ANCHORING REAL FACES TO PRETRAINED FEATURES FOR MULTI-DATASET DEEPFAKE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models that synthesize faces indistinguishable from real ones are proliferating rapidly. Deepfake detectors have kept pace by mining spatial and frequency artifacts, seeking cues shared across generators, and adapting large pretrained models. However, even when trained on multiple datasets, existing detectors learn which source each real face comes from: they score real faces from unseen sources closer to forgeries and reject many of them, even though their AUC remains high. In this work, we present **P**retrained-feature-**A**nchored **L**earning (PAL), which anchors each training real face to its frozen pretrained feature and learns the directions along which forgeries depart from that anchor. PAL requires only binary labels and attaches to a pretrained encoder without changing its architecture. On multi-dataset protocols, PAL accepts more real faces from unseen sources than existing detectors at a comparable AUC. As a by-product, a single threshold set halfway between the score of the anchor and those of the weakest validation forgeries, without any test data, yields a higher accuracy across benchmarks than existing detectors reach even with their best single threshold. We believe PAL eases the concern over rejecting unseen real faces and, by separating the real side from the forgery side, offers a first step toward detectors that generalize to unseen forgeries.

1 INTRODUCTION

Generative models have advanced to the point where synthesized faces are hard to distinguish from real ones, and their misuse for impersonation and disinformation has turned them into a real threat (Verdoliva, 2020). This has motivated efforts to make deepfake detection reliable beyond controlled settings (Chandra et al., 2026), including work on exposing forgery traces, improving generalization across manipulations, and reducing disparities across demographic groups (Qian et al., 2020; Li et al., 2020; Yan et al., 2023a; Ojha et al., 2023; Dong et al., 2023; Ju et al., 2024).

As deepfake generators continue to grow in number and diversity, however, refining a detector for the forgeries it has already seen is no longer enough, since it may encounter manipulations unlike those in its training set. The challenge thus shifts from optimizing for individual forgery types to training on sufficiently diverse forgery sources, and a natural way to widen this range is to draw on the many datasets the community has produced. Yet many detectors are developed and evaluated under protocols dominated by a limited set of datasets or manipulation sources (Rossler et al., 2019), which keeps the training distribution narrow and conceals how a model behaves once many sources are combined. To be trusted in practice, a detector should therefore learn from forgeries as diverse as possible, which naturally leads to training across multiple datasets (Wang et al., 2025b; Yang et al., 2025), while flagging as fake only what this evidence supports and keeping inputs it has no evidence against, such as genuine faces from unseen sources, on the real side.

Training across multiple datasets, however, reveals an unexpected problem. We observe that, although its overall AUC remains high, an existing detector often assigns high fake scores to real faces from sources unseen during training. This is because not only fakes but also real faces carry source-specific fingerprints (Appendix A.3), allowing the detector to use the source of a real face as a shortcut for classification. The paired construction of deepfake datasets can further exacerbate this problem, and as datasets are added, the diversity of forgeries expands rapidly while that of real sources grows much more slowly. The failure has gone largely unnoticed because generalization

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

is framed around unseen forgeries: real faces from unseen sources enter evaluation mainly through AUC, which is blind to where the threshold falls, and accuracy on forgery-dominated test sets credits a detector that calls every unfamiliar face fake. A detector saturated toward the fake side can thus look strong on paper while rejecting the genuine faces that make up most of its inputs in practice.

We argue that this problem stems from treating real faces the same way as fake ones during training. When the real class is learned only from the training data, nothing prevents the detector from modeling real faces through whatever separates them in that data, and dataset origin is often the easiest such signal. This raises the question: can a detector keep learning from ever more forgery sources while keeping real faces from unseen sources on the real side, using only the protocol's training data, only real/fake labels, and no test data? Our answer starts from the observation that real faces need not be learned from the training data alone, since a pretrained encoder already provides a stable representation of real facial content. We therefore propose **P**retrained-feature-**A**nchored **L**earning (**PAL**), which, rather than learning "real" anew, anchors each training real face to its frozen pretrained feature and learns only how forgeries depart from that anchor. Constraining the real side in this way leaves the detector less room to encode source-specific information through the real class and directs it toward how forgeries deviate from real faces. PAL requires only binary labels and leaves the encoder architecture unchanged, so it can be attached to existing detection pipelines at no annotation cost.

We validate PAL on multiple multi-dataset protocols and pretrained backbones, and find that it accepts more real faces from unseen sources than existing detectors at a comparable AUC, most clearly where real sources are scarce relative to forgery sources. As a by-product, PAL also pays off at deployment. The threshold that best separates real from fake differs from one dataset to another, so a threshold tuned on one dataset rarely transfers to the next; yet with a single threshold fixed from validation forgeries alone, without any test data, PAL reaches a higher accuracy across benchmarks than existing detectors attain even with their best single threshold. By separating the real side from the forgery side, PAL thus offers a first step toward detectors that scale with the number of forgery sources.

Our main contributions are summarized as follows:

- We show that deepfake detectors trained across multiple datasets learn to recognize the source of real faces and reject real faces from unseen sources while their AUC stays high, a failure that common evaluation leaves unpenalized.
- We propose PAL, which anchors each training real face to its frozen pretrained feature and learns only how forgeries depart from it, so that a face is flagged as fake only when it departs in the ways seen forgeries do and otherwise stays on the real side.
- On multi-dataset protocols, PAL accepts more real faces from unseen sources than existing detectors at a comparable AUC and, as a by-product, with a single threshold fixed from validation forgeries alone, reaches a higher accuracy across benchmarks than they attain with their best single threshold.

2 Related Work

2.1 Task-Specific Deepfake Detection

Early face forgery detectors relied on generic CNN classifiers (Chollet, 2017; Rossler et al., 2019) that collapsed on unseen manipulations. Subsequent work improved generalization by mining frequency-domain cues (Qian et al., 2020; Liu et al., 2021) and blending traces (Li et al., 2020; Shiohara & Yamasaki, 2022), yet these cues remain tied to the manipulation types seen during training. Rather than modeling such forgery cues directly, reconstruction-based methods instead characterize the distribution of real faces and flag deviations from it (Cao et al., 2022). Another line reduces this dependence at the representation level, disentangling forgery-related features to broaden the decision boundary (Yan et al., 2023a; 2024a). Across these directions, however, representations learned solely from task-specific forgery data remain bound to the training distribution and transfer poorly to unseen forgeries, motivating a turn toward the capacity of large-scale pre-trained models.

2.2 Foundation Model-based Detection

Large-scale pre-trained representations are increasingly adopted to leverage transferable visual knowledge and improve generalization beyond the training distribution (Ojha et al., 2023; Liu et al., 2024; Tan et al., 2025; Cui et al., 2025). Some approaches adapt these representations for forgery detection, which may distort pre-trained semantic knowledge and overfit to limited forgery patterns. Yan et al. (2025) address this issue by preserving the principal components of the pre-trained representation while constraining forgery-related learning to the orthogonal subspace. An alternative direction learns transferable facial representations from large-scale real-face data through self-supervised pre-training (Wang et al., 2025a). However, detectors can still exploit spurious training cues, with studies exploring data composition and learning strategies, such as source-consistent real-fake pairs (Yermakov et al., 2026) and token-level shuffling and mixing (Fu et al., 2025).

2.3 Scalable Deepfake Detection

Detectors trained on a narrow range of manipulations collapse on unseen manipulations and domains, revealing their dependence on limited training diversity (Tan et al., 2026). This has motivated efforts to scale detection across more diverse sources, and detection error has indeed been shown to shrink predictably as forgery diversity grows (Wang et al., 2025b). Yet, simply adding more generators can widen the in-domain/out-of-domain gap, which prior work addresses by learning generator-agnostic artifacts from image discrepancies (Yang et al., 2025). Pushing this further, scaling across heterogeneous datasets can instead degrade performance, and SD^2 (Jung et al., 2026) addresses this by unifying benchmarks and incorporating text-based forgery knowledge.

3 Methodology

Real data carry source fingerprints as fakes do, and the real sources met in deployment form an open set of which training covers few: each real video is reused by many manipulation methods, so fakes far outnumber reals (up to 18:1 in our pools) and detectors overfit to the real sources they see. We therefore take a pretrained representation, rather than the few training sources, as the reference for real and learn forgery only as a departure from it, so that a fake decision is meant to rest on forgery evidence, not on source unfamiliarity.

3.1 Fingerprints of real and fake data

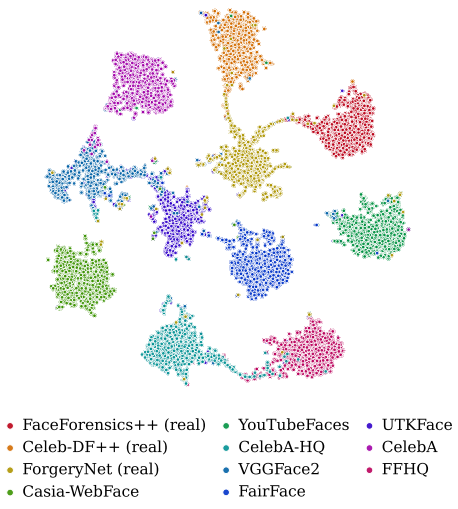

Figure 1: t-SNE of the 11 real sources.

In deepfake research, each generative model is well known to leave its own characteristic artifacts, but much less attention has been paid to the real side, even though detection performance changes markedly when the real test data are collected from different sources. Liu & He (2025) showed that modern networks can still tell which dataset an image comes from, even among today's large-scale and diverse datasets, so scale and diversity alone do not remove dataset-specific signatures. Following their setup, we train ConvNeXt V2-Nano (Woo et al., 2023) to classify 11 real sources and, separately, 52 forgery sources. It reaches 93.7% and 82.5% accuracy, respectively (chance: 9.1% and 1.9%), so not only the fakes but also the reals consist of clearly distinguishable sources, which suggests that the real class carries many latent fingerprints. Binary training treats real as a single class, and some detectors add terms that explicitly pull same-class features together; either way, the decision boundary risks responding to the fingerprints of particular sources rather than to forgery itself.

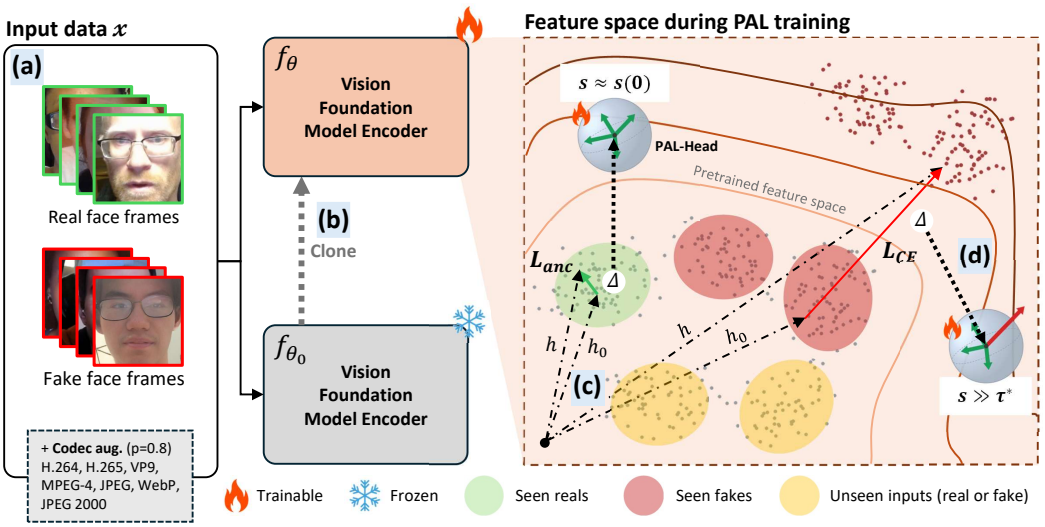

Figure 2: Overview of PAL. (a) First, input frames x are re-encoded during training by one of seven codecs with probability 0.8. (b) Next, the trainable f_θ is cloned from the frozen pretrained f_{θ_0}. (c) Then, in the feature space during training, $\mathcal{L}_{\text{anc}}$ keeps seen reals near their pretrained features ($h \approx h_0$), while $\mathcal{L}_{\text{CE}}$ lets seen fakes leave the pretrained space (red dots). For unseen inputs, reals are meant to stay real, and fakes are caught only if they depart along a learned direction. (d) Finally, the PAL head reads $\Delta = h - h_0$ along K learned forgery directions ($K = 4$ shown, $K = 8$ used), so a real shows no evidence ($s \approx s(0)$) while a fake exceeds the threshold ($s \gg \tau^*$).

3.2 Anchoring Training Reals for Unseen Real Sources

Learning to separate the fakes should move the training reals as little as possible from their pretrained features, since otherwise reals from unseen sources risk being rejected for lacking the fingerprints of the training real sources. For a face frame x, let $h = f_\theta(x)$ and $h_0 = f_{\theta_0}(x)$ in $\mathbb{R}^d$ be the features from the fine-tuned encoder f_θ and its frozen pretrained copy f_{θ_0}; we define the *departure*

$$\Delta(x) \;=\; h(x) - h_0(x), \tag{1}$$

and tie each training real to its pretrained feature by penalizing its departure. With binary labels $y \in \{0, 1\}$ (0: real, 1: fake), the anchor loss is

$$\mathcal{L}_{\text{anc}} \;=\; \frac{\mathbb{E}_{x:\, y=0} \, \|\Delta(x)\|_2^2}{\mathbb{E}_{x:\, y=0} \, \|h_0(x)\|_2^2}, \tag{2}$$

where both expectations run over the reals of a mini-batch and the normalization makes the loss independent of the backbone's feature scale. Training minimizes

$$\mathcal{L} \;=\; \mathcal{L}_{\text{CE}} \;+\; \lambda \, \mathcal{L}_{\text{anc}}, \qquad \mathcal{L}_{\text{CE}} \;=\; -\,\mathbb{E}\big[\, y \log \sigma(s) + (1 - y) \log \big(1 - \sigma(s)\big) \,\big], \tag{3}$$

where σ is the logistic function and $s(x)$ the forgery score computed from $\Delta(x)$ by the head defined next. Both terms use only the binary label y; generator labels and real-source names are never used. We use $\lambda = 1$ for every backbone and select the model by in-domain validation AUC only.

Because each training real is tied to its own pretrained feature, the anchor does not pull the training reals into one cluster, outside which any real, including one from an unseen source, would risk looking fake. The anchor penalizes only training reals, so the encoder may move the fakes freely as long as those reals stay near their pretrained features.

3.3 Scoring Departures for Unseen Forgeries

With the training reals anchored, the head reads forgery evidence from the departure itself. Its score exceeds $s(\Delta{=}0)$ only when x departs along a direction learned from the training forgeries, so an

unseen forgery yields evidence only if it departs as they do. Given K learned forgery directions $u_1, \ldots, u_K \in \mathbb{R}^d$ ($K = 8$ by default) and $\hat{u}_k = u_k / \|u_k\|$, the head computes

$$r_k(x) \;=\; \hat{u}_k^\top \Delta(x), \qquad s(x) \;=\; g \, \log \sum_{k=1}^{K} \exp\big(r_k(x)\big) \;+\; b, \tag{4}$$

with learnable scalars g and b initialized to $g = 1$ and $b = -\log K$, so that $s(\Delta{=}0)$ starts at 0; the two-class logits $[\,0,\, s(x)\,]$ give $p(\text{fake} \mid x) = \sigma(s(x))$. Every 200 mini-batches, a direction whose r_k was the largest for fewer than 1% of the fakes since the last check is replaced by the normalized departure of a fake randomly selected from the current mini-batch.

By equation 4, cross-entropy separates the classes only through the evidence read along the K directions, so the encoder learns to make the departure from the frozen features a measure of forgery evidence, and a real is meant to be recognized by its absence rather than by resemblance to the training reals. Fakes are pushed far along these directions, whereas training reals can only be shifted slightly against them within the anchor's slack, so that only seen forgeries leave the frozen feature space and the decision stays anchored to the pretrained space. Unseen reals receive this shift only in part and spread along the forgery directions, so the score still separates seen from unseen reals. When the lower tail of the validation forgeries lies above $s(\Delta{=}0)$, so does τ^*, and forgeries that barely depart or depart off all learned directions are missed, which delimits the method's scope.

3.4 Fixing one threshold for unseen datasets

So far we have anchored the training reals and scored each input by its departure along the learned forgery directions. Here we fix the single threshold that turns this score into a decision. Since the best threshold differs across datasets and no data from an unseen dataset may be used, we set one threshold after training and apply it unchanged to every test set, accepting that it cannot be the best for each. It lies between two reference scores. The first is the score of an input with no departure, which follows in closed form from the head because every r_k in equation 4 vanishes at $\Delta = 0$,

$$s_0 \;=\; s(\Delta{=}0) \;=\; g \log K + b. \tag{5}$$

The second is the 0.05-quantile $Q_{0.05}(S_f)$ of the scores S_f of the validation forgeries, pooled without regard to source, which marks the weakest seen forgeries. The threshold is their midpoint,

$$\tau^* \;=\; \tfrac{1}{2}\big[\, s_0 + Q_{0.05}(S_f) \,\big], \tag{6}$$

and x is declared fake if and only if $s(x) \geq \tau^*$.

The anchor keeps the training reals near $\Delta = 0$ and hence slightly below s_0, whereas on unseen data the two classes move toward each other. Unseen reals drift upward from the training reals, many of them past s_0, and many unseen forgeries depart less along the learned directions than seen ones and fall below the seen-forgery tail. The midpoint leaves the same room beyond each reference score without assuming which drift is larger. Since the rule only sets where the score is cut, the ranking and every AUC of the head are unchanged, and folding τ^* into the offset, $b \leftarrow b - \tau^*$, lets the detector's own rule $p(\text{fake} \mid x) \geq 0.5$ apply it on every test set without test-time adjustment.

4 Experiment

4.1 Experimental Setup

Datasets. We train on DF40 (Yan et al., 2024b), Celeb-DF++ (Li et al., 2025), DeepSpeak v2 (Barrington et al., 2026), and FaceForensics++ (FF++) (Rossler et al., 2019), which form three settings with an increasing number of real sources: Restricted (FF++ reals and DF40 forgeries on FF++), Default (adding Celeb-DF++), and Extended (adding DeepSpeak v2 and FF++ forgeries). We evaluate on five datasets from sources unseen in training, UTKFace (Zhang et al., 2017), RedFace (Shi et al., 2025), Deepfake-Eval-2024 (DFE24) (Chandra et al., 2026), DFDC (Dolhansky et al., 2020), and FFHQ-Fake, an identity-controlled benchmark we build from FFHQ (Karras et al., 2019) (Appendix A.4).

Baselines. We compare with Xception (Chollet, 2017), F3Net (Qian et al., 2020), RECCE (Cao et al., 2022), UCF (Yan et al., 2023a), UDD (Fu et al., 2025), FSFM (Wang et al., 2025a), GenD (Yermakov et al., 2026), and Effort (Yan et al., 2025), each retrained in every setting with our data, preprocessing, and augmentation (Appendix A.2). As a same-backbone control, we also report full fine-tuning of our backbone with a linear head, that is, PAL without the anchor and the head.

Implementation details. We fine-tune all layers of PE-Core B/16 (Bolya et al., 2026) (93.7M parameters), taking its attention-pooled 1024-dimensional output as h and that of a frozen copy as h_0, with $K = 8$ randomly initialized directions and $\lambda = 1$. We use AdamW (Loshchilov & Hutter, 2017) (learning rate 10^{-5}, weight decay 10^{-4}, cosine decay, gradient clipping at norm 1) on mini-batches of 32 real and 32 fake frames with 4-step accumulation and 50,000 frames per epoch. We train for 15 epochs (early-stopping patience 3) in Restricted and 30 (patience 5) otherwise, and keep the checkpoint with the highest AUC on 20,000 in-domain validation frames.

Faces are cropped around FaceXFormer (Narayan et al., 2025) landmarks. Because training mixes video and image sources, every training frame, real or fake, is re-encoded with probability 0.8 by one of seven image or video codecs, so that compression cannot act as a shortcut (Appendix A.1).

Evaluation metrics. All metrics are per frame. We report threshold-free AUC and thresholded accuracy, for which each model applies one threshold τ, chosen on validation data instead of the conventional 0.5, to every test source. Baselines take the τ that maximizes the balanced accuracy averaged over validation sources, and PAL uses its own rule (Section 3.4). We also report R_{real}, the fraction of real faces scored below τ, and R_{fake}, the fraction of fakes scored at or above it.

4.2 Evaluation on Restricted and Default Settings

We first compare the Restricted and Default settings, grouping FF, CDF, and FFHQ-Fake as in-domain sets, whose forgeries come wholly or partly from generators seen in training, and RedFace, DFDC, and DFE24 as out-domain sets, with the real-only UTKFace reported separately. We ask whether detectors accept real faces from unseen sources while still separating them from forgeries.

PAL accepts more unseen real faces than the baselines, most clearly in the Restricted setting with a single real source, and adding Celeb-DF++ narrows the gap mainly by raising the baselines, which suggests they treat unfamiliar reals as fake (Table 1). One exception is out-domain R_{fake}, which is lower for PAL under its own threshold. Applying the baseline threshold rule brings it into the baseline range without changing AUC, indicating an operating-point rather than ranking difference. Overall, PAL is less sensitive to real-source shifts while keeping real–fake separation across domains.

4.3 Evaluation on Extended Setting

We next ask whether PAL's behavior persists as more real sources are added, which should also help the baselines accept unseen real faces. The Extended setting adds FF++ forgeries and DeepSpeak v2 to the Default setting, giving three real sources, and uses the same test sets as Table 1.

PAL keeps the best AUC on both groups and the highest in-domain real acceptance (Table 2). Unlike before, Effort accepts more real faces on UTK and out-domain sets but detects fewer out-domain fakes. Its small residual update may keep CLIP features near pretrained ones, letting three real sources broaden its real region. PAL accepts fewer of these faces partly because its threshold tightens as compressed and lip-synced fakes enter validation. Threshold sweeping removes this factor (Figure 3c, f). PAL then stays above the baselines almost everywhere, even at Effort's real acceptance. Thus, PAL retains the better trade-off, though its unseen-real advantage shrinks as real sources increase.

4.4 PAL on Other Encoders and Existing Detectors

To test whether PAL depends on PE-Core, we train it on five other encoders, CLIP ViT-B/16 and ViT-L/14 (Radford et al., 2021), DINOv3 (Siméoni et al., 2025), FSFM (Wang et al., 2025a), and Xception (Chollet, 2017), and compare each with *pure*, the same encoder fine-tuned with a linear head (Table 3). PAL raises out-domain accuracy on all six encoders shown with little in-domain

Table 1: Detection performance in the Restricted and Default settings. PAL (val. thr.): PAL at the baselines' validation-optimal threshold, not ranked. **Bold**: best, underline: second best per column within each setting.

		UTK	in-domain avg.			out-domain avg.		
Method	τ	R_{real}	AUC	R_{real}	R_{fake}	AUC	R_{real}	R_{fake}
Restricted setting								
Xception	0.125	0.031	0.870	0.417	0.974	0.568	0.230	**0.821**
F3Net	0.222	0.085	0.817	0.449	0.959	0.583	0.261	0.794
RECCE	0.442	0.083	0.841	0.451	0.945	0.568	0.293	0.733
UCF	0.062	0.054	0.845	0.436	0.954	0.556	0.248	0.782
UDD	0.271	0.037	0.914	0.469	**0.986**	0.624	0.292	0.818
FSFM	0.980	0.315	0.902	0.649	0.897	0.634	0.430	0.703
GenD	0.293	0.166	0.887	0.615	0.914	0.581	0.305	0.759
Effort	0.526	0.139	0.931	0.610	0.963	0.647	0.382	0.749
PAL	0.988	**0.647**	**0.956**	**0.750**	0.918	**0.691**	**0.785**	0.485
PAL (val. thr.)	0.512	0.253	0.956	0.617	0.981	0.691	0.461	0.752
Default setting								
Xception	0.806	0.315	0.914	0.765	**0.916**	0.611	0.373	**0.718**
F3Net	0.160	0.261	0.879	0.740	0.893	0.583	0.377	0.698
RECCE	0.398	0.445	0.885	0.848	0.813	0.581	0.521	0.579
UCF	0.296	0.407	0.887	0.842	0.826	0.556	0.447	0.590
UDD	0.369	0.453	0.945	0.841	0.913	0.652	0.487	0.663
FSFM	0.975	0.694	0.934	**0.949**	0.821	0.647	0.664	0.560
GenD	0.380	0.581	0.939	0.903	0.853	0.614	0.536	0.591
Effort	0.481	0.676	0.940	0.880	0.877	0.690	0.716	0.547
PAL	0.928	**0.848**	**0.968**	0.945	0.891	**0.698**	**0.796**	0.502
PAL (val. thr.)	0.509	0.658	0.968	0.879	0.944	0.698	0.597	0.649

Table 2: Detection performance in the Extended setting.

		UTK	in-domain avg.			out-domain avg.		
Method	τ	R_{real}	AUC	R_{real}	R_{fake}	AUC	R_{real}	R_{fake}
Xception	0.375	0.283	0.892	0.713	**0.917**	0.629	0.313	**0.805**
F3Net	0.408	0.338	0.869	0.754	0.866	0.600	0.371	0.722
RECCE	0.675	0.409	0.801	0.572	0.865	0.575	0.372	0.706
UCF	0.288	0.512	0.852	0.741	0.837	0.643	0.492	0.698
UDD	0.453	0.545	0.937	0.838	0.890	0.703	0.524	0.723
FSFM	0.379	0.536	0.939	0.880	0.901	0.695	0.521	0.714
GenD	0.386	0.616	0.930	0.813	0.874	0.655	0.557	0.619
Effort	0.452	**0.866**	0.947	0.903	0.856	0.706	**0.754**	0.542
PAL	0.797	0.798	**0.975**	**0.955**	0.910	**0.713**	0.656	0.622
PAL (val. thr.)	0.643	0.726	0.975	0.926	0.937	0.713	0.559	0.689
Veritas[†]		0.327	–	–	–	0.670	0.606	0.742

[†] Reference only, not ranked. Released checkpoint of Veritas (Tan et al., 2026), a reasoning-based detector, with its own preprocessing and prompt, as its SFT→MiPO→P-GRPO training exceeds our computational resources. Lower accuracy with our tighter face crop.

change, and PE-Core gives the highest out-domain accuracy. The gain is smallest on FSFM, likely because its face-only pretraining leaves less general knowledge to preserve, and on PE-Core and FSFM out-domain AUC dips slightly, so their gains lie in the operating point.

We then add PAL to existing foundation-model detectors, replacing only their classification head and adding the anchor to the loss (Table 4). PAL raises out-domain accuracy for all four and every average for Effort and GenD. FSFM changes little, as at the encoder level, and UDD loses some AUC, possibly because its original loss terms are removed. PAL thus works across encoders and as a drop-in component for existing detectors.

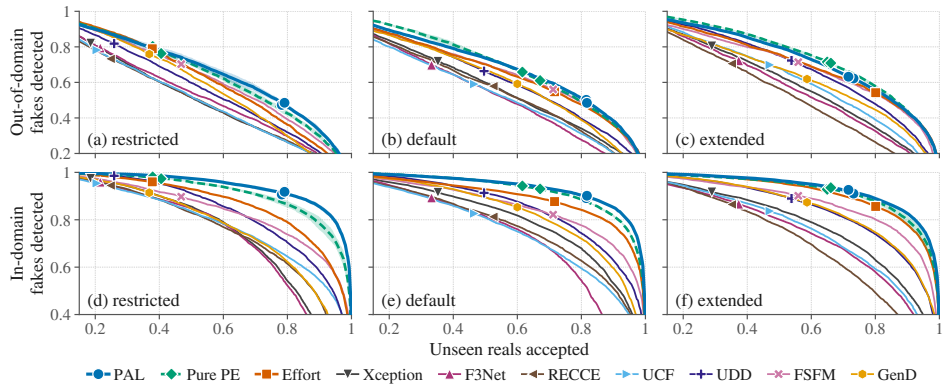

Figure 3: Real-fake trade-off in the three settings. Each curve sweeps one detector's threshold; markers show the deployed thresholds. PAL and pure PE: mean of two seeds, seed range shaded.

4.5 Ablation Study

We ablate the anchor and the head (Table 5). With FF and CDF nearly saturated, the differences lie in FFHQ-Fake, whose real faces are unseen. The head alone hurts, accepting far fewer real faces, since departures from pretrained features indicate forgery only when the anchor keeps training reals near them. The anchor alone helps slightly, and both together give the best FFHQ-Fake and in-domain results, so the head needs the anchor. Appendices A.9 and A.8 cover λ, K, and robustness.

4.6 Preserving only the reals versus both classes

WiSE-FT (Wortsman et al., 2022) interpolates the fine-tuned weights toward the pretrained ones, $\theta = \alpha\theta_{\text{ft}} + (1-\alpha)\theta_0$, and Effort (Yan et al., 2025) tunes only a low-rank residual of the pretrained weights. Both hold the pretrained model for reals and forgeries alike, whereas PAL holds only the reals. Table 6 compares them under the same threshold rule. Unlike in WiSE-FT, where the zero-shot model already solves the task, the frozen model here cannot separate forgeries, so Acc rises steadily with α and Effort also stays below full fine-tuning. Holding the forgeries holds back what separates them from reals, whereas PAL, which holds only the reals, keeps the separation of full fine-tuning in every setting and has the highest Acc in Restricted and Default settings.

Table 3: PAL on other pretrained encoders (Default setting). First rows fine-tune the encoder with a linear head only (*pure*), with thresholds as in Table 4. Shaded: our encoder.

		in-domain avg.		out-domain avg.	
Encoder	τ	Acc.	AUC	Acc.	AUC
PE-Core	0.470	0.907	0.963	0.626	**0.702**
+ PAL	0.928	**0.918**	**0.968**	**0.649**	0.698
CLIP-B16	0.553	0.882	0.934	0.576	0.627
+ PAL	0.675	**0.889**	**0.944**	**0.598**	**0.645**
CLIP-L14	0.729	**0.927**	0.969	0.612	0.681
+ PAL	0.914	0.919	**0.970**	**0.631**	**0.687**
DINOv3	0.419	0.885	0.950	0.617	0.669
+ PAL	0.420	**0.894**	**0.952**	**0.629**	**0.678**
FSFM	0.373	**0.881**	**0.934**	0.618	**0.659**
+ PAL	0.485	0.873	0.931	**0.623**	0.655
Xception	0.431	**0.847**	0.906	0.533	0.570
+ PAL	0.482	0.838	**0.907**	**0.545**	**0.592**

Table 4: PAL on existing detectors (Default setting). Original rows use the validation-optimal threshold, +PAL rows the rule of PAL. **Bold**: better of the two rows.

		in-domain avg.		out-domain avg.	
Method	τ	Acc.	AUC	Acc.	AUC
FSFM	0.975	**0.885**	**0.934**	0.612	0.647
+ PAL	0.985	0.878	0.934	**0.622**	**0.657**
Effort	0.481	0.878	0.940	0.632	0.690
+ PAL	0.536	**0.893**	**0.947**	**0.639**	**0.705**
UDD	0.369	0.877	**0.945**	0.575	**0.652**
+ PAL	0.656	**0.884**	0.938	**0.578**	0.633
GenD†	0.338	0.904	0.960	0.585	0.669
+ PAL	0.960	**0.919**	**0.975**	**0.645**	**0.689**

† Our GenD baseline trains only its head (Table 8). As the anchor constrains the full feature, we unfreeze all layers in both rows.

Table 5: Removing components of PAL (Default setting). Rows without the head use a linear head. All rows use the threshold rule of PAL, with $s_0 = 0$ for rows without the head.

Components		FF			CDF			FFHQ-Fake			in-domain avg.	
Anchor	Head	AUC	R_{real}	R_{fake}	AUC	R_{real}	R_{fake}	AUC	R_{real}	R_{fake}	Acc.	AUC
✗	✗	**0.998**	**1.000**	**0.928**	**0.993**	**0.998**	0.907	0.899	0.807	0.845	0.914	0.963
✓	✗	0.997	**1.000**	0.923	0.992	0.993	0.905	0.908	0.790	**0.865**	0.913	0.966
✗	✓	0.996	0.996	0.916	0.991	0.988	**0.915**	0.840	0.596	0.854	0.878	0.942
✓	✓	0.997	0.999	0.926	0.992	0.996	0.893	**0.915**	**0.839**	0.855	**0.918**	**0.968**

Table 6: Preserving the pretrained model for both classes versus for reals only. All rows use the threshold rule of PAL, with $s_0 = 0$ for rows without its head.

		Restricted		Default		Extended	
Method	Preserved	Acc	AUC_{in}	Acc	AUC_{in}	Acc	AUC_{in}
Frozen PE + linear probe ($\alpha = 0$)	real + fake	0.633	0.827	0.699	0.864	0.655	0.876
WiSE-FT ($\alpha = 0.5$)	real + fake	0.656	0.808	0.717	0.898	0.710	0.879
Effort	real + fake	0.684	0.931	0.751	0.940	0.760	0.947
Full fine-tuning ($\alpha = 1$)	none	0.721	0.944	0.775	0.963	**0.791**	0.965
PAL (ours)	real	**0.735**	**0.956**	**0.783**	**0.968**	0.786	**0.975**

5 Discussion

PAL changes how the backbone is trained, not its input, so its separation is bounded by the backbone. On PE-Core B/16, out-domain AUC stays near 0.7, and Acc stays within 0.014 of full fine-tuning, falling 0.005 below it in the Extended setting (Table 6). PAL thus mainly shifts where errors fall, trading out-domain R_{fake} for R_{real} against most baselines. This lead on the real side narrows in the Extended setting and falls behind FSFM and Effort under heavy degradation (Figure 5), and trained on FF++ alone PAL misses many forgeries (Appendix A.7).

PAL flags an unseen forgery only when it departs along a direction learned from seen forgeries. Since the number of directions barely affects out-domain accuracy (Appendix A.9), the bottleneck lies in the input and representation rather than the head. Signals from several layers or beyond the RGB frame, learned from more diverse forgeries, could widen this evidence.

With PE-Core L/14, Acc and out-domain AUC keep rising up to the Extended setting, where S/16 and B/16 drop, suggesting that capacity and data should grow together (Appendix A.6). Preserving the pretrained model for both classes falls behind full fine-tuning, whereas preserving only the reals keeps pace with it (Section 4.6), so the two classes are best treated differently.

6 Conclusion

Detectors trained across multiple datasets learn the source of real faces and reject unseen-source reals while their AUC stays high. We presented PAL, which anchors each training real face to its frozen pretrained feature and learns forgery as a departure from it, so that a face is called fake only on evidence from seen forgeries and is otherwise treated as real. With a simple head on a strong pretrained backbone, PAL reached the highest AUC among existing detectors in every setting, accepted more unseen real faces when real sources were few, and, with one threshold fixed without test data, exceeded the baselines at their best single threshold. It also raised out-domain accuracy on most other backbones and four existing detectors. As forgery sources multiply while unfamiliar real faces dominate deployment, we hope the field treats the two classes differently and judges detectors by the real faces they reject as well as the forgeries they catch.

Reproducibility Statement

The paper specifies the backbone, hyperparameters, training settings, and threshold rule, and every benchmark except FFHQ-Fake is publicly available, with the construction of FFHQ-Fake described in full. The threshold follows in closed form from the trained head and the validation forgeries, so it can be recomputed from any checkpoint. Upon acceptance, we will release the code, configurations, data splits, and trained checkpoints for PAL and all baselines.

Ethics Statement

This work aims to reduce false alarms on authentic media, which can wrongly discredit real people, while keeping forgeries detectable. All training and test data come from public benchmarks used under their licenses, and no new data were collected from people. FFHQ-Fake, which we synthesize from the public FFHQ dataset, is used only for internal evaluation and will not be released, to avoid distributing forged faces of real individuals.

AI Use Statement

We used Claude Opus 5.5 as a writing assistant for rephrasing and grammar and as a coding assistant for implementing and debugging experiment code. All experiments were run on the authors' hardware under their direction, every reported number comes from these runs, and the authors verified all results and text and take full responsibility for the paper.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Sarah Barrington, Maty Bohacek, and Hany Farid. The deepspeak dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1893–1902, 2026.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, et al. Perception encoder: The best visual embeddings are not at the output of the network. *Advances in Neural Information Processing Systems*, 38:60884–60937, 2026.

Stella Bounareli, Christos Tzelepis, Vasileios Argyriou, I. Patras, and Georgios Tzimiropoulos. Hyperreenact: One-shot reenactment via jointly learning to refine and retarget faces. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7115–7125, 2023. URL https://api.semanticscholar.org/CorpusID:259991520.

Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4113–4122, June 2022.

Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pp. 67–74. IEEE, 2018.

Nuria Alina Chandra, Hannah Lee, Ryan Murtfeldt, Lin Qiu, Arnab Karmakar, Emmanuel Tanumihardja, Kevin Farhat, Ben Caffee, Changyeon Lee, Jongwook Choi, et al. Deepfake-eval-2024: A multi-modal in-the-wild benchmark of deepfakes circulated in 2024. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10668–10678, 2026.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for

photorealistic text-to-image synthesis. In *International conference on learning representations*, volume 2024, pp. 57611–57640, 2024.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Xinjie Cui, Yuezun Li, Ao Luo, Jiaran Zhou, and Junyu Dong. Forensics adapter: Adapting clip for generalizable face forgery detection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19207–19217, 2025. doi: 10.1109/CVPR52734.2025.01789.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4685–4694, 2019. doi: 10.1109/CVPR.2019.00482.

Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020.

Shichao Dong, Jin Wang, Renhe Ji, Jiajun Liang, Haoqiang Fan, and Zheng Ge. Implicit identity leakage: The stumbling block to improving deepfake detection generalization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3994–4004. IEEE, 2023.

Xinghe Fu, Zhiyuan Yan, Taiping Yao, Shen Chen, and Xi Li. Exploring unbiased deepfake detection via token-level shuffling and mixing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3040–3048, 2025.

Alexander Groshev, Anastasia Maltseva, Daniil Chesakov, Andrey Kuznetsov, and Denis Dimitrov. Ghost—a new face swap approach for image and video domains. *IEEE Access*, 10:83452–83462, 2022. doi: 10.1109/ACCESS.2022.3196668.

Jianzhu Guo, Dingyun Zhang, Xiaoqiang Liu, Zhizhou Zhong, Yuan Zhang, Pengfei Wan, and Di Zhang. Liveportrait: Efficient portrait animation with stitching and retargeting control. *arXiv preprint arXiv:2407.03168*, 2024.

Yinan He, Bei Gan, Siyu Chen, Yichun Zhou, Guojun Yin, Luchuan Song, Lu Sheng, Jing Shao, and Ziwei Liu. Forgerynet: A versatile benchmark for comprehensive forgery analysis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4358–4367. IEEE, 2021.

Yan Ju, Shu Hu, Shan Jia, George H Chen, and Siwei Lyu. Improving fairness in deepfake detection. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4643–4653. IEEE, 2024.

Inho Jung, Hyeongjun Choi, Minh Binh Le, Hohyun Na, and Simon Woo. A rich knowledge space for scalable deepfake detection. In *International Conference on Learning Representations*, volume 2026, pp. 96506–96535, 2026.

Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1548–1558, 2021.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4396–4405. IEEE, 2019.

Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5001–5010, 2020.

Yuezun Li, Delong Zhu, Xinjie Cui, and Siwei Lyu. Celeb-df++: A large-scale challenging video deepfake benchmark for generalizable forensics. *arXiv preprint arXiv:2507.18015*, 2025.

Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai 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*, pp. 772–781, 2021.

Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10770–10780, 2024.

Zhuang Liu and Kaiming He. A decade's battle on dataset bias: Are we there yet? In *International Conference on Learning Representations*, volume 2025, pp. 41433–41449, 2025.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11966–11976. IEEE, 2022.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Kartik Narayan, Vibashan VS, Rama Chellappa, and Vishal M Patel. Facexformer: A unified transformer for facial analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11369–11382, 2025.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24480–24489, 2023.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2085–2094, October 2021.

Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pp. 86–103. Springer, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1–11, 2019.

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. *ACM SIGGRAPH 2022 Conference Proceedings*, 2022. URL `https://api.semanticscholar.org/CorpusID:246441861`.

Junyu Shi, Minghui Li, Junguo Zuo, Zhifei Yu, Yipeng Lin, Shengshan Hu, Ziqi Zhou, Yechao Zhang, Wei Wan, Yinzhe Xu, et al. Towards real-world deepfake detection: A diverse in-the-wild dataset of forgery faces. *arXiv preprint arXiv:2510.08067*, 2025.

Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18720–18729, 2022.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Chuangchuang Tan, Renshuai Tao, Huan Liu, Guanghua Gu, Baoyuan Wu, Yao Zhao, and Yunchao Wei. C2p-clip: Injecting category common prompt in clip to enhance generalization in deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7184–7192, 2025.

Hao Tan, Zichang Tan, Senyuan Shi, Ajian Liu, Chuanbiao Song, Huijia Zhu, Weiqiang Wang, Jun Wan, Zhen Lei, et al. Veritas: Generalizable deepfake detection via pattern-aware reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 66420–66477, 2026.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on graphics (TOG)*, 40(4):1–14, 2021.

Luisa Verdoliva. Media forensics and deepfakes: an overview. *IEEE journal of selected topics in signal processing*, 14(5):910–932, 2020.

Gaojian Wang, Feng Lin, Tong Wu, Zhenguang Liu, Zhongjie Ba, and Kui Ren. Fsfm: A generalizable face security foundation model via self-supervised facial representation learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24364–24376. IEEE, 2025a.

Wenhao Wang, Longqi Cai, Taihong Xiao, Yuxiao Wang, and Ming-Hsuan Yang. Scaling laws for deepfake detection. *arXiv preprint arXiv:2510.16320*, 2025b.

Yuhan Wang, Xu Chen, Junwei Zhu, Wenqing Chu, Ying Tai, Chengjie Wang, Jilin Li, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Hififace: 3d shape and semantic prior guided high fidelity face swapping. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1136–1142. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/157. URL https://doi.org/10.24963/ijcai.2021/157. Main Track.

Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pp. 529–534. IEEE, 2011.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16133–16142. Ieee, 2023.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7949–7961. IEEE, 2022.

Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22412–22423, 2023a.

Zhiyuan Yan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. Deepfakebench: A comprehensive benchmark of deepfake detection. *arXiv preprint arXiv:2307.01426*, 2023b.

Zhiyuan Yan, Yuhao Luo, Siwei Lyu, Qingshan Liu, and Baoyuan 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*, pp. 8984–8994, 2024a.

Zhiyuan Yan, Taiping Yao, Shen Chen, Yandan Zhao, Xinghe Fu, Junwei Zhu, Donghao Luo, Chengjie Wang, Shouhong Ding, Yunsheng Wu, et al. Df40: Toward next-generation deepfake detection. *Advances in Neural Information Processing Systems*, 37:29387–29434, 2024b.

Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable ai-generated image detection, 2025. URL https://arxiv.org/abs/2411.15633.

Yongqi Yang, Zhihao Qian, Ye Zhu, Olga Russakovsky, and Yu Wu. Dˆ 3: scaling up deepfake detection by learning from discrepancy. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23850–23859, 2025.

Andrii Yermakov, Jan Cech, Jiri Matas, and Mario Fritz. Deepfake detection that generalizes across benchmarks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 773–783, 2026.

Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4352–4360. IEEE, 2017.

A Appendix

A.1 Datasets

A.1.1 Training datasets

- **DF40** (Yan et al., 2024b) is a large-scale deepfake detection dataset with 40 forgery techniques across four manipulation types, containing over 100K fake videos and over 1M fake images.
- **FaceForensics++ (c23)** (Rossler et al., 2019) is a widely used face forgery dataset of 1,000 real YouTube videos and 5,000 manipulated videos from five methods, released at three compression levels (raw, c23, and c40). We use the c23 version.
- **Celeb-DF++** (Li et al., 2025) extends Celeb-DF with 53,196 fake videos generated by 22 methods across face swap, face reenactment, and talking face, while keeping its real celebrity videos.
- **DeepSpeak v2** (Barrington et al., 2026) is a talking-head dataset of 16,585 videos from 280 consenting participants, pairing self-recorded webcam videos with lip-sync, face-swap, and avatar deepfakes of the same participants.

A.1.2 Evaluation datasets

- **UTKFace** (Zhang et al., 2017) is an in-the-wild face dataset of over 20,000 images covering a wide range of ages, genders, and ethnicities. It contains only real faces, so we use it to measure whether genuine faces from an unseen source are accepted as real.
- **RedFace** (Shi et al., 2025) is a real-world-oriented dataset of over 60,000 forged images whose forgeries are produced on commercial online platforms from authentic CelebA faces.
- **FFHQ-Fake** is an identity-controlled benchmark that we construct from FFHQ (Karras et al., 2019), pairing demographically balanced FFHQ photographs with forgeries from eight generators; its construction is described in Appendix A.4.
- **DFDC** (Dolhansky et al., 2020) is a large-scale dataset released for the Deepfake Detection Challenge, containing over 128K videos sourced from 3,426 paid actors who consented to the manipulation of their likeness.
- **Deepfake-Eval-2024** (Chandra et al., 2026) is an in-the-wild benchmark of deepfakes circulated online in 2024, collected from social media and a deepfake detection platform. It contains 1,975 images, 1,820 audio files, and 2,036 videos. We use only its image subset.

A.1.3 Preprocessing

Face-centric preprocessing pipeline. For videos, we sample 32 frames uniformly over the entire clip, and for image datasets, we use the single image of each sample. Each frame then passes through three stages. First, we detect faces with MTCNN (Zhang et al., 2016) (`min_face_size`=20, cascade thresholds $[0.6, 0.7, 0.7]$, pyramid factor 0.709). When multiple faces are detected we keep the one with the largest bounding-box area, and frames with no detection are discarded. Second, we filter non-frontal faces using FaceXFormer (Narayan et al., 2025): the detected face region is cropped, resized to 224×224 using bicubic interpolation and normalized with ImageNet statistics (mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$), and a frame is kept only if its face-parsing output contains both eyes and the nose (LaPa classes 4, 5, 6) together with at least one lip region (class 7 or 9); non-frontal frames are discarded without resampling, so the effective yield varies by source. Third, we align the face with a least-squares similarity transform that maps five of the 68 FaceXFormer landmarks (outer eye corners, nose tip, and mouth corners; indices 36, 45, 30, 48, 54) onto a face-centric variant of the ArcFace five-point template (Deng et al., 2019), described below. Aligned faces are saved as lossless 256×256 PNG images.

Face-centric cropping. This face-centric design reflects the fact that many face manipulation methods primarily alter facial regions. Rather than retaining substantial background and hair regions, we enlarge the facial region to preserve the main facial area while excluding most hair and background. When high-resolution media are downscaled to the model input size, retaining excessive background lowers the informational density of facial artifacts; cropping tightly around the face instead maximizes the resolution allocated to where manipulation cues are most concentrated.

Concretely, starting from the 112×112 ArcFace template, we shift the five reference points upward by 12 pixels to leave room for the chin, and then enlarge the face by a factor of 1.4 around the five-point centroid. We acknowledge that some detection signals, such as blending artifacts at face-background boundaries or hair-region inconsistencies, fall outside this scope. These, however, are broader image-level manipulation cues rather than face-specific forgery patterns, and are orthogonal to our focus on facial manipulation; our design instead prioritizes preserving and enhancing facial artifacts, which aligns with our core objective of detecting face-centric manipulations.

A.1.4 Data splits and sampling

Data splits. DF40 uses its official training and test splits. The other training datasets are randomly split into 90% training, 5% validation, and 5% test frames. The evaluation datasets are used only for testing: DFDC and RedFace use their official test splits, Deepfake-Eval-2024 uses all images, FFHQ-Fake uses $1{,}500$ randomly drawn real images and up to $1{,}500$ images per generator, and UTKFace uses a random 5% test portion.

Class-balanced sampling. Because the training sources differ widely in size, we keep $\min(N_g, C)$ randomly drawn training frames from each fake generator g with N_g frames, where $C = 15{,}000$, while real frames are not capped. Each mini-batch then contains equal numbers of real and fake images. This cap and real–fake balance are applied to all methods, including the baselines. For UCF, UDD, FSFM, GenD, and Effort, each class is additionally drawn equally from every training dataset (Table 8), whereas the other detectors and PAL sample frames uniformly within each class.

A.1.5 Augmentation

Training-time input and augmentation. During training, the 256×256 crops are resized (bilinear) to the input size of the backbone and normalized using a per-channel mean of 0.5 and standard deviation of 0.5. Augmentations are applied only on the training split, in order: horizontal flip, Gaussian blur, random brightness/contrast, and JPEG compression. In addition, we apply a codec re-encoding augmentation that re-encodes each image through a randomly chosen video or image codec, described next.

Codec augmentation. Training data come from multiple datasets and generation sources, which can introduce differences in their compression characteristics. These differences can let a detector exploit compression cues as a shortcut instead of relying on forgery evidence. Standard pipelines such as DeepfakeBench (Yan et al., 2023b) primarily operate within video-based datasets, where this mismatch is less pronounced; accordingly, they apply JPEG compression as a generic augmentation but do not include video-codec re-encoding. We therefore re-encode each training image through a randomly chosen codec (H.264, H.265, VP9, MPEG-4, JPEG, WebP, or JPEG2000) at a randomized quality before decoding it back, with probability 0.8, so that compression traces are observed across both classes and become less predictive of the label. We isolate this component in the Restricted setting by toggling only the codec re-encoding, scoring and thresholding every row as in Table 1 (Table 7). Enabling it trades some R_{fake} on FF for higher CDF AUC and higher acceptance of unseen real faces for nearly every method (e.g., for PAL, CDF AUC $0.811 \rightarrow 0.915$ and R_{real} on UTKFace $0.130 \rightarrow 0.647$).

A.2 Hyperparameters of the compared detectors

Table 8 lists the training configuration of each detector in the Default setting, read from the configuration saved with its checkpoint. Each baseline starts from its official implementation, except that GenD trains only its head on the frozen encoder and Effort is trained without its auxiliary regularizers. All detectors share the training data, the cap of $15{,}000$ frames per generator, batches with equal numbers of real and fake frames, horizontal flipping, codec re-encoding with probability 0.8 (Appendix A.1), and checkpoint selection by in-domain validation AUC.

Table 7: Codec re-encoding augmentation, on versus off, under the Restricted training setting. **Bold** marks the better of the two conditions for each method.

			FF			CDF			UTK	RedFace		
Method	Codec	τ	AUC	R_{real}	R_{fake}	AUC	R_{real}	R_{fake}	R_{real}	AUC	R_{real}	R_{fake}
Xception	off	0.438	**1.000**	**0.997**	**0.990**	0.648	0.001	**1.000**	0.012	0.517	0.000	0.921
	on	0.125	0.996	0.993	0.965	**0.829**	**0.050**	0.999	**0.031**	**0.536**	**0.005**	**0.955**
F3Net	off	0.185	**0.999**	**0.998**	**0.988**	0.569	0.002	**1.000**	0.054	**0.557**	0.000	0.809
	on	0.222	0.997	**0.998**	0.965	**0.774**	**0.138**	0.996	**0.085**	0.543	**0.036**	**0.901**
RECCE	off	0.809	**0.999**	**0.999**	**0.983**	0.543	0.005	**1.000**	0.039	0.327	0.000	0.804
	on	0.442	0.994	0.982	0.945	**0.809**	**0.096**	0.995	**0.083**	**0.526**	**0.026**	**0.874**
UCF	off	0.423	**0.999**	**0.999**	**0.992**	0.500	0.001	**1.000**	0.018	**0.529**	0.000	**0.889**
	on	0.062	0.997	0.980	0.974	**0.831**	**0.099**	0.998	**0.054**	0.514	**0.016**	0.887
UDD	off	0.780	**1.000**	**0.999**	**0.997**	0.815	0.004	**1.000**	**0.038**	0.410	0.001	0.772
	on	0.271	0.999	0.996	0.977	**0.854**	**0.029**	**1.000**	0.037	**0.588**	**0.025**	**0.918**
FSFM	off	1.000 (+8.66)	**0.999**	**0.999**	**0.990**	0.801	0.059	**1.000**	0.196	0.541	0.106	**0.791**
	on	0.980	0.993	0.991	0.933	**0.883**	**0.221**	0.993	**0.315**	**0.582**	**0.220**	0.767
GenD	off	0.552	**0.997**	**0.996**	**0.959**	0.691	0.058	**0.994**	0.063	0.376	0.011	0.817
	on	0.293	0.989	0.980	0.927	**0.790**	**0.100**	0.987	**0.166**	**0.472**	**0.059**	**0.832**
Effort	off	0.386	**1.000**	**0.998**	**0.991**	0.778	0.000	**1.000**	0.012	0.505	0.001	**0.819**
	on	0.526	0.997	0.982	0.971	**0.850**	**0.061**	0.999	**0.139**	**0.565**	**0.044**	0.812
PAL	off	1.000 (+7.70)	**1.000**	**1.000**	**0.988**	0.811	0.021	**0.999**	0.130	0.676	0.198	**0.744**
	on	0.988	0.999	**1.000**	0.968	**0.915**	**0.299**	0.995	**0.647**	**0.721**	**0.747**	0.634

PAL shows, for each condition, the seed with the higher in-domain validation AUC, so the two rows of a method may come from different seeds.

Table 8: Training configuration of each detector in the Default setting. CLIP-B, CLIP-L, ViT-B, and PE-B denote CLIP ViT-B/16, CLIP ViT-L/14, ViT-B/16, and PE-Core B/16. Batch sizes are per step, with gradient-accumulation steps after $\times$. Source-balanced batches draw each class equally from every training dataset. Aug. prob. gives the probabilities of blur, brightness/contrast, and JPEG compression. Additional losses give the weight of each term added to cross-entropy (rec. reconstruction, con. contrastive, align. alignment, gen. generator classification).

	Xception	F3Net	RECCE	UCF	UDD	FSFM	GenD	Effort	PAL
Backbone	Xception	Xception	Xception	Xception	CLIP-B	ViT-B	CLIP-L	CLIP-L	PE-B
Initialization	ImageNet	ImageNet	ImageNet	ImageNet	CLIP	FSFM	CLIP	CLIP	PE-Core
Trained params	all	all	all	all	LoRA	all	head	SVD res.	all
Input size	256	256	256	256	224	224	224	224	224
Optimizer	AdamW	Adam	Adam	Adam	AdamW	AdamW	AdamW	Adam	AdamW
Learning rate	2e-4	2e-4	1e-4	2e-4	5e-4	5e-5	1e-3	2e-4	1e-5
Weight decay	1e-2	1e-4	5e-4	5e-4	1e-2	5e-2	0	5e-4	1e-4
LR schedule	cosine	cosine	cosine	cosine	cosine	cosine	cosine	constant	cosine
Batch size	32	32	32	32×2	16×4	32	512	32	64×4
Epochs (patience)	30 (5)	30 (5)	30 (5)	30 (5)	30 (5)	30 (5)	30 (5)	10 (3)	30 (5)
Frames per epoch	50k	50k	50k	20k	20k	100k	50k	50k	50k
Source-balanced	–	–	–	✓	✓	✓	✓	✓	–
Aug. prob.	.5/.5/.5	.5/.5/.5	.5/.5/.5	.5/.5/.5	.1/.5/.5	.3/.3/.3	.1/.5/.5	.1/.5/.5	.5/.5/.5
Additional losses	–	–	rec. 0.3 con. 0.1	rec. 0.3 con. 0.05 gen. 0.1	con. 0.1 align. 0.1	–	–	–	anchor 1 (K=8)

A.3 Fingerprint separability: real sources, fake generators

Our approach rests on the premise that face images carry source-specific fingerprints. Real faces retain traces of the dataset they were collected from, and fake faces retain traces of the generator that produced them. We verify this premise directly for both real and fake images. We train a standard convolutional network from scratch to classify which dataset a real face image came from, or which generator produced a fake image, and show that both classifiers succeed far above chance. For both tasks, we further show that this result is robust across sample size, model capacity, and aggressive low-level image corruption, and is not attributable to memorization.

Table 9: Held-out accuracy for the two fingerprint-separability tasks.

Task	Chance	Held-out accuracy
Real sources (11-way)	9.09%	93.7%
Fake generators (52-way)	1.92%	82.5%

Setup. We use the same architecture and training recipe for both experiments: ConvNeXt V2-Nano (Woo et al., 2023) (15.0M parameters), trained from scratch (no ImageNet pretraining). We follow Liu & He (2025), who cast dataset-of-origin classification as direct evidence of dataset bias, and apply the same classification-based test to real images from our sources. We then extend it to fake images, labeled by generator identity instead of source dataset. We choose this architecture over the original ConvNeXt-Tiny (Liu et al., 2022) (27.8M) used by Liu & He (2025) because the default training recipe in Liu & He (2025) assumes 1M images per class, which substantially exceeds the number available from any single real face source or fake generator in our setting. A smaller architecture therefore provides a more favorable parameter-to-data ratio given the available data. This architectural choice does not affect the results for either task. Reevaluating both the real-source and fake-generator experiments with ConvNeXt V2-Pico (8.6M) and ConvNeXt-Tiny leaves accuracy essentially unchanged.

For real sources, we assemble 11 real face sources spanning a range of collection pipelines and demographics: FaceForensics++ (Rossler et al., 2019) (real), Celeb-DF++ (Li et al., 2025) (real), UTKFace (Zhang et al., 2017), FFHQ (Karras et al., 2019), YouTubeFaces (Wolf et al., 2011), VGGFace2 (Cao et al., 2018), CelebA (Liu et al., 2015), CelebA-HQ (Karras et al., 2017), Casia-WebFace (Yi et al., 2014), FairFace (Karkkainen & Joo, 2021), and ForgeryNet (He et al., 2021), balanced at N=8,000 images/class. This totals 88,000 images, with chance accuracy at $1/11 = 9.09\%$. For fake images, we assemble 52 generator classes from DF40 (Yan et al., 2024b) and Celeb-DF++. Real images and generators with undisclosed methods are excluded, while those shared across both datasets are merged into a single class each. We balance at N=1,500 images/class, totaling 78,000 images, with chance accuracy at $1/52 = 1.92\%$.

Results. On the 11-way real-source and 52-way fake-generator tasks, the classifier reaches held-out accuracy far above chance on both (Table 9). These results indicate that source- and generator-specific information is readily recoverable from individual face images. Figure 4 (t-SNE) provides a corresponding visualization: the 11 real sources (left) and 52 fake generators (right) occupy largely disjoint regions of feature space, suggesting that source-specific signals form distinct fingerprints. For the fake generators, we further examine whether this fingerprint reflects the generation algorithm or the dataset from which the fake images originate. Among the 9 generators shared between both datasets ($\circ$ = DF40, $\triangle$ = Celeb-DF++ in the fake panel), most show the two datasets' points largely intermixed within a single cluster, indicating that the fingerprint tracks the generation algorithm rather than the source dataset.

To confirm that this separability reflects a genuine, learnable signal rather than an artifact of the setup, we conduct three additional controls for both the real-source and fake-generator tasks. *Sample-size scaling* (Table 10): even at the smallest per-class sample sizes, held-out accuracy already far exceeds chance for both tasks, and it climbs monotonically without saturating as samples increase, indicating a genuine, learnable signal rather than a ceiling effect. *Memorization control* (Table 11): when the same recipe is used to classify a random 3-way split within a single source or generator, accuracy stays near the 33.3% chance level (34.0% mean for real sources, 34.7% for fake generators), so the observed separability is not explained by memorizing incidental per-image cues. *Low-level corruption* (Table 12): we corrupt both training and validation images using color jitter, additive Gaussian noise, Gaussian blur, or resolution reduction, each at two severities, and retrain from scratch under each corruption. For both tasks, accuracy stays flat under mild color jitter and drops progressively with severity elsewhere, but remains far above chance under all tested corruptions, consistent with the pattern reported by Liu & He (2025). This indicates that the source- and generator-specific fingerprints are not carried by superficial low-level statistics such as color, noise, or resolution alone, but persist as a deeper signal that low-level corruption cannot remove.

Taken together, both real sources and fake generators are reliably, not just incidentally, distinguishable from face images alone: this holds for a standard convolutional architecture trained entirely

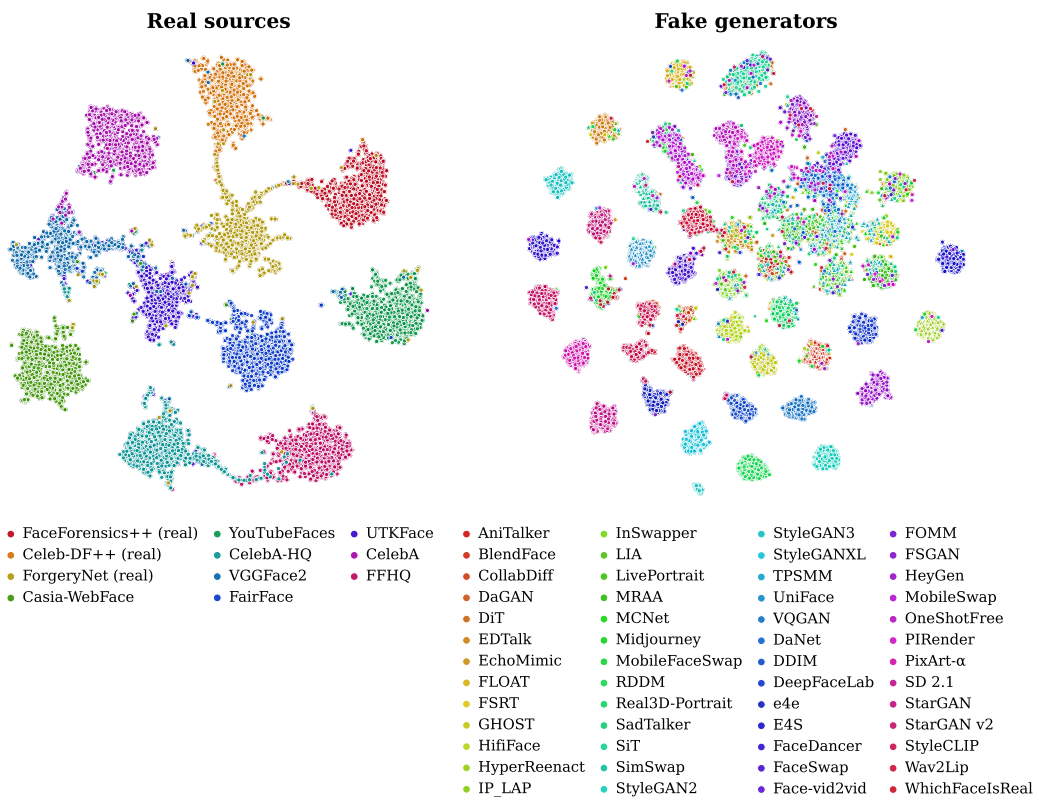

Figure 4: t-SNE of pooled ConvNeXt V2-Nano features, trained from scratch, colored by source: real sources (left, 11-way) and fake generators (right, 52-way). For the 9 fake generators shared between DF40 and Celeb-DF++, marker shape indicates dataset of origin (○ = DF40, △ = Celeb-DF++). In both cases, clusters are largely disjoint.

Table 10: Sample-size scaling: held-out accuracy vs. per-class sample size N, for both the 11-way real-source and 52-way fake-generator tasks.

Real sources (11-way)		Fake generators (52-way)	
N/class	Held-out accuracy	N/class	Held-out accuracy
2,000	82.5%	500	64.2%
5,000	90.5%	1,000	74.8%
8,000	93.7%	1,500	82.5%

from scratch, is not an artifact of model capacity, survives aggressive corruption of low-level color and texture cues, and remains far above chance even at the smallest sample sizes tested. In neither case is it attributable to memorization, since an identical recipe applied to a meaningless within-source split fails to learn anything beyond chance. Just as fake images carry generator-specific fingerprints, real faces carry source-specific ones: each occupies its own region of feature space rather than a shared "real" or "fake" locus.

A.4 FFHQ-BASED IDENTITY-CONTROLLED DEEPFAKE BENCHMARK

We construct FFHQ-Fake, an evaluation benchmark that pairs real FFHQ photographs with forgeries whose source identities are controlled across generators. This section describes how its real and fake images are built.

Table 11: Memorization control: held-out accuracy classifying a meaningless random 3-way split of each single source/generator, summarized across the 11 real sources and 52 fake generators (chance= 33.3%).

Task	Mean	Max
Real sources (11-way)	34.0%	35.3%
Fake generators (52-way)	34.7%	39.2%

Table 12: Low-level corruption: held-out accuracy after retraining from scratch on corrupted train+val images.

Corruption	Real sources (11-way)	Fake generators (52-way)
color jittering (strength 1.0)	93.7%	82.7%
color jittering (strength 2.0)	91.4%	81.9%
Gaussian noise (std 0.2)	72.8%	68.9%
Gaussian noise (std 0.3)	71.6%	66.0%
Gaussian blur (radius 3)	88.2%	77.6%
Gaussian blur (radius 5)	80.8%	76.7%
low resolution (64×64)	90.2%	80.6%
low resolution (32×32)	84.7%	76.5%

Real source construction. The real image pool is drawn from FFHQ (Karras et al., 2019) (70,000 images) in three filtering stages. First, minors are excluded using a FairFace age classifier (Karkkainen & Joo, 2021), which outputs a probability distribution over age bins rather than a single age label. We define p_{minor} as the sum of the predicted probability over the three bins under 20, and exclude an image if $p_{\text{minor}} > 0.20$, which is more conservative than argmax-based filtering since it also catches borderline cases with substantial but non-majority probability mass under 20. This removes 36.2% of the pool. Second, near-duplicate identities are filtered via ArcFace embedding similarity (Deng et al., 2019): any image with cosine similarity > 0.45 to a previously retained image is excluded, applied globally across demographic groups rather than within groups. Third, the remaining pool is subsampled to a final 10,000 using an 8-cell (gender $\times$ race) balanced quota with water-filling across cells with insufficient samples (Table 13), given FFHQ's racial imbalance toward White subjects.

Generator selection. Generator candidates were first narrowed to those with publicly available, non-commercial pretrained weights (excluding services such as MidJourney or HeyGen) requiring no fine-tuning. Within that pool, generators were selected to preserve the FFHQ source identity where applicable and, where possible, to include methods reported to be challenging for existing detectors. The two entire-face-synthesis generators are an exception to the identity-preservation preference, since no input identity exists for them to preserve. The eight resulting generators (Table 14) split evenly across four forgery categories (face-swap, FS; face-reenactment, FR; face-editing, FE; entire-face-synthesis, EFS), two per category.

Generator inputs. Six of the eight generators consume a source image drawn from a separate, demographically balanced pool of 1,250 identities, sampled from the 10,000-image real pool constructed above (Table 13) and shared across all six generators. Because all six use the same source identities, differences in generator difficulty are not confounded by differences in source identities. Among these six, the FS and FR generators additionally require a second image (target or donor), drawn under opposite pairing constraints. For FS, source and target are required to come from the *same* demographic cell and to have ArcFace cosine similarity < 0.3. Among candidates satisfying both constraints, we select the target minimizing $|\Delta\text{yaw}|$ with the source. For FR, the same-cell constraint is dropped and no pose constraint is applied, since the donor supplies pose only. The remaining two generators, both EFS, consume no source image, but are still grounded in the FFHQ domain. StyleGAN-XL uses the official, publicly released FFHQ-pretrained checkpoint (`ffhq1024.pkl`), and each PixArt-α prompt is derived from an FFHQ image using Qwen2.5-VL (Bai et al., 2025).

Table 13: Real-image cell allocation after the 3-stage filtering pipeline (gender $\times$ race, N=10,000 total).

Cell	Available	Selected
Male $\times$ White	12,082	1,303
Female $\times$ White	12,692	1,304
Male $\times$ Asian	2,465	1,303
Female $\times$ Asian	2,929	1,303
Male $\times$ Others	2,493	1,303
Female $\times$ Others	1,750	1,303
Male $\times$ Black	1,186	1,186
Female $\times$ Black	995	995

Table 14: Generator roster: eight generators spanning four forgery categories, two per category.

Category	Generator	Input
FS	Ghost (Groshev et al., 2022)	source + target
FS	HiFiFace (Wang et al., 2021)	source + target
FR	HyperReenact (Bounareli et al., 2023)	source + donor
FR	LivePortrait (Guo et al., 2024)	source + donor
FE	StyleCLIP (Patashnik et al., 2021)	source (+ template text)
FE	e4e (Tov et al., 2021)	source
EFS	StyleGAN-XL (Sauer et al., 2022)	seed only
EFS	PixArt-α (Chen et al., 2024)	text prompt

A.5 Threshold for baselines

Every baseline is read at a single threshold chosen from the validation split of its *own training data*; no test data, no unseen source, and no label beyond real/fake is used. Let $\mathcal{R}$ and $\mathcal{F}$ be the training real sources and fake manifests (fakes are capped to c=120 frames per generator). We pick $\tau^\star = \arg\max_t \frac{1}{2}\big(\mathrm{Acc_r}(t) + \mathrm{Acc_f}(t)\big)$, where $\mathrm{Acc_r}$ averages the acceptance rate over real sources and $\mathrm{Acc_f}$ averages the detection rate over fake manifests (Alg. 1). Because the objective is a mean of per-source rates, it is invariant to the real/fake ratio and to source sizes, so the well-known fake-heavy imbalance of deepfake validation splits does not shift $\tau^\star$. The same $\tau^\star$ is then applied unchanged to all test sets. PAL uses its own rule instead (Section 3.4).

A.6 Backbone capacity and the benefit of broader training data

We train PAL with three sizes of PE-Core, S/16 at 384 pixels (23.8M parameters), B/16 at 224 (93.7M), and L/14 at 336 (317.2M), in each of the three settings. All nine runs share one recipe, an effective batch of 256 and 10 epochs, so the B/16 rows differ from the main results. Table 15 reports Acc and the out-domain averages of AUC, R.real, and R.fake. A larger backbone raises Acc and out-domain AUC in every setting, and at the fixed threshold it accepts more unseen reals while catching fewer unseen forgeries. While S/16 and B/16 drop when Extended adds data to Default, L/14 keeps improving, and its lead over B/16 in Acc grows from 0.013 in Restricted to 0.037 in Default and 0.084 in Extended. This suggests that capacity and training data need to grow together, a pattern that holds consistently throughout Table 15.

A.7 Results under the Standard FF++ Protocol

For completeness, we report results under the standard FaceForensics++ (FF++) (Rossler et al., 2019) protocol with its official splits. To ensure a fair comparison, every baseline is retrained following its original implementation as closely as possible and tuned for its best performance under this protocol, rather than using pretrained checkpoints directly. Table 16 reports in-domain performance under this protocol.

Algorithm 1 Threshold selection for a detector

Require: Score function $s(\cdot)$; validation real sets $\{R_i\}_{i=1}^{m}$ and fake sets $\{F_j\}_{j=1}^{n}$
Ensure: Threshold τ

$S_i^{\mathrm{r}} \leftarrow \{\, s(x) : x \in R_i \,\}$ for $i = 1, \ldots, m$
$S_j^{\mathrm{f}} \leftarrow \{\, s(x) : x \in F_j \,\}$ for $j = 1, \ldots, n$
$\mathcal{C} \leftarrow \mathrm{unique}\big(\bigcup_i S_i^{\mathrm{r}} \cup \bigcup_j S_j^{\mathrm{f}}\big)$
for $t \in \mathcal{C}$ **do**
    $A_{\mathrm{r}}(t) \leftarrow \frac{1}{m}\sum_{i=1}^{m} \Pr_{s \in S_i^{\mathrm{r}}}\big[s < t\big]$
    $A_{\mathrm{f}}(t) \leftarrow \frac{1}{n}\sum_{j=1}^{n} \Pr_{s \in S_j^{\mathrm{f}}}\big[s \geq t\big]$
    $J(t) \leftarrow \frac{1}{2}\big(A_{\mathrm{r}}(t) + A_{\mathrm{f}}(t)\big)$
end for
$\tau \leftarrow \arg\max_{t \in \mathcal{C}} J(t)$
return τ ▷ predict FAKE iff $s(x) \geq \tau$

Table 15: PAL across PE-Core backbone capacities and training settings (one seed).

			out-domain avg.		
Setting	Backbone	Acc	AUC	R.real	R.fake
Restricted	PE-Core S/16	0.648	0.595	0.478	**0.565**
	PE-Core B/16	<u>0.730</u>	<u>0.687</u>	<u>0.689</u>	<u>0.546</u>
	PE-Core L/14	**0.743**	**0.741**	**0.828**	0.469
Default	PE-Core S/16	0.686	0.629	0.511	**0.631**
	PE-Core B/16	<u>0.776</u>	<u>0.713</u>	<u>0.719</u>	<u>0.568</u>
	PE-Core L/14	**0.813**	**0.773**	**0.887**	0.481
Extended	PE-Core S/16	0.670	0.596	0.354	**0.729**
	PE-Core B/16	<u>0.756</u>	<u>0.686</u>	<u>0.566</u>	<u>0.659</u>
	PE-Core L/14	**0.840**	**0.820**	**0.848**	0.601

A.8 Robustness to Combined Blur, Noise, and JPEG Compression

Since deployed images are often degraded, we test whether the main trends hold under blur, noise, and JPEG compression combined at increasing severity (Figure 5), using the Default-setting models with each threshold fixed to its value in Table 1. On FF and CDF, all detectors lose accuracy, but most detectors built on pretrained encoders degrade more gradually than the CNN-based ones, which fall close to chance by L2. This is consistent with CNN-based detectors relying on high-frequency artifacts that blur and compression remove. Among them, PAL stays second only to FSFM, which is pretrained on faces, at every perturbed level. The exception is real acceptance on the real-only sets. PAL accepts the most real faces on clean images and remains competitive at L1, but at L2 and L3 FSFM and Effort accept more. The anchor keeps a real face near its pretrained feature only for the degradations seen in training, so heavier ones likely open a departure that PAL reads as forgery. PAL thus stays robust in separating real from fake, while making its real acceptance robust to heavy degradation is left for future work.

A.9 Sensitivity to λ and K

We vary the anchor weight λ and the number of directions K in the Restricted setting (Tables 17 and 18). The in-domain and out-domain averages vary by at most 0.034 across both sweeps, so PAL is not sensitive to either choice in these terms. R_{real} on UTKFace is more sensitive and peaks at our defaults, $\lambda = 1$ and $K = 8$ (0.647). It falls most when the anchor is removed ($\lambda = 0$, 0.383), as training reals are then free to drift from their pretrained features, or made too strong ($\lambda = 10$, 0.363), which shifts the operating point toward fakes and raises out-domain fake detection instead.

Table 16: FF++ standard protocol.

Method	τ	AUC	R_{real}	R_{fake}
Xception	0.201	0.898	0.995	0.314
F3Net	0.989	0.935	0.999	0.293
RECCE	0.769	0.849	0.960	0.398
UCF	0.774	0.864	0.999	0.226
UDD	0.620	0.925	0.999	0.455
FSFM	0.989	0.942	0.993	0.440
GenD	0.839	0.915	0.956	0.661
Effort	0.978	0.903	0.992	0.489
PAL	1.000 (+8.13)	0.882	1.000	0.280

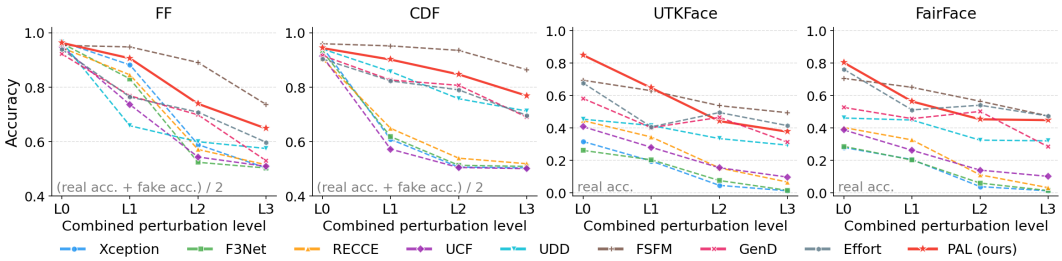

Figure 5: Robustness to combined image-level perturbations (Gaussian blur + Gaussian noise + JPEG compression) at four severity levels, L0–L3, where L0 is unperturbed and L1–L3 use kernel/σ/QF of 7/0.01/90, 9/0.03/75, and 13/0.05/60. UTKFace and FairFace (Karkkainen & Joo, 2021) contain only real faces.

Table 17: Weight λ of the anchor term (Restricted setting). $\lambda = 0$ removes the anchor. The shaded row is the default used in all other tables.

		in-domain avg.		out-domain avg.	
λ	UTK	Acc.	AUC	Acc.	AUC
0	0.383	0.820	0.944	0.611	0.685
0.1	0.572	0.832	0.946	0.613	0.673
0.3	0.610	0.824	0.947	0.615	0.670
1	**0.647**	0.834	0.956	0.635	0.691
3	0.626	**0.837**	**0.960**	0.639	0.697
10	0.363	0.822	0.956	**0.643**	**0.704**

Table 18: Number of directions K in the PAL head (Restricted setting). The shaded row is the default used in all other tables.

		in-domain avg.		out-domain avg.	
K	UTK	Acc.	AUC	Acc.	AUC
1	0.557	0.832	0.952	0.627	0.676
2	0.568	0.827	0.952	0.628	0.685
4	0.533	0.837	0.955	0.636	0.699
8	**0.647**	0.834	0.956	0.635	0.691
16	0.567	0.830	0.953	**0.644**	**0.710**
24	0.594	**0.838**	**0.957**	0.639	0.695
32	0.481	0.824	0.948	0.629	0.678