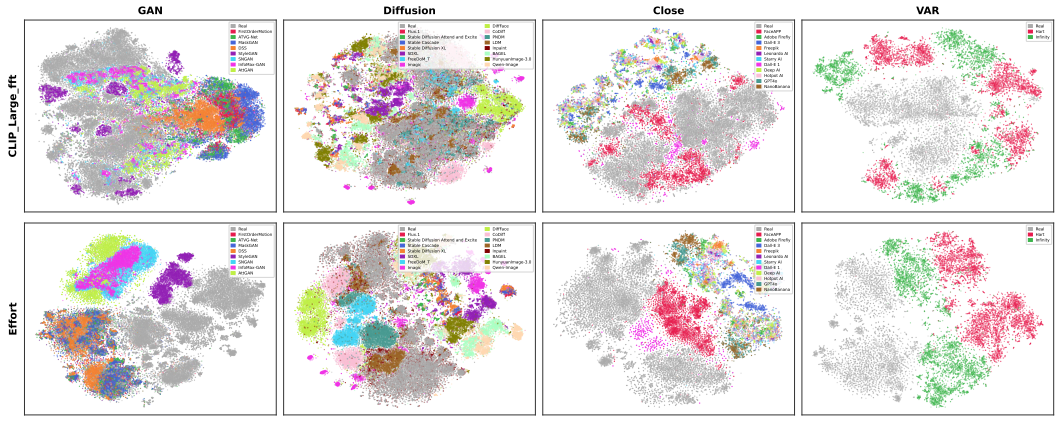


Figure 5: t-SNE visualization of CLIP-ViT-FFT [27] and Effort [3] features on GAN, Diffusion, Close-source and VAR architectures.

is very prone to errors. This is confirmed by the performance statistics from architecture level in Tab. 3. The clip-style models [27, 3] outperforms all other models in GAN and CG frameworks, but lags far behind LSDA [67] and IID [69] models in the Diffusion, Close-source and VAR frameworks (the performance gap even reaches 0.2). To qualitatively evaluate the performance of the model on different architectures, we visualize t-SNE of clip-style models in Fig. 5. Note that some algorithms are closed considering the clarity. Real samples are colored in gray dots for display. It can be found that the two models can be clearly distinguished on the GAN architectures. However, there is confusion in the Diffusion architectures.

Finding-3. **Latent space augmentation and representation learning strategies can significantly improve the generalization on new attacks**. LSDA [67] and IID [69] achieve the highest AUC performance in Diffusion, Close-source and VAR architectures respectively. And these performances are much higher than other models. This demonstrates that the latent augmentation and representation learning on FF++ can effectively improve the generalization performance on new forgery architectures. This finding has important guiding significance for future AIGC detection. While the specific techniques are also very important. For example, ProDet [66] adopts a progressive feature augmentation strategy, while LSDA directly combines features across forgery types.

4.3 In&Out-of-Domain Attribution Results (Pro-2&3)

Finding-4. **DINOv2 performed best in attribution, while CLIP performed best in OOD retrieval**. As shown in Tab. 5, fully fine-tuned DINOv2 [39] achieves the best attribution performance on Pro-2&3 (*e.g.*, *ACC* and *mAP*). This indicates that the low-level multi-crop pretraining are more effective for capturing subtle deepfake fingerprints generated by different manipulation algorithms. This phenomenon has also been reported in [74]. In the retrieval task, the fine-tuning CLIP-ViT [27] demonstrates superior performance in retrieval metrics (*e.g.*, *Ret-mAP* and *Ret-Recall*). This advantage stems from the strong global semantic representation ability of CLIP model, which provides a good start for the learning of model fingerprints. One obvious phenomenon is that the Effort [3] lags behind the fine-tuned CLIP and DINO models in terms of attribution performance. This indicates that the conservative SVD fine-tuning strategy retains most of the semantic features in CLIP, but it does not provide much help in attribution learning.

Table 4: mAP & Recall of backbones and attribution methods on Protocol-4.The default rank in LORA is 1.

Methods	Ret-mAP	Ret-Recall @1%	Ret-Recall @0.1%	Avg
Visual BackBones				
Xception [60]	0.2493	0.4586	0.2916	0.3331
F3Net [63]	0.2736	0.4289	0.2567	0.3197
Resnet-34 [72]	0.2745	0.4242	0.276	0.3249
Effort [3]	**0.5753**	**0.8607**	**0.7727**	**0.7362**
CLIP-ViT_FFT [27]	0.5507	0.8355	0.7358	0.7073
CLIP-ViT_LORA [73]	0.4687	0.7819	0.6805	0.6437
DINOv2-ViT_FFT [39]	0.3618	0.6459	0.4542	0.4873
DINOv3-ViT_FFT [40]	0.3788	0.5703	0.4054	0.4515
Attribution Methods				
DCT [28]	0.2742	0.4997	0.3948	0.3895
DNA-DET [48]	0.1535	0.2151	0.1257	0.1647
Universal [6]	0.1974	0.3343	0.1592	0.2303
LoRAX [7]	0.3069	0.5365	0.3635	0.4023
OCC-CLIP [8]	0.3131	0.5482	0.3463	0.4025
NPR [9]	**0.3312**	**0.6345**	**0.4707**	**0.4788**
POSE [49]	0.2476	0.3522	0.2284	0.2760
HIFI-IFD [10]	0.2819	0.5397	0.3992	0.4069
Repmix [11]	0.2950	0.4588	0.2788	0.3442
GAN Attribution [42]	0.2061	0.2756	0.1470	0.2095

Finding-5. **RepMix and NPR achieve the best performance among the existing attribution methods**. Tab. 5 shows that RepMix [11] and NPR [9] have achieved the best attribution performance in Protocol-2&3 respectively. The RepMix [11] adopted a latent feature mixing strategy to synthesize data from interpolation between existing samples. Unlike the LSDA [67], the mixing operation occurs between real and fake samples, not between fake and fake. The latent space feature augmentation can significantly improve the attribution ability of the model in the same domain, but its performance is very poor in the OOD setting, as shown in Tab. 4. The core idea of NPR [9] is to detect the forgery fingerprints generated by the up-sampling operation. In Protocol-4, NPR [9] still achieved the highest performance results, which demonstrates the effectiveness of adopting the architecture trace. However, their performance still has a large performance gap compared with the CLIP and DINO series model. This is mainly caused by the model size and the lack of pre-training knowledge.

Table 5: ACC, mAP & Recall on Protocol-2 & 3 in terms of backbones and attribution methods.r denotes the rank used in LoRA and † refers to CLIP-ViT without pretraining on large-scale data.

Methods	Protocol-2						Protocol-3					
	Acc	mAP	Ret-mAP	Ret-Recall @1%	Ret-Recall @0.1%	Avg	Acc	mAP	Ret-mAP	Ret-Recall @1%	Ret-Recall @0.1%	Avg
Visual BackBones												
Xception [60]	0.8899	0.9176	0.9149	0.9515	0.8762	0.910	0.7997	0.8501	0.8519	0.8316	0.6824	0.8031
F3Net [63]	0.8884	0.9130	0.9106	0.9525	0.8904	0.9110	0.8220	0.8606	0.8395	0.8433	0.7177	0.8166
ResNet-34 [72]	0.8755	0.9044	0.8899	0.9426	0.8923	0.9009	0.7446	0.8116	0.8158	0.8170	0.6140	0.7606
Effort [3]	0.8848	0.9154	0.7909	0.9327	0.8943	0.8836	0.7595	0.8302	0.7337	0.9002	0.8199	0.8087
CLIP-ViT-FFT [27]	0.9021	0.9334	0.9128	**0.9786**	0.9339	0.9322	0.8091	0.8711	0.8776	0.9294	0.8597	0.8694
CLIP-ViT-LORA [73](r=1)	0.9091	0.9459	0.8548	0.9555	0.9254	0.9181	0.8147	0.8825	0.8029	**0.9367**	**0.8786**	0.8631
CLIP-ViT-LORA(r=64)	0.8947	0.9296	0.7744	0.9182	0.8785	0.8791	0.7715	0.8520	0.7334	0.8837	0.7765	0.8034
DINOv2-L-FFT [39]	**0.9159**	**0.9511**	0.9052	0.9631	0.9329	0.9337	**0.8355**	**0.9008**	0.8654	0.9259	0.8552	**0.8766**
DINOv3-L-FFT [40]	0.9130	0.9467	**0.9245**	0.9768	**0.9507**	**0.9423**	0.8198	0.8826	**0.8818**	0.9075	0.8365	0.8656
Attribution Methods												
DCT [28]	0.5306	0.5561	0.3710	0.6288	0.5343	0.5242	0.5211	0.4964	0.4124	0.4564	0.3598	0.4492
DNA-DET [48]	0.6235	0.5921	0.7391	0.8176	0.7277	0.7000	0.4877	0.5337	0.6295	0.6277	0.4223	0.5402
Universal [6]	0.7996	0.8456	0.5367	0.5801	0.4945	0.6513	0.6441	0.6955	0.4847	0.4811	0.3451	0.5301
LoRAX [7]	0.8621	0.8916	0.8173	0.8792	0.7924	0.8485	0.7361	0.8004	0.7730	0.8234	0.6550	0.7576
OCC-CLIP [8]	0.8774	0.8952	0.8871	0.9366	0.8894	0.8971	0.7319	0.7820	0.7690	0.7661	0.5666	0.7231
NPR [9]	0.8727	0.9028	0.8884	0.9578	**0.9058**	0.9055	0.7597	0.8171	0.8086	0.7975	0.6120	0.7590
POSE [49]	0.8453	0.8782	0.7713	0.8320	0.7319	0.8117	0.6977	0.7469	0.6593	0.7475	0.5540	0.6811
HIFI-IFDL [10]	0.8260	0.8661	0.8588	0.9359	0.8567	0.8687	0.7422	0.7883	0.8025	0.7842	0.6020	0.7438
Repmix [11]	**0.8904**	**0.9173**	**0.8993**	**0.9603**	0.8914	**0.9117**	**0.7655**	**0.8436**	**0.8365**	**0.8317**	**0.6561**	**0.7867**
GAN Attribution [42]	0.7548	0.8017	0.6221	0.6772	0.5585	0.6829	0.5784	0.6826	0.5484	0.5098	0.3145	0.5267

4.4 Out-of-Domain Retrieval Results (Pro-4)

Finding-6. **CLIP generalizes while DINOv2 collapses in OOD settings.** Tab. 4 shows the performance of each model under Protocol-4 settings, *i.e.*, the retrieval performance under the *Cross Domain, Cross Algorithm* setting. Surprisingly, although the DINO models achieve high overall performance in Protocol-2&3, it lags far behind the CLIP models in Protocol-4. This shows that the DINO model has a serious over-fitting problem, and the generalization boundary is extremely sharp. However, the fine-tuned CLIP variants (full fine-tuned, LoRA, and SVD fine-tuned) consistently outperform DINO models by 20-25% across all metrics on Protocol-4. This phenomenon highlights a fundamental architectural difference: the dense local representations ability of DINO tends to overfit specific low-level fingerprints, whereas CLIP's large-scale contrastive pre-training knowledge provides generalized semantic priors that prevent collapse on unseen generative methods.

Finding-7. **Frequency and structural traces generalize better than specific spatial features.** Tab. 5 evaluates the attribution and retrieval performance of common visual backbones [60, 63, 72, 3, 27, 39, 40] and previous attribution methods [28, 5–9, 49, 10, 11, 42]. Surprisingly, the performance of these carefully designed attribution models is inferior to basic ViT models, especially in the retrieval task. We also find that models that relying on low-level fingerprints can achieve better generalization performance. For example, although DCT [28] performed worst on seen domains (Protocol-2&3), it outperforms more than half of the baselines on Protocol-4. This divergence in performance shows that capturing different frequency-domain features through discrete cosine transform provides more advantages in OOD retrieval scenarios than relying on specific spatial features. The excellent performance of NPR [9] in Protocol-4 confirms this finding. By amplifying the high-frequency fingerprints introduced by the up-sampling operation, NPR [9] achieves the best performance in Protocol-4.

4.5 How to Improve Poor OOD Performance?

To explore effective strategies for alleviating poor OOD generalization performance, we conduct a series of experiments based on CLIP variants.

Finding-8. **Contrastive learning biases the feature space and hurts OOD generalization performance.** Traditional attribution methods [5, 75] usually adopt contrastive learning (CL) to enhance the feature representation ability, while the counterintuitive phenomenon in Tab. 7 can not support this experience. Compared with the naive full fine-tune model, CL brings nearly 9% performance drop. To explain this phenomenon, we carried out a series of explorations and found that this performance is caused by *Overfitting* and *Semantic Space Collapse* during contrastive learning. Fig. 6 visualizes the pairwise cosine similarity heatmaps between feature representations in Protocol-2 (in-domain evaluation). The diagonal represents intra-class similarity and other regions represent inter-class similarity. It can be found that the model with CL has higher similarity between different classes, which

Table 6: mAP & Recall of CLIP variants on Protocol-2&3.

Methods	Process	Protocol-2						Protocol-3					
		Acc	mAP	Ret-mAP	Ret-Recall @1%	Ret-Recall @0.1%	Avg	Acc	mAP	Ret-mAP	Ret-Recall @1%	Ret-Recall @0.1%	Avg
CLIP-ViT	-	0.9021	0.9334	0.9128	**0.9786**	0.9339	0.9322	0.8091	**0.8711**	0.8776	0.9294	0.8597	**0.8694**
Vanilla-ViT	-	0.8667	0.9038	0.8749	0.9290	0.8254	0.8799	0.7367	0.8099	0.8163	0.8087	0.6454	0.7634
CLIP-ViT	*w.* CL	0.9023	0.9320	**0.9191**	0.9703	0.9220	0.9291	**0.8193**	0.8674	**0.8800**	0.8965	0.7690	0.8464
CLIP-ViT	*w.*LSDA[67]	0.8975	0.9377	0.9107	0.9748	0.9391	0.9320	0.8048	0.8587	0.8674	0.9213	0.8144	0.8533
CLIP-ViT	*w.* aug	**0.9128**	**0.9449**	0.9091	0.9756	**0.9535**	**0.9392**	0.7912	0.8660	0.8643	0.9154	0.8607	0.8595
Effort[3]	*w.* aug	0.8823	0.9096	0.7730	0.9204	0.8768	0.8724	0.7539	0.8214	0.7282	0.9038	0.8165	0.8048
CLIP-ViT	*w.* ps=14	0.8783	0.9170	0.8894	0.9622	0.9217	0.9137	0.7816	0.8381	0.8314	0.8831	0.7908	0.8250
CLIP-ViT	*w.* ps=7	0.7902	0.8385	0.8141	0.8961	0.8034	0.8284	0.6136	0.6805	0.6868	0.7509	0.6005	0.6665
Effort	*w.* ps=14	0.8105	0.8646	0.6914	0.8191	0.7655	0.7902	0.6782	0.7455	0.6774	0.8278	0.6865	0.7231

Table 7: mAP & Recall of CLIP variants on Protocol-4. CL denotes the contrastive learning and ps indicates the patch size in patch shuffle.

Methods	Process	Ret-mAP	Ret-Recall @1%	Ret-Recall @0.1%	Avg
CLIP-ViT	-	0.5507	0.8355	0.7358	0.7073
CLIP-ViT	*w/o* FFT	0.2758	0.6220	0.4892	0.4623
Vanilla-ViT	-	0.2553	0.4050	0.2199	0.2934
CLIP-ViT	*w.* CL	0.4233	0.7692	0.6458	0.6127
CLIP-ViT	*w.* LSDA[67]	0.4306	0.7162	0.6070	0.5846
CLIP-ViT	*w.* aug	0.3672	0.6676	0.5332	0.5226
Effort [3]	*w.* aug	**0.5849**	**0.8649**	**0.7915**	**0.7471**
CLIP-ViT	*w.* ps=14	0.5428	0.8196	0.7456	0.7026
CLIP-ViT	*w.* ps=7	0.4349	0.7390	0.6459	0.6066
Effort	*w.* ps=14	0.4234	0.6059	0.4582	0.4958

Table 8: mAP & Recall of combining features of two methods on Protocol-4. † represents the incorporation of Patch Shuffle. By default, patch size is set to 14 to align with the ViT's token size. ‡ indicates the use of augmented input data.

Method1	Method2	Ret-mAP	Ret-Recall @1%	Ret-Recall @0.1%	Avg
CLIP-ViT	-	0.5507	0.8355	0.7358	0.7073
CLIP-ViT†	CLIP-ViT	0.6138	0.8651	0.8019	0.7602
CLIP-ViT	Effort‡	0.6294	**0.9070**	0.8469	0.7944
CLIP-ViT†	Effort‡	**0.6536**	0.9056	**0.8598**	**0.8063**
DINOv2	Effort‡	0.5248	0.8101	0.6946	0.6765
DINOv2	CLIP-ViT	0.5034	0.8078	0.7186	0.6766
DINOv2	CLIP-ViT†	0.5263	0.8329	0.7639	0.7077

implies higher confusion between known classes. Consequently, this confusion slightly degrades the in-domain performance, resulting 0.9291 average performance on Protocol-2 (CLIP-ViT achieves 0.9322), as show in Tab. 6. In addition, we adopted t-SNE diagram to visually compare the features before and after using comparative learning on Protocol-4 in OOD setting. The degradation of the feature separability confirms the aforementioned analysis. Due to length constraints, more detailed and comprehensive analysis is listed in the Appendix C.

Findings-9. The OOD generalization ability of CLIP-ViTs relies heavily on pre-trained semantic priors. We compared three ViT model variants to assess the importance of pre-trained knowledge: a *Vanilla ViT* [59] without contrastive language-image pre-training, a standard *CLIP-ViT with full fine-tune*, and a *Frozen CLIP-ViT* (zero-shot setting). As shown in Tab. 6, the fine-tuned vanilla ViT achieves respectable 0.8799 and 0.7634 average performance on Protocol-2&3, respectively. This performance lags behind the fine-tuned CLIP-ViT with average performance 0.9322 and 0.8694. However, the fine-tuned vanilla ViT only achieve 0.2934 on Protocol-4, which is extremely lower than the fine-tuned CLIP-ViT with 0.7073, as shown in Tab. 7. Even without any fine-tuning, the zero-shot CLIP-ViT achieves 0.4623 on Protocol-4. These evidences directly confirm that the attribution robustness of CLIP-ViT models benefits from the large-scale pre-trained priors.

Finding-10. **Low-level and high-level features are complementary for OOD generalization.** We conducted a series of experiments to study the complementarity between *Low-level Fingerprints* and *High-level Semantics*, as shown in Tab. 8. To separate these features, we adopt two specialized models: Effort [3] with data augmentations that emphasizes high-level semantics, while a patch-shuffled CLIP-ViT [27] is forced to focus on only low-level fingerprints (due to space limitations, more details are provided in Appendix E). Individually, the standard CLIP-ViT achieves an average performance of 70.73% on Protocol-4, whereas the patch-shuffled CLIP-ViT shows an average performance drop of about 2% across Protocols. Firstly, combining the standard CLIP-ViT with models focusing on either high-level semantics (Effort) or low-level fingerprints (patch-shuffled CLIP-ViT) both bring performance improvements. Specifically, fusing the standard CLIP-ViT with the patch-shuffled CLIP-ViT brings a 6% improvement on Protocol-4, while fusing it with Effort gains a 9% improvement.Secondly, the combination of specialized high-level and low-level models achieves the best results. Combining the patch-shuffled CLIP-ViT with the Effort achieves a remarkable 10% boost compared to a single standard CLIP-ViT. Interestingly, combining Effort with

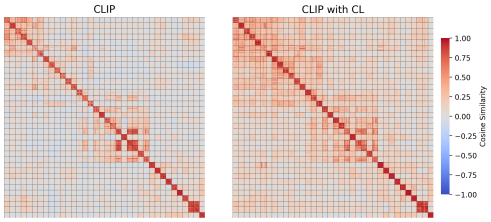


Figure 6: Cosine similarity heatmaps of features on Protocol-2. CL indicates contrastive learning. A redder color indicates a higher similarity.

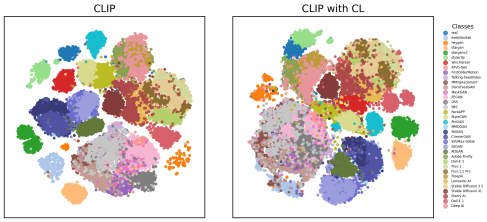


Figure 7: t-SNE visualization of features on 36 out-of-domain algorithms on Protocol 4. CL indicates contrastive learning.

the stronger standard CLIP-ViT performs worse than pairing it with the weaker patch-shuffled variant. We attribute this to feature redundancy and domain bias: standard CLIP-ViT contains mixed features with domain-specific bias, which interferes with Effort's robust semantics, whereas the patch-shuffled model provides orthogonal low-level cues. These results suggest that effective OOD generalization is not achieved by naively stacking strong models (see Appendix D for further analysis). Pairing pure high-level semantics with pure low-level fingerprints forms a synergistic representation, enabling more robust and accurate source attribution.

5 Limitation and Discussion

The semantics of the collected data are not strictly aligned. Because the entire dataset largely refers to the existing deepfake datasets, and only 8 algorithms (strictly semantic alignment on CelebA) have been synthesized. Therefore, the performance metrics may not reflect the true attribution performance. The retrieval *mAP* and *Recall* may be affected by semantic information. A straightforward solution is to re-synthesize all algorithms based on fixed real semantic data. However, this would incur an enormous workload. We leave this challenging task for future work.

No novel framework are proposed in this paper. This paper presents a systematic evaluation of deepfake detection and attribution tasks, and explores how classical strategies (*e.g.*, patch shuffle and data augmentation) can be adopted to explore the performance upper bound of existing models. Through model combination, we further conduct a complementary analysis of high-level semantic and low-level fingerprint features. Due to space and time limitations, we did not propose novel mechanisms and frameworks for deepfake attribution task. Based on our empirical findings, we highlight several promising research directions, including *Latent Space Feature Augmentation*, the *Disentanglement of High-level Semantic and Low-level Fingerprint*, and *Advanced Data Augmentation Strategies*. We believe this work can provide valuable insights and serve as a solid foundation for future research on deepfake attribution architecture exploring.

A temporal gap exists between DFG and the latest generative models. This work was initiated in 2025, during which we collected a series of SOTA open-source and closed-source algorithms. Representative examples include open-source methods such as *HunyuanImage-3.0* [22] and *Qwen-Image* [1], as well as closed-source solutions including *Nano Banana* [2] and *GPT-4o* [25]. Over the past year, the iteration speed of AIGC algorithms has far exceeded general expectations. Since adjusting the algorithm composition of the DFG benchmark is rather cumbersome, we retain the original 100 algorithms and do not supplement the most recently released generative models. Nevertheless, this limitation does not undermine the validity and generalization of the core findings in this paper.

6 Conclusions and Limitations

This work presents a comprehensive DFG benchmark for deepfake detection and attribution. The DFG covers 100 facial forgery methods and 2M images with a much larger scale than prior DFA benchmarks. Additionally, 4 practical evaluation protocols are established, including a novel retrieval-based attribution paradigm that better fits active defense scenarios such as blacklist registration. Extensive experiments with over 2,000+ evaluations are conducted to summarize 10 insightful findings. DFG provides a unified benchmark for systematically evaluating deepfake attribution methods and guiding the development of robust and generalizable attribution systems.

References

[1] C. Wu, J. Li, J. Zhou, J. Lin, K. Gao, K. Yan, S.-m. Yin, S. Bai, X. Xu, Y. Chen *et al.*, "Qwen-image technical report," *arXiv preprint arXiv:2508.02324*, 2025.

[2] Google, "Nano banana." [Online]. Available: https://github.com/haofanwang/inswapper

[3] Z. Yan, J. Wang, P. Jin, K.-Y. Zhang, C. Liu, S. Chen, T. Yao, S. Ding, B. Wu, and L. Yuan, "Orthogonal subspace decomposition for generalizable ai-generated image detection," *ICML*, 2024.

[4] Z. Yan, T. Yao, S. Chen, Y. Zhao, X. Fu, J. Zhu, D. Luo, C. Wang, S. Ding, Y. Wu *et al.*, "Df40: Toward next-generation deepfake detection," *Advances in Neural Information Processing Systems*, vol. 37, pp. 29 387–29 434, 2024.

[5] T. Yang, Z. Huang, J. Cao, L. Li, and X. Li, "Deepfake network architecture attribution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4662–4670.

[6] D. Cioni, C. Tzelepis, L. Seidenari, and I. Patras, "Are clip features all you need for universal synthetic image origin attribution?" in *European Conference on Computer Vision*. Springer, 2024, pp. 363–382.

[7] D. Sullivan-Pao, N. Tian, and P. Khorrami, "Lorax: Lora expandable networks for continual synthetic image attribution," *arXiv preprint arXiv:2504.08149*, 2025.

[8] F. Liu, H. Luo, Y. Li, P. Torr, and J. Gu, "Which model generated this image? a model-agnostic approach for origin attribution," in *European Conference on Computer Vision*. Springer, 2024, pp. 282–301.

[9] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, "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*, 2024, pp. 28 130–28 139.

[10] X. Guo, X. Liu, Z. Ren, S. Grosz, I. Masi, and X. Liu, "Hierarchical fine-grained image forgery detection and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 3155–3165.

[11] T. Bui, N. Yu, and J. Collomosse, "Repmix: Representation mixing for robust attribution of synthesized images," in *European Conference on Computer Vision*. Springer, 2022, pp. 146–163.

[12] B. Zhang, J. P. Zhou, I. Shumailov, and N. Papernot, "On attribution of deepfakes," *arXiv preprint arXiv:2008.09194*, 2020.

[13] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1–11.

[14] Y. He, B. Gan, S. Chen, Y. Zhou, G. Yin, L. Song, L. Sheng, J. Shao, and Z. Liu, "Forgerynet: A versatile benchmark for comprehensive forgery analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4360–4369.

[15] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, 2020, pp. 5781–5790.

[16] T. Yang, D. Wang, F. Tang, X. Zhao, J. Cao, and S. Tang, "Progressive open space expansion for open-set model attribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 856–15 865.

[17] P. Bongini, S. Mandelli, A. Montibeller, M. Casu, O. Pontorno, C. V. Ragaglia, L. Zanchetta, M. Aquilina, T. M. Wani, L. Guarnera *et al.*, "Wild: a new in-the-wild image linkage dataset for synthetic image attribution," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–8.

[18] H. Cheng, Y. Guo, T. Wang, L. Nie, and M. Kankanhalli, "Diffusion facial forgery detection," in *Proceedings of the 32nd ACM international conference on multimedia*, 2024, pp. 5939–5948.

[19] Z. Chen, K. Sun, Z. Zhou, X. Lin, X. Sun, L. Cao, and R. Ji, "Diffusionface: Towards a comprehensive dataset for diffusion-based face forgery analysis," *arXiv preprint arXiv:2403.18471*, 2024.

[20] C. Deng, D. Zhu, K. Li, C. Gou, F. Li, Z. Wang, S. Zhong, W. Yu, X. Nie, Z. Song *et al.*, "Emerging properties in unified multimodal pretraining," *arXiv preprint arXiv:2505.14683*, 2025.

[21] T. H. Team, "Hunyuanimage 2.1: An efficient diffusion model for high-resolution (2k) text-to-image generation," 2025.

[22] S. Cao, H. Chen, P. Chen, Y. Cheng, Y. Cui, X. Deng, Y. Dong, K. Gong, T. Gu, X. Gu *et al.*, "Hunyuanimage 3.0 technical report," *arXiv preprint arXiv:2509.23951*, 2025.

[23] J. Han, J. Liu, Y. Jiang, B. Yan, Y. Zhang, Z. Yuan, B. Peng, and X. Liu, "Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 15 733–15 744.

[24] H. Tang, Y. Wu, S. Yang, E. Xie, J. Chen, J. Chen, Z. Zhang, H. Cai, Y. Lu, and S. Han, "Hart: Efficient visual generation with hybrid autoregressive transformer," *arXiv preprint arXiv:2410.10812*, 2024.

[25] OpenAI, "Gpt-4o." [Online]. Available: https://openai.com/index/hello-gpt-4o/

[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[27] 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*. PmLR, 2021, pp. 8748–8763.

[28] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*. PMLR, 2020, pp. 3247–3258.

[29] J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang, "Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6458–6467.

[30] K. Shiohara and T. Yamasaki, "Detecting deepfakes with self-blended images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 720–18 729.

[31] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia, "Learning self-consistency for deepfake detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 023–15 033.

[32] X. Zhu, H. Wang, H. Fei, Z. Lei, and S. Z. Li, "Face forgery detection by 3d decomposition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2929–2939.

[33] H. Khalid and S. S. Woo, "Oc-fakedect: Classifying deepfakes using one-class variational autoencoder," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 656–657.

[34] N. Larue, N.-S. Vu, V. Struc, P. Peer, and V. Christophides, "Seeable: Soft discrepancies and bounded contrastive learning for exposing deepfakes," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 21 011–21 021.

[35] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3207–3216.

[36] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The deepfake detection challenge (dfdc) dataset," *arXiv preprint arXiv:2006.07397*, 2020.

[37] R. Chen, J. Xi, Z. Yan, K.-Y. Zhang, S. Wu, J. Xie, X. Chen, L. Xu, I. Guan, T. Yao *et al.*, "Dual data alignment makes ai-generated image detector easier generalizable," *arXiv preprint arXiv:2505.14359*, 2025.

[38] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 24 480–24 489.

[39] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[40] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa *et al.*, "Dinov3," *arXiv preprint arXiv:2508.10104*, 2025.

[41] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[42] N. Yu, L. S. Davis, and M. Fritz, "Attributing fake images to gans: Learning and analyzing gan fingerprints," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7556–7566.

[43] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Do gans leave artificial fingerprints?" in *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2019, pp. 506–511.

[44] K. Xu, L. Zhang, and J. Shi, "Detecting origin attribution for text-to-image diffusion models," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 8775–8785.

[45] Z. Sun, S. Chen, T. Yao, R. Yi, S. Ding, and L. Ma, "Rethinking open-world deepfake attribution with multi-perspective sensory learning," *International Journal of Computer Vision*, vol. 133, no. 2, pp. 628–651, 2025.

[46] T. Yang, D. Wang, F. Tang, X. Zhao, J. Cao, and S. Tang, "Progressive open space expansion for open-set model attribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 856–15 865.

[47] S. Girish, S. Suri, S. S. Rambhatla, and A. Shrivastava, "Towards discovery and attribution of open-world gan generated images," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 094–14 103.

[48] T. Yang, Z. Huang, J. Cao, L. Li, and X. Li, "Deepfake network architecture attribution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4662–4670.

[49] T. Yang, D. Wang, F. Tang, X. Zhao, J. Cao, and S. Tang, "Progressive open space expansion for open-set model attribution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 856–15 865.

[50] Z. Sun, S. Chen, T. Yao, B. Yin, R. Yi, S. Ding, and L. Ma, "Contrastive pseudo learning for open-world deepfake attribution," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 20 882–20 892.

[51] J. Li, H. Wang, S. Li, Z. Qian, X. Zhang, and A. V. Vasilakos, "Are handcrafted filters helpful for attributing ai-generated images?" in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 10 698–10 706.

[52] Z. Sun, S. Chen, T. Yao, R. Yi, S. Ding, and L. Ma, "Rethinking open-world deepfake attribution with multi-perspective sensory learning," *International Journal of Computer Vision*, vol. 133, no. 2, pp. 628–651, 2025.

[53] M.-H. Liu, X.-Q. Liu, X. Luo, and X.-S. Xu, "Data: Multi-disentanglement based contrastive learning for open-world semi-supervised deepfake attribution," *IEEE Transactions on Multimedia*, 2025.

[54] Y. Zhang, J. Zhang, C. Ma, W. Guan, T. Gan, and Z. Gao, "Pvlm: Parsing-aware vision language model with dynamic contrastive learning for zero-shot deepfake attribution," *arXiv preprint arXiv:2504.14129*, 2025.

[55] "Deepfakes." [Online]. Available: https://github.com/deepfakes/faceswap

[56] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.

[57] M. Kowalski, "Faceswap." [Online]. Available: https://github.com/MarekKowalski/FaceSwap

[58] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *Acm Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[59] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[60] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[61] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[62] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, and X. Yang, "End-to-end reconstruction-classification learning for face forgery detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4113–4122.

[63] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *European conference on computer vision*. Springer, 2020, pp. 86–103.

[64] Y. Luo, Y. Zhang, J. Yan, and W. Liu, "Generalizing face forgery detection with high-frequency features," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 317–16 326.

[65] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu, "Spatial-phase shallow learning: rethinking face forgery detection in frequency domain," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 772–781.

[66] J. Cheng, Z. Yan, Y. Zhang, Y. Luo, Z. Wang, and C. Li, "Can we leave deepfake data behind in training deepfake detector?" *Advances in Neural Information Processing Systems*, vol. 37, pp. 21 979–21 998, 2024.

[67] Z. Yan, Y. Luo, S. Lyu, Q. Liu, and B. Wu, "Transcending forgery specificity with latent space augmentation for generalizable deepfake detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8984–8994.

[68] Z. Yan, Y. Zhang, Y. Fan, and B. Wu, "Ucf: Uncovering common features for generalizable deepfake detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 22 412–22 423.

[69] B. Huang, Z. Wang, J. Yang, J. Ai, Q. Zou, Q. Wang, and D. Ye, "Implicit identity driven deepfake face swapping detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 4490–4499.

[70] Y. Ni, D. Meng, C. Yu, C. Quan, D. Ren, and Y. Zhao, "Core: Consistent representation learning for face forgery detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12–21.

[71] L. Chen, Y. Zhang, Y. Song, L. Liu, and J. Wang, "Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 710–18 719.

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[73] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *Iclr*, vol. 1, no. 2, p. 3, 2022.

[74] T. Zhang, P. Yu, Z. Xia, L. Dai, X. Zhou, and H. Gao, "Fine-grained dino tuning with dual supervision for face forgery detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 15, 2026, pp. 12 780–12 788.

[75] Z. Li, Z. Teng, B. Zhang, and J. Fan, "Boosting non-causal semantic elimination: An unconventional harnessing of lvm for open-world deepfake interpretation," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 3528–3537.

[76] S. Bai, Y. Cai, and et.al., "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[77] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International conference on machine learning*. PMLR, 2020, pp. 9929–9939.

[78] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[79] O. Roy and M. Vetterli, "The effective rank: A measure of effective dimensionality," in *2007 15th European signal processing conference*. IEEE, 2007, pp. 606–610.

[80] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," *arXiv preprint arXiv:1912.13457*, 2019.

[81] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7184–7193.

[82] R. Chen, X. Chen, B. Ni, and Y. Ge, "Simswap: An efficient framework for high fidelity face swapping," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 2003–2011.

[83] K. Shiohara, X. Yang, and T. Taketomi, "Blendface: Re-designing identity encoders for face-swapping," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 7634–7644.

[84] Z. Liu, M. Li, Y. Zhang, C. Wang, Q. Zhang, J. Wang, and Y. Nie, "Fine-grained face swapping via regional gan inversion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 8578–8587.

[85] F. Rosberg, E. E. Aksoy, F. Alonso-Fernandez, and C. Englund, "Facedancer: Pose-and occlusion-aware high fidelity face swapping," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 3454–3463.

[86] Z. Xu, Z. Hong, C. Ding, Z. Zhu, J. Han, J. Liu, and E. Ding, "Mobilefaceswap: A lightweight framework for video face swapping," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 2973–2981.

[87] Roop, "Inswapper." [Online]. Available: https://github.com/haofanwang/inswapper

[88] C. Xu, J. Zhang, Y. Han, G. Tian, X. Zeng, Y. Tai, Y. Wang, C. Wang, and Y. Liu, "Designing one unified framework for high-fidelity face reenactment and swapping," in *European conference on computer vision*. Springer, 2022, pp. 54–71.

[89] M. Kowalski, "Deepfacelab." [Online]. Available: https://github.com/iperov/DeepFaceLab

[90] W. Zhang, X. Cun, X. Wang, Y. Zhang, X. Shen, Y. Guo, Y. Shan, and F. Wang, "Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 8652–8661.

[91] A. Siarohin, O. J. Woodford, J. Ren, M. Chai, and S. Tulyakov, "Motion representations for articulated animation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 653–13 662.

[92] Y. Wang, D. Yang, F. Bremond, and A. Dantcheva, "Latent image animator: Learning to animate images via latent space navigation," *arXiv preprint arXiv:2203.09043*, 2022.

[93] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," *arXiv preprint arXiv:1910.12713*, 2019.

[94] K. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar, "A lip sync expert is all you need for speech to lip generation in the wild," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 484–492.

[95] F.-T. Hong, L. Zhang, L. Shen, and D. Xu, "Depth-aware generative adversarial network for talking head video generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3397–3406.

[96] S. Bounareli, C. Tzelepis, V. Argyriou, I. Patras, and G. Tzimiropoulos, "Hyperreenact: one-shot reenactment via jointly learning to refine and retarget faces," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 7149–7159.

[97] F.-T. Hong and D. Xu, "Implicit identity representation conditioned memory compensation network for talking head video generation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 23 062–23 072.

[98] T.-C. Wang, A. Mallya, and M.-Y. Liu, "One-shot free-view neural talking-head synthesis for video conferencing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 039–10 049.

[99] Y. Ren, G. Li, Y. Chen, T. H. Li, and S. Liu, "Pirenderer: Controllable portrait image generation via semantic neural rendering," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 13 759–13 768.

[100] J. Zhao and H. Zhang, "Thin-plate spline motion model for image animation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3657–3666.

[101] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[102] "heygen." [Online]. Available: https://www.heygen.com/

[103] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[104] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," *Advances in neural information processing systems*, vol. 34, pp. 852–863, 2021.

[105] A. Sauer, K. Schwarz, and A. Geiger, "Stylegan-xl: Scaling stylegan to large diverse datasets," in *ACM SIGGRAPH 2022 conference proceedings*, 2022, pp. 1–10.

[106] "whichfaceisreal." [Online]. Available: https://www.whichfaceisreal.com/

[107] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[108] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[109] J. Liu, Q. Wang, H. Fan, Y. Wang, Y. Tang, and L. Qu, "Residual denoising diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 2773–2783.

[110] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu *et al.*, "Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis," *arXiv preprint arXiv:2310.00426*, 2023.

[111] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205.

[112] S. Atito, M. Awais, and J. Kittler, "Sit: Self-supervised vision transformer," *arXiv preprint arXiv:2104.03602*, 2021.

[113] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for stylegan image manipulation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[114] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[115] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8188–8197.

[116] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2085–2094.

[117] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7832–7841.

[118] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," *Advances in neural information processing systems*, vol. 32, 2019.

[119] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala, "Text-based editing of talking-head video," *acm Transactions on Graphics (tog)*, vol. 38, no. 4, pp. 1–14, 2019.

[120] Y. Deng, J. Yang, D. Chen, F. Wen, and X. Tong, "Disentangled and controllable face image generation via 3d imitative-contrastive learning," in *IEEE Computer Vision and Pattern Recognition*, 2020.

[121] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[122] Y. Jo and J. Park, "Sc-fegan: Face editing generative adversarial network with user's sketch and color," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[123] "faceapp." [Online]. Available: https://faceapp.com/app

[124] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[125] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[126] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," *arXiv preprint arXiv:1801.01401*, 2018.

[127] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[128] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos, "The cramer distance as a solution to biased wasserstein gradients," *arXiv preprint arXiv:1705.10743*, 2017.

[129] K. S. Lee, N.-T. Tran, and N.-M. Cheung, "Infomax-gan: Improved adversarial image generation via information maximization and contrastive learning," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3942–3952.

[130] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-supervised gans via auxiliary rotation loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 154–12 163.

[131] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "Attgan: Facial attribute editing by only changing what you want," *IEEE transactions on image processing*, vol. 28, no. 11, pp. 5464–5478, 2019.

[132] Adobe. (2023) Adobe firefly. Accessed: 2026-3-9. [Online]. Available: https://firefly.adobe.com/

[133] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo *et al.*, "Improving image generation with better captions," *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, vol. 2, no. 3, p. 8, 2023.

[134] Freepik. (2024) "freepik ai image generator". Accessed: 2026-3-9. [Online]. Available: https://docs.freepik.com/api-reference/mystic/post-mystic

[135] L. AI. (2024) "Leonardo AI: AI-powered creative image generation platform". Accessed: 2026-3-9. [Online]. Available: https://leonardo.ai

[136] "Starry AI," https://starryai.com/, accessed: 2026-3-9.

[137] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.

[138] D. AI, "DeepAI Text-to-Image Generator," https://deepai.org/, accessed: 2026-3-9.

[139] H. AI, "HotPot AI: AI-Powered Image and Text Generation Tools," https://hotpot.ai/, accessed: 2026-3-9.

[140] B. F. Labs, "Flux," https://github.com/black-forest-labs/flux, 2024, accessed: 2026-3-9.

[141] B. F. Labs. (2024) "FLUX 1.1 [pro]: Advanced Text-to-Image Generation Model". Accessed: 2026-3-9. [Online]. Available: https://flux1-1.ai/

[142] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[143] E. Xie, J. Chen, J. Chen, H. Cai, H. Tang, Y. Lin, Z. Zhang, M. Li, L. Zhu, Y. Lu *et al.*, "Sana: Efficient high-resolution image synthesis with linear diffusion transformer, 2024," *URL https://arxiv.org/abs/2410.10629*, vol. 4, 2024.

[144] P. Pernias, D. Rampas, M. L. Richter, C. J. Pal, and M. Aubreville, "Würstchen: An efficient architecture for large-scale text-to-image diffusion models," *arXiv preprint arXiv:2306.00637*, 2023.

[145] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or, "Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models," *ACM transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–10, 2023.

[146] Z. Li, J. Zhang, Q. Lin, J. Xiong, Y. Long, X. Deng, Y. Zhang, X. Liu, M. Huang, Z. Xiao *et al.*, "Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding," *arXiv preprint arXiv:2405.08748*, 2024.

[147] MidJourney. (2024) "MidJourney: An AI-powered image generation tool". Accessed: 2026-3-9. [Online]. Available: https://www.midjourney.com

[148] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023.

[149] J. Yu, Y. Wang, C. Zhao, B. Ghanem, and J. Zhang, "Freedom: Training-free energy-guided conditional diffusion model," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 23 174–23 184.

[150] X. Wu, K. Sun, F. Zhu, R. Zhao, and H. Li, "Human preference score: Better aligning text-to-image models with human preference," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2096–2105.

[151] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 500–22 510.

[152] K. Kim, Y. Kim, S. Cho, J. Seo, J. Nam, K. Lee, S. Kim, and K. Lee, "Diffface: Diffusion-based face swapping with facial guidance," *Pattern Recognition*, vol. 163, p. 111451, 2025.

[153] M. Kim, F. Liu, A. Jain, and X. Liu, "Dcface: Synthetic face generation with dual condition diffusion model," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2023, pp. 12 715–12 725.

[154] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani, "Imagic: Text-based real image editing with diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 6007–6017.

[155] Z. Huang, K. C. Chan, Y. Jiang, and Z. Liu, "Collaborative diffusion for multi-modal face generation and editing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 6080–6090.

[156] C. H. Wu and F. De la Torre, "A latent space of stochastic diffusion models for zero-shot image editing and guidance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7378–7387.

[157] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, "Pseudo numerical methods for diffusion models on manifolds," *arXiv preprint arXiv:2202.09778*, 2022.

[158] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon, "Perception prioritized training of diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 472–11 481.

[159] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," *arXiv preprint arXiv:2108.01073*, 2021.

[160] W. Zhao, Y. Rao, W. Shi, Z. Liu, J. Zhou, and J. Lu, "Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8568–8577.

A Detailed Information of the DFG Benchmark

Detailed list of 100 algorithms. Tab. 9 provides a detailed enumeration of the 100 algorithms in the DFG benchmark, along with the corresponding domains. Since the first 92 algorithms are collected from existing Deepfake and DFA datasets [13, 4, 14, 15, 5, 16–19], the domain is relatively rich. In the *Domain* column, *Multi-1* represents CREMA-D, VoxCeleb2, RAVDESS, and AVSpeech. *Multi-2* represents FFHQ, CelebA and FF++. *Multi-3* represents VoxCeleb2 and CelebA.

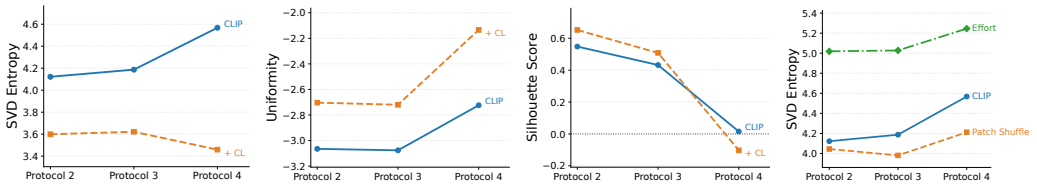
Detailed data partition in 4 Protocols. Tab. 10-13 provide a detailed display of the algorithm settings under 4 evaluation protocols. The DFG benchmark not only enables extended evaluation in traditional Deepfake tasks but also provides rich settings on DFA evaluations. The Protocol-2 and Protocol-3 focus on *Same Domain, Same Algorithm* and *Cross Domain, Same Algorithm*, respectively. Protocol-4 focuses on novel forgery attacks in *Cross Domain, Cross Algorithm* setting, which is most similar to threats in the real world.

Visualization of 8 Synthesized Algorithms. We adopt real facial images from the CelebA and leverage the Qwen3-VL [76] to generate corresponding textual descriptions. In Fig. 9, we present 10 representative cases for intuitive illustration, with the detailed prompts listed in Tab. 16. It can be observed that NanoBanana [2] and GPT-4o [25] achieve an extremely high level of visual realism. Meanwhile, open-source algorithms including Qwen-Image [1], Infinity [23], and HunyuanImage-3.0 [22] are also capable of producing highly realistic human portraits. By contrast, the visual quality of Hart [24], HunyuanImage-2.1 [21] and Bagel [20] is relatively inferior.

B Detailed Results in Protocol-1

Tab. 2 and Tab. 3 in the main text categorize and summarize the results of the deepfake methods in Protocol-1. We list all detailed results in Tab. 14 and Tab. 15. It should be noted that in Protocol-1, only 59 new algorithms were evaluated, excluding FF++ [13] and DF40 [4]. This design is mainly based on two considerations: the algorithms adopted in DF40 are no longer innovative, and this dataset has been widely adopted for evaluating the generalization ability of deepfake models. To avoid duplicate evaluation efforts, we exclude all DF-40 algorithms in Protocol-1.

C More Details of Finding-8



(a) SVD_Entropy (b) Uniformity [77] (c) Silhouette_Score [78] (d) SVD_Entropy

Figure 8: **Comparison of feature space properties across evaluation protocols.** (a-c) SVD Entropy, Uniformity, and Silhouette Score for the CLIP-ViT and CLIP-ViT equipped with contrastive learning. (d) SVD Entropy comparison among Effort, CLIP, and CLIP-ViT equipped with patch shuffle.

C.1 Analysis

To further verify the feature-space degradation discussed in Finding-8, we quantitatively evaluate the learned representation space using *Singular Value Decomposition Entropy* (SVD Entropy), *Uniformity* [77], and *Silhouette Score* [78], as shown in Fig. 8.

SVD entropy. Similar to effective rank [79], SVD entropy measures the information richness and effective dimensionality of the feature space. A higher SVD entropy indicates that the representation occupies more informative dimensions, while a lower value suggests dimensional collapse. As shown in Fig. 8a, the addition of contrastive learning results in a sharp drop in SVD entropy. For example, on Protocol 2, the entropy decreases from 4.122 to 3.599. This confirms that contrastive learning

leads to a severe loss of informative dimensions and compresses the representation space into a lower-dimensional subspace.

Uniformity. We further use the uniformity metric [77] to evaluate how evenly features are distributed on the hypersphere. A lower uniformity value indicates a more uniformly distributed feature space. However, as shown in Fig. 8b, contrastive learning worsens the uniformity score, increasing it from -3.06 to -2.70. This indicates that the feature space becomes less continuous and less evenly distributed. Instead of spreading samples across the hypersphere, contrastive learning causes features to clump together in restricted and crowded regions.

Silhouette score. We also compute the silhouette score [78], which measures clustering quality by comparing intra-class compactness and inter-class separation. A negative silhouette score indicates severe overlap between different classes. As shown in Fig. 8c, although contrastive learning artificially improves the clustering score on seen domains by forcing training classes to be compact, its score drops drastically on unseen domains. In particular, on Protocol 4, the silhouette score of the contrastive model decreases to -0.103, whereas the baseline remains positive at 0.015. This result demonstrates that the compactness learned by contrastive learning does not generalize to unseen generative methods. Instead, it amplifies inter-class confusion in out-of-domain scenarios.

Overall, these quantitative results support the conclusion in Finding-8: contrastive learning overfits the model to the seen training taxonomy, reduces feature diversity, and collapses the continuous semantic space. This explains why contrastive learning slightly hurts in-domain discrimination but severely damages out-of-domain generalization.

The negative impact of this distorted space becomes more obvious when evaluating the model on unseen domains. While the baseline maintains reasonable robustness with scores of 0.8694 on Protocol 3 and 0.7073 on Protocol 4, the contrastive learning model suffers a severe performance drop, yielding only 0.8464 and 0.6127, respectively. This decline occurs because the continuous feature space has been disrupted, causing the features of unseen new generative models to scatter unpredictably in the space. This phenomenon is clearly visualized using t-SNE in Fig. 7. While the baseline (Fig. 7a) keeps unseen methods relatively separated, the contrastive model (Fig. 7b) mixes them into a chaotic and overlapping state. Additional quantitative analyses using SVD entropy, Uniformity [77], and Silhouette score [78] are provided in Appendix C, which further confirm the feature-space collapse caused by contrastive learning.

C.2 Implementation of Uniformity and SVD Entropy

In this section, we provide the implementation details of the two representation-space metrics used in our analysis: uniformity and Singular Value Decomposition (SVD) entropy. Given a feature matrix $\mathbf{Z} \in \mathbb{R}^{N \times d}$, where N is the number of samples and d is the feature dimension, all features are first converted to floating-point tensors. For uniformity, features are additionally ℓ_2-normalized before computing pairwise distances.

C.2.1 Uniformity

Algorithm 1 Uniformity Calculation

Require: Feature matrix $\mathbf{Z} \in \mathbb{R}^{N \times d}$, temperature $t = 2.0$, maximum sample size $M = 20{,}000$
Ensure: Uniformity score $\mathcal{L}_{\text{uniform}}$
1: **if** $N > M$ **then**
2: Randomly sample M rows from $\mathbf{Z}$
3: **end if**
4: Normalize each feature: $\hat{\mathbf{z}}_i \leftarrow \mathbf{z}_i / \|\mathbf{z}_i\|_2$
5: Compute pairwise squared distances: $d_{ij} \leftarrow \|\hat{\mathbf{z}}_i - \hat{\mathbf{z}}_j\|_2^2$ for all $i < j$
6: Compute $\mathcal{L}_{\text{uniform}} \leftarrow \log\left(\frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \exp(-t d_{ij})\right)$
7: **return** $\mathcal{L}_{\text{uniform}}$

Uniformity measures how evenly features are distributed on the unit hypersphere. Following [77], we compute the uniformity score as:

$$\mathcal{L}_{\text{uniform}}(\mathbf{Z}) = \log \mathbb{E}_{i \neq j} \left[\exp \left(-t \left\| \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2} - \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_2} \right\|_2^2 \right) \right], \tag{3}$$

where t is a temperature hyperparameter and is set to 2.0 in our experiments. A smaller, i.e., more negative, value indicates that features are more uniformly spread over the hypersphere. For computational efficiency, when the number of samples exceeds $20{,}000$, we randomly subsample $20{,}000$ features to estimate this metric.

C.2.2 SVD Entropy

Algorithm 2 SVD Entropy Calculation

Require: Feature matrix $\mathbf{Z} \in \mathbb{R}^{N \times d}$, numerical threshold $\epsilon = 10^{-7}$
Ensure: SVD entropy $\mathcal{H}_{\text{SVD}}$
1: Center the features: $\tilde{\mathbf{Z}} \leftarrow \mathbf{Z} - \text{mean}(\mathbf{Z}, \text{dim} = 0)$
2: Compute singular values $\{\sigma_i\}_{i=1}^{r}$ of $\tilde{\mathbf{Z}}$
3: Compute spectral energy: $\lambda_i \leftarrow \sigma_i^2$
4: Normalize energy: $p_i \leftarrow \lambda_i / \sum_{j=1}^{r} \lambda_j$
5: Remove numerically unstable entries where $p_i \leq \epsilon$
6: Compute $\mathcal{H}_{\text{SVD}} \leftarrow -\sum_i p_i \log p_i$
7: **return** $\mathcal{H}_{\text{SVD}}$

SVD entropy measures the effective utilization of feature dimensions. Given the feature matrix $\mathbf{Z}$, we first center it by subtracting the feature mean:

$$\bar{\mathbf{z}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i, \qquad \tilde{\mathbf{z}}_i = \mathbf{z}_i - \bar{\mathbf{z}}. \tag{4}$$

We then compute the singular values of $\tilde{\mathbf{Z}}$:

$$\sigma_1, \sigma_2, \ldots, \sigma_r = \text{SVD}(\tilde{\mathbf{Z}}), \tag{5}$$

where $r = \text{rank}(\tilde{\mathbf{Z}})$. The normalized spectral energy is defined as:

$$p_i = \frac{\sigma_i^2}{\sum_{j=1}^{r} \sigma_j^2}. \tag{6}$$

The SVD entropy is then computed as:

$$\mathcal{H}_{\text{SVD}}(\mathbf{Z}) = -\sum_{i=1}^{r} p_i \log p_i. \tag{7}$$

A higher SVD entropy suggests that the representation uses more informative dimensions, while a lower value indicates that the feature space is concentrated in fewer dominant directions, suggesting potential dimensional collapse.

D More Analysis of Finding-10

More details of the fingding-10

A further comparison reveals that combining Effort with the stronger standard CLIP-ViT performs worse than pairing it with the weaker patch-shuffled CLIP-ViT. We attribute this phenomenon to feature redundancy and domain bias. The standard CLIP-ViT learns a mixture of low-level and high-level features, where high-level semantics are often biased toward seen domains. When combined with Effort, these biased features overlap with Effort's semantic representations and introduce interference. In contrast, the patch-shuffled CLIP-ViT provides purely low-level signals, as spatial semantics are

removed by patch shuffling. This minimizes feature overlap while maximizing complementarity between models. Nevertheless, not all model combinations are beneficial. For example, as shown in Tab. 8, combining DINOv2 with other models degrades performance, likely due to severe feature interference and redundancy. These observations further confirm that naive ensembling of strong models is suboptimal for OOD generalization. Instead, constructing orthogonal feature spaces—by combining pure low-level fingerprints and high-level semantics—is the key to achieving robust cross-domain generalization.

SVD Entropy Analysis of Effort

To further support the analysis in Finding-10, we measure the Singular Value Decomposition (SVD) entropy of different models, as shown in Fig. 8d. SVD entropy serves as a metric for the information richness and effective dimensionality of the feature manifold. A higher SVD entropy indicates that the representation preserves more informative directions and is less likely to suffer from feature-space collapse. The results show that Effort maintains consistently higher SVD entropy across all protocols. This suggests that Effort better preserves the pre-trained CLIP-ViT representation space, while full fine-tuning tends to reduce the diversity and dimensionality of the feature manifold. This difference becomes especially clear on the fully unseen Protocol 4. The entropy score of Effort reaches 5.2455, significantly surpassing the 4.5680 of the fully fine-tuned CLIP-ViT and the 4.2117 of its patch-shuffled variant. This quantitative evidence supports our conclusion that Effort's out-of-domain robustness comes from preserving the pre-trained semantic structure. Unlike CLIP-ViT-FFT, which aggressively adapts the feature space to seen-domain fingerprints, Effort retains a richer and more continuous representation manifold. Therefore, when low-level artifacts are unreliable or corrupted, Effort can still rely on robust high-level semantic cues, leading to better generalization on unseen generative methods.

E Analysis on Low-level Fingerprints and High-level Semantics

To investigate the differences between the fully fine-tuned CLIP-ViT and Effort, which preserves the vast majority of pre-trained semantic information, we conducted patch shuffling and data augmentation experiments as shown in Tab. 6 and Tab. 7. First, for CLIP-ViT-FFT, when applying patch shuffling at the default 14×14 resolution, which disrupts global spatial semantics while preserving local details, CLIP-ViT-FFT exhibits remarkable robustness. It shows minor performance drops of 1.85% and 4.44% on Protocols 2 and 3, and a negligible drop of merely 0.47% on the unseen Protocol 4. This indicates that destroying spatial semantics barely affects its out-of-domain performance. However, reducing the shuffle granularity from 14×14 to 7×7 leads to a significant performance decline, as finer shuffling destroys not only global semantics but also the local spatial consistency aligned with the Vision Transformer tokenization scale. Conversely, corrupting low-level information yields the exact opposite effect. While standard data augmentations and latent space domain augmentation(*i.e.*, LSDA) achieve comparable or even better performance on the seen domains of Protocol- 2&3, they cause massive performance drops of 18.47% and 12.27% respectively on Protocol-4. This clear difference exposes a critical mechanism: full fine-tuning causes the model's high-level representations to severely overfit to the seen domains. As a result, the model's generalization ability relies almost entirely on local, low-level fingerprints. However, when data augmentation corrupts these low-level details, the model is forced to rely more on its high-level features. Since these high-level features are already overfitted, this forced reliance inevitably leads to a drop in robustness and poor generalization. In contrast, the Effort exhibits improved OOD generalization when subjected to data augmentations. Unlike CLIP_L_FFT, Effort freezes the vast majority of CLIP's pre-trained semantic features, this allows the model to map unseen generative methods into distinct clusters within this broad semantic space and effectively prevents the high-level overfitting observed in CLIP-ViT. When augmentations like blurring corrupt low-level details, Effort is forced to concentrate on its well-preserved, high-level semantic features, which inherently enhances its robustness.

Table 9: Summary of 100 facial forgery algorithms in DeepfakeGenome benchmark.

Dataset	Task	ID	Algorithm	Venue	Type	Domain	FF++ Train	FF++ Test	Other Real	Other Fake
FF++ [13]	FS	001	Deepfakes [55]	None	GAN	FF++	22928	4454	-	-
	FS	002	FaceShifter [80]	ArXiv 2019	CNN	FF++	22968	4479	-	-
	FR	003	NeuralTextures [58]	TOG 2019	GAN	FF++	22973	4479	-	-
	FR	004	Face2Face [56]	CVPR 2016	CG	FF++	22991	4480	-	-
DF40 [4]	FS	005	FSGAN [81]	ICCV 2019	GAN	FF++ & CDF	21815	4455	CDF-5.5k	19601
	FS	006	SimSwap [82]	MM 2020	GAN	FF++ & CDF	31350	4468	CDF-5.5k	20322
	FS	007	BlendFace [83]	ICCV 2023	GAN	FF++ & CDF	22513	4467	CDF-5.5k	20323
	FS	008	e4s [84]	CVPR 2023	GAN	FF++ & CDF	22779	3663	CDF-5.5k	9198
	FS	009	FaceDancer [85]	WACV 2023	GAN	FF++ & CDF	22747	4458	CDF-5.5k	20252
	FS	010	MobileSwap [86]	AAAI 2022	GAN	FF++ & CDF	22824	27279	CDF-5.5k	20243
	FS	011	InSwapper [87]	None	GAN	FF++ & CDF	27988	3762	CDF-5.5k	13433
	FS	012	UniFace [88]	ECCV 2022	GAN	FF++ & CDF	11794	4465	CDF-5.5k	19881
	FS	013	FaceSwap [57]	None	CG	FF++ & CDF	22852	4461	CDF-5.5k	20173
	FS	014	DeepFaceLab [89]	None	Close	UADFV	-	-	1547	3122
	FR	015	Sadtalker [90]	CVPR 2023	GAN	FF++ & CDF	22797	4416	CDF-5.5k	20986
	FR	016	MRAA [91]	CVPR 2021	GAN	FF++ & CDF	22811	4456	CDF-5.5k	20273
	FR	017	LIA [92]	ArXiv 2022	GAN	FF++ & CDF	22811	4448	CDF-5.5k	19806
	FR	018	FS_vid2vid [93]	ArXiv 2019	GAN	FF++ & CDF	22815	4468	CDF-5.5k	19890
	FR	019	Wav2Lip [94]	MM 2020	GAN	FF++ & CDF	22682	4379	CDF-5.5k	20814
	FR	020	DaGAN [95]	CVPR 2022	GAN	FF++ & CDF	22646	4412	CDF-5.5k	19284
	FR	021	Hyperreenact [96]	ICCV 2023	GAN	FF++ & CDF	18783	4480	CDF-5.5k	20503
	FR	022	MCNet [97]	ICCV 2023	CNN	FF++ & CDF	22613	4401	CDF-5.5k	19201
	FR	023	OneShot [98]	CVPR 2021	CNN	FF++ & CDF	22959	4471	CDF-5.5k	20625
	FR	024	PIRender [99]	ICCV 2021	CNN	FF++ & CDF	22914	4479	CDF-5.5k	20185
	FR	025	TPSMM [100]	CVPR 2022	CNN	FF++ & CDF	22964	4443	CDF-5.5k	20587
	FR	026	FOMM [101]	NIPS 2019	CNN	FF++ & CDF	22811	4455	CDF-5.5k	20278
	FR	027	HeyGen [102]	None	Close	VFHQ	-	-	7536	1605
	EFS	028	VQGAN [103]	CVPR 2021	GAN	FF++ & CDF	18007	3711	CDF-5.5k	28174
	EFS	029	StyleGAN3 [104]	NIPS 2021	GAN	FF++ & CDF	18007	3711	CDF-5.5k	28174
	EFS	030	StyleGAN-XL [105]	SigGraph 2022	GAN	FF++ & CDF	18007	3711	CDF-5.5k	28174
	EFS	031	WhichisReal [106]	None	GAN	WhichisReal	-	-	1000	1001
	EFS	032	DDPM [107]	NIPS 2020	Diff	FF++ & CDF	18007	3711	CDF-5.5k	28174
	EFS	033	SD-2.1 [108]	CVPR 2022	Diff	FF++ & CDF	18007	3711	CDF-5.5k	28174
	EFS	034	RDDM [109]	CVPR 2024	Diff	FF++ & CDF	18007	3711	CDF-5.5k	14027
	EFS	035	PixArt-α [110]	ArXiv 2023	Diff	FF++ & CDF	18007	3711	CDF-5.5k	28174
	EFS	036	DiT-XL/2 [111]	ICCV 2023	Diff	FF++ & CDF	18007	3711	CDF-5.5k	28174
	EFS	037	SiT-XL/2 [112]	ArXiv 2021	Diff	FF++ & CDF	18007	3711	CDF-5.5k	28174
	FE	038	e4e [113]	TOG 2021	GAN	FF++ & CDF	22993	4479	CDF-5.5k	18650
	FE	039	StarGAN [114]	CVPR 2018	GAN	CelebA	-	-	1984	1984
	FE	040	StarGANv2 [115]	CVPR 2020	GAN	CelebA	-	-	2000	2000
	FE	041	StyleCLIP [116]	ICCV 2021	GAN	CelebA	-	-	1000	2824
ForgeryNet [14]	FR	042	ATVG-Net [117]	CVPR 2019	GAN	Multi-1	-	-	ForgeryNet-1w	5209
	FR	043	FirstOrderMotion [118]	NIPS 2019	GAN	Multi-1	-	-	ForgeryNet-1w	10000
	FR	044	Talking-headVideo [119]	TOG 2019	GAN	Multi-1	-	-	ForgeryNet-1w	10000
	FE	045	DiscoFaceGAN [120]	CVPR 2020	GAN	Multi-1	-	-	ForgeryNet-1w	10000
	FE	046	MaskGAN [121]	CVPR 2020	GAN	Multi-1	-	-	ForgeryNet-1w	10000
	FE	047	SC-FEGAN [122]	ICCV 2019	GAN	Multi-1	-	-	ForgeryNet-1w	2600
	FS	048	MMreplacement [14]	CVPR 2021	CG	Multi-1	-	-	ForgeryNet-1w	10000
	FS	049	DSS [14]	CVPR 2021	GAN	Multi-1	-	-	ForgeryNet-1w	10000
	FS	050	SBS [14]	CVPR 2021	GAN	Multi-1	-	-	ForgeryNet-1w	10000
DFFD [15]	FE	051	FaceAPP [123]	None	Close	Multi-2	-	-	CelebA-1w	6309
	EFS	052	StyleGAN [124]	CVPR 2019	GAN	Multi-2	-	-	CelebA-1w	9999
	EFS	053	ProGAN [125]	ArXiv 2017	GAN	Multi-2	-	-	CelebA-1w	9982
DNA-Det [5]	EFS	054	MMDGAN [126]	ArXiv 2018	GAN	CelebA	-	-	CelebA-1w	10000
	EFS	055	SNGAN [127]	ArXiv 2018	GAN	CelebA	-	-	CelebA-1w	10000
	EFS	056	CramerGAN [128]	ArXiv 2017	GAN	CelebA	-	-	CelebA-1w	10000
	EFS	057	InfoMax-GAN [129]	WACV 2021	GAN	CelebA	-	-	CelebA-1w	10000
	EFS	058	SSGAN [130]	CVPR 2019	GAN	CelebA	-	-	CelebA-1w	10000
OSMA [16]	FE	059	AttGAN [131]	TIP 2019	GAN	CelebA	-	-	CelebA-5k	5301
WILD-20 [17]	EFS	060	Adobe Firefly [132]	None	Close	1k Prompts	-	-	CelebA-1k	1000
	EFS	061	DALL·E3 [133]	None	Close	1k Prompts	-	-	CelebA-1k	1000
	EFS	062	Freepik [134]	None	Close	1k Prompts	-	-	CelebA-1k	1000
	EFS	063	Leonardo AI [135]	None	Close	1k Prompts	-	-	CelebA-1k	1000
	EFS	064	Starry AI [136]	None	Close	1k Prompts	-	-	CelebA-1k	1000
	EFS	065	DALL·E 1 [137]	ArXiv 2021	Close	1k Prompts	-	-	CelebA-1k	1000
	EFS	066	Deep AI [138]	None	Close	1k Prompts	-	-	CelebA-1k	1000
	EFS	067	Hotpot AI [139]	None	Close	1k Prompts	-	-	CelebA-1k	1000
	EFS	068	FLUX.1 [140]	None	Diff	1k Prompts	-	-	CelebA-1k	1000
	EFS	069	FLUX 1.1 Pro [141]	None	Diff	1k Prompts	-	-	CelebA-1k	1000
	EFS	070	SD 3.5 Large [142]	CVPR 2022	Diff	1k Prompts	-	-	CelebA-1k	1000
	EFS	071	SD XL Turbo [142]	CVPR 2022	Diff	1k Prompts	-	-	CelebA-1k	1000
	EFS	072	NVIDIA Sana PAG [143]	ICLR 2025	Diff	1k Prompts	-	-	CelebA-1k	1000
	EFS	073	Stable Cascade [144]	Arxiv 2023	Diff	1k Prompts	-	-	CelebA-1k	1000
	EFS	074	SD Attend&Excite [145]	SigGraph 2023	Diff	1k Prompts	-	-	CelebA-1k	1000
	EFS	075	Hunyuan-DiT [146]	ArXiv 2024	Diff	1k Prompts	-	-	CelebA-1k	1000
DiFF [18]	EFS	076	Midjourney [147]	None	Diff	Multi-3	-	-	CelebA-1w	10000
	EFS	077	SD XL [148]	CoRR 2023	Diff	Multi-3	-	-	CelebA-1w	10000
	EFS	078	FreeDoM_T [149]	ICCV 2023	Diff	Multi-3	-	-	CelebA-1w	10000
	EFS	079	HPS [150]	ICCV 2023	Diff	Multi-3	-	-	CelebA-1w	10000
	FE	080	SDXL Refiner [148]	ICLR 2024	Diff	Multi-3	-	-	CelebA-1w	10000
	FE	081	DeamBooth [151]	CVPR 2023	Diff	Multi-3	-	-	CelebA-1w	10000
	FS	082	DiffFace [152]	ArXiv 2022	Diff	Multi-3	-	-	CelebA-1w	10000
	FS	083	DCFace [153]	CVPR 2023	Diff	Multi-3	-	-	CelebA-1w	10000
	FE	084	Imagic [154]	CVPR 2023	Diff	Multi-3	-	-	CelebA-1w	10000
	FE	085	CollabDiff [155]	CVPR 2023	Diff	Multi-3	-	-	CelebA-1w	10000
	FE	086	CycleDiff [156]	ICCV 2023	Diff	Multi-3	-	-	CelebA-1w	10000
DiffFace [19]	EFS	087	PNDM [157]	ICLR 2022	Diff	CelebA	-	-	CelebA-1w	10000
	EFS	088	P2/ADM [158]	CVPR 2022	Diff	CelebA	-	-	CelebA-1w	10000
	EFS	089	LDM [142]	CVPR 2022	Diff	CelebA	-	-	CelebA-1w	10000
	FE	090	SD v1.5 [142]	CVPR 2022	Diff	Prompt+Img	-	-	CelebA-1w	10000
	FE	091	SDEdit [159]	ICLR 2022	Diff	Prompt+Img	-	-	CelebA-1w	10000
	FS	092	DiffSwap [160]	CVPR 2023	Diff	MM-CelebA-HQ	-	-	CelebA-1w	10000
Synthetized	EFS	093	Qwen-Image [1]	ArXiv 2025	Diff	CelebA	-	-	CelebA-5k	4869
	EFS	094	BAGEL [20]	ArXiv 2025	Diff	CelebA	-	-	CelebA-5k	4937
	EFS	095	HunyuanImage-2.1 [21]	ArXiv 2025	Diff	CelebA	-	-	CelebA-5k	4808
	EFS	096	HunyuanImage-3.0 [22]	ArXiv 2025	Diff	CelebA	-	-	CelebA-5k	5000
	EFS	097	Infinity [23]	CVPR 2025	VAR	CelebA	-	-	CelebA-5k	4955
	EFS	098	Hart [24]	ArXiv 2024	VAR	CelebA	-	-	CelebA-5k	4884
	EFS	099	Nano Banana [2]	None	Close	CelebA	-	-	CelebA-1k	1000
	EFS	100	GPT-4o [25]	None	Close	CelebA	-	-	CelebA-1k	977

Table 10: Protocol-1 training and testing algorithms.

Training Set	Algorithms
DeepFakes [55], Face2Face [56], FaceSwap [57], NeuralTextures [58]	ATVG-Net [117], FirstOrderMotion [118], Talking-headVideo [119], DiscoFaceGAN [120], MaskGAN [121], SC-FEGAN [122], MMreplacement [14], DSS [14], SBS [14], FaceAPP [123], StyleGAN [124], ProGAN [125], MMDGAN [126], SNGAN [127], CramerGAN [128], InfoMax-GAN [129], SSGAN [130], AttGAN [131], Adobe Firefly [132], DALL·E3 [133], Freepik [134], Leonardo AI [135], Starry AI [136], DALL·E 1 [137], Deep AI [138], Hotpot AI [139], FLUX.1 [140], FLUX 1.1 Pro [141], SD 3.5 Large [142], SD XL Turbo [142], NVIDIA Sana PAG [143], Stable Cascade [144], SD Attend&Excite [145], Hunyuan-DiT [146], Midjourney [147], SD XL [148], FreeDoM_T [149], HPS [150], SDXL Refiner [148], DreamBooth [151], DiffFace [152], DCFace [153], Imagic [154], CollabDiff [155], CycleDiff [156], PNDM [157], P2/ADM [158], LDM [142], SD v1.5 [142], SDEdit [159], DiffSwap [160], Qwen-Image [1], BAGEL [20], HunyuanImage-2.1 [21], HunyuanImage-3.0 [22], Infinity [23], Hart [24], Nano Banana [2], GPT-4o [25]

Table 11: Protocol-2 training and testing algorithms.

Training Set	Test Set
Deepfakes [55], FaceShifter [80], NeuralTextures [58], Face2Face [56], FSGAN [81], SimSwap [82], BlendFace [83], e4s [84], FaceDancer [85], MobileSwap [86], InSwapper [87], UniFace [88], FaceSwap [57], Sadtalker [90], MRAA [91], LIA [92], FS_vid2vid [93], Wav2Lip [94], DaGAN [95], Hyperreenact [96], MCNet [97], OneShot [98], PIRender [99], TPSMM [100], FOMM [101], HeyGen [102], VQGAN [103], StyleGAN3 [104], StyleGAN-XL [105], DDPM [107], SD-2.1 [108], RDDM [109], PixArt-α [110], DiT-XL/2 [111], SiT-XL/2 [112], e4e [113]	Deepfakes [55], FaceShifter [80], NeuralTextures [58], Face2Face [56], FSGAN [81], SimSwap [82], BlendFace [83], e4s [84], FaceDancer [85], MobileSwap [86], InSwapper [87], UniFace [88], FaceSwap [57], Sadtalker [90], MRAA [91], LIA [92], FS_vid2vid [93], Wav2Lip [94], DaGAN [95], Hyperreenact [96], MCNet [97], OneShot [98], PIRender [99], TPSMM [100], FOMM [101], HeyGen [102], VQGAN [103], StyleGAN3 [104], StyleGAN-XL [105], DDPM [107], SD-2.1 [108], RDDM [109], PixArt-α [110], DiT-XL/2 [111], SiT-XL/2 [112], e4e [113]

Table 12: Protocol-3 training and testing algorithms.

Training Set	Testing Set
Deepfakes [55], FaceShifter [80], NeuralTextures [58], Face2Face [56], FSGAN [81], SimSwap [82], BlendFace [83], e4s [84], FaceDancer [85], MobileSwap [86], InSwapper [87], UniFace [88], FaceSwap [57], Sadtalker [90], MRAA [91], LIA [92], FS_vid2vid [93], Wav2Lip [94], DaGAN [95], Hyperreenact [96], MCNet [97], OneShot [98], PIRender [99], TPSMM [100], FOMM [101], HeyGen [102], VQGAN [103], StyleGAN3 [104], StyleGAN-XL [105], DDPM [107], SD-2.1 [108], RDDM [109], PixArt-α [110], DiT-XL/2 [111], SiT-XL/2 [112], e4e [113]	FSGAN [81], SimSwap [82], BlendFace [83], e4s [84], FaceDancer [85], MobileSwap [86], InSwapper [87], UniFace [88], FaceSwap [57], Sadtalker [90], MRAA [91], LIA [92], FS_vid2vid [93], Wav2Lip [94], DaGAN [95], Hyperreenact [96], MCNet [97], OneShot [98], PIRender [99], TPSMM [100], FOMM [101], HeyGen [102], VQGAN [103], StyleGAN3 [104], StyleGAN-XL [105], DDPM [107], SD-2.1 [108], RDDM [109], PixArt-α [110], DiT-XL/2 [111], SiT-XL/2 [112], e4e [113]

Table 13: Protocol-4 training and testing algorithms.

Training Set	Testing Set
Deepfakes [55], FaceShifter [80], NeuralTextures [58], Face2Face [56], FSGAN [81], SimSwap [82], BlendFace [83], e4s [84], FaceDancer [85], MobileSwap [86], InSwapper [87], UniFace [88], FaceSwap [57], Sadtalker [90], MRAA [91], LIA [92], FS_vid2vid [93], Wav2Lip [94], DaGAN [95], Hyperreenact [96], MCNet [97], OneShot [98], PIRender [99], TPSMM [100], FOMM [101], HeyGen [102], VQGAN [103], StyleGAN3 [104], StyleGAN-XL [105], DDPM [107], SD-2.1 [108], RDDM [109], PixArt-α [110], DiT-XL/2 [111], SiT-XL/2 [112], e4e [113]	ATVG-Net [117], FirstOrderMotion [118], Talking-headVideo [119], DiscoFaceGAN [120], MaskGAN [121], SC-FEGAN [122], MMreplacement [14], DSS [14], SBS [14], FaceAPP [123], StyleGAN [124], ProGAN [125], MMDGAN [126], SNGAN [127], CramerGAN [128], InfoMax-GAN [129], SSGAN [130], AttGAN [131], Adobe Firefly [132], DALL·E3 [133], Freepik [134], Leonardo AI [135], Starry AI [136], DALL·E 1 [137], Deep AI [138], Hotpot AI [139], FLUX.1 [140], FLUX 1.1 Pro [141], SD 3.5 Large [142], SD XL Turbo [142], NVIDIA Sana PAG [143], Stable Cascade [144], SD Attend&Excite [145], Hunyuan-DiT [146], Midjourney [147], SD XL [148], FreeDoM_T [149], HPS [150], SDXL Refiner [148], DreamBooth [151], DiffFace [152], DCFace [153], Imagic [154], CollabDiff [155], CycleDiff [156], PNDM [157], P2/ADM [158], LDM [142], SD v1.5 [142], SDEdit [159], DiffSwap [160], Qwen-Image [1], BAGEL [20], HunyuanImage-2.1 [21], HunyuanImage-3.0 [22], Infinity [23], Hart [24], Nano Banana [2], GPT-4o [25]

Table 14: ACC on 59 new algorithms in Protocol-1.

Algorithm	Basic Backbone		Finetuned CLIP		Reconstruction-based		Frequency-domain			Latent Space		Representation			Aug Adv.
	Xception [60]	Eff-B4 [61]	CLIP-L-FFT [27]	Effort [3]	RECCE [62]	SBI [30]	F3Net [63]	SRM [64]	SPSL [65]	ProDet [66]	LSDA [67]	UCF [68]	IID [69]	CORE [70]	sladd [71]
ATVG-Net [117]	0.7110	0.7480	0.7601	0.8360	0.7412	0.8347	0.7588	0.7037	0.5620	0.6876	0.7553	0.7827	0.7287	0.7453	0.7494
FirstOrderMotion [118]	0.7910	0.7955	0.8190	0.8679	0.7922	0.6973	0.8116	0.7681	0.6698	0.7347	0.8019	0.8256	0.7811	0.8069	0.7743
Talking-headVideo [119]	0.5541	0.5997	0.6177	0.7717	0.5593	0.5491	0.5514	0.5214	0.6152	0.6181	0.5996	0.5390	0.5203	0.5782	0.5863
MMreplacement [14]	0.5252	0.5430	0.5837	0.5824	0.4954	0.4823	0.5072	0.5006	0.5359	0.4819	0.4863	0.5064	0.4751	0.5206	0.4821
DiscoFaceGAN [120]	0.8069	0.8244	0.8138	0.8565	0.8161	0.7264	0.8394	0.7673	0.6821	0.7516	0.8086	0.8346	0.7829	0.8214	0.7948
MaskGAN [121]	0.6173	0.6460	0.8086	0.8616	0.6484	0.6246	0.6376	0.6445	0.6002	0.6572	0.6472	0.6839	0.6640	0.6631	0.6918
SC-FEGAN [122]	0.7311	0.7459	0.7254	0.8021	0.7399	0.8052	0.8093	0.6717	0.5063	0.6104	0.7270	0.7872	0.7304	0.7634	0.7231
DSS [14]	0.8111	0.7979	0.8200	0.8667	0.7928	0.7761	0.8092	0.7677	0.6790	0.7567	0.8233	0.8122	0.8060	0.8241	0.8012
SBS [14]	0.6126	0.5893	0.6942	0.7637	0.5975	0.5593	0.6047	0.5809	0.5747	0.5801	0.5697	0.5987	0.5896	0.6006	0.6051
FaceAPP [123]	0.5560	0.7267	0.6308	0.7367	0.6335	0.5168	0.6448	0.5484	0.4916	0.4890	0.6435	0.6634	0.6213	0.5918	0.6145
ProGAN [125]	0.5541	0.7232	0.7725	0.8750	0.6100	0.5755	0.6158	0.6062	0.5173	0.5133	0.5827	0.6427	0.5956	0.5704	0.5802
StyleGAN [124]	0.5000	0.7153	0.5854	0.8099	0.6101	0.5109	0.5625	0.6077	0.4867	0.4977	0.6090	0.5716	0.5796	0.5345	0.6243
MMDGAN [126]	0.4090	0.5946	0.7205	0.8763	0.4512	0.5276	0.5204	0.3906	0.5801	0.4281	0.4994	0.5341	0.4459	0.4561	0.3723
SNGAN [127]	0.4159	0.5926	0.6548	0.8749	0.4400	0.5509	0.5003	0.3937	0.5648	0.4312	0.4763	0.5304	0.4501	0.4596	0.3769
CramerGAN [128]	0.4516	0.5877	0.7561	0.8762	0.4916	0.5543	0.5497	0.4759	0.6040	0.4802	0.5461	0.6068	0.5379	0.5141	0.4135
InfoMax-GAN [129]	0.5495	0.4752	0.7118	0.8763	0.5206	0.5226	0.5438	0.4862	0.6310	0.4635	0.6118	0.5841	0.5427	0.6579	0.4907
SSGAN [130]	0.5381	0.4460	0.7377	0.8763	0.5206	0.5254	0.5372	0.4518	0.6335	0.4282	0.5800	0.5713	0.5709	0.6449	0.4873
AttGAN [131]	0.5065	0.6767	0.7662	0.8727	0.5196	0.4769	0.6360	0.4530	0.5235	0.3348	0.5950	0.6664	0.4697	0.5448	0.4067
Adobe Firefly [132]	0.5697	0.5195	0.5378	0.4795	0.5819	0.3862	0.6133	0.5712	0.5079	0.3882	0.6731	0.5712	0.6163	0.5991	0.5712
Dall-E 3 [133]	0.7122	0.6403	0.7086	0.5911	0.6983	0.4351	0.7768	0.6434	0.6039	0.5480	0.8758	0.6660	0.8199	0.7814	0.7358
Flux.1.1 Pro [141]	0.5055	0.4672	0.4446	0.3916	0.5554	0.3095	0.5464	0.5131	0.4567	0.5504	0.7021	0.4889	0.6704	0.5943	0.5973
Flux.1 [140]	0.5479	0.4830	0.4810	0.3975	0.5850	0.3445	0.5839	0.4490	0.4073	0.5355	0.7111	0.5247	0.6596	0.6421	0.5901
Freepik [134]	0.6917	0.5335	0.4404	0.4114	0.6717	0.2868	0.6472	0.6607	0.6241	0.5130	0.8263	0.5420	0.7437	0.6687	0.6687
Leonardo AI [135]	0.4884	0.4178	0.4284	0.3891	0.5202	0.3649	0.5524	0.5756	0.4325	0.3664	0.6890	0.4587	0.5726	0.5917	0.5751
Stable Diffusion 3.5 [142]	0.5617	0.4990	0.5071	0.4337	0.5658	0.3741	0.5846	0.5347	0.4526	0.4888	0.7125	0.5209	0.5770	0.6177	0.6183
SD Attend&Excite [145]	0.6222	0.5222	0.5192	0.5076	0.6242	0.3429	0.6621	0.5763	0.5556	0.4995	0.7687	0.5222	0.7066	0.6985	0.6323
Starry AI [136]	0.5280	0.4710	0.4564	0.4081	0.5436	0.3703	0.5466	0.5718	0.5038	0.4378	0.6786	0.4932	0.5819	0.5844	0.5788
Dall-E 1 [137]	0.7419	0.5897	0.7057	0.7651	0.7243	0.4698	0.7435	0.6815	0.5857	0.5565	0.8407	0.6502	0.7808	0.7802	0.7152
Deep AI [138]	0.6122	0.5257	0.4728	0.4529	0.5791	0.3461	0.6392	0.5858	0.5405	0.5094	0.7863	0.5308	0.7089	0.6117	0.6178
Hotpot AI [139]	0.4600	0.4104	0.4154	0.4059	0.4825	0.2918	0.4830	0.5260	0.4840	0.3499	0.5586	0.4550	0.4685	0.4885	0.4740
Nvidia Sana PAG [143]	0.6267	0.5360	0.5361	0.4619	0.6303	0.3208	0.6792	0.5237	0.4923	0.5324	0.8115	0.5458	0.7405	0.6900	0.6936
Stable Cascade [144]	0.6734	0.5736	0.6295	0.4315	0.6759	0.4214	0.7067	0.6492	0.5736	0.5418	0.8191	0.6240	0.7303	0.7571	0.7314
Tencent Hunyuan [146]	0.6970	0.6304	0.6126	0.5196	0.7077	0.3859	0.7626	0.6518	0.5857	0.5435	0.8485	0.6411	0.7809	0.7570	0.7240
Stable Diffusion XL [142]	0.4671	0.4414	0.4837	0.4590	0.5354	0.3836	0.5073	0.6084	0.4822	0.4450	0.5510	0.4877	0.6089	0.5953	0.6204
Midjourney [147]	0.6191	0.5692	0.5793	0.4037	0.6869	0.5073	0.6851	0.7123	0.6061	0.6151	0.6905	0.5093	0.7215	0.7283	0.7386
SDXL [148]	0.5975	0.5617	0.5998	0.5470	0.6794	0.5492	0.6948	0.7327	0.6004	0.6134	0.6722	0.5789	0.7489	0.7183	0.7280
SDXL_Refine [148]	0.6294	0.5857	0.5659	0.4253	0.7154	0.5292	0.7134	0.7359	0.6099	0.6303	0.6825	0.6025	0.7801	0.7571	0.7283
FreeDoM_T [149]	0.6815	0.6211	0.6235	0.8533	0.6485	0.3977	0.5679	0.7138	0.7699	0.6888	0.6919	0.6756	0.7105	0.7466	0.6273
HPS [150]	0.6108	0.6307	0.6559	0.7448	0.6865	0.4592	0.6990	0.7205	0.6365	0.6603	0.6299	0.6287	0.7779	0.7132	0.6990
Imagic [154]	0.4759	0.4099	0.5643	0.8499	0.4663	0.3167	0.3743	0.6037	0.4763	0.4748	0.4859	0.4115	0.5340	0.4567	0.6064
DreamBooth [151]	0.4712	0.5703	0.5625	0.5981	0.5557	0.4441	0.5731	0.6529	0.5007	0.5713	0.5829	0.4779	0.7010	0.5739	0.6657
DiffFace [152]	0.4102	0.5492	0.7921	0.9087	0.3973	0.3377	0.3285	0.4516	0.6577	0.5133	0.4536	0.3779	0.3604	0.4259	0.4668
DCFace [153]	0.7455	0.7963	0.8933	0.9129	0.7158	0.5285	0.7099	0.6511	0.7776	0.7203	0.6454	0.8466	0.7329	0.7834	0.5581
CoDiff [155]	0.4383	0.3505	0.3603	0.8575	0.3297	0.3341	0.3239	0.3268	0.6618	0.5759	0.3803	0.4620	0.3989	0.5086	0.3615
cycle_diff [156]	0.4270	0.4869	0.6076	0.8490	0.4063	0.4929	0.4159	0.5154	0.6299	0.5430	0.4171	0.3875	0.4812	0.4468	0.5349
PNDM [157]	0.5406	0.5783	0.6165	0.7538	0.5365	0.5258	0.5385	0.5589	0.6258	0.5799	0.6093	0.6059	0.5426	0.5774	0.5554
ADM [158]	0.6484	0.5653	0.6307	0.8475	0.6112	0.4922	0.5969	0.5960	0.5870	0.5468	0.5354	0.6352	0.6713	0.6254	0.5923
LDM [142]	0.7390	0.6316	0.6480	0.7572	0.7306	0.5716	0.6950	0.7054	0.6097	0.5757	0.7719	0.6838	0.7335	0.7674	0.7252
SDv15_DS0.7 [142]	0.5254	0.4295	0.4661	0.5996	0.5319	0.4907	0.5084	0.5628	0.5314	0.5041	0.5572	0.5033	0.5378	0.5466	0.5368
Inpaint [159]	0.5144	0.5101	0.5407	0.6350	0.5072	0.5276	0.5193	0.5117	0.5247	0.5256	0.5293	0.5226	0.5251	0.5083	0.5288
DiffSwap [160]	0.7451	0.7068	0.8678	0.8760	0.7180	0.7505	0.6992	0.7206	0.6540	0.6480	0.6989	0.7508	0.6885	0.6816	0.7207
BAGEL [20]	0.6202	0.5302	0.4788	0.5518	0.6176	0.3203	0.6183	0.5364	0.5147	0.5053	0.6805	0.5663	0.6583	0.6069	0.5603
GPT4o [25]	0.6980	0.5569	0.4512	0.3971	0.7011	0.3111	0.6439	0.6480	0.4694	0.3829	0.7522	0.5306	0.6201	0.6995	0.5832
Hart [24]	0.7104	0.6131	0.5715	0.6458	0.7245	0.4197	0.7204	0.6726	0.5876	0.5243	0.8361	0.6114	0.6936	0.7596	0.6736
HunyuanImage-2.1 [21]	0.6308	0.4724	0.4210	0.4053	0.5399	0.2956	0.5323	0.5407	0.4784	0.4830	0.6818	0.4866	0.7230	0.6005	0.5205
HunyuanImage-3.0 [22]	0.6229	0.5439	0.4373	0.3890	0.6132	0.4059	0.5846	0.5890	0.4502	0.4790	0.6857	0.4955	0.6091	0.6527	0.5048
Infinity [23]	0.5722	0.5378	0.4716	0.4523	0.5914	0.3065	0.5469	0.5493	0.4996	0.4857	0.6590	0.5295	0.6864	0.5457	0.5268
NanoBanana [2]	0.7195	0.6500	0.6853	0.6313	0.7690	0.4847	0.7858	0.5690	0.4428	0.3692	0.7973	0.7720	0.6965	0.7630	0.6268
Qwen-Image [1]	0.6390	0.5078	0.4340	0.4403	0.6115	0.3356	0.5977	0.5594	0.5194	0.5221	0.6704	0.5821	0.6793	0.6052	0.5652

Table 15: AUC on 59 new algorithms in Protocol-1.

Algorithm	Basic Backbone		Finetuned CLIP		Reconstruction-based		Frequency-domain			Latent Space		Representation			Aug Adv.
	Xception [60]	**Eff-B4** [61]	**CLIP-L-FFT** [27]	**Effort** [3]	**RECCE** [62]	**SBI** [30]	**F3Net** [63]	**SRM** [64]	**SPSL** [65]	**ProDet** [66]	**LSDA** [67]	**UCF** [68]	**IID** [69]	**CORE** [70]	**sladd** [71]
ATVG-Net [117]	0.8039	0.8392	0.9452	0.9808	0.8267	0.8875	0.8246	0.8346	0.7674	0.9519	0.8559	0.8725	0.8331	0.8392	0.8996
FirstOrderMotion [118]	0.9016	0.8880	0.9734	0.9824	0.8925	0.7724	0.9000	0.8911	0.8559	0.8773	0.9034	0.9184	0.8844	0.9082	0.8910
Talking-headVideo [119]	0.5984	0.6486	0.6682	0.8397	0.5975	0.6163	0.6191	0.5438	0.6766	0.6627	0.6489	0.5970	0.5359	0.6316	0.6139
MMreplacement [14]	0.5622	0.5732	0.6284	0.6336	0.5038	0.4873	0.5467	0.5026	0.5384	0.4746	0.4908	0.5493	0.4624	0.5562	0.4672
DiscoFaceGAN [120]	0.9160	0.9263	0.9490	0.9605	0.9213	0.8242	0.9243	0.9143	0.8895	0.9117	0.9159	0.9217	0.8847	0.9199	0.9143
MaskGAN [121]	0.6770	0.6936	0.9534	0.9721	0.7188	0.6725	0.7078	0.6945	0.7040	0.7414	0.7013	0.7563	0.7207	0.7333	0.7686
SC-FEGAN [122]	0.8917	0.8593	0.9618	0.9785	0.8393	0.8064	0.9129	0.8328	0.9313	0.8267	0.8191	0.9028	0.8988	0.9196	0.9503
DSS [14]	0.9175	0.8858	0.9715	0.9825	0.8878	0.8687	0.8922	0.8848	0.8863	0.9261	0.9132	0.9052	0.9108	0.9222	0.9274
SBS [14]	0.6547	0.6141	0.7648	0.8195	0.6399	0.6167	0.6501	0.6262	0.6253	0.6078	0.5998	0.6516	0.6233	0.6546	0.6495
FaceAPP [123]	0.6123	0.8275	0.6324	0.7948	0.6386	0.5137	0.6968	0.6177	0.5079	0.5530	0.6744	0.6733	0.6395	0.6767	0.6358
ProGAN [125]	0.6463	0.8108	0.8557	0.9837	0.6734	0.5924	0.7186	0.6494	0.4897	0.4711	0.6221	0.7220	0.6000	0.6741	0.5989
StyleGAN [124]	0.6231	0.7901	0.6421	0.9018	0.6873	0.5168	0.6796	0.6597	0.4429	0.4771	0.7065	0.6363	0.6009	0.6708	0.6440
MMDGAN [126]	0.3996	0.6452	0.7981	0.9838	0.5249	0.5513	0.6727	0.3778	0.6299	0.3601	0.4951	0.6564	0.4151	0.6197	0.1836
SNGAN [127]	0.3821	0.6415	0.7192	0.9750	0.5000	0.5800	0.6314	0.3824	0.6159	0.3672	0.4288	0.6248	0.4120	0.5879	0.1878
CramerGAN [128]	0.5047	0.6380	0.8382	0.9885	0.6213	0.5877	0.7115	0.5089	0.6716	0.4173	0.5793	0.7447	0.5532	0.7054	0.2996
InfoMax-GAN [129]	0.5364	0.5086	0.7886	0.9915	0.5982	0.5492	0.6657	0.5457	0.7020	0.3867	0.6589	0.7009	0.5767	0.7348	0.4433
SSGAN [130]	0.5358	0.4625	0.8157	0.9951	0.6106	0.5505	0.6561	0.4905	0.6965	0.3569	0.6185	0.6838	0.6084	0.7215	0.4244
AttGAN [131]	0.5249	0.7304	0.8550	0.9836	0.6524	0.5025	0.7959	0.4880	0.5515	0.2655	0.6625	0.7958	0.4494	0.7137	0.2765
Adobe Firefly [132]	0.5971	0.5032	0.5773	0.4660	0.5727	0.3648	0.6519	0.6280	0.5372	0.3787	0.7407	0.5127	0.6744	0.6295	0.6152
Dall-E 3 [133]	0.8002	0.7080	0.8067	0.6097	0.7870	0.4336	0.8616	0.7475	0.6590	0.6235	0.9562	0.7572	0.9090	0.8776	0.7963
Flux.1.1 Pro [141]	0.5069	0.4519	0.4708	0.2318	0.6233	0.2184	0.5748	0.5508	0.4156	0.6158	0.8052	0.4264	0.7367	0.6324	0.6207
Flux.1 [140]	0.5623	0.4501	0.4933	0.2142	0.6475	0.3059	0.6318	0.4729	0.3660	0.5734	0.8185	0.4855	0.7338	0.6998	0.6197
Freepik [134]	0.7294	0.5397	0.4656	0.3297	0.7470	0.1609	0.7223	0.7725	0.6336	0.5339	0.9020	0.5676	0.8255	0.7326	0.7154
Leonardo AI [135]	0.5292	0.3679	0.4551	0.2345	0.5524	0.2984	0.6250	0.6134	0.3926	0.3150	0.7826	0.3237	0.6129	0.6565	0.6000
Stable Diffusion 3.5 [142]	0.5599	0.4554	0.5262	0.3971	0.5714	0.3363	0.6055	0.5780	0.4141	0.4953	0.7955	0.4464	0.6104	0.6390	0.6634
SD Attend&Excite [145]	0.6803	0.5450	0.5803	0.4343	0.6767	0.3242	0.7613	0.6464	0.5834	0.5290	0.8658	0.5460	0.7743	0.7665	0.6837
Starry AI [136]	0.5905	0.4656	0.4808	0.2489	0.5878	0.3109	0.6124	0.6216	0.4751	0.4267	0.7698	0.4462	0.6374	0.6400	0.6118
Dall-E 1 [137]	0.8092	0.6280	0.7911	0.8377	0.7994	0.4720	0.8296	0.7756	0.6127	0.6279	0.9193	0.7160	0.8693	0.8518	0.7952
Deep AI [138]	0.6660	0.5400	0.4981	0.3578	0.6418	0.2883	0.7212	0.6609	0.5489	0.5538	0.8878	0.5230	0.7786	0.6794	0.6548
Hotpot AI [139]	0.4349	0.3836	0.3484	0.2507	0.4490	0.2123	0.5070	0.5511	0.4359	0.2956	0.6054	0.3118	0.4945	0.4776	0.4854
Nvidia Sana PAG [143]	0.7132	0.5527	0.6022	0.4275	0.7198	0.2873	0.7738	0.5847	0.4845	0.6194	0.9057	0.5677	0.8087	0.7643	0.7445
Stable Cascade [144]	0.7215	0.5919	0.7254	0.3686	0.7330	0.4110	0.7660	0.7453	0.6020	0.6104	0.9060	0.6053	0.8225	0.8285	0.7893
Tencent Hunyuan [146]	0.7969	0.6960	0.6787	0.4961	0.8028	0.3564	0.8464	0.7518	0.6030	0.6384	0.9343	0.7344	0.8673	0.8479	0.7797
Stable Diffusion XL [142]	0.3972	0.4320	0.5021	0.3725	0.5497	0.3396	0.5231	0.6576	0.4393	0.4341	0.6074	0.4370	0.6446	0.6371	0.6629
Midjourney [147]	0.7222	0.6405	0.7316	0.5210	0.7912	0.5298	0.8039	0.7892	0.5643	0.6374	0.7374	0.6254	0.7755	0.8410	0.7584
SDXL [148]	0.7003	0.6168	0.7376	0.6514	0.7912	0.5764	0.8090	0.8035	0.5709	0.6353	0.6959	0.7042	0.7929	0.8364	0.7464
SDXL_Refine [148]	0.7007	0.6347	0.7198	0.5452	0.8128	0.5544	0.8127	0.8138	0.5847	0.6738	0.7041	0.7109	0.8287	0.8571	0.7515
FreeDoM_T [149]	0.7287	0.6761	0.7434	0.8972	0.7589	0.4511	0.7164	0.7701	0.8303	0.6654	0.8014	0.8137	0.7665	0.8189	0.6699
HPS [150]	0.7230	0.7110	0.7640	0.8005	0.7840	0.4839	0.8156	0.7933	0.5914	0.6734	0.7080	0.7517	0.8070	0.8264	0.7116
Imagic [154]	0.6744	0.4992	0.6182	0.8836	0.5902	0.3070	0.5822	0.6465	0.5105	0.4858	0.4714	0.5785	0.5900	0.7153	0.6405
DreamBooth [151]	0.6603	0.6592	0.6881	0.6636	0.6901	0.4616	0.7412	0.7014	0.4564	0.5268	0.6518	0.6641	0.7285	0.7548	0.6553
DiffFace [152]	0.4923	0.6292	0.8766	0.9838	0.5526	0.3257	0.4855	0.4624	0.6284	0.4369	0.5701	0.5708	0.3910	0.5710	0.5070
DCFace [153]	0.7933	0.8501	0.9573	0.9939	0.7963	0.5603	0.8382	0.7014	0.8400	0.6633	0.7613	0.9245	0.7903	0.8514	0.5513
CoDiff [155]	0.4588	0.4067	0.4556	0.9062	0.4212	0.3551	0.4023	0.3334	0.6909	0.5037	0.5734	0.6516	0.4818	0.5960	0.3732
cycle_diff [156]	0.5249	0.5674	0.7193	0.9037	0.5283	0.5176	0.5918	0.5136	0.5984	0.4880	0.4986	0.5463	0.5446	0.6180	0.5767
PNDM [157]	0.5684	0.6129	0.6692	0.8244	0.5973	0.5453	0.5995	0.6155	0.7167	0.6167	0.6842	0.7166	0.5499	0.6697	0.5778
ADM [158]	0.6666	0.5720	0.6854	0.9354	0.6534	0.4889	0.6375	0.6485	0.6499	0.5910	0.5637	0.6917	0.7364	0.6404	0.6274
LDM [142]	0.7994	0.6709	0.7131	0.8286	0.8025	0.5993	0.7549	0.8225	0.6641	0.6418	0.8505	0.7562	0.8225	0.8352	0.8121
SDv15_DS0.7 [142]	0.5266	0.3723	0.4341	0.6340	0.5530	0.4981	0.5101	0.5914	0.5349	0.4869	0.5656	0.5303	0.5653	0.5714	0.5471
Inpaint [159]	0.5015	0.5077	0.5543	0.6705	0.4993	0.5407	0.5186	0.5212	0.5379	0.5427	0.5275	0.5394	0.5288	0.5057	0.5372
DiffSwap [160]	0.8137	0.7800	0.9673	0.9967	0.7943	0.9058	0.7855	0.8593	0.7863	0.8600	0.7781	0.8404	0.7544	0.7376	0.8174
BAGEL [20]	0.6108	0.5182	0.4209	0.5319	0.6551	0.2476	0.6412	0.5972	0.5178	0.5597	0.7387	0.5650	0.7264	0.5926	0.5560
GPT4o [25]	0.7467	0.5716	0.4356	0.3226	0.7666	0.2116	0.7243	0.7107	0.4243	0.3541	0.8383	0.5770	0.6763	0.7529	0.6100
Hart [24]	0.7692	0.6405	0.6192	0.6771	0.7882	0.4062	0.7973	0.7620	0.6179	0.5636	0.9095	0.6804	0.7755	0.8328	0.7320
HunyuanImage-2.1 [21]	0.6551	0.4533	0.2828	0.2900	0.5621	0.1172	0.5228	0.5947	0.4416	0.4890	0.7542	0.4861	0.8036	0.6255	0.5168
HunyuanImage-3.0 [22]	0.6649	0.5458	0.4011	0.2300	0.6633	0.3691	0.6386	0.6445	0.4198	0.4486	0.7712	0.4674	0.6677	0.7020	0.5042
Infinity [23]	0.5631	0.5195	0.4206	0.4065	0.6310	0.2004	0.5319	0.6123	0.4793	0.4917	0.7347	0.5233	0.7614	0.5401	0.5501
NanoBanana [2]	0.8167	0.6745	0.4688	0.3986	0.8011	0.2802	0.7655	0.7794	0.5821	0.4988	0.8854	0.6649	0.8562	0.8536	0.6503
Qwen-Image [1]	0.6479	0.4759	0.3464	0.3278	0.6278	0.2564	0.6192	0.6415	0.5315	0.5823	0.7426	0.5856	0.7608	0.6033	0.5696

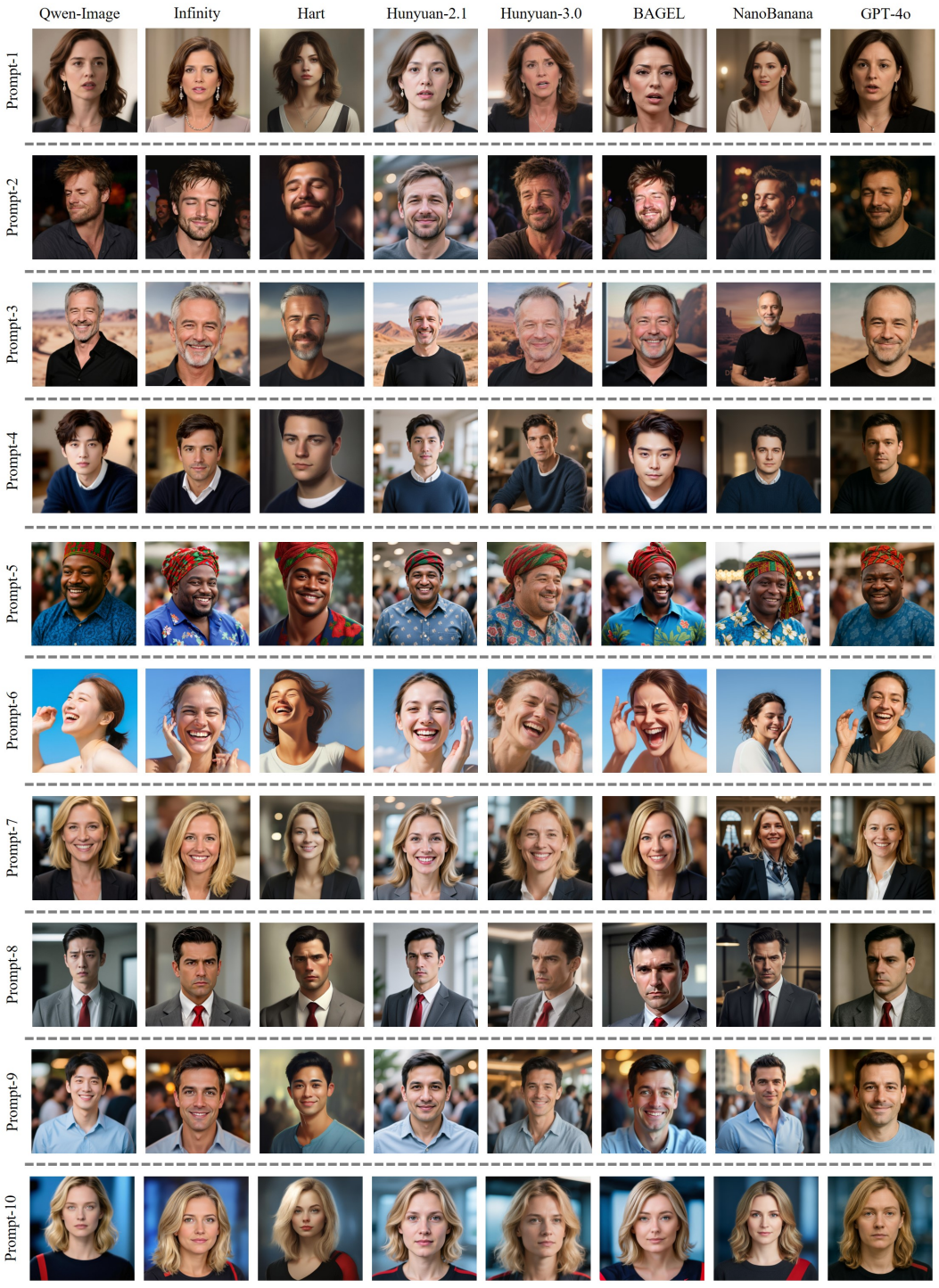


Figure 9: Visualization of 8 synthesized algorithms.

Prompt ID	Textual Description Prompt
1	The individual has shoulder-length brown hair styled with soft waves. They wear dangling earrings and a necklace, suggesting a formal or semi-formal occasion. Their expression is serious, with slightly parted lips as if mid-speech, indicating engagement in conversation. The posture appears upright, though only the head and shoulders are visible. The background is softly lit, featuring neutral tones that suggest an indoor setting, possibly a studio or event space. The overall appearance conveys professionalism and poise.
2	The individual has short, tousled brown hair and a stubbled beard, giving him a rugged appearance. His eyes are partially closed, and he has a slight smile, suggesting a relaxed or content mood. He is wearing a dark, casual shirt. The background is dimly lit, with indistinct shapes and colors, possibly indicating an evening event or gathering. His posture appears casual and confident.
3	The individual has short, graying hair with a receding hairline, a light beard, and a relaxed smile. His eyes are partially squinted, suggesting a friendly demeanor. He wears a black shirt and stands upright with a casual posture. The background features a blurred image of a desert landscape, possibly from a movie poster, indicating the setting might be a film-related event. The overall atmosphere is informal and approachable.
4	The individual has short, dark brown hair with a slight wave, styled neatly. Their face is oval-shaped with a strong jawline, and they have a calm, neutral expression. They are wearing a dark blue sweater over a white shirt. The posture is relaxed yet confident, leaning slightly forward. The background is blurred, suggesting an indoor setting with warm lighting, possibly a studio or home environment. The overall mood is serene and composed.
5	The individual wears a vibrant red and green patterned headwrap, complementing his blue shirt with a floral design. His face is adorned with a warm smile, showcasing a friendly demeanor. He has a robust build and appears to be in mid-conversation, suggesting an engaging personality. The blurred background hints at a public or social setting, possibly an event or gathering. His posture is upright, exuding confidence and approachability. The overall image conveys a sense of cultural pride and community involvement.
6	The individual has light skin, brown hair tied back, and is smiling widely, showing teeth, with eyes slightly squinted in joy. Their posture suggests movement, possibly dancing or celebrating, as one hand is raised near the face. The background is a bright blue sky, indicating an outdoor setting during daytime. The overall impression is one of happiness and spontaneity, captured in a candid moment.
7	The individual has shoulder-length blonde hair, a warm smile, and a friendly demeanor. Their posture is upright, exuding confidence. The background suggests an indoor setting with blurred elements, possibly a professional or social event. The person appears approachable and engaged, likely participating in a formal gathering or interview. Their attire seems professional, aligning with the setting's formality.
8	The individual has neatly combed dark hair, a sharp jawline, and a serious expression, suggesting contemplation or concern. He wears a gray blazer over a white shirt with a red tie, indicating formality. His posture is upright, exuding confidence. The background appears to be an indoor setting with soft lighting, possibly a studio or office, enhancing the focus on his face. The overall image conveys professionalism and intensity.
9	The individual has short, dark hair with a slight smile, showcasing a friendly demeanor. His eyes are bright and engaging, suggesting confidence. He wears a light blue shirt, indicating a casual setting. The background is blurred, with indistinct figures and warm lighting, hinting at a social or public event. His posture is upright, exuding ease and approachability. The overall impression is one of warmth and sociability in a relaxed environment.
10	The individual has shoulder-length blonde hair with subtle highlights, styled in soft waves. Their face is oval-shaped with a calm, composed expression, featuring light-colored eyes and a neutral mouth. They are wearing a dark top with a visible red stripe on the shoulder. The posture is upright, suggesting confidence or attentiveness. The background is blurred with blue tones, possibly indicating an indoor setting with professional lighting, emphasizing the subject as the focal point.

Table 16: Detailed textual description prompts (Prompt 1–10) adopted in our experiments.