PROMPT PROTOTYPE LEARNING (PPL): RETHINKING TEXT SUPERVISION FOR GENERALIZABLE AI-GENERATED IMAGE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive vision-language models (VLMs) have shown promise for generalizable AI-generated image (AIGI) detection, yet existing approaches rely on fixed single-token class labels or per-sample captions, a brittle form of supervision that fails to capture diverse cues to generalize across unseen generators. We propose **Prompt Prototype Learning (PPL)**, a novel framework that replaces fixed class tokens with ensemble-averaged prototype embeddings. For each class, we construct multiple prompts capturing complementary aspects of visual realism, including texture, boundary consistency, and geometric cues. These prompts are encoded and aggregated into a normalized class prototype, providing richer and more stable contrastive supervision. Training images are aligned with these prototypes, improving generalization to variations in generation artifacts. We apply lightweight LoRA to the vision encoder and train exclusively on ProGAN with four object categories. Despite this limited training distribution, PPL generalizes strongly to unseen generators in the UniversalFakeDetect (97.3% mACC and 99.8% mAP) benchmark, as well as across four additional datasets covering recent generators and real-world transmitted images (94.1% mACC and 97.7% mAP), spanning GANs, diffusion models, and commercial tools, surpassing all prior methods. Ablation studies confirm that PPL is the primary driver of performance gains, with texture, boundary, and geometric prompts each contributing complementary signals.

1 INTRODUCTION

The rapid proliferation of AI-generated images (AIGIs) from generative adversarial networks (GANs) Karras et al. (2019), diffusion Rombach et al. (2022), and commercial synthesis pipelines has created a pressing need for detectors that generalize beyond their training distribution. A detector trained on one generator family must remain reliable against generators that it has never encountered, a challenge that purely discriminative classifiers consistently often fail to meet Wang et al. (2020).

Contrastive vision-language models (VLMs), trained on web-scale image-text pairs, offer a promising foundation: their joint embedding space encodes rich semantic priors about visual appearance that can be leveraged for zero-shot or few-shot detection Radford et al. (2021). Early VLM-based approaches Ojha et al. (2023); Cozzolino et al. (2024); Tasnim et al. (2025; 2026) explored the vision encoder in VLMs for AIGI detection, yet suffer from limited generalization and fail to leverage the rich textual knowledge, as shown in Fig. 1a. Recent works Tan et al. (2025); Liu et al. (2024) have exploited VLMs for AIGI detection by fine-tuning a vision-text encoder providing class supervision, as depicted in Fig. 1b. Yet a key limitation of these approaches is their reliance on fixed, single-token, or per-sample captions, which map each class to a single embedding (e.g., "Camera" or "Deepfake") and capture generic image properties rather than synthetic behavior. In addition, a few methods Liu et al. (2024) used visual embeddings with a single token (e.g., "Camera" or "Deepfake") to supervise the vision encoder, thereby improving generalization. A single token occupies a narrow, low-variance region of text embedding space, providing a weak and potentially unstable contrastive target, particularly for a high-capacity vision encoder, such as a perception encoder Bolya et al. (2025).

We propose **Prompt Prototype Learning (PPL)**, a principled drop-in replacement for single-token, visual, or per-sample caption supervision. Inspired by the CLIP zero-shot ensembling paradigm Radford et al. (2021), PPL forms class prototypes by encoding multiple semantically diverse prompts per class, averaging their normalized embeddings, and re-normalizing, as visualized in Fig. 1c. Each prompt captures complementary properties of real or AIGI, such as texture consistency, boundary structure, semantic cues, and synthesis artifacts (Sec. 3.3 and A.4, Fig. 3), so the resulting prototype

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.

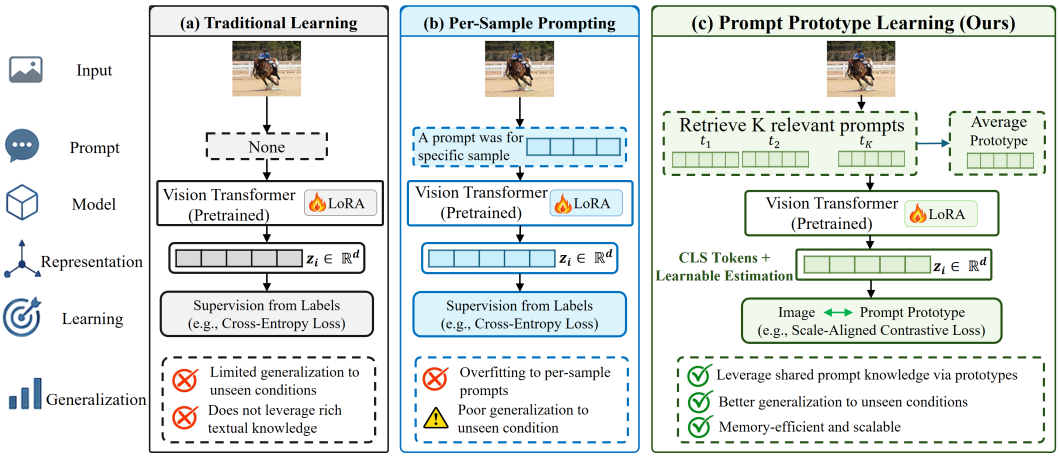


Figure 1: **Comparison of three learning paradigms for VLM representation learning.** (a) Traditional Learning relies solely on fixed class labels without leveraging textual descriptions, leading to limited semantic supervision. (b) Per-sample prompting introduces a fixed or single prompt per image, providing additional context but suffering from instability and sensitivity to prompt choice. (c) PPL retrieves multiple semantically diverse prompts (e.g., texture, boundary, and geometric cues), aggregates them into a stable class prototype, and aligns image features with this prototype via contrastive learning. This approach provides richer supervision and improves generalization.

represents a richer and more informative region of the text embedding space rather than a single point. The major contributions are listed as follows:

- We propose PPL, which replaces fixed single-token labels with ensemble-averaged prototype embeddings, providing richer and more stable contrastive supervision without per-sample annotations.
- We design a structured prompt library spanning texture, boundary, and geometric axes, and show via ablation that full axis coverage is essential for cross-generator generalization.
- We demonstrate that PPL, trained exclusively on ProGAN with only 4 categories, generalizes strongly to unseen GAN, diffusion, and commercial generators, surpassing prior VLM and non-VLM detectors for AIGI detection.

2 Related Work

We review prior work on traditional approaches, artifact-based, VLM-based, and LoRA adaptation.

2.1 AI-Generated Image Detection

Early AIGI detection showed that CNN classifiers trained on GAN outputs can generalize across generator families Wang et al. (2020), with ProGAN Karras et al. (2017) serving as a canonical benchmark. Subsequent studies identified structural sources of artifacts, including spectral inconsistencies from upsampling operations Durall et al. (2020); Frank et al. (2020), neighborhood pixel statistics Tan et al. (2024b), and gradient signatures Tan et al. (2023). Frequency-domain approaches such as FreqNet Tan et al. (2024a) improved generalization by operating directly in spectral space. Patch-level and statistical methods further enhanced cross-model detection Chai et al. (2020); Nataraj et al. (2019); Barni et al. (2020); Shiohara & Yamasaki (2022).

The emergence of diffusion models introduced new challenges. DIRE Wang et al. (2023) detects synthetic images via diffusion inversion and reconstruction error, based on the observation that diffusion-generated images are reconstructed more faithfully by a pretrained diffusion model than real images. Corvi et al. Corvi et al. (2023) provided early cross-model evaluations, showing that diffusion-generated images exhibit distinct statistical properties from GAN outputs. More recent work highlights the importance of data alignment Rajan et al. (2025), questions over-reliance on real-image features Rajan & Lee (2025), and proposes data-efficient forgery-focused representations Chen et al. (2025b). A systematic sanity-check study Yan et al. (2025) reveals fundamental limitations of current detectors, motivating stronger supervision and improved generalization strategies.

2.2 VLM-based Detection

The introduction of large vision-language models, particularly CLIP Radford et al. (2021), shifted the detection paradigm. UniD Ojha et al. (2023) showed that frozen CLIP features achieve strong zero-shot generalization, establishing VLMs as a powerful inductive bias. Cozzolino et al. Cozzolino et al. (2024) further demonstrated that CLIP encodes artifact-relevant cues even without task-specific fine-tuning. RINE Koutlis & Papadopoulos (2024) improves sensitivity by mapping intermediate

CLIP features, reducing reliance on final-layer representations.

Subsequent work explored parameter-efficient adaptation. FatF Liu et al. (2024) integrates forgery-aware adapters with frequency analysis while still relying on fixed class tokens (e.g., "Camera" or "Deepfake"). C2PC Tan et al. (2025) introduces category-common prompts to enrich supervision but ultimately retains fixed tokens for classification. Effort Yan et al. (2024) decomposes CLIP features into domain-specific and invariant components, while MiraGe Shi et al. (2025) applies multimodal discriminative learning. SAFE Li et al. (2025) improves generalization via image transformations, and IAPL Li et al. (2026a) proposes image-adaptive prompts that are updated at inference time. However, these methods do not fundamentally address the limited expressiveness of text supervision.

2.3 Prompt Learning and Parameter-Efficient Adaptation

Prompt learning replaces hand-crafted templates with learnable tokens Liu et al. (2021); Zhou et al. (2022b), extended to vision transformers via visual prompt tuning (VPT) Jia et al. (2022) and input-conditioned prompts via CoCoOp Zhou et al. (2022a). In AIGI, prompt-based CLIP adaptation has proven effective Liu et al. (2024); Tan et al. (2025); Lin et al. (2025), though prior methods treat prompts as class-label surrogates rather than enriching supervision. PPL instead brings CLIP's prompt ensembling Radford et al. (2021) into the training objective. For parameter-efficient fine-tuning, LoRA Hu et al. (2022) constrains updates to a low-rank subspace to preserve generalization, and has been applied to CLIP-based AIGI detectors Liu et al. (2024); Tan et al. (2025).

2.4 Positioning of PPL

PPL lies at the intersection of VLM-based AIGI detection and prompt ensembling, and differs from prior work along three axes. *What the text target encodes:* FatF Liu et al. (2024) supervises with a single class token (e.g., "Camera" or "Deepfake"), and C2PC Tan et al. (2025) with category-common captions that describe image content. PPL instead builds each class target from multiple artifact-aware descriptions spanning texture, boundary, and geometric cues, so the target encodes *why* an image is real or synthetic rather than *what* it depicts. *How the target is formed:* learnable prompts Zhou et al. (2022b;a) and image-adaptive prompts Li et al. (2026a) optimize the text input during training or at inference. PPL keeps its prompts fixed and aggregates them into a precomputed prototype, which prevents the text target from drifting toward source-generator artifacts and requires no test-time optimization. *Where text is used:* CLIP's prompt ensembling Radford et al. (2021) averages prompts at inference to build zero-shot classifiers. PPL moves ensembling into the training objective as a contrastive target and discards the text branch after training, so inference reduces to the vision encoder and a linear head. Together, these choices make PPL a drop-in replacement for existing text supervision that adds no annotation burden and no inference cost.

3 Method

This section details the motivation, architecture, prompt prototype, LoRA, and training objective.

3.1 Motivation

Existing VLM-based detectors supervise contrastive training with either (i) a single fixed token Liu et al. (2024) (e.g., "Camera" or "Deepfake"), which maps to a narrow, low-variance point in text embedding space and provides a weak contrastive target, or (ii) per-sample captions Tan et al. (2025), which focus on generic image descriptions with no direct connection to synthesis artifacts and still rely on fixed tokens for the decision boundary. Both strategies fail to capture the diverse visual properties that distinguish real from AIGIs.

PPL replaces both with class-level prototype embeddings derived from ensembles of diverse, artifact-aware prompts, which consistently outperforms single-prompt detection Radford et al. (2021). Rather than relying on a single caption or token, each prototype is grounded in prompts spanning three orthogonal forensic axes, texture, boundary, and semantic cues, so that no single generator-specific artifact dominates the supervision and the resulting target remains stable across generation pipelines.

3.2 Architecture

Fig. 2 provides an overview of the proposed PPL. PPL comprises four components: a vision encoder adapted with LoRA, a frozen text encoder that supplies prototype supervision, a learnable projection with scale-matching to align the two modalities, and a lightweight detection head.

We use PE-Core Bolya et al. (2025), a ViT-G/14 pretrained at 448×448 with $d_v = 1536$ dimensions and 1.88B parameters as a vision encoder. All parameters are frozen except the LoRA adapters. Given an input $I \in \mathbb{R}^{3\times448\times448}$, the vision encoder produces CLS token features:

$$\mathbf{v} = f_v(I) \in \mathbb{R}^{L\times d_v}, \quad d_v = 1536. \tag{1}$$

We adopt **CLIP-ViT-bigG** Cherti et al. (2023), a frozen text encoder with $d_t = 1280$ dimensions, which is used exclusively during training to precompute class prototype embeddings (Sec. 3.3); it is

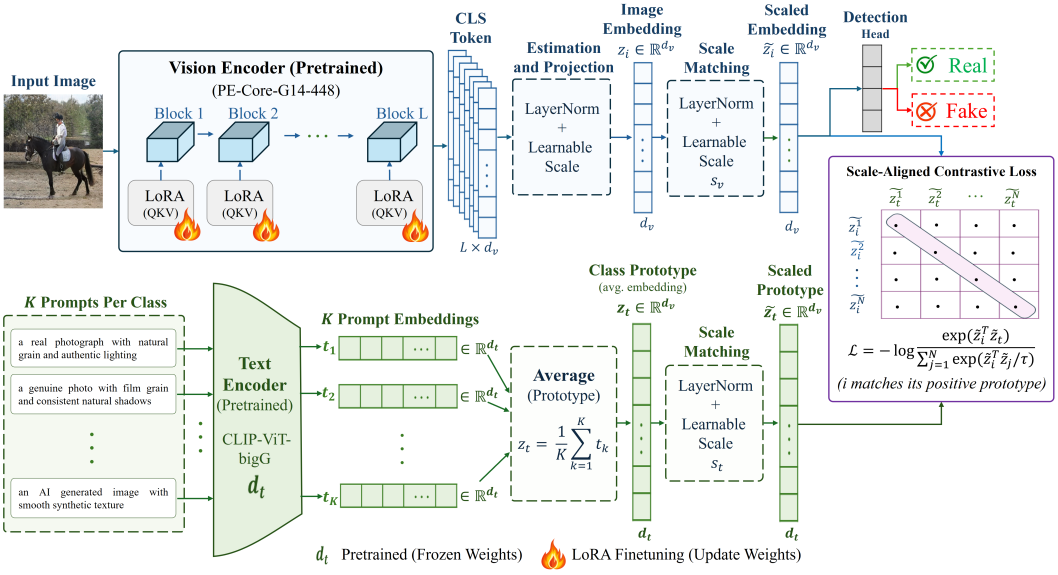


Figure 2: **Overview of the proposed PPL.** Top: an input image is encoded by a vision transformer, adapted via LoRA on its QKV projections, to produce an ℓ_2-normalized image embedding $\tilde{z}_i \in \mathbb{R}^d$. Bottom: for each class, K prompts describing complementary properties of real and AIGIs are encoded by a frozen text encoder, then averaged and re-normalized into a class prototype $\tilde{z}_t \in \mathbb{R}^d$. A scale-aligned contrastive loss aligns each image embedding with its class prototype, with temperature τ controlling distribution sharpness. The text branch and prototypes are used only during training; at inference, detection relies solely on the image branch with a fully connected layer.

not invoked at inference time. Given an input prompt p_k, the frozen text encoder produces:

$$\mathbf{t}_k = f_t(p_k) \in \mathbb{R}^{d_t}, \quad d_t = 1024. \tag{2}$$

The L patch tokens output by the vision encoder are aggregated and passed through a LayerNorm followed by a learnable scale to produce the image embedding $\mathbf{z}_i \in \mathbb{R}^{d_v}$. A learnable linear projection $W \in \mathbb{R}^{d_t \times d_v}$ then maps the normalized image embedding into the text encoder's output space:

$$\mathbf{z}_i = \mathrm{LN}(\mathbf{v}), \qquad \hat{\mathbf{v}}_{\text{proj}} = \frac{W\mathbf{z}_i}{\|W\mathbf{z}_i\|_2}. \tag{3}$$

W is initialized with $\mathcal{N}(0,\, 0.02^2)$ and serves as the primary bridge aligning the two modalities. Since the vision and text encoders operate at different embedding scales, we apply a scale matching block to each branch, consisting of a LayerNorm followed by a learnable scalar (s_v for the vision branch, s_t for the text branch):

$$\tilde{\mathbf{z}}_i = s_v \cdot \mathrm{LN}(\hat{\mathbf{v}}_{\text{proj}}) \in \mathbb{R}^{d_v}, \qquad \tilde{\mathbf{z}}_t = s_t \cdot \mathrm{LN}(\mathbf{z}_t) \in \mathbb{R}^{d_t}. \tag{4}$$

The scaled embeddings $\tilde{\mathbf{z}}_i$ and $\tilde{\mathbf{z}}_t$ are used in the scale-aligned contrastive loss (Sec. 3.4). A separate linear layer $g : \mathbb{R}^{d_v} \to \mathbb{R}$ maps the normalized vision embedding directly to a binary detection logit:

$$s = g(\mathbf{z}_i) \in \mathbb{R}. \tag{5}$$

The head is initialized with $\mathcal{N}(0,\, 0.02^2)$. At inference, detection is performed using only this head: $\hat{y} = \sigma(s)$. The projection W, scale matching blocks, and text encoder are not involved, making inference identical in cost to a standard linear probe on top of the frozen PE features.

We fine-tune the PE-Core-G14-448 encoder via LoRA Hu et al. (2022), constraining weight updates to a low-dimensional subspace: $W = W_0 + \frac{\alpha}{r}BA$, where $r \ll d$. LoRA is applied to `qkv` and projections with rank r=8, α=6, and dropout p=0.3 based on ablation studies (Sec. 5.2). $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{d \times r}$ are the down- and up-projection matrices, respectively.

3.3 Prompt Prototype Learning

PPL constructs class prototypes from ensembles of K=6 diverse, artifact-aware prompts per class from a prompt library of 12 prompts per class (A.4) through ablation (Sec. 5.3, Figs 7a- 7c), following CLIP's zero-shot ensemble formulation Radford et al. (2021). Prompts are designed along three orthogonal forensic axes: texture (grain, noise, sensor statistics), boundary (blur, bokeh, chromatic aberration), and geometric (lens distortion, perspective, depth) (Sec. 5.3, Fig. 9). Real-class prompts (RP-1–12) describe universal physical camera properties, while fake-class prompts cover three

Real Prompt (RP) Properties

Target physical optics and photochemical properties absent in synthesized images:
Texture Realism (RP-1,2,3,4)
Natural grain, sensor noise, film grain, consistent material properties.
Boundary Optics (RP-5,6,7,8)
Real optical blur, natural bokeh, depth-of-field, chromatic aberration.
Geometric Effects (RP-9,10,11,12)
Authentic lens distortion, perspective curvature, depth discontinuities, natural shadow gradients.

Fake Prompt (FP) Properties

Target artifact patterns across GAN, diffusion, and commercial pipelines:
General Synthesis Artifacts (FP-1,2,3,4)
Synthetic texture, unnatural edges, smooth surfaces, deepfake artifacts.
GAN Artifacts (FP-5,6,7,8)
Unnatural skin boundaries, synthetic hair, color bleeding, spectral inconsistencies.
Diffusion Artifacts (FP-9,10,11,12)
Blended transitions, inconsistent lighting, fake bokeh, over-smoothed surfaces.

Figure 3: Properties used to design real and fake prompts in PPL (full prompt list in A.4, Fig. 9). Real prompts target physical imaging properties: texture realism, boundary optics, and geometric effects. Fake prompts cover synthesis artifacts: AI, GAN-, and diffusion-specific.

synthesis paradigms: general artifacts (FP-1–4), GAN artifacts (FP-5–8), and diffusion artifacts (FP-9–12). Prompts describe artifact families rather than specific models, to encourage generalization. Each prompt p_k^c is encoded by the frozen text encoder f_t to produce $\mathbf{t}_k^c = f_t(p_k^c) \in \mathbb{R}^{d_t}$. Embeddings are ℓ_2-normalized, averaged, and re-normalized to form the class prototype:

$$\mathbf{z}_t^c = \frac{\bar{\mathbf{t}}_c}{\|\bar{\mathbf{t}}_c\|_2}, \quad \bar{\mathbf{t}}_c = \frac{1}{K}\sum_{k=1}^{K} \frac{\mathbf{t}_k^c}{\|\mathbf{t}_k^c\|_2}, \quad c \in \{\text{real, fake}\}. \tag{6}$$

The prototype $\mathbf{z}_t^c$ is then scaled via $\tilde{\mathbf{z}}_t^c = s_t \cdot \mathrm{LN}(\mathbf{z}_t^c)$ for use in the contrastive loss. Prototypes are fixed before training and not updated. Ablations (Sec. 5.3, Figs. 7a- 7c) confirm that axis coverage is the primary driver of generalization for AIGI detection.

3.4 Training Objective

We train on ProGAN Karras et al. (2017)-generated images (four object classes: car, cat, chair, horse) for 8000 steps with batch size 16, using the Adam optimizer (β_1=0.9, β_2=0.999) with learning rate 1×10^{-4} and a step-decay schedule (factor 0.9 every 1000 steps). Mixed-precision training is used throughout with total loss:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \lambda\, \mathcal{L}_{\text{contrastive}}, \tag{7}$$

where $\mathcal{L}_{\text{BCE}}$ is binary cross-entropy (BCE), $\mathcal{L}_{\text{contrastive}}$ is a symmetric InfoNCE loss between image projections and class prototypes. We set λ=3 based on the ablation study (Sec. 5.4).

4 Results

This section presents the experimental setup, performance comparisons, and detailed ablation study.

4.1 Experimental Details

All experiments are conducted on Ubuntu 22.04 using two 48GB GPUs. PPL is trained exclusively on ProGAN Karras et al. (2017) and evaluated, without any retraining or adaptation, on five benchmarks: UniversalFakeDetect (UFD) Ojha et al. (2023), AIGCDetect (AIGC) Yan et al. (2025), AIGI-Bench (AIGB) Li et al. (2026b), AIGI-Holmes (AIGH) Zhou et al. (2025), and TrueFake (TruFak) Dell'Anna et al. (2025). Implementation details and dataset statistics are provided in A.1 and A.2 (Tables 6- 7). Following standard protocol, we report mean accuracy (mACC) and mean average precision (mAP) for PPL and all state-of-the-art AIGI detectors (A.3).

4.2 Performance Evaluations and Comparisons

Tables 1 and 2 report detection ACC and AP across 5 benchmarks spanning GAN, low-level vision, and Diffusion families to compare the proposed PPL with prior detectors.

4.2.1 Evaluation on UniversalFakeDetect Benchmark

GAN Family: Most detectors perform strongly on canonical GAN generators, since many were originally designed around GAN-specific spectral and upsampling artifacts. EFFT, FatF, and ForL achieve near-perfect ACC (>93%) on ProGAN, CycleGAN, BigGAN, StyleGAN, GauGAN, and StarGAN, benefiting from these shared artifacts, while CNND and RINE show notable drops on individual generators (e.g., CNND on BigGAN: 70.2%; RINE on StyleGAN: 88.9%). PPL stays within 0.8 points of the best method on every GAN generator (98.9–100.0%).
Other Family: This family is more heterogeneous and exposes clearer separations among baselines. On Deepfakes, several methods fall well below 80% ACC (CNND: 53.5%, AIDE: 54.1%, UniD: 66.6%), and here FrqNet is the strongest performer (97.4%/99.9%), with PPL close behind at 91.7% ACC / 99.2% AP. On SITD and SAN, where real/fake boundaries are subtle, CNND and UniD again

Table 1: Detection ACC/AP (%) across all generators on UFD benchmark. All methods use the ProGAN-trained model. Best=**bold**, Second best=underline.

	CNND		NPR		FrqNet		UniD		EFFT		FatF		RINE		AIDE		C2PC		ForL		SAFE		MiraGe		IAPL		**PPL**	
Generator	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP
GAN Family																												
ProGAN	**100.0**	**100.0**	99.8	**100.0**	97.9	99.9	**100.0**	**100.0**	**100.0**	**100.0**	99.9	**100.0**	**100.0**	**100.0**	99.9	**100.0**	99.9	**100.0**	**100.0**	**100.0**	99.9	**100.0**	**100.0**	**100.0**	**100.0**	99.9	**100.0**	**100.0**
CycleGAN	85.2	93.5	95.0	99.5	95.8	99.6	98.5	98.1	**99.9**	**100.0**	93.3	**100.0**	99.3	**100.0**	98.5	99.9	97.3	**100.0**	99.5	**100.0**	98.9	99.8	94.3	**100.0**	98.6	99.9	99.0	**100.0**
BigGAN	70.2	84.5	87.6	94.5	90.5	96.1	94.5	94.5	**99.6**	**100.0**	99.5	99.9	**99.6**	99.9	83.9	94.4	99.1	99.9	99.4	99.9	89.7	95.9	96.5	99.9	98.7	99.9	99.4	**100.0**
StyleGAN	85.7	99.5	96.2	99.9	97.6	99.9	82.0	86.7	95.1	99.8	97.2	99.8	88.9	99.4	**99.7**	**100.0**	96.4	99.5	96.6	99.2	98.0	99.9	96.8	99.8	94.9	99.8	98.9	**100.0**
GauGAN	79.0	89.5	86.6	88.8	50.2	99.7	99.5	99.3	99.6	**100.0**	99.4	**100.0**	**99.8**	**100.0**	73.2	97.7	99.2	**100.0**	95.9	99.5	91.5	97.2	93.6	99.9	99.4	**100.0**	99.7	**100.0**
StarGAN	91.7	98.2	99.8	**100.0**	93.4	98.6	97.0	99.5	**100.0**	**100.0**	99.8	**100.0**	99.5	**100.0**	99.9	**100.0**	99.6	**100.0**	99.0	**100.0**	99.9	**100.0**	96.1	99.9	96.7	**100.0**	99.7	**100.0**
Other Family																												
Deepfakes	53.5	89.0	76.9	84.4	**97.4**	**99.9**	66.6	91.7	87.6	99.0	93.2	97.9	80.6	97.9	54.1	76.4	93.8	98.6	81.2	95.1	93.1	97.5	88.7	96.0	95.9	97.6	91.7	99.2
SITD	66.7	73.8	66.9	97.9	88.9	94.4	63.0	78.5	92.5	97.5	81.1	97.9	90.6	97.2	68.1	83.6	**95.6**	98.9	82.4	94.7	85.6	85.1	75.8	93.9	90.8	97.3	85.3	**99.4**
SAN	48.7	59.5	**98.6**	**99.9**	59.0	74.6	57.5	67.5	81.5	97.5	68.0	81.2	68.3	94.9	71.1	88.8	64.4	84.6	78.0	98.6	95.9	99.3	71.9	84.7	93.8	98.1	95.1	**99.9**
CRN	86.3	98.2	50.0	50.2	71.9	80.1	59.5	83.1	**98.9**	**100.0**	69.5	99.8	89.2	97.3	60.9	90.7	93.3	99.9	98.3	99.8	50.1	41.3	92.9	99.9	92.5	**100.0**	96.5	**100.0**
IMLE	86.3	98.4	50.0	50.2	67.4	75.7	72.0	91.1	**98.9**	**100.0**	69.5	99.9	90.6	99.7	61.0	99.9	93.3	99.9	98.1	**100.0**	50.1	47.2	92.9	**100.0**	92.7	**100.0**	94.5	**100.0**
Diffusion Family																												
Guided	60.1	73.7	84.6	**98.3**	86.7	96.3	70.0	79.2	69.2	95.4	76.0	91.9	76.1	96.4	88.5	97.3	69.1	94.1	67.1	93.3	82.4	95.8	82.0	96.4	72.8	96.3	**90.3**	98.1
LDM 200	54.0	70.6	97.7	99.9	84.6	96.1	94.2	95.8	99.3	**100.0**	98.6	99.8	98.3	99.8	98.2	99.3	99.3	99.9	99.2	99.9	98.8	99.9	98.3	99.9	99.5	99.9	**99.6**	99.9
LDM 200 CFG	55.0	71.0	98.0	99.9	**99.6**	**100.0**	73.8	79.8	96.8	99.9	94.9	99.1	88.2	98.3	97.5	99.1	97.3	99.8	96.7	99.8	98.7	99.9	94.6	99.1	97.7	99.6	**99.6**	99.9
LDM 100	54.1	70.5	98.2	99.9	65.6	62.3	94.4	95.9	99.5	**100.0**	98.7	99.9	98.6	99.9	98.4	99.3	99.3	99.9	99.3	99.9	98.8	99.9	98.8	99.9	99.2	99.7	**99.7**	99.9
Glide 100/27	60.8	80.7	96.3	99.9	85.7	99.8	79.1	93.9	97.5	99.9	94.4	99.1	88.9	98.8	98.2	99.3	95.3	99.7	84.1	99.7	95.8	98.9	97.5	99.8	98.0	99.4	**99.6**	**100.0**
Glide 50/27	63.8	84.9	97.2	99.9	97.4	99.8	79.9	95.1	97.8	99.9	94.7	99.4	92.6	99.3	98.4	99.3	95.3	99.8	86.5	99.8	96.6	99.2	97.5	99.7	98.3	99.7	**99.6**	**100.0**
Glide 100/10	65.7	82.1	97.4	99.9	88.2	96.4	78.1	94.6	97.2	**100.0**	94.2	99.2	90.7	98.9	97.9	99.2	96.1	99.8	84.4	99.9	97.3	99.4	98.0	99.8	98.4	99.9	**99.9**	**100.0**
DALL-E	55.6	70.6	87.2	99.3	59.1	77.8	86.8	88.5	98.1	**100.0**	98.8	99.8	95.0	99.3	97.5	99.0	98.9	99.9	98.8	99.9	97.5	99.7	98.6	99.9	98.9	99.9	**99.7**	**100.0**
Mean	69.6	83.6	87.6	92.8	83.0	91.9	81.4	90.1	95.2	99.4	90.6	98.1	91.3	98.8	86.6	96.0	93.8	98.6	91.8	98.9	90.5	92.4	92.9	98.3	95.6	99.3	**97.3**	**99.8**

Table 2: Detection ACC/AP (%) on other datasets, including recent generators and real-world transmission. For true generalization, all methods use ProGAN-trained models, as given in Table 1.

	CNND		NPR		FrqNet		UniD		EFFT		FatF		RINE		AIDE		C2PC		ForL		SAFE		MiraGe		IAPL		**PPL**	
Dataset	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP
AIGC	53.5	63.7	87.7	93.3	76.9	82.2	64.5	84.2	90.8	95.2	79.8	93.6	74.8	95.8	88.3	94.8	74.6	94.9	84.7	96.3	85.9	93.8	89.8	95.1	91.5	96.5	**95.6**	**98.3**
AIGB	57.7	65.0	72.7	72.8	75.6	84.3	60.9	67.1	72.2	77.2	70.2	74.7	69.5	75.9	77.1	79.8	70.0	80.0	72.1	75.8	82.4	85.9	84.2	91.1	70.2	75.6	**93.4**	**96.1**
AIGH	51.6	58.3	64.5	72.7	66.6	69.1	63.4	86.8	72.8	84.7	64.0	66.9	69.6	92.2	81.7	90.2	62.3	83.5	73.8	79.4	86.6	92.7	91.0	96.3	65.2	69.6	**97.8**	**99.2**
TruFak	38.4	79.2	48.4	76.4	38.6	75.7	35.0	69.7	36.4	71.2	43.5	74.7	41.3	81.2	41.4	77.8	60.0	80.0	37.4	76.4	38.7	72.5	63.6	89.0	55.0	78.9	**89.7**	**97.3**
Mean	50.3	66.6	68.3	78.8	64.4	77.8	56.0	77.0	68.1	82.1	64.4	77.5	63.8	86.3	72.1	85.7	66.7	84.6	67.0	82.0	73.4	86.2	82.2	92.9	70.5	80.2	**94.1**	**97.7**

Detailed per-dataset results across generators are provided in A.5 and Tables 9- 12.

struggle (48.7–66.7% ACC), and PPL attains the best AP on both (99.4%, 99.9%), while C2PC (95.6%) and NPR (98.6%) edge out PPL on raw ACC. On CRN and IMLE, PPL attains the best or joint-best AP (100.0%) with strong ACC (96.5% and 94.5%, respectively), though EFFT achieves the top ACC on both generators (98.9%).

Diffusion Family: Diffusion models remain the most discriminative setting for separating baselines. Guided diffusion is the hardest case in this family: several strong baselines collapse toward 60–76% ACC (CNND: 60.1%, EFFT: 69.2%, C2PC: 69.1%), while PPL achieves the best ACC at 90.3% (AP: 98.1%, second only to NPR's 98.3%), a clear margin over the next-best ACC baseline, AIDE (88.5%). Across the LDM and Glide variants, PPL attains the best or joint-best ACC in every setting (99.6–99.9%) with near-perfect AP (99.9–100.0%), and on DALL-E it again leads with 99.7% ACC and 100.0% AP, matching or exceeding all competing methods.

Across all nineteen generators, PPL achieves the best overall mACC (97.3%) and mAP (99.8%), outperforming every baseline. The second-best mACC is achieved by IAPL (95.6%), while the second-best mAP is achieved by EFFT (99.4%), underscoring PPL's consistency and generalizability across the GAN, Other, and Diffusion families.

4.2.2 Evaluation on Other Recent and Real-World Benchmarks

To assess generalization beyond UFD, we evaluate all ProGAN-trained detectors on four additional datasets (Table 2). AIGC, AIGB, and AIGH cover more recent and diverse generators, including text-to-image diffusion, identity-conditioned, and autoregressive models. TruFak contains real and synthetic images shared through social-media platforms, which apply re-compression and re-encoding. This setting is considerably harder than UFD: the mean ACC of most baselines falls to 50–73%. The drop is most severe on TruFak, where ten of the thirteen baselines score below 50% ACC while retaining 69.7–81.2% AP. This indicates that their decision thresholds, rather than only their ranking ability, fail to transfer under this shift.

PPL achieves the best ACC and AP on all four datasets, with a mean of 94.1% ACC and 97.7% AP. It outperforms the second-best method, MiraGe (82.2%/92.9%), by +11.9 ACC and +4.8 AP on average. Relative to the second-best method on each dataset, the ACC gain is +4.1 on AIGC (IAPL), +9.2 on AIGB, +6.8 on AIGH, and +26.1 on TruFak (all MiraGe). The largest gain is therefore on TruFak, where PPL reaches 89.7% ACC and 97.3% AP despite the combination of unseen generators and lossy transmission. PPL is also the only method whose ACC exceeds 89% on every dataset, whereas the second-best method, MiraGe, ranges from 63.6% to 91.0%. These results suggest that the cues learned by PPL transfer better than those of prior detectors to both newer architectures and real-world degradation. Per-generator results for each dataset are provided in A.5.

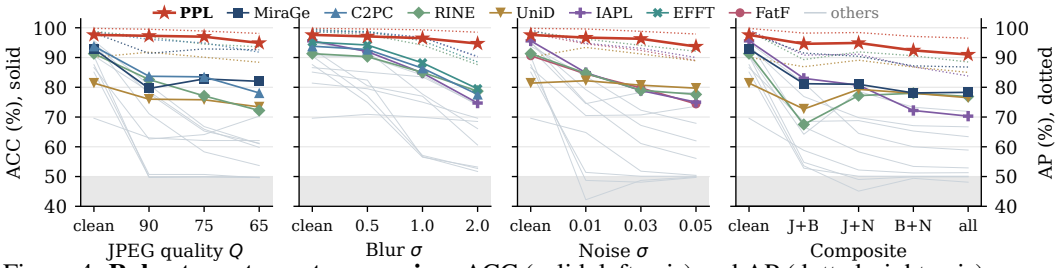


Figure 4: **Robustness to post-processing.** ACC (solid, left axis) and AP (dotted, right axis) versus perturbation strength; gray curves are the remaining baselines, the shaded band marks chance.

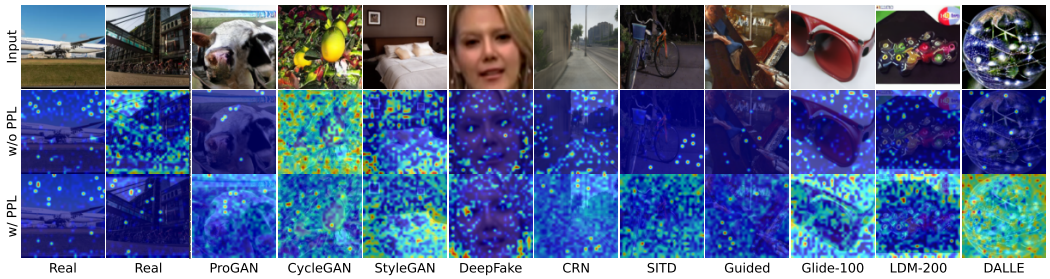


Figure 5: Grad-CAM Selvaraju et al. (2017) visualizations comparing the model's prediction without (middle row) and with (bottom row) PPL on images from two real sources and ten generators.

4.3 Robustness to Post-Processing Perturbations

Images circulated online are rarely pristine: they are recompressed, resized, and denoised before they reach a detector. We therefore evaluate all methods under three canonical perturbations, JPEG compression ($Q \in \{90, 75, 65\}$), Gaussian blur ($\sigma \in \{0.5, 1.0, 2.0\}$), and additive Gaussian noise ($\sigma \in \{0.01, 0.03, 0.05\}$), as well as their pairwise and triple compositions, without retraining or perturbation-aware augmentation, so that the results measure intrinsic rather than adapted robustness (Table 13; degradation curves in Fig. 4, solid: ACC, dotted: AP). Methods that rely on high-frequency residuals or spectral artifacts collapse as soon as those cues are filtered out: NPR and FrqNet fall from 87.6/83.0 ACC on clean data to ≈ 50 under $Q = 90$ alone, and SAFE behaves similarly. Stronger recent detectors survive single perturbations but not compositions; EFFT drops from 95.2 to 52.9 ACC and IAPL from 95.6 to 70.3 under all settings, losses of 42.3 and 25.3 points. Notably, AP degrades far more gracefully than ACC for most baselines, indicating that the perturbations shift the decision threshold at least as much as they destroy the underlying signal.

In contrast, PPL attains the best ACC and AP in every one of the fourteen conditions. It loses only 6.3 ACC (97.3 → 91.0) and 3.3 AP (99.8 → 96.5) between the clean and the fully composed setting, the smallest degradation of any method by a wide margin. Per family, the gap over the strongest competing detector is +12.9 ACC at $Q = 65$, +15.2 at blur $\sigma = 2.0$, +14.0 at noise $\sigma = 0.05$, and +12.7 under all. The near-flat red curves in Fig. 4, contrasted with the steep descent of the gray baseline envelope, show that the cues PPL exploits are not confined to the fragile high-frequency band that most prior detectors depend on, and therefore remain available after compression, smoothing, and noise. The detailed results per-perturbation across detectors are provided in A.6 and Table 13.

4.4 Grad-CAM Visualization

Figure 5 shows Grad-CAM attention maps that indicate which image regions the model relies on. Without the PPL loss, activations concentrate on isolated, high-frequency point artifacts. This is likely a brittle strategy, since such local statistics vary across generators and may not be present in unseen ones. With PPL, attention instead spreads over coherent structural regions, and this pattern holds across real images and all ten fake sources. This shift is consistent with the prototype alignment objective: by pulling visual embeddings toward semantically grounded text prototypes, PPL appears to reduce reliance on source-specific low-level cues and to encourage a more generator-agnostic representation of authenticity. Together with the ablation results, this visual evidence supports PPL as a key contributor to the cross-generator generalization gains in Tables 1 and 2.

5 Ablation Study

We performed sequential ablation studies on UFD benchmark with ProGAN-trained model. Each was evaluated under the best prior setting, with the final model integrating all optimal choices.

5.1 Encoder Selection

We evaluated prior foundation models and selected the best-performing vision and text encoders.

Table 3: Vision Encoder w/o PPL

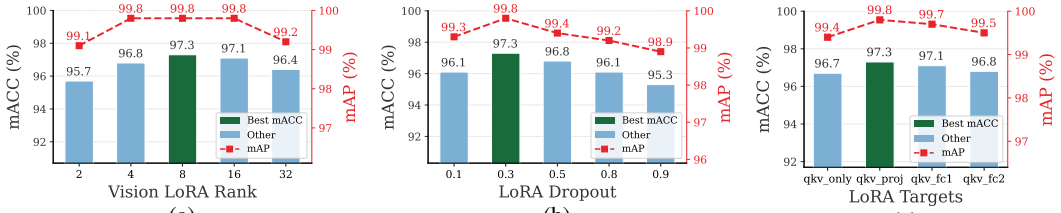
Backbone	mACC	mAP
CLIP-L/14	92.9	98.5
DINOv2-L/14	90.5	98.3
PE-Core-L/14	93.1	98.6
PE-Core-G/14	**93.8**	**99.2**

Table 4: Vision Encoder with PPL

Encoder	mACC	mAP
CLIP-L/14	94.4	99.1
DINOv2-L/14	93.1	99.0
PE-Core-L/14	96.3	99.3
PE-Core-G/14	**97.3**	**99.8**

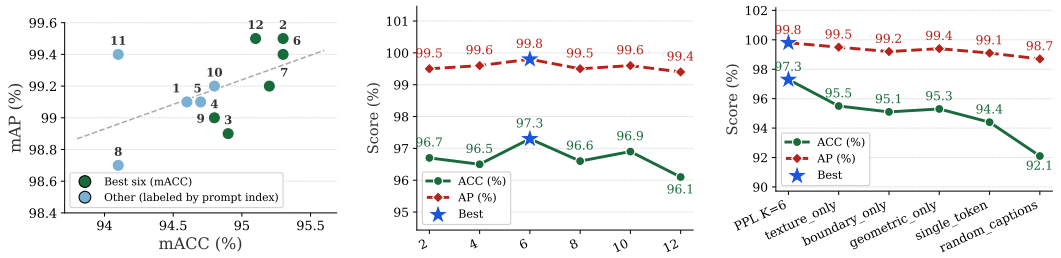
Table 5: Text Encoder

Encoder	mACC	mAP
CLIP-H/14	96.7	99.7
CLIP-bigG/14	**97.3**	**99.8**
SigLIP-L/16	94.7	99.1
BLIP-L	95.1	99.5



(a) (b) (c)

Figure 6: Ablation on LoRA configuration (a) rank selection, (b) dropout rate, and (c) target modules



(a) (b) (c)

Figure 7: Ablation studies on (a) per-prompt, (b) prompt ensemble, and (c) prompt content.

Vision Encoder: Tables 3–4 compare vision backbones without and with PPL. PE-Core-G/14 performs best in both settings (93.8%/97.3% mACC w/o and w/ PPL), followed by PE-Core-L/14 (93.1%/96.3%) and CLIP-L/14 Cherti et al. (2023) (92.9%/94.4%), while DINOv2-L/14 Oquab et al. (2023) performs worst (90.5%/93.1%), confirming that language-aligned features generalize better for forgery detection and that PPL consistently boosts accuracy across backbones.

Text Encoder: Table 5 evaluates text encoders paired with PE-Core-G/14. CLIP-bigG/14 Cherti et al. (2023) achieves the best performance (97.3% mACC, 99.8% mAP), followed by CLIP-H/14 Cherti et al. (2023) (96.7%/99.7%), BLIP-L Li et al. (2022) (95.1%/99.5%), and SigLIP-L/16 Zhai et al. (2023) (94.7%/99.1%), indicating CLIP-style contrastive pretraining aligns best with forgery-relevant semantics for AIGI detection.

5.2 LoRA Adaptation for Fine-Tuning

LoRA Rank: As shown in Fig. 6a, accuracy peaks at rank r=8 (97.3% mACC, 99.8% mAP) and falls off on either side: too small a rank (r=2: 95.7%) limits representational capacity, while too large a rank (r=32: 96.4%) invites overfitting. A moderate rank thus offers the best capacity–regularization trade-off for generalized AIGI detection.

LoRA Dropout: A similar trend holds for dropout (Fig. 6b): 0.3 is optimal (97.3% mACC, 99.8% mAP), whereas both under-regularizing (0.1: 96.1%) and over-regularizing (0.9: 95.3%) hurt performance. Moderate stochastic regularization is therefore preferable when adapting a large pre-trained encoder to limited data.

LoRA Target Modules: Fig. 6c compares which modules LoRA is applied to. Adapting the QKV and output projections jointly (qkv_proj) is most effective (97.3% mACC, 99.8% mAP), ahead of adapting QKV alone (96.7%) or extending further into the feed-forward block (qkv_fc1: 97.1%; qkv_fc2: 96.8%). Attention-projection adaptation alone is therefore sufficient, and reaching into the feed-forward layers adds complexity without a corresponding gain.

5.3 Prompt Selection

Prompt Index: Individually, none of the 12 crafted prompts is sufficient on its own (Fig. 7a): the best two, prompts 2 and 6, tie at 95.3% mACC, while the rest span 94.1%–95.2%. Within this narrow 1.2-point band even the weakest prompt (prompt 8, 94.1% mACC, 98.7% mAP) stays competitive, so each template appears to capture a similar, partial view of the forgery signal, the gap between prompts is one of degree, not kind.

Prompt Selection: We next combine the top-performing prompts by count K (Fig. 7b). Accuracy peaks at K=6 (97.3% mACC, 99.8% mAP) and falls off on either side, to 96.7% at K=2 and 96.1%

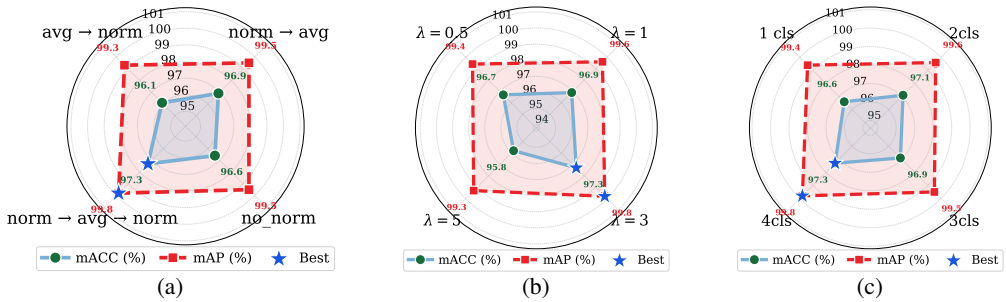


Figure 8: Ablation studies on (a) prototype normalization, (b) loss weight, and (c) training classes.

at K=12; the intermediate points fluctuate by a few tenths, but the envelope is clearly single-peaked. Six prompts capture enough complementary cues from Fig. 7a's strongest templates without diluting the ensemble with redundant or conflicting signals.

Prompt Content: Finally, Fig. 7c isolates what these six prompts contribute beyond their surface wording. Our full formulation (PPL, K=6) reaches 97.3% mACC, ahead of every single-aspect variant: texture_only (95.5%), geometric_only (95.3%), boundary_only (95.1%), single_token (94.4%), and random_captions, which collapses to 92.1% without any forgery-specific content. No single aspect is decisive – the three lie within 0.4 points – yet the full formulation clears the best of them by 1.8. Combining the individually strongest, semantically grounded prompts, not merely adding text, yields the strongest representation.

5.4 Prototype Normalization, Contrastive Weight, and Training Data

Prototype Normalization: Fig. 8a shows normalizing before and after averaging (norm_then_avg_then_norm) performs best (97.3% mACC, 99.8% mAP), ahead of normalizing only before averaging (norm_then_avg: 96.9%), no normalization (no_norm: 96.6%), and averaging before a single final normalization (avg_then_norm: 96.1%). Normalizing before averaging equalizes each prompt's contribution, preventing high-norm prompts from dominating, while the trailing normalization restores unit-norm prototypes for cosine similarity. Averaging first forfeits this equalization, making it the weakest setting.

Loss Weight: Fig. 8b ablates the contrastive loss weight λ. Performance improves steadily from λ=0.5 (96.7% mACC) to a peak at λ=3 (97.3% mACC, 99.8% mAP), then drops sharply when overweighted at λ=5 (95.8%), confirming the contrastive loss is most effective at a moderate strength.

Training Classes: Fig. 8c studies the effect of the number of ProGAN object categories used during training. Performance is robust across all settings, with all configurations at or above 96.6% mACC. Using 4 classes achieves the best results (97.3% mACC, 99.8% mAP), while even a single class yields a strong 96.6% mACC, demonstrating that the proposed method generalizes well even with minimal training diversity.

6 Conclusion

We presented PPL, a framework that replaces fixed single-token and per-sample caption supervision in VLM-based AIGI detection with ensemble-averaged prototype embeddings derived from structured, artifact-aware prompt libraries. By encoding complementary forensic axes, texture, boundary, and semantic cues into stable class prototypes, PPL provides richer contrastive supervision without per-sample annotations. We further showed that pairing the large PE-Core-G14-448 vision encoder with a CLIP-ViT-bigG text encoder yields the best performance. Trained exclusively on ProGAN, PPL achieves 97.3% mACC and 99.8% mAP on 19 unseen generators spanning GANs, diffusion models, and commercial tools, surpassing all prior methods by a clear margin. PPL further generalizes to four additional datasets covering recent generators and real-world transmitted images, achieving 94.1% mACC and 97.7% mAP, and maintains strong robustness under common image perturbations, underscoring its practical applicability beyond curated benchmarks. Ablation studies confirm that prototype-based supervision, normalization strategy, and structured prompt coverage are the primary drivers of generalization.

The prompt library is manually designed, which may introduce bias and limit coverage of emerging synthesis artifacts. Future work will explore automated prompt discovery, multi-source training to improve coverage of challenging generators, and extending PPL to video and audio deepfake detection, where temporal artifacts introduce additional information.

REFERENCES

Mauro Barni, Kassem Kallas, Ehsan Nowroozi, and Benedetta Tondi. Cnn detection of gan-generated face images based on cross-band co-occurrences analysis. In *IEEE international workshop on information forensics and security (WIFS)*, pp. 1–6. IEEE, 2020.

Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024a.

Black Forest Labs. Flux.1-dev. `https://huggingface.co/black-forest-labs/FLUX.1-dev`, 2024b.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European conference on computer vision*, pp. 103–120. Springer, 2020.

Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3291–3300, 2018.

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.

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1511–1520, 2017.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025a.

Yingjian Chen, Lei Zhang, and Yakun Niu. Forgelens: Data-efficient forgery focus for generalizable forgery image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2818–2829, 2023.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul 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*, pp. 8789–8797, 2018.

Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the bar of ai-generated image detection with clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4356–4366, 2024.

Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11065–11074, 2019.

Stefano Dell'Anna, Andrea Montibeller, and Giulia Boato. Truefake: A real world case dataset of last generation fake images also shared on social networks. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2025.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7890–7899, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pp. 3247–3258. PMLR, 2020.

Google DeepMind. Imagen 3. `https://deepmind.google/technologies/imagen-3`, 2024.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10696–10706, 2022.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15733–15744. IEEE, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Yiwen Huang, Aaron Gokaslan, Volodymyr Kuleshov, and James Tompkin. The gan is dead; long live the gan! a modern gan baseline. *Advances in neural information processing systems*, 37: 44177–44215, 2024.

Huawei. Wukong. `https://xihe.mindspore.cn/modelzoo/wukong`, 2022. Accessed: 2026-08-20.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727, 2022.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34:852–863, 2021.

Christos Koutlis and Symeon Papadopoulos. Leveraging representations from intermediate encoder-blocks for synthetic image detection. In *European Conference on Computer Vision*, pp. 394–411. Springer, 2024.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. 2019 ieee. In *CVF International Conference on Computer Vision (ICCV)*, pp. 4219–4228, 2019.

Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Fuli Feng. Improving synthetic image detection towards generalization: An image transformation perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2405–2414, 2025.

Yiheng Li, Zichang Tan, Guoqing Xu, Zhen Lei, Xu Zhou, and Yang Yang. Towards generalizable ai-generated image detection via image-adaptive prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21262–21272, 2026a.

Zhen Li, Mingdeng Cao, Xintao Wang, Zhongang Qi, Ming-Ming Cheng, and Ying Shan. Photomaker: Customizing realistic human photos via stacked id embedding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8640–8650. IEEE, 2024.

Ziqiang Li, Jiazhen Yan, Ziwen He, Kai Zeng, Weiwei Jiang, Lizhi Xiong, and Zhangjie Fu. Is artificial intelligence generated image detection a solved problem? *Advances in neural information processing systems*, 38, 2026b.

Kaiqing Lin, Yuzhen Lin, Weixiang Li, Taiping Yao, and Bin Li. Standing on the shoulders of giants: Reprogramming visual-language model for general deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5262–5270, 2025.

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.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

Midjourney. Midjourney. `https://www.midjourney.com`, 2022. Accessed: 2026-08-20.

Midjourney Team. Midjourney v6.1. `https://www.midjourney.com/home`, 2024.

Lakshmanan Nataraj, Tajuddin Manhar Mohammed, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H Bappy, Amit K Roy-Chowdhury, and BS Manjunath. Detecting gan generated fake images using co-occurrence matrices. *arXiv preprint arXiv:1903.06836*, 2019.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

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.

OpenAI Team. Dall-e 3 ai image generator. `https://dalle3.ai/`, 2024.

Maxime Oquab et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2337–2346, 2019.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Anirudh Sundara Rajan and Yong Jae Lee. Stay-positive: A case for ignoring real image features in fake image detection. In *Proceedings of the International Conference on Machine Learning*, 2025.

Anirudh Sundara Rajan, Utkarsh Ojha, Jedidiah Schloesser, and Yong Jae Lee. Aligned datasets improve detection of latent diffusion-generated images. In *Proceedings of the International Conference on Learning Representations*, 2025.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Kuo Shi, Jie Lu, Shanshan Ye, Guangquan Zhang, and Zhen Fang. Mirage: Multimodal discriminative representation learning for generalizable ai-generated image detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 353–361, 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.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12105–12114, 2023.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In *Proceedings of the AAAI Conference*, pp. 5052–5060, 2024a.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao 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*, pp. 28130–28139, 2024b.

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.

Nusrat Tasnim, Kutub Uddin, and Khalid Mahmood Malik. Ai-generated image detection: An empirical study and future research directions. *arXiv preprint arXiv:2511.02791*, 2025.

Nusrat Tasnim, Khalid Malik, et al. Grex-bench: Benchmarking generalization, robustness, and explainability in ai-generated image detection. *https://doi.org/10.21203/rs.3.rs-8633550/v1*, 2026.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, Anthony Chen, Huaxia Li, Xu Tang, and Yao Hu. Instantid: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*, 2024.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8695–8704, 2020.

Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22445–22455, 2023.

Jevin West and Carl Bergstrom. Which face is real? `https://www.whichfaceisreal.com/`, 2019.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12966–12977. IEEE, 2025.

Yi Wu, Ziqiang Li, Heliang Zheng, Chaoyue Wang, and Bin Li. Infinite-id: Identity-preserved personalization via id-semantics decoupling paradigm. In *European Conference on Computer Vision*, pp. 279–296. Springer, 2024.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, volume 2025, pp. 28240–28264, 2025.

Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for ai-generated image detection. In *Proceedings of the Internation Conference on Learning Representations*, 2025.

Zhiyuan Yan, Jiangming Wang, Zhendong Wang, Peng Jin, Ke-Yue Zhang, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Effort: Efficient orthogonal modeling for generalizable ai-generated image detection. *arXiv preprint arXiv:2411.15633*, 2024.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Xiaohua Zhai et al. Sigmoid loss for language image pre-training. *ICCV*, 2023.

Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11294–11304. IEEE, 2022.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. In *International Journal of Computer Vision*, volume 130, pp. 2337–2348, 2022b.

Ziyin Zhou, Yunpeng Luo, Yuanchen Wu, Ke Sun, Jiayi Ji, Ke Yan, Shouhong Ding, Xiaoshuai Sun, Yunsheng Wu, and Rongrong Ji. Aigi-holmes: Towards explainable and generalizable ai-generated image detection via multimodal large language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18746–18758. IEEE, 2025.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.

A Appendix

This appendix provides additional technical details and supplementary results to complement the main paper. The main paper is self-contained; the appendix offers extended analysis for interested readers. The appendix is organized as follows:

- Section A.1: Training Configurations (Table 6)
- Section A.2: Dataset Descriptions (Table 7)
- Section A.3: State-of-the-Art AIGI Detectors (Table 8)
- Section A.4: Prompt Library (Figure 9)
- Section A.5: Detailed Results on Recent and Real-World Benchmarks (Tables 9- 12)
- Section A.6: Details Robustness Results on UFD Benchmark (Table 13)
- Section A.7: Prompt Design Choices (Table 14)
- Section A.8: Additional Ablation Studies (Figures 10a- 10b)
- Section A.9: Computational Cost (Table 17)
- Section A.10: Broader Impact, Ethics, and Responsible Disclosure

A.1 Training Configurations

Table 6 summarizes the training configuration used for PPL. We build on a large-scale pretrained vision backbone, PE-Core-G14-448, paired with a CLIP-ViT-bigG-14 text encoder, and adapt this frozen backbone using lightweight LoRA modules (rank $r = 8$, dropout 0.3) rather than full fine-tuning, keeping the number of trainable parameters small relative to the backbone. PPL is trained exclusively on real and fake images from four ProGAN object categories, car, cat, chair, and horse, with a batch size of 16, at an input resolution of 448×448 (matching the backbone's native resolution) and a corresponding crop size of 448×448. We optimize with a learning rate of 1×10^{-4}, decayed by a factor of 0.9 every 1000 steps, and apply a class-alignment loss weighted at 3 to encourage the learned representation to remain discriminative for the real/fake distinction while preserving the semantic structure of the pretrained embedding space. All other implementation details, including the optimizer and hardware setup, are reported in Sec. 4.1.

Table 6: Training configuration for PPL.

Hyperparameter	**Value**
Vision encoder	PE-Core-G14-448
Text encoder	CLIP-ViT-bigG-14
Training classes	car, cat, chair, horse
Batch size	16
Learning rate	1×10^{-4}
LR decay frequency	10
LR decay factor	0.9
Input load size	448 $\times$ 448
Crop size	448 $\times$ 448
Loss weight (λ)	3
LoRA rank (r)	8
LoRA dropout	0.3

Table 7: Details of Datasets used for training and cross-generator generalization. The model is trained on ProGAN and evaluated zero-shot on every remaining benchmark, without retraining, threshold calibration, or adaptation.

Dataset	Role	#Gen.	Generators
ProGAN (4-class)	Train	1	ProGAN Karras et al. (2017)
UFD	Eval	19	ProGAN Karras et al. (2017), CycleGAN Zhu et al. (2017), BigGAN Brock et al. (2018), StyleGAN Karras et al. (2019), GauGAN Park et al. (2019), StarGAN Choi et al. (2018), Deepfakes Rossler et al. (2019), SITD Chen et al. (2018), SAN Dai et al. (2019), CRN Chen & Koltun (2017), IMLE Li et al. (2019), Guided Dhariwal & Nichol (2021), LDM-200/-CFG/-100 Rombach et al. (2022), Glide-100/27, -50/27, -100/10 Nichol et al. (2021), DALL-E Ramesh et al. (2021)
AIGC	Eval	9	ADM Dhariwal & Nichol (2021), DALL-E2 Ramesh et al. (2022), GLIDE Nichol et al. (2021), Midjourney Midjourney (2022), SD-XL Podell et al. (2024), SD v1.4/v1.5 Rombach et al. (2022), VQDM Gu et al. (2022), Wukong Huawei (2022)
AIGB	Eval	19	BLIP Li et al. (2022), CommunityAI Li et al. (2026b), DALLE-3 OpenAI Team (2024), FLUX1-dev Black Forest Labs (2024b), GLIDE Nichol et al. (2021), IP-Adapter Ye et al. (2023), Imagen3 Google DeepMind (2024), Infinite-ID Wu et al. (2024), InstantID Wang et al. (2024), Midjourney Midjourney Team (2024), PhotoMaker Li et al. (2024), R3GAN Huang et al. (2024), SD3 Esser et al. (2024), SDXL Podell et al. (2024), SocialRF Li et al. (2026b), StyleGAN3 Karras et al. (2021), StyleGAN-XL Sauer et al. (2022), StyleSwin Zhang et al. (2022), WFIR West & Bergstrom (2019)
AIGH	Eval	10	FLUX Black Forest Labs (2024a), Infinity Han et al. (2025), Janus Wu et al. (2025), Janus-Pro-1B Chen et al. (2025a), Janus-Pro-7B Chen et al. (2025a), LlamaGen Sun et al. (2024), PixArt-XL Chen et al. (2024), SD35-L Esser et al. (2024), Show-o Xie et al. (2025), VAR Tian et al. (2024)
TrueFake	Eval	–	Social-media re-encoded images (Facebook, Twitter, Telegram) Dell'Anna et al. (2025)

A.2 Dataset Descriptions

Table 7 summarizes the datasets used for training and evaluation. PPL is trained exclusively on the 4-class ProGAN Karras et al. (2017) split, following the standard protocol established by prior work Wang et al. (2020), and is evaluated zero-shot, without retraining, threshold calibration, or any form of test-time adaptation, on five independent benchmarks spanning 19 to 57 generators in aggregate. UniversalFakeDetect (UFD) Ojha et al. (2023) provides the canonical cross-generator testbed, covering 19 generators across GAN, low-level vision, and diffusion families: ProGAN, CycleGAN Zhu et al. (2017), BigGAN Brock et al. (2018), StyleGAN Karras et al. (2019), GauGAN Park et al. (2019), and StarGAN Choi et al. (2018) for GAN-based synthesis; Deepfakes Rossler et al. (2019), SITD Chen et al. (2018), SAN Dai et al. (2019), CRN Chen & Koltun (2017), and IMLE Li et al. (2019) for low-level and perceptual manipulations; and Guided Dhariwal & Nichol (2021), LDM Rombach et al. (2022), Glide Nichol et al. (2021), and DALL-E Ramesh et al. (2021) for diffusion-based synthesis. AIGC Yan et al. (2025) extends evaluation to nine additional generators (ADM, DALL-E2 Ramesh et al. (2022), GLIDE, Midjourney Midjourney (2022), SD-XL Podell et al. (2024), SD v1.4/v1.5, VQDM Gu et al. (2022), Wukong Huawei (2022)), emphasizing more recent text-to-image diffusion pipelines than those covered by UFD.

AIGB Li et al. (2026b) substantially broadens the evaluation scope to 19 sources, combining recent text-to-image diffusion models (FLUX1-dev Black Forest Labs (2024b), SD3 Esser et al. (2024), SDXL, DALLE-3 OpenAI Team (2024), Midjourney Midjourney Team (2024), GLIDE, Imagen3 Google DeepMind (2024)), GAN variants (R3GAN Huang et al. (2024), StyleGAN3 Karras et al. (2021), StyleGAN-XL Sauer et al. (2022), StyleSwin Zhang et al. (2022), WFIR West & Bergstrom (2019)), identity-conditioned personalized generation methods (BLIP Li et al. (2022), IP-Adapter Ye et al. (2023), InstantID Wang et al. (2024), PhotoMaker Li et al. (2024), Infinite-ID Wu et al. (2024)), and, critically, two in-the-wild subsets, CommunityAI, sourced from AI-art platforms such as ArtStation, Civitai, and Liblib, and SocialRF, sourced from social media platforms, that introduce realistic distributional shifts absent from curated benchmarks. AIGH combines ten recent autoregressive and hybrid generative architectures (FLUX Black Forest Labs (2024a), Infinity Han et al. (2025), Janus Wu et al. (2025), Janus-Pro-1B/7B Chen et al. (2025a), LlamaGen Sun et al. (2024), PixArt-XL Chen et al. (2024), SD35-L, Show-o Xie et al. (2025), VAR Tian et al. (2024))

Table 8: Summary of baseline AIGI detectors compared against PPL. Foundation model-based methods leverage large-scale vision-language pretraining, while traditional methods rely on low-level artifact statistics.

Method	Venue (Year)	Key Mechanism	Backbone
NPR Tan et al. (2024b)	CVPR (2024)	Neighboring pixel relationships targeting up-sampling artifacts	ResNet-50
CNND Wang et al. (2020)	CVPR (2020)	CNN with augmentation for generalization	ResNet-50
FrqNet Tan et al. (2024a)	AAAI (2024)	Frequency-domain spectral inconsistency analysis	ResNet-50
UniD Ojha et al. (2023)	CVPR (2023)	Frozen CLIP features with linear probe	CLIP ViT-L/14
EFFT Yan et al. (2024)	ICML (2025)	Orthogonal decomposition of forgery-relevant/irrelevant subspaces	CLIP ViT-L/14
FatF Liu et al. (2024)	CVPR (2024)	Forgery-aware adapters + frequency analysis	CLIP ViT-L/14
RINE Koutlis & Papadopoulos (2024)	ECCV (2024)	Aggregates intermediate representations	CLIP ViT-L/14
AIDE Yan et al. (2025)	ICLR (2025)	Multi-feature-extractor ensemble	CLIP + ResNet
C2PC Tan et al. (2025)	AAAI (2025)	Category-common prompts injected into CLIP	CLIP ViT-L/14
ForL Chen et al. (2025b)	ICCV (2025)	Data-efficient representation learning	CLIP ViT-L/14
SAFE Li et al. (2025)	KDD (2025)	Image transformation perspective for generalization	CLIP ViT-L/14
MiraGe Shi et al. (2025)	ACM MM (2025)	Multimodal discriminative representation learning	CLIP ViT-L/14
IAPL Li et al. (2026a)	CVPR (2026)	Image-adaptive prompts updated at inference	CLIP ViT-L/14
PPL (Ours)	–	Ensemble-averaged prompt prototypes	PE-Core-G/14 + CLIP-G/14

that diverge structurally from the GAN- and diffusion-based generators seen during training, probing generalization to fundamentally different generative paradigms. Finally, TrueFake Dell'Anna et al. (2025) departs from synthetic-source generalization entirely, instead evaluating robustness to real-world social-media transmission artifacts by collecting images re-encoded through Facebook, Twitter, and Telegram.

Collectively, these five benchmarks probe three largely orthogonal axes of generalization: (i) generator architecture, spanning GAN-based, diffusion-based, and autoregressive/hybrid synthesis; (ii) generation task, ranging from unconditional and class-conditional synthesis to open-vocabulary text-to-image generation and identity-conditioned personalization; and (iii) deployment realism, moving from curated laboratory benchmarks to in-the-wild AI-art community content and, ultimately, images degraded through real social-media transmission. Evaluating a single ProGAN-trained model across all three axes without any retraining or adaptation constitutes a stringent test of whether a detector has learned generator-agnostic forgery cues, as opposed to overfitting to artifacts specific to its training distribution.

A.3 State-of-the-Art AIGI Detectors

Table 8 summarizes the thirteen baseline detectors compared against PPL, spanning traditional CNN- and frequency-based methods and recent foundation model-based approaches. Traditional detectors, including NPR Tan et al. (2024b), CNND Wang et al. (2020), and FrqNet Tan et al. (2024a), exploit low-level artifact statistics, such as image gradients, pixel-neighborhood correlations, and frequency-domain spectral irregularities, extracted from CNN backbones trained directly on GAN outputs. While effective within GAN families, these methods degrade sharply on diffusion and commercial generators, whose artifact signatures diverge substantially from GAN-specific up-sampling patterns. Foundation model-based detectors, including UniD Ojha et al. (2023), EFFT Yan et al. (2024), FatF Liu et al. (2024), RINE Koutlis & Papadopoulos (2024), AIDE Yan et al. (2025), C2PC Tan et al. (2025), ForL Chen et al. (2025b), SAFE Li et al. (2025), MiraGe Shi et al. (2025), and IAPL Li et al. (2026a), instead build on large-scale CLIP pretraining, adapting the vision encoder via linear probing, adapters, or LoRA, and supervising with fixed class tokens, category-common prompts, or test-time-adaptive prompts. As Table 8 shows, most foundation-model baselines use CLIP ViT-L/14; of these, only a few (e.g., FatF, C2PC, IAPL) use text supervision, typically a class token, category prompt, or per-sample caption. PPL departs from this design by pairing a substantially larger vision encoder with ensemble-averaged prompt prototypes spanning multiple forensic axes, yielding richer and more stable contrastive supervision than any single-token or single-prompt alternative.

A.4 Prompt Library

The prompt library consists of two pools of twelve prompts, one per class, each organized into three groups of four (Fig. 9). The real pool describes properties that arise from the physics of camera capture, grouped into texture realism (RP-1–4: natural grain, film grain, sensor noise, and consistent material properties), boundary optics (RP-5–8: optical and motion blur, bokeh, chromatic aberration, and vignetting), and geometric effects (RP-9–12: depth of field, lens distortion and curvature, perspective, specular reflectance, depth discontinuities, and shadow gradients). These prompts use the

Real Prompts (RP) Class

Target properties of physical optics and photochemical capture absent in synthesized images:

Texture Realism

RP-1: "a real photograph with natural grain and authentic lighting"

RP-2: "genuine photo with film grain and consistent natural shadows"

RP-3: "a real camera image with natural noise and consistent material properties"

RP-4: "a real image captured by a digital camera with sensor noise"

Boundary Optics

RP-5: "a camera photo with real optical blur and natural skin texture"

RP-6: "photograph taken with a camera showing natural imperfections and real bokeh"

RP-7: "natural photograph with authentic color fringing and chromatic aberration"

RP-8: "a genuine photo with real motion blur and natural vignetting"

Geometric Effects

RP-9: "an authentic photograph with natural depth of field and lens distortion"

RP-10: "camera-captured image with authentic perspective and lens curvature"

RP-11: "real photograph showing natural specular highlights and material reflectance"

RP-12: "authentic image with real depth discontinuities and natural shadow gradients"

Fake Prompts (FP) Class

Target artifact types recurring across GAN and diffusion generation pipelines:

General Synthesis Artifacts

FP-1: "an AI generated image with synthetic texture and deepfake artifacts"

FP-2: "a computer generated image with smooth texture and unnatural edges"

FP-3: "a machine-generated image with repeating texture patterns and grid artifacts"

FP-4: "an AI-generated face with symmetric artifacts and unnaturally perfect skin"

GAN Artifacts

FP-5: "a GAN generated face with unnatural skin boundary and synthetic hair"

FP-6: "a deepfake image with unnatural skin smoothness and blurred hair boundaries"

FP-7: "a synthetic image with spectral inconsistencies and unnatural frequencies"

FP-8: "a GAN image with color bleeding artifacts at semantic boundaries"

Diffusion Artifacts

FP-9: "a diffusion model image with blended artifacts and inconsistent lighting"

FP-10: "an AI synthesized image with inconsistent specular highlights and fake depth"

FP-11: "a generated image with blurred background transitions and fake bokeh"

FP-12: "a diffusion-generated image with oversmoothed surfaces"

Figure 9: The PPL prompt library. From these 24 prompts, $K=6$ per class are selected to form the real (left, blue) and fake (right, red) class prototypes. Real prompts are grouped into texture realism (RP-1–4), boundary optics (RP-5–8), and geometric effects (RP-9–12). Fake prompts are grouped into general synthesis artifacts (FP-1–4), GAN-specific artifacts (FP-5–8), and diffusion-specific artifacts (FP-9–12). Constructing the prototypes requires no annotation beyond the ProGAN real/fake labels.

vocabulary of photography and optics and emphasize the imperfections that signal authenticity. The fake pool describes artifacts of generative pipelines, grouped into general synthesis artifacts that name no specific generator family (FP-1–4: synthetic and repeating textures, grid artifacts, unnatural edges, and unnaturally perfect or symmetric faces), GAN-specific artifacts (FP-5–8: skin and hair boundary failures, spectral inconsistencies, and color bleeding at semantic boundaries), and diffusion-specific artifacts (FP-9–12: blended transitions, inconsistent lighting and specular highlights, fake depth and bokeh, and over-smoothed surfaces). Together, the two pools span cues from low-level statistics, such as noise, grain, and frequency content, to mid-level scene properties, such as depth, lighting, and specular highlights.

PPL does not use all 24 prompts. Following the ablation in Sec. 5.3, $K=6$ prompts per class are selected and averaged into each class prototype (Eq. 6). The prompts serve only as training targets: the text encoder and prototypes are discarded after training, and detection at inference relies on the image branch alone.

A.5 Detailed Results on Recent and Real-World Benchmarks

This section analyzes the per-generator results on the four benchmarks beyond UFD (Section A.2), each probing a different axis of generalization: recent text-to-image diffusion generators (AIGC); recent diffusion, GAN, identity-conditioned, and in-the-wild sources (AIGB); recent autoregressive and diffusion-transformer generators (AIGH); and real-world social-media transmission (TrueFake).

Table 9: ACC (%) and AP (%) on the AIGC Yan et al. (2025) benchmark. To assess true generalization, all detectors use the same ProGAN Karras et al. (2017)-trained weights from Tables 1.

	NPR		CNND		FrqNet		UniD		EFFT		FatF		RINE		AIDE		C2PC		ForL		SAFE		MiraGe		IAPL		PPL	
Dataset	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**
ADM	82.6	89.9	60.2	76.2	78.8	85.0	66.9	87.1	93.7	93.7	79.5	95.0	74.6	96.3	**93.8**	**97.1**	65.1	92.2	79.9	96.9	85.4	91.2	91.2	96.7	90.4	95.3	91.7	94.5
DALL-E2	94.6	97.7	50.4	53.6	59.2	61.8	50.7	63.2	90.7	94.8	71.1	91.6	54.8	89.8	81.9	88.4	69.6	97.3	87.3	98.9	92.1	**99.8**	86.4	90.2	89.1	94.2	**96.3**	97.3
GLIDE	90.3	95.5	56.9	71.5	77.5	83.5	62.5	84.3	92.2	92.0	89.0	97.7	80.7	97.9	95.2	98.6	89.0	99.0	95.0	**99.7**	91.4	95.2	92.4	94.8	94.7	97.4	**99.1**	**99.7**
Midjourney	76.8	85.9	50.8	58.7	62.5	66.8	56.3	74.6	82.1	96.8	55.5	73.8	57.1	87.7	89.3	95.0	54.9	80.4	58.3	82.2	87.5	92.3	85.2	94.1	**93.1**	**97.4**	87.6	98.0
SD-XL	85.1	94.0	54.9	69.8	80.5	87.5	60.1	88.0	85.4	97.9	71.0	90.5	64.3	96.3	91.2	95.4	63.7	91.4	73.6	91.8	76.9	84.1	86.4	93.6	91.4	95.1	**97.8**	**99.9**
SD v1.4	90.4	94.7	51.1	59.2	85.9	91.1	63.8	86.6	95.3	95.2	88.1	98.4	83.9	98.4	84.7	90.2	84.2	98.7	94.8	99.7	85.4	96.3	93.2	97.1	90.3	97.8	**96.7**	**99.1**
SD v1.5	90.5	94.4	51.2	60.0	85.2	90.5	63.5	86.2	96.1	93.8	87.9	98.1	83.3	98.3	78.3	95.2	84.3	98.6	94.6	99.6	81.9	96.8	89.4	95.9	93.1	98.5	**94.3**	**99.9**
VQDM	90.0	94.3	55.1	67.6	77.0	83.1	85.4	96.7	88.7	94.7	88.0	98.5	89.8	**99.2**	92.5	96.6	79.1	97.9	86.1	98.2	84.7	94.8	90.5	96.2	88.9	96.1	**97.1**	98.4
Wukong	89.3	93.3	51.1	57.0	85.4	90.4	71.0	91.3	93.3	97.8	88.1	98.6	84.9	98.6	88.9	96.3	81.9	98.2	92.3	99.4	87.4	93.8	93.1	97.4	92.4	96.8	**99.9**	**99.9**
Average	87.7	93.3	53.5	63.7	76.9	82.2	64.5	84.2	90.8	95.2	79.8	93.6	74.8	95.8	88.3	94.8	74.6	94.9	84.7	96.3	85.9	93.8	89.8	95.1	91.5	96.5	**95.6**	**98.3**

Table 10: ACC (%) and AP (%) on the AIGB Li et al. (2026b) benchmark. To assess true generalization, all detectors use the same ProGAN Karras et al. (2017)-trained weights as in Table 1.

	NPR		CNND		FrqNet		UniD		EFFT		FatF		RINE		AIDE		C2PC		ForL		SAFE		MiraGe		IAPL		PPL	
Dataset	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**
BLIP	89.7	91.5	52.8	72.0	86.7	91.5	58.5	73.6	90.0	99.3	82.7	92.2	90.7	98.7	97.0	99.8	84.9	93.8	90.5	96.6	99.4	**99.9**	97.9	**99.9**	92.6	97.6	**99.7**	**99.9**
CommunityAI	55.2	56.7	50.5	56.2	60.9	64.0	51.4	57.9	51.0	51.8	51.9	71.3	51.2	58.7	53.9	50.5	51.0	54.5	51.4	45.6	54.5	61.2	**84.6**	**95.2**	52.0	48.1	78.6	86.0
DALLE-3	66.6	64.4	47.5	41.5	47.1	40.0	53.5	53.8	50.3	35.4	50.0	46.7	51.0	51.3	44.5	41.8	64.4	75.4	48.5	41.0	41.9	39.3	94.9	**98.8**	55.8	57.9	**96.3**	98.0
FLUX1-dev	79.5	82.5	50.7	52.9	84.2	92.4	48.4	37.1	53.7	56.8	48.2	41.8	51.5	57.3	85.5	89.2	47.5	73.9	60.5	64.9	91.4	96.4	**93.0**	**98.1**	45.2	42.6	85.0	91.0
GLIDE	79.8	81.3	54.9	66.9	83.5	90.6	59.9	72.9	93.5	98.1	84.4	92.2	78.2	93.4	90.8	97.1	85.2	93.4	89.5	95.6	90.5	94.1	84.7	94.5	83.6	92.4	**99.0**	**99.6**
IP-Adapter	79.0	79.8	50.1	53.4	76.8	87.4	52.6	56.7	64.7	82.5	67.1	74.1	60.0	77.4	91.2	96.8	55.1	65.9	72.4	81.4	88.9	93.3	97.4	99.7	64.6	69.0	**99.3**	**99.9**
Imagen3	79.5	82.0	57.0	63.8	83.6	92.1	50.7	46.3	50.6	56.3	46.9	43.1	49.1	51.3	88.9	91.9	45.2	73.6	49.5	52.5	**89.6**	92.7	85.7	**94.8**	41.2	40.5	75.1	88.7
Infinite-ID	76.7	68.4	49.6	53.1	73.7	84.3	62.6	73.9	67.7	78.6	83.3	91.2	64.5	77.7	93.0	94.7	85.2	93.7	65.7	70.5	90.9	95.3	97.8	**100.0**	84.7	93.6	**98.6**	99.4
InstantID	68.8	68.9	49.6	64.5	92.4	96.3	51.6	55.9	52.9	55.2	69.3	77.3	48.8	44.5	92.0	95.8	86.8	93.9	49.6	49.5	91.7	96.6	**97.8**	**99.9**	87.2	94.9	90.2	93.7
Midjourney	61.3	65.9	46.7	39.0	70.3	82.5	52.3	47.4	61.8	66.9	47.5	43.5	52.8	55.4	79.9	78.3	51.2	50.2	62.3	67.4	76.1	81.5	**96.6**	**99.7**	41.8	43.0	84.9	89.5
PhotoMaker	57.8	52.8	47.5	42.8	78.0	87.2	51.9	54.4	51.9	39.7	46.4	37.4	49.3	46.8	77.8	81.2	50.6	52.0	46.0	34.4	**91.0**	94.6	84.1	**95.3**	51.4	47.7	81.3	86.7
R3GAN	77.7	70.5	72.0	87.3	67.7	80.9	78.5	90.5	99.0	**99.9**	94.3	99.3	96.9	99.8	93.8	99.5	95.4	99.8	93.3	99.6	91.6	93.3	96.0	99.0	89.6	99.1	**99.4**	**99.9**
SD3	78.3	82.0	58.3	70.0	61.8	73.5	56.3	64.7	74.0	89.6	62.8	68.5	67.7	85.3	94.0	97.3	53.5	65.6	74.6	84.9	92.4	96.1	97.7	**100.0**	62.3	69.0	**99.3**	99.8
SDXL	78.3	82.0	60.7	73.5	90.4	94.3	59.1	70.4	77.4	93.8	75.1	82.9	72.2	88.3	93.2	95.2	59.4	76.0	79.4	88.7	93.3	97.7	97.7	**100.0**	74.4	82.5	**99.6**	**100.0**
SocialRF	54.8	56.3	50.9	48.8	54.9	60.4	54.6	57.8	56.9	64.8	56.9	65.1	56.0	59.8	57.6	65.7	56.4	62.3	59.5	73.0	58.4	66.0	81.8	89.9	59.3	64.8	**90.9**	**94.4**
StyleGAN3	79.0	81.0	74.1	88.3	83.8	92.3	64.2	77.1	96.9	99.5	91.5	96.8	91.3	97.9	78.2	86.4	93.8	98.5	95.3	99.2	90.4	92.6	59.2	81.6	90.3	98.6	**99.6**	**100.0**
StyleGAN-XL	75.8	72.6	57.2	75.3	95.9	96.9	80.5	92.7	91.7	99.3	93.5	98.0	96.5	99.3	92.0	95.9	94.1	98.1	91.2	96.2	91.0	94.7	48.1	37.2	88.9	98.3	**99.4**	**99.9**
StyleSwin	77.0	79.7	79.2	91.4	94.6	97.6	84.1	94.8	98.3	99.9	93.9	99.8	96.5	99.5	92.5	96.8	85.4	99.9	93.5	98.7	91.6	95.8	52.5	63.6	89.6	99.9	**99.6**	**100.0**
WFIR	65.8	65.3	86.8	94.8	50.9	96.9	87.1	97.3	90.2	99.8	88.0	98.5	97.0	99.5	94.7	99.6	84.9	99.5	97.7	**100.0**	51.9	51.7	51.5	83.7	79.0	97.6	**99.5**	**100.0**
Average	72.7	72.8	57.7	65.0	75.6	84.3	60.9	67.1	72.2	77.2	70.2	74.7	69.5	75.9	77.1	79.8	70.0	80.0	72.1	75.8	82.4	85.9	84.2	91.1	70.2	75.6	**93.4**	**96.1**

Table 11: ACC (%) and AP (%) on the AIGH Zhou et al. (2025) benchmark. To assess true generalization, all detectors use the same ProGAN Karras et al. (2017)-trained weights as in Table 1.

	NPR		CNND		FrqNet		UniD		EFFT		FatF		RINE		AIDE		C2PC		ForL		SAFE		MiraGe		IAPL		PPL	
Dataset	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**
FLUX	79.1	88.4	51.5	60.8	74.9	82.2	51.6	69.7	61.2	85.3	54.6	61.7	56.3	92.5	74.1	86.3	54.4	81.0	65.9	82.6	79.4	84.6	77.2	84.7	54.1	59.6	**89.3**	**97.3**
Infinity	79.8	89.1	49.5	57.7	86.2	91.8	56.9	82.9	83.1	95.2	53.9	55.9	65.6	90.8	**97.2**	**99.5**	59.3	88.8	85.6	98.0	86.9	93.2	94.9	99.1	62.9	70.7	96.0	98.0
Janus	30.1	33.4	49.0	57.7	43.2	37.8	70.6	94.1	50.5	53.6	48.8	38.3	66.1	87.6	55.1	81.0	53.0	59.3	51.1	43.4	67.9	74.5	88.5	94.9	54.6	46.9	**99.3**	**99.6**
Janus-Pro-1B	29.8	37.6	48.7	52.6	42.9	36.2	68.2	92.7	50.6	56.1	48.6	36.4	69.4	88.3	55.3	79.7	62.0	80.2	52.8	52.1	75.1	88.3	96.3	98.8	58.3	56.3	**99.5**	**99.7**
Janus-Pro-7B	29.8	39.6	48.3	44.6	42.7	35.0	59.4	86.7	54.1	75.4	48.8	32.6	57.3	95.2	50.5	58.2	51.0	53.4	50.5	37.7	81.6	90.8	88.7	95.9	50.4	34.8	**99.2**	**99.5**
LlamaGen	79.5	88.3	52.0	63.5	73.2	82.7	75.2	94.5	97.5	99.9	84.2	95.0	93.4	99.8	98.9	99.9	71.5	97.8	98.4	99.9	94.3	98.4	96.3	98.5	93.4	98.5	**99.9**	**100.0**
PixArt-XL	78.7	85.6	48.9	53.1	68.8	76.9	53.9	76.8	69.7	89.9	67.1	78.9	59.6	93.4	96.1	99.1	64.3	93.3	72.2	86.0	97.7	99.4	95.4	99.2	67.1	77.0	**99.4**	**99.8**
SD35-L	79.4	88.0	63.4	84.0	54.8	59.4	74.0	94.6	79.2	94.8	71.0	86.6	72.1	88.7	**99.1**	**99.9**	58.9	92.6	76.8	97.0	95.7	99.5	97.0	99.1	60.0	78.2	97.5	99.0
Show-o	79.6	89.1	53.4	59.8	90.8	95.9	57.9	84.4	90.6	99.4	73.2	86.8	75.9	94.7	95.4	99.2	64.4	91.5	93.5	99.5	91.7	98.3	94.8	98.3	65.9	78.3	**99.9**	**100.0**
VAR	79.3	87.9	51.1	57.2	88.9	92.8	66.6	91.5	91.5	97.3	90.2	97.3	80.1	91.1	95.4	99.6	84.1	97.3	90.7	98.2	95.3	**99.8**	80.4	95.6	85.6	95.7	**98.6**	99.0
Average	64.5	72.7	51.6	58.3	66.6	69.1	63.4	86.8	72.8	84.7	64.0	66.9	69.6	92.2	81.7	90.2	62.3	83.5	73.8	79.4	86.6	92.7	91.0	96.3	65.2	69.6	**97.8**	**99.2**

Table 12: ACC (%) and AP (%) on the TrueFake Dell'Anna et al. (2025) benchmark. To assess true generalization, all detectors use the same ProGAN Karras et al. (2017)-trained weights as in Table 1.

	NPR		CNND		FrqNet		UniD		EFFT		FatF		RINE		AIDE		C2PC		ForL		SAFE		MiraGe		IAPL		PPL	
Dataset	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**	**ACC**	**AP**
Facebook	41.7	70.8	36.9	76.2	34.6	75.4	33.8	68.7	33.4	67.1	35.4	68.1	34.7	78.7	38.9	75.0	60.6	78.6	33.8	76.2	33.4	67.9	60.5	86.9	48.9	75.8	**85.7**	**97.3**
Telegram	45.0	77.1	37.7	80.1	34.8	70.5	34.9	70.5	33.6	69.0	43.1	74.6	39.4	81.0	35.0	73.6	57.3	79.8	34.3	72.9	33.4	67.9	62.4	89.3	56.2	79.2	**92.3**	**97.6**
Twitter	58.5	81.4	40.7	81.4	46.5	81.3	36.4	69.9	42.2	77.5	52.0	81.5	49.8	84.0	50.4	84.8	62.1	81.6	44.0	80.1	49.3	81.8	67.8	90.9	60.0	81.7	**91.1**	**96.9**
Average	48.4	76.4	38.4	79.2	38.6	75.7	35.0	69.7	36.4	71.2	43.5	74.7	41.3	81.2	41.4	77.8	60.0	80.0	37.4	76.4	38.7	72.5	63.6	89.0	55.0	78.9	**89.7**	**97.3**

For each benchmark, we identify where PPL's gains are largest and where existing detectors remain more competitive. All detectors use the same ProGAN-trained weights as in Table 1.

A.5.1 Evaluation on AIGC Benchmark

Table 9 reports results on the nine generators of AIGC. PPL achieves the best ACC on six of them (DALL-E2, GLIDE, SD-XL, SD v1.4, VQDM, and Wukong), reaching 99.1% and 99.9% ACC on GLIDE and Wukong, and attains AP of at least 99.1% on five generators (GLIDE, SD-XL, SD v1.4, SD v1.5, and Wukong). It obtains the highest average ACC/AP of 95.6%/98.3%, exceeding the second-best method, IAPL (91.5%/96.5%), by +4.1 ACC and +1.8 AP. PPL is less competitive in ACC on three generators. On ADM, it ranks third in ACC (91.7% vs. 93.8% for AIDE and 93.7% for EFFT), and its AP (94.5%) trails six other methods, making ADM the only generator where both metrics lag. On Midjourney and SD v1.5, PPL also ranks third in ACC (87.6% vs. 93.1% for IAPL and 89.3% for AIDE on Midjourney; 94.3% vs. 96.1% for EFFT and 94.6% for ForL on SD v1.5), yet it achieves the highest AP on both (98.0% and 99.9%). This gap between ACC and AP indicates that PPL still ranks these images correctly, but its fixed decision threshold does not transfer well to these generators.

A.5.2 Evaluation on AIGB Benchmark

Table 10 reports results on the nineteen sources of AIGB, which include recent diffusion models (DALLE-3, FLUX1-dev, GLIDE, Imagen3, Midjourney, SD3, SDXL), GAN-based generators (R3GAN, StyleGAN3, StyleGAN-XL, StyleSwin, WFIR), identity- and subject-conditioned generators (BLIP, IP-Adapter, Infinite-ID, InstantID, PhotoMaker), and two in-the-wild sources (CommunityAI and SocialRF). AIGB is markedly harder than AIGC: eleven of the thirteen baselines average

below 80% ACC, and many fall to near-chance ACC on DALLE-3, Imagen3, PhotoMaker, CommunityAI, and SocialRF. PPL achieves the best ACC on thirteen of the nineteen sources and the highest average ACC/AP of 93.4%/96.1%, exceeding the second-best method, MiraGe (84.2%/91.1%), by +9.2 ACC and +5.0 AP. On the two in-the-wild sources, the results are mixed: PPL leads clearly on SocialRF (90.9% vs. 81.8% for MiraGe) but trails MiraGe on CommunityAI (78.6% vs. 84.6%). PPL is also outperformed on Imagen3 (75.1% vs. 89.6% for SAFE), FLUX1-dev (85.0% vs. 93.0% for MiraGe), Midjourney (84.9% vs. 96.6% for MiraGe), PhotoMaker (81.3% vs. 91.0% for SAFE), and InstantID (90.2% vs. 97.8% for MiraGe). These results indicate that PPL's gains are strongest on GAN-based, identity-conditioned, and open-source diffusion generators, whereas several commercial and community-generated sources remain challenging.

A.5.3 Evaluation on AIGH Benchmark

Table 11 reports results on the ten generators of AIGH, which include autoregressive and unified multimodal models (Infinity, Janus, Janus-Pro-1B/7B, LlamaGen, Show-o, VAR) as well as diffusion and flow-based transformers (FLUX, PixArt-XL, SD35-L). Many baselines degrade sharply on this benchmark: NPR, CNND, UniD, FatF, and C2PC average below 65% ACC, and most baselines fall to near or below chance on the Janus family, with NPR dropping to about 30% ACC. PPL achieves the best ACC on eight of the ten generators and the highest average ACC/AP of 97.8%/99.2%, exceeding the second-best method, MiraGe (91.0%/96.3%), by +6.8 ACC and +2.9 AP. The largest gains are on the Janus family, where PPL reaches 99.2–99.5% ACC compared with at most 96.3% for any baseline. PPL is outperformed only by AIDE on Infinity (96.0% vs. 97.2%) and SD35-L (97.5% vs. 99.1%), and its lowest accuracy is on FLUX (89.3%), which is nonetheless the best among all methods. These results indicate that PPL transfers to autoregressive and transformer-based generators despite being trained only on ProGAN images.

A.5.4 Evaluation on TrueFake Benchmark

Table 12 reports results on TrueFake, which contains real and synthetic images shared through Facebook, Telegram, and Twitter, each applying platform-specific re-compression and re-encoding. This is the most challenging of the four benchmarks for existing detectors: ten of the thirteen baselines score below 50% average ACC, while their AP remains in the 69.7–81.2% range. The gap between ACC and AP indicates that platform processing shifts the score distribution relative to the fixed decision threshold, rather than removing the discriminative signal entirely. PPL achieves the best ACC and AP on all three platforms, with an average of 89.7% ACC and 97.3% AP, exceeding the second-best method, MiraGe (63.6%/89.0%), by +26.1 ACC and +8.3 AP. The ACC margin over MiraGe is consistently large across platforms (+25.2 on Facebook, +29.9 on Telegram, and +23.2 on Twitter), suggesting that PPL's robustness is not specific to the processing pipeline of any single platform.

A.6 Detailed Robustness Results on UFD Benchmark

Table 13 reports the mean ACC and AP over the UFD generators for all fourteen perturbation settings, extending the summary in Section 4.3. On clean data, most CLIP-based detectors are competitive: eight of the thirteen baselines exceed 95% AP, and the strongest reach 95.2–95.6% ACC (EFFT and IAPL), compared with 97.3%/99.8% for PPL. Clean performance alone therefore separates the methods only modestly.

Under single perturbations, the baselines diverge sharply. NPR, FrqNet, and SAFE fall to near-chance ACC ($\approx$50%) under every JPEG setting and every noise level, and under blur from σ=1.0 onward, with AP dropping to a similar level (44–61%). This indicates that the signal these methods rely on is largely removed, consistent with a dependence on high-frequency artifacts. Other baselines lose accuracy while retaining considerably higher AP: EFFT, for example, falls from 95.2% to 53.7% ACC at Q=65 (a loss of 41.5 points) while keeping 76.5% AP, and to 56.1% ACC at noise σ=0.05 while keeping 88.6% AP. For these methods, the perturbations shift the score distribution relative to the fixed decision threshold more than they destroy the ranking signal. IAPL, the strongest baseline on clean data, similarly falls to 61.0% ACC at Q=65. Blur degrades most baselines progressively, with losses of 10–31 ACC points between σ=0.5 and σ=2.0 (CNND is the exception, losing only 2.4 points from an already low level). Noise produces less uniform behavior: some methods degrade steadily (EFFT loses 18.7 points and ForL 17.2 points from σ=0.01 to 0.05), whereas others remain nearly flat (UniD) or even improve slightly (AIDE, MiraGe). Under the composed settings, the strongest baselines reach 83.0% (JPEG+Blur, IAPL), 81.0% (JPEG+Noise, MiraGe), 78.1% (Blur+Noise, MiraGe), and 78.3% ACC (All, MiraGe).

PPL achieves the best ACC and AP in every one of the fourteen conditions and never falls below

Table 13: Robustness under commonly used image perturbations (ACC / AP, %) on UFD benchmark. To assess true generalization, all detectors use the same ProGAN Karras et al. (2017)-trained weights as in Table 1. **Bold** indicates the best result and underline the second-best result per row.

Condition	NPR		CNND		FrqNet		UniD		EFFT		FatF		RINE		AIDE		C2PC		ForL		SAFE		MiraGe		IAPL		**PPL**	
	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP
Clean	87.6	92.8	69.6	83.6	83.0	91.9	81.4	90.1	95.2	99.4	90.6	98.1	91.3	98.8	86.6	96.0	93.8	98.6	91.8	98.9	90.5	92.4	92.9	98.3	95.6	99.3	**97.3**	**99.8**
JPEG Q=90	50.7	46.2	63.0	81.0	49.9	50.8	76.0	91.8	71.1	94.2	78.6	91.9	82.6	96.9	62.6	89.5	83.7	96.1	77.9	93.7	49.7	55.1	79.6	91.4	80.8	92.6	**97.3**	**99.5**
JPEG Q=75	50.7	45.5	62.1	80.3	49.9	47.8	75.8	90.1	58.3	82.2	65.9	85.0	77.1	94.5	64.1	88.8	83.5	94.9	65.3	85.1	49.9	56.3	82.8	92.9	66.8	77.6	**97.0**	**98.9**
JPEG Q=65	49.6	44.4	62.1	79.5	49.8	45.8	73.4	88.4	53.7	76.5	59.8	80.1	72.2	93.6	70.3	90.4	78.1	91.8	61.2	80.7	49.7	57.6	82.0	92.6	61.0	80.7	**94.9**	**98.1**
Blur σ=0.5	74.9	79.3	70.9	86.6	82.4	91.1	79.5	92.0	94.2	98.7	90.2	98.4	90.3	97.7	85.1	94.1	92.7	98.7	92.0	98.2	77.1	83.0	80.5	91.2	92.0	98.3	**97.1**	**99.6**
Blur σ=1.0	56.5	57.5	69.9	84.5	56.7	60.5	75.0	88.4	88.2	95.6	84.1	96.0	85.0	94.2	83.5	92.0	86.4	96.7	77.7	94.4	57.1	60.8	77.0	89.9	84.8	96.8	**96.5**	**99.3**
Blur σ=2.0	53.2	53.4	68.5	80.6	51.7	51.7	69.6	81.0	79.5	88.5	74.0	87.3	78.7	87.6	74.3	86.1	77.6	90.4	66.1	86.1	52.7	46.4	60.6	82.5	74.7	90.2	**94.7**	**98.5**
Noise σ=.01	48.7	48.3	64.8	72.4	51.4	54.4	82.2	93.4	74.8	95.6	84.5	94.6	84.7	96.2	70.5	81.4	83.7	96.1	79.2	94.3	42.2	45.4	74.5	87.1	84.9	94.8	**96.7**	**99.6**
Noise σ=.03	48.1	59.9	51.8	71.1	50.1	50.6	80.7	91.2	61.1	91.6	79.8	92.4	79.0	94.1	70.7	90.6	72.3	94.2	67.2	93.1	53.7	53.7	78.3	89.0	78.9	93.1	**96.3**	**98.8**
Noise σ=.05	49.7	63.1	50.4	68.2	50.1	49.2	79.7	88.9	56.1	88.6	74.4	89.0	77.6	92.1	73.6	88.7	67.9	90.9	62.0	85.0	49.9	45.1	78.2	88.7	75.0	90.0	**93.7**	**97.9**
JPEG+Blur	53.6	53.6	58.8	77.3	52.8	52.9	72.7	86.9	66.7	87.8	64.2	81.7	67.5	89.3	68.5	85.6	76.9	89.9	81.5	92.0	54.8	61.9	81.2	91.5	83.0	90.8	**94.6**	**98.2**
JPEG+Noise	45.1	58.6	52.2	69.9	50.1	50.2	79.3	89.1	58.3	88.9	78.6	90.1	77.2	91.9	68.8	88.1	69.8	92.6	64.5	85.9	49.0	55.6	81.0	90.4	80.7	91.1	**94.9**	**98.4**
Blur+Noise	49.4	61.8	51.2	66.9	50.1	51.2	78.0	86.8	53.4	85.9	73.3	84.4	78.0	90.6	65.3	86.1	67.1	88.6	60.0	77.4	49.9	60.6	78.1	87.2	72.2	86.9	**92.4**	**97.1**
All	48.1	59.9	51.3	66.7	50.1	50.8	76.5	85.0	52.9	83.2	71.9	82.8	76.9	88.5	63.4	84.4	66.7	87.6	58.9	77.8	49.9	59.0	78.3	86.8	70.3	83.8	**91.0**	**96.5**

Table 14: Effect of prompt content on class-prototype construction. Every structured configuration outperforms random captions. Using all prompts from all three categories underperforms the pairwise combinations, whereas selecting the six strongest prompts (PPL) performs best.

Prompt Configuration	mACC (%)	mAP (%)
random_captions	92.1	98.7
texture_only	95.5	99.5
boundary_only	95.1	99.2
geometric_only	95.3	99.4
texture + boundary	96.2	99.4
texture + geometric	96.7	99.6
boundary + geometric	96.8	99.5
texture + boundary + geometric (all 12)	96.1	99.4
PPL, K=6 (Ours)	**97.3**	**99.8**

91.0% ACC or 96.5% AP. Its largest drop relative to clean data is 6.3 ACC and 3.3 AP under the combined setting, while its drop under any single perturbation is at most 3.6 ACC (noise σ=0.05). The ACC margin over the strongest competing method is small on clean data (+1.7) and under mild blur (+2.9 at σ=0.5), but grows to 12.9–13.6 points under all JPEG settings, 15.2 points at blur σ=2.0, 11.8–15.6 points across noise levels, and 11.6–14.3 points under the composed settings, including +12.7 ACC and +8.0 AP under All. This advantage holds across compression, blur, and noise alike. It suggests that the cues exploited by PPL are less concentrated in the high-frequency components removed by these perturbations, which makes PPL better suited to real-world images, where re-compression and noise are common.

A.7 Additional Ablations on Prompt Selection

A.7.1 Prompt Design Choices

Table 14 reports the effect of prompt content on detection performance. All structured configurations outperform the random-caption baseline (92.1% mACC, 98.7% mAP) by 3.0–5.2 mACC points, indicating that the content of the prompts, and not merely the presence of text supervision, determines the usefulness of the class prototypes. Among single-category configurations, texture-only prompts perform best (95.5% mACC, 99.5% mAP), followed by geometric-only (95.3%, 99.4%) and boundary-only (95.1%, 99.2%). The three lie within 0.4 mACC points of each other, suggesting that each cue type carries a comparable amount of discriminative signal on its own and that no single type dominates.

Combining two categories improves mACC over both constituents in every case: texture + boundary reaches 96.2% (+0.7 over texture-only), texture + geometric 96.7% (+1.2), and boundary + geometric 96.8% (+1.5 over geometric-only). The two pairs that include geometric prompts outperform texture + boundary by 0.5–0.6 mACC points, suggesting that geometric cues are the most complementary to the other two types. In terms of mAP, the gains are smaller and not uniform: texture + boundary (99.4%) is slightly below texture-only (99.5%), reflecting that mAP is already close to saturation (99.2–99.8%) across all structured configurations. Adding the third category, however, does not help when all prompts are used: combining all twelve prompts from the three categories yields 96.1% mACC, below every pairwise combination and consistent with the K=12 result in Figure 7b. In contrast, the PPL configuration (K=6), which keeps only the six individually strongest prompts

Table 15: Example prompts generated by different LLMs for real and fake image descriptions.

LLM	Real	Fake
ChatGPT	A genuine camera photograph with natural sensor noise, film grain, realistic textures, physically consistent lighting and shadows, authentic depth of field, lens distortion, bokeh, chromatic aberration, perspective, material reflectance, and natural optical imperfections.	An AI-generated image with synthetic textures, unnatural edges and skin, oversmoothed surfaces, inconsistent lighting and depth, repeating and spectral artifacts, color bleeding, artificial bokeh, blurred transitions, and other GAN or diffusion artifacts.
Claude	A real camera photograph with natural sensor noise and film grain, authentic lighting and consistent shadows, real optical blur, bokeh and depth of field, lens distortion, vignetting and chromatic aberration, natural skin texture and material reflectance.	An AI-generated GAN or diffusion image with synthetic over-smoothed texture, unnaturally perfect skin, blurred hair and unnatural edges, color bleeding and grid artifacts, inconsistent lighting and specular highlights, fake bokeh and depth, and unnatural frequency statistics.
DeepSeek	An authentic camera photograph with natural film grain and sensor noise, real optical blur and bokeh, consistent natural lighting and shadows, authentic perspective with lens distortion and chromatic aberration, and genuine material reflectance with natural imperfections.	An AI-generated image with synthetic textures and unnatural edges, GAN-style skin boundary artifacts and color bleeding, diffusion-style over-smoothing and blended transitions, plus fake bokeh, inconsistent lighting, and spectral/frequency inconsistencies.
Gemini	A real photograph captured by a digital camera, featuring natural sensor noise, film grain, authentic optical blur, realistic depth of field, consistent natural shadows, and accurate material specular highlights.	An AI-generated synthetic image with deepfake artifacts, unnatural skin smoothing, over-smoothed surfaces, inconsistent lighting and shadows, fake bokeh, and color bleeding at semantic boundaries.

Table 16: Optimal Prompt Results.

LLM	mACC (%)	mAP (%)
ChatGPT	93.8	98.7
Claude	92.5	98.5
DeepSeek	94.6	99.0
Gemini	93.9	99.1
PPL, K=6 (Ours)	**97.3**	**99.8**

(Figure 7a), achieves the best result on both metrics (97.3% mACC, 99.8% mAP), improving over the best pairwise combination by 0.5 mACC and over the full three-category library by 1.2 mACC. These six prompts include two from each category (texture: 2, 3; boundary: 6, 7; geometric: 9, 12). Together, these results indicate that covering all three cue types is beneficial only when combined with prompt selection: a balanced subset of strong prompts yields a better prototype than any subset of categories, whereas including every prompt dilutes the prototype with weaker or redundant descriptions.

A.7.2 LLM-Consolidated Prompts

A natural alternative to ensembling multiple prompts is to write a single, more comprehensive prompt per class. To test this, we asked four LLMs (ChatGPT, Claude, DeepSeek, and Gemini) to condense our prompt library into one description per class. Each LLM received the 12 real and 12 fake prompts (Fig. 9) and was instructed to produce a single real and a single fake description capturing the most discriminative visual cues across all categories without redundancy. The resulting prompts, all of which fit within the 77-token context of the text encoder, are listed in Table 15. Each prompt pair was then used on its own to construct the class prototypes, with all other settings identical to the main configuration (Secs. 3.4 and 4.1).

Table 16 reports the results. All four single-prompt configurations fall well below PPL. DeepSeek achieves the highest mACC (94.6%, 99.0% mAP), followed by Gemini (93.9%, 99.1%), which attains the highest mAP, and ChatGPT (93.8%, 98.7%), while Claude performs lowest (92.5%, 98.5%). The

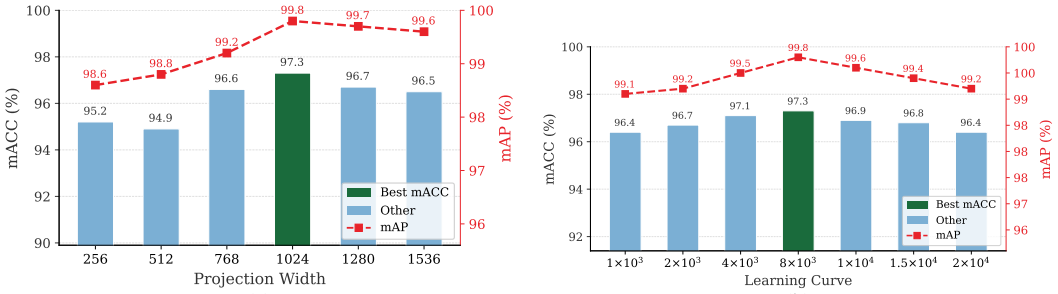


Figure 10: Ablation on (a) projection width and (c) training steps on the UFD benchmark.

spread across LLMs (2.1 mACC points) shows that performance is sensitive to the exact wording of a single prompt, even when all prompts are derived from identical source material. This ranking also does not follow prompt specificity: Claude's description is among the most detailed, explicitly naming cues such as grid artifacts and unnatural frequency statistics, yet it yields the lowest accuracy. Packing more cues into one description therefore does not reliably produce a better prototype.

PPL outperforms the best LLM-consolidated prompt by 2.7 mACC points (97.3% vs. 94.6%) and the best mAP by 0.7 points (99.8% vs. 99.1%). Moreover, every single-prompt configuration falls below each single-category prompt ensemble in Table 14 (95.1–95.5% mACC), even though each consolidated prompt draws on all categories. Combined with the random-caption result in Table 14, these findings indicate that prompt content is necessary but not sufficient: relevant cues must be present, and encoding them as multiple separate prompts that are averaged into a prototype is more effective than combining them into a single description, however comprehensive.

A.8 Additional Ablation Studies

A.8.1 Projection Width

Fig. 10a visualizes mACC and mAP across a range of projection head widths. Performance improves substantially as width increases from 256 to 1024, rising from 94.9% to 97.3% mACC and 98.8% to 99.8% mAP, before declining at wider settings. At the narrowest width (256), the projection head lacks sufficient capacity to disentangle the rich semantic and forgery-relevant features encoded in the PE backbone, resulting in the largest performance gap relative to the optimum. Widening the projection layer to 1024 yields the best trade-off between representational capacity and overfitting, achieving the peak performance of 97.3% mACC and 99.8% mAP.

Beyond 1024, performance declines gradually, dropping to 96.7% mACC at 1280 and 96.5% mACC at 1536. This degradation suggests that excess projection capacity allows the model to overfit to spurious, high-dimensional correlations in the ProGAN training distribution rather than learning compact, generalizable forgery representations. The relatively modest decline (0.8 points in mACC from peak to the widest setting) indicates the model is fairly robust to over-parameterization of the projection head, though performance never exceeds the 1024-dimensional optimum. We therefore adopt a projection width of 1024 as the default.

A.8.2 Training Steps

Fig. 10b shows how performance evolves as a function of training duration. The model improves steadily in the early stages, reaching 96.4% mACC at 1,000 steps and 96.7% at 2,000 steps, and continues to improve through 4,000 steps (97.1% mACC, 99.5% mAP) before peaking at 8,000 steps (97.3% mACC, 99.8% mAP). This monotonic improvement suggests that the LoRA and projection parameters require a sustained number of iterations to fully align with the detection task before reaching their optimal configuration.

Beyond 8,000 steps, performance declines steadily and monotonically. mACC drops to 96.9% at 10,000 steps, 96.8% at 15,000 steps, and further to 96.4% at 20,000 steps, with mAP following a similar gradual decline (99.6%→99.4%→99.2%). This consistent degradation with extended training is indicative of overfitting to the ProGAN training distribution, where the model increasingly memorizes generator-specific artifacts at the expense of more generalizable visual features captured by the PPL prototypes. Notably, the degradation remains modest even at 20,000 steps (2.5× the optimal duration), suggesting the model is reasonably robust to moderate over training, though performance never recovers to the 8,000-step optimum. We therefore adopt 8,000 training steps as the default, as it provides the best generalization performance.

Property	Value
Total params (training, incl. frozen text enc.)	2,578.5M
Params at inference (vision + head)	1,881.2M
Trainable params	3.4M
Trainable ratio	0.13%
GFLOPs (per sample)	1,900.1
Inference Time	99.5 ms/image
Throughput	10.1 img/s

Table 17: Computational cost of PPL. Only the lightweight adaptation modules are trainable; the backbone is kept frozen.

A.9 Computational Cost

Table 17 summarizes the computational profile of PPL. Although the underlying backbone totals 2.56B parameters, only 3.4M (0.13%) are trainable, confirming that the generalization gains reported in Secs. 4.2 and 4.3 arise from a compact, parameter-efficient adaptation on top of a frozen large-scale representation rather than from full fine-tuning. This design choice keeps training fast and memory-efficient, and avoids the catastrophic forgetting and overfitting to source-generator artifacts that commonly afflict fully fine-tuned detectors when evaluated out-of-distribution.

At inference time, PPL requires 1,900.1 GFLOPs per sample and runs at 99.5 ms/image (10.1 images/s) on a single GPU, without batching or FP16 optimization. While the frozen backbone dominates the FLOP count, the added trainable modules contribute negligible overhead, so the runtime cost is effectively determined by the backbone alone, the same cost any method built on a comparable large-scale pretrained encoder would incur. This places PPL well within the range of practical deployment for offline or near-real-time moderation pipelines, and the cost can be further reduced through standard techniques such as batching, quantization, or knowledge distillation, which we leave to future work.

A.10 Broader Impact, Ethics, and Responsible Disclosure

Positive Societal Impact. PPL contributes to the detection of AIGIs, supporting digital media integrity and public trust in visual content. As generative models become more capable and accessible, robust detectors are needed to counter disinformation, protect individuals from non-consensual synthetic media, and support forensic analysis in journalistic and legal contexts. Because PPL is trained on a single generator and generalizes to many unseen ones, it may reduce the data-collection burden of deploying detectors in settings where training coverage is inherently limited.

Potential Negative Impact and Misuse. Detectors can be misused or over-trusted. A released detector may serve as a discriminator for adversaries seeking to generate or post-process images that evade detection, and PPL has not been evaluated against adaptive or adversarial attacks. False positives can wrongly discredit authentic images, including legitimate journalism or personal photographs, while false negatives can lend unwarranted credibility to synthetic content. Detector outputs should therefore be treated as probabilistic evidence to support, not replace, human judgment, and should not be used as the sole basis for consequential decisions such as content removal, legal findings, or accusations against individuals.

Contributions to the Research Community. This work (i) shows that ensemble-averaged prompt prototypes provide a simple, annotation-free form of text supervision that improves cross-generator generalization in VLM-based detectors, and (ii) provides ablations isolating the effect of prompt content, prompt count, prototype construction, and adaptation settings, which may guide the design of future detection pipelines.

Fairness Considerations. PPL is trained on four ProGAN object categories, and several prompts in the library describe human-specific attributes (e.g., skin texture and hair boundaries). The detector may therefore behave unevenly across demographic groups, skin tones, cultural content, or image domains that are underrepresented in training. We have not systematically evaluated such disparities, and we encourage evaluation on demographically diverse benchmarks before deployment in sensitive contexts.

Limitations. First, the prompt library is manually designed, which may bias supervision toward known artifact types and limit coverage of future synthesis methods whose failure modes differ from

those of current GAN and diffusion models. Second, because PPL is trained on a single source (ProGAN), its fixed decision threshold does not always transfer under distribution shift: ACC drops on some generators and in-the-wild sources, including Midjourney (87.6% on AIGC), Imagen3 (75.1%), and CommunityAI (78.6%), even where AP remains high (e.g., 98.0% on Midjourney, the best among all methods). Adaptive threshold calibration or multi-source training may mitigate this. Third, PPL relies on a large vision encoder (PE-Core-G14-448, $\sim$1.88B parameters), which limits deployment on resource-constrained devices. Fourth, robustness was evaluated only against benign post-processing (JPEG compression, blur, and noise), not against adversarial perturbations. Finally, evaluation is restricted to static images; extending PPL to video, where temporal artifacts introduce additional complexity, is left for future work.

Use of Large Language Models. LLMs were used in two ways. (i) As writing aids, for grammatical correction and rephrasing to improve clarity. (ii) As an experimental component in the prompt-aggregation study (A.7.1), where ChatGPT, Claude, DeepSeek, and Gemini were asked to consolidate our prompt library into single real/fake descriptions; their outputs are reported verbatim in Table 15. The prompt library used by PPL itself, the method, experimental design, implementation, analysis, and conclusions are the authors' own work, and the authors take full responsibility for all content of this paper.

Code and Model Availability. Code, trained weights, the full prompt library, and evaluation scripts will be released publicly upon acceptance to support reproducibility.