This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PPM-CLIP: Probabilistic Prompt Modeling for Generalizable AI-Generated Image Detection

Xinyuan Wang[1]*, Yingxin Lai[1]*, Zhiming Luo[1,2]†, Zhihui Liu[3]
[1] Department of Artificial Intelligence, Xiamen University, Xiamen, China
[2] Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, Xiamen, China
[3] Truesight Technology Co. Ltd, Xiamen, China

Abstract

The rapid rise of highly realistic AI-generated images necessitates reliable and generalizable detection methods. However, existing methods are constrained by their discriminative nature: by learning a single static decision boundary, they tend to memorize generator-specific artifacts and consequently fail to generalize to the unseen distributions of new generative models. To overcome this limitation, we propose PPM-CLIP, a new framework that shifts from static classification to conditional generative modeling based on the CLIP vision-language model. Instead of learning a fixed decision boundary, a Probabilistic Prompt Modeling (PPM) module is used as a generator that produces an adaptive distribution of prompts according to the input image. This allows the model to flexibly capture novel artifacts, rather than matching them against fixed templates. In addition, to enhance the visual encoder's sensitivity to subtle artifacts, a Patch-Wise Contrastive Learning (PWCL) strategy is introduced. Extensive experiments on Ojha, GenImage, and DRCT benchmarks demonstrate that our generative paradigm significantly outperforms state-of-the-art methods, especially in cross-domain detection. The code will be released at https://github.com/bandaidssssss/PPM_CLIP.

1. Introduction

With the rapid advancement of image generation technologies, particularly Generative Adversarial Networks (GANs) [4, 9, 10] and diffusion-based models [5, 6, 18, 24], producing high-fidelity forged images has become alarmingly feasible. The potential misuse of these technologies poses significant societal risks, ranging from fabricating disinformation to manipulating public opinion and interfering in political elections [25, 37, 42]. Consequently, developing robust detection methods for AI-generated images is critical to protect the integrity of online information.

To tackle this challenge, researchers have developed var-

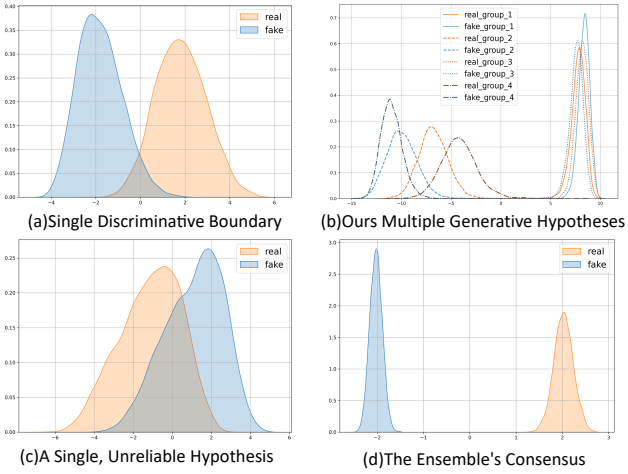


Figure 1. PCA visualization of text embeddings, where prompts are selected according to the image's ground truth ("real" prompt for real images, "fake" prompt for fake images). Distributions are computed on the GenImage dataset by visualizing the text embeddings of a large batch of inputs. (a) Single Discriminative Boundary: Derived from FatFormer, this shows that conventional methods relying on a single, static boundary often fail to create a separable feature space. (b) Our Multiple Generative Hypotheses: Displaying a subset of our generated prompt pairs, our generative approach produces a distribution of hypotheses, revealing the underlying data structure. (c) A Single, Unreliable Hypothesis: Illustrating a single, randomly sampled prompt pair, we observe that an individual hypothesis from the distribution is often unreliable and leads to poor separation. (d) The Ensemble Consensus: By averaging all hypotheses, the final prediction effectively cancels out the noise from individual hypotheses to achieve final class separation.

*Equal contribution
†Corresponding author: zhiming.luo@xmu.edu.cn

ious techniques including pixel-level and semantic-level approaches. Pixel-level methods [3, 35] primarily focus on detecting low-level artifacts and achieve high accuracy on known generators. However, their reliance on learning fixed patterns from training artifacts often leads to overfitting and poor cross-generator generalization. In contrast, semantic-level approaches leverage Vision-Language Models (VLMs) such as CLIP [21] to spot high-level inconsistencies [7, 12, 23, 27]. Within this stream, several studies [13, 31] have specifically explored the role of the text prompt by learning adaptive textual features to guide detection. Despite architectural variations, these methods remain fundamentally constrained by a discriminative paradigm that relies on a single decision boundary.

The inherent fragility of these discriminative methods is starkly visualized in Figure 1. Even with a meticulously designed prompt pair, the feature space remains entangled, demonstrating that a single, static decision boundary is often insufficient to handle the nuanced differences between real and generated images. This failure stems from a fundamental flaw in the static, discriminative paradigm: by forcing the model to find a single, optimal separation, it inevitably learns to "memorize" a fixed set of generator-specific artifacts. However, the landscape of image generation is a diverse and rapidly evolving ecosystem. A static template is inherently ill-equipped for this diversity, leading to a performance collapse against unseen generators.

To overcome this generalization bottleneck, we propose a crucial shift in perspective, inspired by how human experts operate. Instead of searching for a single, perfect forgery cue, an expert reasons about an image's authenticity from multiple, diverse perspectives. To realize this expert-inspired reasoning process, we introduce PPM-CLIP, a framework that shifts from static classification to dynamic, probabilistic reasoning. The core of our approach, the Probabilistic Prompt Modeling (PPM) module, acts as a conditional generator, producing a tailored distribution of semantic hypotheses for any given image. Grounded in our hierarchical Prompt Flow formulation, each "hypothesis" corresponds to a probabilistically sampled text prompt that dynamically synthesizes three distinct semantic levels: image-generic structural priors, image-specific visual evidence, and class-indicative semantics. While any single hypothesis may be unreliable, the ensemble's consensus, obtained by averaging their outcomes, marginalizes out noise and converges on a highly robust and accurate final decision. This approach allows our model to adaptively reason about novel artifacts on the fly, rather than failing to match them against a fixed, outdated template.

On the other hand, the standard visual encoder trained on high-level semantics often overlooks the subtle traces of digital forgery. To address this limitation, we further introduce a Patch-Wise Contrastive Learning (PWCL) strategy that refines the visual encoder by leveraging frequency-domain cues to enhance its sensitivity to the subtle, high-frequency artifacts. This refinement ensures that PPM operates on a feature space enriched with the fine-grained forensic details necessary for reliable detection.

The contributions of this work are three-fold:

- **A New Generative Perspective.** We shift from the static discriminative paradigm to a generative-powered approach, which first models the forgery distribution and then performs a robust probabilistic inference. We have initially explored how generation and discrimination can be unified.
- **Synergistic Model Design.** We propose a PPM module, a generative core that produces a diverse ensemble of hypotheses, and a PWCL strategy that grounds them in fine-grained visual evidence.
- **State-of-the-Art Performance.** PPM-CLIP sets a new state-of-the-art in accuracy and, critically, in cross-generator robustness, empirically validating our generative, probabilistic paradigm.

2. Related Work

Pixel-level Detection Methods: Early methods primarily detect low-level artifacts in the spatial or frequency domains. Spatial approaches train classifiers directly on pixels [33] or specific patch anomalies [43]. Frequency-aware methods [3, 12, 38] target spectral inconsistencies invisible to the human eye. Another major stream, reconstruction-based detection [1, 17, 35], identifies forgeries via the reconstruction error of pre-trained generators. However, by training models to learn fixed discriminative patterns from these low-level cues, these methods inevitably memorize source-specific artifacts, leading to poor generalization against novel generative paradigms.

Semantic-level Detection Methods: To improve generalization, recent studies leverage pre-trained models such as CLIP [21]. UnivFD [19] applies a simple classifier on frozen CLIP features. Subsequent works have successfully employed Parameter-Efficient Fine-Tuning (PEFT) [16, 36, 41] to fine-tune CLIP's behavior, while others have explored sophisticated prompting strategies [13, 31, 40] to guide the detection. Furthermore, other studies have extended this paradigm by leveraging different pre-trained models [26] or by integrating knowledge from multiple models [34] to enhance detection capabilities. However, these methods remain bound by a discriminative framework. Whether optimizing a static adapter or a fixed prompt vector, they fundamentally compress the complex forgery landscape into a rigid decision boundary. This deterministic mapping lacks the capacity to handle the diversity of unseen domains. In contrast, our framework shifts to a generative

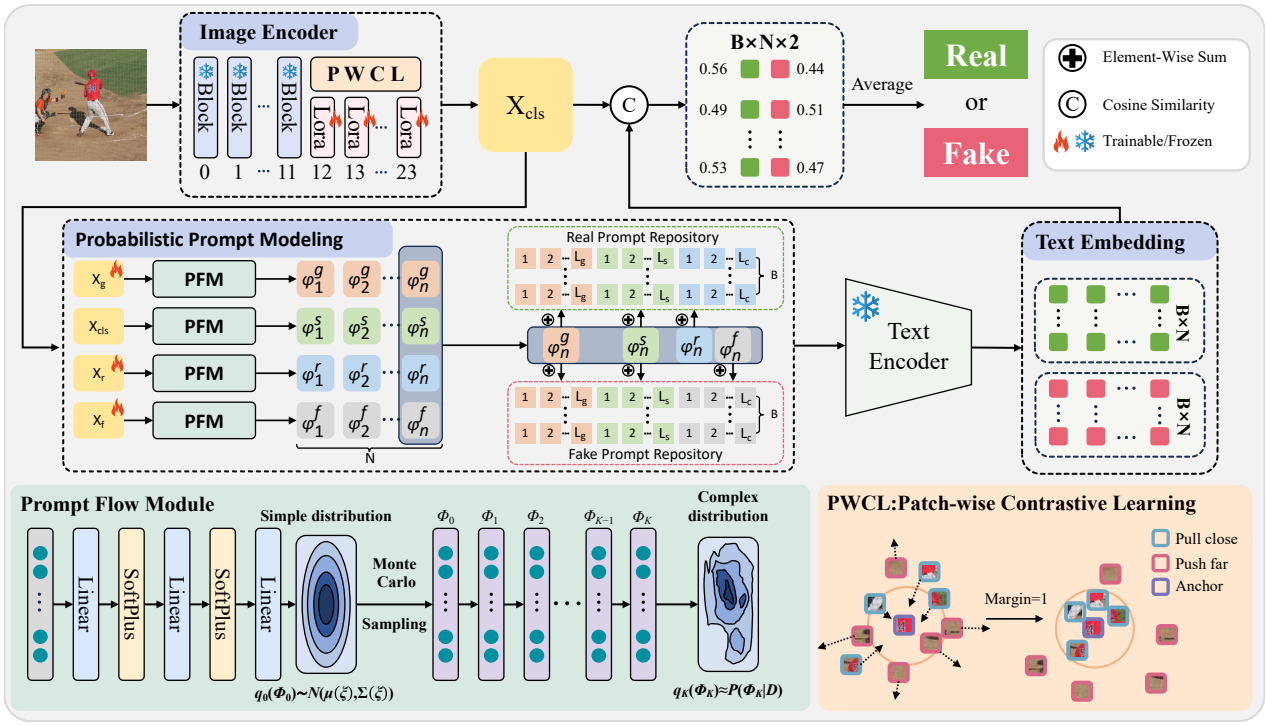


Figure 2. Overview of **PPM-CLIP**. An input image is processed by a visual encoder enhanced by Patch-Wise Contrastive Learning (PWCL), yielding a feature vector, $\mathbf{X}_{cls}$, that is rich in fine-grained forensic details. Concurrently, the Prompt Flow Module learns a probabilistic distribution from its conditioning inputs and generates N prompt samples from it via Monte Carlo sampling. These samples are fused with a repository of B base prompt sets, yielding $B \times N$ text embeddings. The final prediction is determined by averaging the cosine similarities between the visual token (X_{cls}) and the resulting text embeddings.

paradigm, modeling the *distribution* of forgery prompts to enable robust probabilistic inference.

3. Methodology

As illustrated in Figure 2, our PPM-CLIP consists of two modules. First, an input image $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$ is processed by a visual encoder enhanced by our Patch-Wise Contrastive Learning (PWCL) strategy (Sec. 3.1), yielding a final visual feature vector, $\mathbf{X}_{\text{cls}}$.

Concurrently, our core innovation is the Probabilistic Prompt Modeling (PPM) framework. Departing from prior discriminative methods, the Prompt Flow Module employs a generative approach to produce adaptive adjustment vectors. These dynamically generated vectors are fused with a two-class prompt repository and passed through a frozen text encoder, yielding two distinct sets of text embeddings: one for "real" and one for "fake".

Finally, the prediction is determined by aligning the visual feature $\mathbf{X}_{\text{cls}}$ with both sets of text embeddings. The classification of the image as real or fake is determined by which set yields a higher average cosine similarity score.

3.1. Patch-wise Contrastive Learning

To supply our Probabilistic Prompt Modeling (PPM) module with the necessary fine-grained visual details, a standard visual encoder is insufficient as it often overlooks subtle forgery artifacts. Therefore, we propose Patch-Wise Contrastive Learning (PWCL), an auxiliary strategy to refine the encoder. As shown in Figure 2, PWCL leverages frequency-domain cues to guide the encoder's focus toward high-frequency patterns within local patches, which is a known indicator of digital forgery [2, 43]. This yields a visual backbone highly sensitive to the forensic details required by our generative hypotheses.

For a given RGB image, we first view it as a grid of non-overlapping patches, denoted as $\mathcal{I} = \{p_1, p_2, \ldots, p_n\}$, where $p_i \in \mathbb{R}^{M \times M \times 3}$. We employ the AIDE [38] method to compute the frequency score for each image patch using the following equation, with full details in Appendix 6:

$$G_m = \sum_{k=0}^{N_f-1} 2^k \cdot \sum_{c=0}^{2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} F_{i,j}^{(k)} \cdot \log\left(\left|p_m^{\text{dct}}(i,j,c)\right| + 1\right), \tag{1}$$

where $|p_m^{\text{dct}}(i,j,c)|$ is the absolute DCT coefficient of the m-th patch, and N_f is the number of frequency band-pass

filters. The term $F_{i,j}^{(k)} \in \{0,1\}$ is a binary indicator for the k-th frequency band.

We rank all patches by their frequency scores and use a selection ratio, α, to partition them into a high-frequency set (top $n \times \alpha$) and a low-frequency set. Within the high-frequency set, the highest-scoring serves as the anchor, while the others become positive samples. All patches in the low-frequency set are treated as negative samples.

A distance-based contrastive learning objective is then applied to the embeddings of these patches to pull the anchor closer to its positive peers and push it away from the negative ones. This encourages the model to learn discriminative representations that focus on informative, high-frequency forgery artifacts.

$$\mathcal{L}_{\text{con}} = \sum_{\mathbf{e}_p \in \mathcal{P}} ||\mathbf{e}_a - \mathbf{e}_p||_2^2 + \sum_{\mathbf{e}_n \in \mathcal{N}} \max(0, m - ||\mathbf{e}_a - \mathbf{e}_n||_2^2), \tag{2}$$

where $\mathbf{e}_a$ is the anchor embedding (the highest-scoring patch), $\mathcal{P}$ is the set of positive embeddings (other top-scoring patches), and $\mathcal{N}$ is the set of negative embeddings (bottom-scoring patches). The m represents the margin, and $||\cdot||_2^2$ denotes the squared L2-distance.

3.2. Probabilistic Prompt Modeling

Our Probabilistic Prompt Modeling (PPM) framework is designed to generate a diverse ensemble of image-adaptive prompts (hypotheses). It consists of two core, synergistic components: **Prompt Flow Module (PFM)**, a generative model that produces dynamic, length-1 adjustment vectors tailored to the input image. **Prompt Repository**, which provides the stable, full-length structural templates necessary to ensure prompt diversity and coherence.

3.2.1. Prompt Flow Module:

Inspired by normalizing flows [22], the Prompt Flow Module is designed to construct a highly expressive and complex prompt distribution. To this end, the module takes different conditioning inputs to generate three distinct yet complementary distributions, which collectively form a hierarchical representation of semantic information:

- Image-Generic Distribution (IGD): This distribution is conditioned on a single learnable vector ($\mathbf{X}_{\text{g}}$) and is designed to capture coarse-grained, universal attributes common to all prompts, such as syntactic structure.
- Image-Specific Distribution (ISD): In contrast, this distribution is directly conditioned on the input image's visual features ($\mathbf{X}_{\text{cls}}$). Its role is to provide the fine-grained, adaptive details necessary to describe the unique visual evidence within each specific input.
- Image-Class Distribution (ICD): Conditioned on two distinct learnable vectors ($\mathbf{X}_{\text{r}}$ and $\mathbf{X}_{\text{f}}$), this distribution focuses solely on the core discriminative task. It learns to generate class-indicative adjustments that steer the prompt towards either a "real" or "fake" classification.

Notably, for ICD, the two distinct learnable vectors, $\mathbf{X}_{\text{r}}$ ("real") and $\mathbf{X}_{\text{f}}$ ("fake"), are processed by a shared-weight encoder. This deliberate strategy compels the model to learn a unified representation space for authenticity, where "real" and "fake" are treated as opposing semantic poles.

As shown in the bottom-left portion of Figure 2, the process begins with Monte Carlo sampling N times from this base distribution. To ensure proper gradient back-propagation after discrete sampling, the reparameterization method [11] is employed during the optimization process. These samples are then passed through the main flow, which consists of a sequence of K learnable and invertible transformations. The transformation sequence is formally defined as follows:

$$\Phi_{i+1} = \Phi_i + \mathbf{u}\mathbf{h}\left(\mathbf{w}^\top \Phi_i + \mathbf{b}\right), \tag{3}$$

where $i = 0, 1, \cdots, K-1$, $\mathbf{h}(\cdot)$ is $tanh$ activation function, and $\mathbf{w}, \mathbf{u} \in \mathbb{R}^d$, $\mathbf{b} \in \mathbb{R}$ are trainable parameters. After K transformations, the log-probability density of the final distribution, $\log q_K(\Phi_K)$, is given by:

$$\log q_K(\Phi_K) = \log q_0(\Phi_0) - \sum_{k=1}^{K} \log \left|1 + \mathbf{u}_k^{\text{T}} \phi(\Phi_k)\right|, \tag{4}$$

where $\phi(\Phi) = h'(\mathbf{w}^{\text{T}}\Phi + \mathbf{b})\mathbf{w}$. The output of this PFM is a set of N compact adjustment vectors, denoted as $\{\varphi_n^g, \varphi_n^s, \varphi_n^r, \varphi_n^f\}_{n=1}^N$.

3.2.2. Prompt Repository and Dynamic Fusion

To provide a structural foundation for our prompts, we establish a learnable Prompt Repository containing B distinct baseline templates. As illustrated in the center of Figure 2, these templates provide the essential scaffolding for full-length prompts. The structure of the b-th template pair is defined as:

$$\begin{aligned} g_b^r &= [\mathbf{G}_{b,1..L_g}][\mathbf{S}_{b,1..L_s}][\mathbf{C}_{b,1..L_c}^r], \\ g_b^f &= [\mathbf{G}_{b,1..L_g}][\mathbf{S}_{b,1..L_s}][\mathbf{C}_{b,1..L_c}^f]. \end{aligned} \tag{5}$$

Each template is composed of three distinct sequences of learnable vectors, which are designed to serve as the static base for our three dynamic distributions:

- Generic Context ($\mathbf{G}_{b,1..L_g}$): A sequence of L_g learnable vectors that provide a stable, shared foundation for the Image-Generic adjustment (φ^g).
- Specific Content ($\mathbf{S}_{b,1..L_s}$): A sequence of L_s learnable vectors that form the stable content backbone, ready to be modulated by the Image-Specific adjustment (φ^s).
- Class-Indicative Concepts ($\mathbf{C}_{b,1..L_c}^r$ and $\mathbf{C}_{b,1..L_c}^f$): Two distinct sequences of L_c vectors, establishing a stable, discriminative base for the "real" (φ^r) and "fake" (φ^f) class adjustments, respectively.

Adaptability is achieved by dynamically fusing the static templates with the compact, image-adaptive adjustment vectors ($\varphi^g, \varphi^s, \varphi^r, \varphi^f$) from the Prompt Flow Module. For the n-th sample and the b-th template, this fusion is performed by broadcasting and adding each adjustment vector to its corresponding token sequence, yielding two distinct prompts for the "real" and "fake" classes:

$$\begin{aligned} g^r_{b,n} &= [\mathbf{G}_{b,1..L_g} + \varphi^g_n][\mathbf{S}_{b,1..L_s} + \varphi^s_n][\mathbf{C}^r_{b,1..L_c} + \varphi^r_n], \\ g^f_{b,n} &= [\mathbf{G}_{b,1..L_g} + \varphi^g_n][\mathbf{S}_{b,1..L_s} + \varphi^s_n][\mathbf{C}^f_{b,1..L_c} + \varphi^f_n]. \end{aligned} \tag{6}$$

This process expands our initial B template pairs into a rich ensemble of $B \times N$ distinct, image-adaptive prompt pairs with negligible inference overhead.

3.3. Optimization and Inference

During the training stage, a single Monte Carlo sampling is performed to enhance efficiency. Let the text prompts from B samples be denoted as $[g^r_{b,1}, g^f_{b,1}]$, and their corresponding text embeddings as $[t^r_{b,1}, t^f_{b,1}], b \in \{1, 2, \cdots B\}$. To enhance the diversity of learnable prompts in prompt repository, an orthogonal loss is designed on the text embeddings as follows:

$$\mathcal{L}_{\text{ort}} = \sum_{i=1}^{B} \sum_{\substack{j=1 \\ j \neq i}}^{B} \left\{ \cos(t^r_{i,1}, t^r_{j,1})^2 + \cos(t^f_{i,1}, t^f_{j,1})^2 \right\}, \tag{7}$$

Our Prompt Flow Module is trained by optimizing a variational objective based on the Evidence Lower Bound (ELBO), which comprises a KL divergence term ($\mathcal{L}_{kl}$) and a reconstruction term ($\mathcal{L}_{rec}$). The detailed derivation is in Appendix 7.

The first component is the KL divergence ($\mathcal{L}_{KL}$), which is applied to all three distributions (IGD, ISD, and ICD). It acts as a regularizer, encouraging the learned posterior q_K to stay close to a simple prior $p(\Phi)$. Its computational formula is:

$$\mathcal{L}_{KL} = \mathbb{E}_{q_0(\Phi_0)} \left[\log q_K(\Phi_K) - \log p(\Phi_K) \right]. \tag{8}$$

The second component is the reconstruction loss ($\mathcal{L}_{rec}$), which is applied exclusively to the Image-Specific Distribution (ISD). The purpose of this loss is to ensure that the ISD's latent variable Φ^s_K faithfully preserves the salient information from its unique conditioning input. We enforce this by decoding Φ^s_K with a simple decoder to generate a reconstructed vector $\mathbf{X}_{\text{rec}}$ and minimizing the Mean Squared Error (MSE) with the original $\mathbf{X}_{\text{cls}}$:

$$\mathcal{L}_{\text{rec}} = \sum_{i=1}^{d} (x_i - \hat{x}_i)^2, \tag{9}$$

where x_i and $\hat{x}_i$ correspond to elements in the original feature vector $\mathbf{X}_{\text{cls}}$ and the reconstructed vector $\mathbf{X}_{\text{rec}}$.

Following the visual encoding by PWCL and prompt generation by PPM, we obtain the visual feature vector $\mathbf{X}_{\text{cls}}$ and two sets of text embeddings: $\{\mathbf{f}^r_{\text{text},i}\}$ for "real" and $\{\mathbf{f}^f_{\text{text},i}\}$ for "fake", where $i = 1, \ldots, B \times N$.

The prediction is based on cosine similarity cos. For each individual text embedding, we compute its similarity to the visual feature:

$$s^r_i = \cos(\mathbf{X}_{\text{cls}}, \mathbf{f}^r_{\text{text},i}) \quad \text{and} \quad s^f_i = \cos(\mathbf{X}_{\text{cls}}, \mathbf{f}^f_{\text{text},i}). \tag{10}$$

For each prompt pair i, we can then derive the corresponding class probabilities using a softmax function with temperature τ:

$$P^r_i = \frac{\exp(s^r_i/\tau)}{\exp(s^r_i/\tau) + \exp(s^f_i/\tau)}, \quad \text{and} \quad P^f_i = 1 - P^r_i. \tag{11}$$

With these individual probabilities, our training and inference strategies diverge:

During training, for efficiency, we randomly sample a single index i for each input image. The classification loss, $\mathcal{L}_{\text{cls}}$, is then the standard cross-entropy loss computed between this single predicted probability pair $\{P^r_i, P^f_i\}$ and the ground truth label.

During inference, to leverage the full power of our ensemble, we compute the final class probabilities, $\bar{P}^c$, by averaging the individual probabilities from all $B \times N$ prompt pairs:

$$\bar{P}^r = \frac{1}{B \times N} \sum_{i=1}^{B \times N} P^r_i, \quad \text{and} \quad \bar{P}^f = 1 - \bar{P}^r. \tag{12}$$

The overall objective function is a weighted sum of the classification loss and the aforementioned regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}} + \lambda_{\text{ort}}\mathcal{L}_{\text{ort}} + \lambda_{kl}\mathcal{L}_{kl} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}, \tag{13}$$

where λ_{con}, λ_{ort}, λ_{kl}, and λ_{rec} are hyperparameters that balance the contribution of each component. The terms $\lambda_{kl}\mathcal{L}_{kl} + \lambda_{rec}\mathcal{L}_{rec}$ jointly form a weighted approximation of the negative Evidence Lower Bound (ELBO), which is the core variational objective for our prompt flow module.

4. Experiments

4.1. Experimental Setup

Implementation Details. We adopt the CLIP ViT-L/14 model as our backbone, adhering to the standard protocol in Ojha et al. [19]. The model is trained for 1 epochs using the Adam optimizer with a learning rate of 1×10^{-4} and a batch size of 48. Input images are resized to 224×224, applying

Method	Ref	DALLE	Glide			ADM	LDM			mAcc
			100_10	100_27	50_27		100	200	200_cfg	
CNNDet [33]	*CVPR 2020*	51.8	53.3	53.0	54.2	54.9	51.9	52.0	51.6	52.8
FreDect [3]	*ICML 2020*	57.0	53.6	50.4	52.0	53.4	56.6	56.4	56.5	54.5
F3Net [20]	*ECCV 2020*	71.6	88.3	87.0	88.5	69.2	74.1	73.4	80.7	79.1
LGrad [28]	*CVPR 2023*	88.5	89.4	87.4	90.7	86.6	94.8	94.2	95.9	90.9
UnivFD [19]	*CVPR 2023*	89.5	90.1	90.7	91.1	75.7	90.5	90.2	77.3	86.9
PatchCraft [43]	*Arxiv 2023*	83.3	80.1	83.4	77.6	80.9	88.9	89.3	88.1	84.0
FreqNet [29]	*AAAI 2024*	97.2	87.8	84.4	86.6	67.2	97.8	97.4	97.2	89.5
NPR [30]	*CVPR 2024*	94.5	98.2	97.8	98.2	75.8	**99.3**	99.1	**99.0**	95.1
FatFormer [13]	*CVPR 2024*	**98.7**	94.5	94.1	94.2	75.9	98.5	98.5	94.8	93.6
SAFE [12]	*KDD 2025*	97.5	97.2	95.8	96.6	82.4	98.8	98.8	98.7	95.7
COD [8]	*CVPR 2025*	98.1	97.6	97.4	97.7	92.8	98.8	98.9	98.7	97.5
Ours	–	98.4±1.3	**99.3±0.0**	**99.3±0.1**	**99.3±0.1**	**97.0±0.4**	**99.3±0.0**	**99.1±0.1**	98.4±0.3	**98.8±0.2**

Table 1. Comparison with state-of-the-art methods on Ojha datasets [19] using Accuracy (%).

Method	Ref	Midjourney	SD v1.4	SD v1.5	ADM	GLide	Wukong	VQDM	BigGAN	mAcc
CNNDet [33]	CVPR 2020	50.1	50.3	50.3	53.0	51.7	51.4	50.0	69.8	53.3
FreDect [3]	ICML 2020	32.1	28.8	28.9	62.9	42.8	35.9	72.1	26.1	41.2
LGrad [28]	CVPR 2023	73.7	76.3	77.4	51.8	49.8	73.1	52.1	40.5	61.8
UnivFD [19]	CVPR 2023	93.9	96.4	96.2	71.9	85.4	94.3	81.6	90.5	88.8
PatchCraft [43]	Arxiv 2023	89.7	95.0	94.6	81.6	83.5	90.9	88.2	91.5	86.3
FreqNet [29]	AAAI 2024	89.6	98.8	98.6	66.8	86.5	97.3	75.8	81.4	86.8
NPR [30]	CVPR 2024	81.0	98.2	97.9	76.9	89.8	96.9	84.1	84.2	88.6
FatFormer [13]	CVPR 2024	92.7	**100.0**	**99.9**	75.9	88.0	99.9	98.8	55.8	88.9
AIDE [38]	ICLR 2025	79.4	99.7	99.8	78.5	91.8	98.7	80.3	66.9	86.9
Effort [39]	ICML 2025	82.4	99.8	99.8	78.7	93.3	97.4	91.7	77.6	90.1
SAFE [12]	KDD 2025	95.3	99.4	99.3	82.1	96.3	98.2	96.3	97.8	95.6
C2P-CLIP [31]	AAAI 2025	88.2	90.9	97.9	96.4	99.0	98.8	96.5	98.7	95.8
CoD [8]	CVPR 2025	96.0	99.7	99.8	85.2	95.9	98.2	96.8	98.3	96.2
LOTA [32]	ICCV 2025	93.2	99.8	99.8	**99.5**	99.2	99.8	99.5	**99.9**	98.9
Ours	–	**98.7±1.5**	**100.0±0.0**	**99.9±0.0**	99.2±0.9	**99.4±0.7**	**100.0±0.0**	**100.0±0.0**	**99.9±0.2**	**99.6±0.4**

Table 2. Comparison with state-of-the-art methods on GenImage datasets [44] using Accuracy (%).

random cropping during training and center cropping during testing. To enable efficient adaptation, we apply LoRA ($r = 4, \alpha_{lora} = 0.5$) to layers 12-23 in the visual encoder. For the proposed modules, we set the PWCL selection ratio α to 0.5 and the margin m to 1.0. The PPM module utilizes a prompt repository size of $B = 2$ and a flow length of $K = 10$. During inference, we perform the Monte Carlo sampling 4 times.

Datasets. Following previous methods [32, 34], we conduct experiments on three challenging benchmarks: (1) **Ojha** [19]: This dataset is designed to assess the detection of cross-architecture generalization, which models are trained on ProGAN and evaluated on unseen GANs and Diffusion Models. (2) **GenImage** [44]: A large-scale benchmark for cross-generator robustness, where models are trained on SDv1.4 and evaluated on images generated by eight diverse models. (3) **DRCT** [1]: A recent dataset evaluates robustness against high-fidelity diffusion reconstruction and localized inpainting attacks.

4.2. Comparison with State-of-the-Art Methods

Ojha. We first evaluate cross-architecture generalization on Ojha . As shown in Table 1, pixel-level methods like CNNDet and FreDect hard to generalize, overfitting to source-specific artifacts. Although semantic-level methods like UnivFD improve robustness by leveraging CLIP features, they are still lower than our approach (86.9% vs. 98.8%). This gap stems from UnivFD's reliance on a static linear classifier, which tends to learn a rigid decision boundary tied to GAN-induced artifacts. This rigidity hinders adaptability to the diverse artifacts of advanced diffusion models (e.g., 75.7% on LDM). In contrast, PPM-CLIP achieves a mean accuracy of 98.8%, surpassing the baseline by 11.9%. The consistent performance across DALLE (98.4%) and Glide (99.3%) demonstrates that our generative-powered approach effectively mitigates generator-specific biases and captures universal forgery patterns.

GenImage. To further assess cross-generator robustness, we conduct experiments on the GenImage dataset (Table 2). Our method achieves the best average accuracy of 99.6%.

Method	Ref	SD Variants						Turbo Variants		LCM Variants		ControlNet Variants			DR Variants			Avg.
		LDM	SDv1.4	SDv1.5	SDv2	SDXL Refiner	SDXL-Refiner	SD-Turbo	SDXL-Turbo	LCM-SDv1.5	LCM-SDXL	SDv1-Ctrl	SDv2-Ctrl	SDXL-Ctrl	SDv1-DR	SDv2-DR	SDXL-DR	
CNNSpot [33]	CVPR 2020	99.87	99.91	99.90	97.55	66.25	86.55	86.15	72.42	98.26	61.72	97.96	85.89	82.84	60.93	51.41	50.28	81.12
F3Net [20]	ECCV 2020	99.85	99.78	99.79	88.66	55.85	87.37	68.29	63.66	97.39	54.98	97.98	72.39	81.99	65.42	50.39	50.27	77.13
CLIP/RN50 [21]	ICML 2021	99.00	99.99	99.96	94.61	62.08	91.43	83.57	64.40	98.97	57.43	99.74	80.69	82.03	65.83	50.67	50.47	80.05
GramNet [14]	CVPR 2020	99.40	99.01	98.84	95.30	62.63	80.68	71.19	69.32	93.05	57.02	89.97	75.55	82.68	51.23	50.01	50.08	76.62
De-fake [27]	CCS 2023	92.10	99.53	99.51	89.65	64.02	69.24	92.00	93.93	99.13	70.89	58.98	62.34	66.66	50.12	50.16	50.00	75.52
Conv-B [15]	CVPR 2022	99.97	100.0	99.97	95.84	64.44	82.00	80.82	60.75	99.27	62.33	99.80	83.40	73.28	61.65	51.79	50.41	79.11
UnivFD [19]	CVPR 2023	98.30	96.22	96.33	93.83	91.01	93.91	86.38	85.92	90.44	88.99	90.41	81.06	89.06	51.96	51.03	50.46	83.46
DIRE [35]	ICCV 2023	54.62	75.89	76.04	99.87	59.90	93.08	99.77	57.55	87.29	72.53	67.85	99.69	64.40	49.96	52.48	49.92	72.55
DRCT [1]	ICML 2024	94.45	94.35	94.24	95.05	95.61	95.38	94.81	94.48	91.66	95.54	93.86	93.48	93.54	84.34	83.20	**67.61**	91.35
Ours	–	**99.83**	**99.82**	**99.80**	**99.84**	**99.81**	**99.80**	**99.85**	**99.83**	**99.80**	**99.82**	**99.78**	**99.83**	**99.84**	**88.92**	**88.68**	56.67	**95.72**

Table 3. Comparison with state-of-the-art methods on the DRCT datasets [1] using Accuracy (%).

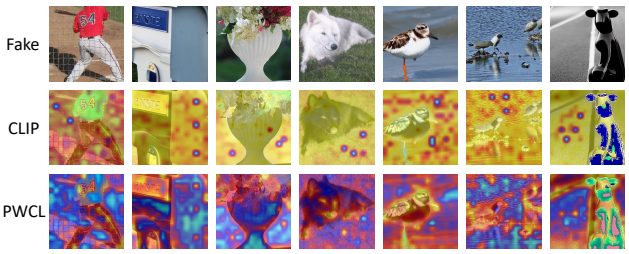


Figure 3. Grad-CAM visualizations comparing the focus of the original CLIP model (middle row) with our PWCL-enhanced model (bottom row).

A closer inspection reveals distinct performance gaps in competing methods. For instance, FatFormer achieves high accuracy on Stable Diffusion but experiences a significant drop on ADM (75.9%) and BigGAN (55.8%). We attribute this decline to its reliance on learning a single pair of prompts, which fails to cover the shifted artifact distributions of unseen generators. In contrast, our PPM-CLIP maintains robust performance across all subsets. This supports our core hypothesis regarding the superiority of a generative approach. Its ability to model the entire distribution of forgery cues enables it to robustly handle the distributional shifts in cross-generator scenarios, a challenge where static, discriminative methods inevitably falter.

DRCT. We further examine robustness against localized manipulations on the DRCT benchmark (Table 3). PPM-CLIP achieves a new state-of-the-art with 95.72% accuracy. Notably, in the "DR Variants" subset (inpainting), where global artifacts are largely absent, traditional detectors like UnivFD collapse to near-chance performance (∼51%). However, our method maintains remarkable resilience (87.80% on SDv1/v2-DR). This robustness is attributed to the synergy between PPM and PWCL: PPM captures high-level semantic inconsistencies, while PWCL compels the encoder to attend to fine-grained anomalies. This capability is visually verified in Figure 3, where our model precisely pinpoints high-frequency artifacts in the forged regions, whereas the baseline CLIP tends to focus on irrelevant semantics.

Prompt Flow	Repository	PWCL	mAcc (%)
		✓	91.2
	✓	✓	92.1
✓		✓	88.9
✓	✓		80.0
✓	✓	✓	99.6

Table 4. Ablation Study on Different Components via Replacement with Zero Tensor.

4.3. Ablation Study

Impact of Core Components. We investigate the contribution of each module on the GenImage dataset, as summarized in Table 4. **First**, we evaluate the Prompt Flow Module (PFM). Disabling this module (reverting to static prompts) results in a significant accuracy drop from 99.6% to 92.1%. This sharp decline mirrors the limitations of previous static methods, confirming that deterministic representations are insufficient for capturing the diverse landscape of forgery artifacts. **Second**, we examine the visual adaptation. Removing the PWCL module leads to the most substantial drop to 80.0%. We attribute this to the "semantic gap": without PWCL, the visual encoder primarily focuses on object semantics rather than forgery artifacts. Furthermore, replacing our frequency-based selection with random selection yields a sub-optimal accuracy of 95.0%, demonstrating that explicit frequency-aware guidance is essential. **Finally**, removing the Prompt Repository leads to a degradation to 88.9%, suggesting that a stable structural prior is necessary to guide the flow optimization.

Performance and Resource Trade-offs. We analyze the trade-off between inference cost and accuracy in Table 5. Even under a minimal sampling budget ($N = 2$), our method reaches 98.6% accuracy, significantly outperforming UnivFD while maintaining high throughput. Increasing the number of samples to 10 yields a peak accuracy of 100.0%, demonstrating the scalability of our approach with respect to computation.

Sensitivity to Hyperparameters. We further analyze the sensitivity of our model to key hyperparameters. For the

Method	Sample(N)	Memory(MB)	Acc(%)	FPS
Univfd		3116	88.8	87.44
Ours	2	3200	98.6	50.51
	4	4764	99.6	35.61
	6	5542	99.8	27.17
	8	6496	99.9	22.19
	10	7976	100.0	18.76

Table 5. Performance and resource trade-offs for our method

λ_{kl}	0.0001	**0.001**	0.01	0.1
mAcc (%)	92.3	**99.6**	98.6	97.1

Table 6. Sensitivity analysis on the KL loss weight (λ_{kl}).

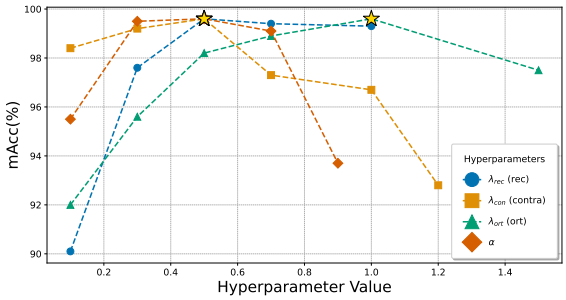


Figure 4. Ablation study on the weights of the reconstruction loss (λ_{rec}), the contrastive loss (λ_{con}), the orthogonal loss weight (λ_{ort}), and the PWCL selection ratio (α).

KL divergence weight, as detailed in Table 6, a small value ($\lambda_{kl} = 0.001$) yields optimal performance. We attribute this to the need for diversity: a larger penalty tends to over-regularize the prompt distribution, suppressing the variance required to model complex forgery patterns. Regarding the auxiliary objectives, Figure 4 illustrates that the weights for reconstruction (λ_{rec}) and contrastive (λ_{con}) losses peak at 0.5, while the orthogonal loss (λ_{ort}) performs best at 1.0. This unified behavior suggests that these objectives should act as gentle regularizers rather than dominating the primary classification task. Finally, for the PWCL selection ratio, a balanced $\alpha = 0.5$ proves most effective (Figure 4), ensuring the model learns from a sufficient variety of both high-frequency artifacts and background textures.

Component Synergy and Sensitivity. Table 7 dissects our Prompt Flow. While ISD captures visual variance, peak performance requires ICD's semantic steering to align adaptive features with discriminative poles, stabilized by IGD priors. For hyperparameters (Table 6, Figure 4), optimal performance relies on balanced regularization. A minimal KL weight ($\lambda_{kl} = 0.001$) preserves prompt diversity against over-regularization. Meanwhile, moderate auxiliary weights ($\lambda_{\text{rec/con}} \approx 0.5$) and a balanced PWCL ratio ($\alpha = 0.5$) guide learning without dominating the primary classification. Together, this delicate balance establishes a robust optimization landscape for universal detection.

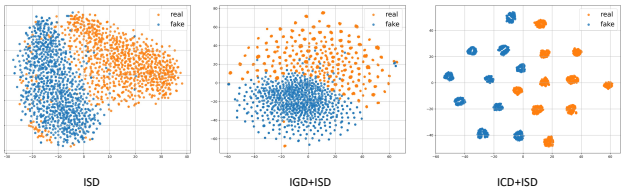


Figure 5. The t-SNE visualization is applied to the fused prompts to confirm that the components capture distinct information as designed.

IGD	ISD	ICD	mAcc (%)
			92.1
		✓	92.6
	✓		95.5
✓			91.3
	✓	✓	98.0
✓		✓	93.1
✓	✓		96.3
✓	✓	✓	99.6

Table 7. Ablation Study(replacing components with zero tensor) on the Contributions and Interplay of the Core Distributions (IGD, ISD, ICD).

Visualization. Figure 5 validates our hierarchical design. While the Image-Specific Distribution (ISD) extracts fine-grained visual details, the Image-Generic Distribution (IGD) further improves separation by providing coarse-grained universal attributes. Finally, the Image-Class Distribution (ICD) acts as a discriminative steering mechanism, successfully disentangling the latent space into linearly separable clusters.

5. Conclusions

In this paper, we introduced PPM-CLIP, a framework that challenges the conventional static, discriminative approach by embracing dynamic, probabilistic reasoning. Its core, Probabilistic Prompt Modeling (PPM), reframes detection as generating and evaluating multiple hypotheses, empowered by a PWCL-enhanced encoder for fine-grained forensic artifacts. Experiments across key benchmarks validate that our paradigm achieves a new SOTA. This work shows the path to universal detection lies in embracing uncertainty and harnessing the power of probabilistic inference.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 62276221).

References

[1] Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. Drct: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *ICML*, 2024. 2, 6, 7

[2] Jiaxuan Chen, Jieteng Yao, and Li Niu. A single simple patch is all you need for ai-generated image detection. *arXiv preprint arXiv:2402.01123*, 2024. 3

[3] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *ICML*, 2020. 2, 6

[4] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1

[5] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1

[7] Zhenglin Huang, Jinwei Hu, Xiangtai Li, Yiwei He, Xingyu Zhao, Bei Peng, Baoyuan Wu, Xiaowei Huang, and Guangliang Cheng. Sida: Social media image deepfake detection, localization and explanation with large multimodal model. In *CVPR*, 2025. 2

[8] Zexi Jia, Chuanwei Huang, Yeshuang Zhu, Hongyan Fei, Xiaoyue Duan, Zhiqiang Yuan, Ying Deng, Jiapei Zhang, Jinchao Zhang, and Jie Zhou. Secret lies in color: Enhancing ai-generated images detection with color distribution analysis. In *CVPR*, 2025. 6

[9] 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. 1

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[12] Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Fuli Feng. Improving synthetic image detection towards generalization: An image transformation perspective. In *KDD*, 2025. 2, 6

[13] Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *CVPR*, 2024. 2, 6

[14] Zhengzhe Liu, Xiaojuan Qi, and Philip H. S. Torr. Global texture enhancement for fake face detection in the wild. In *CVPR*, 2020. 7

[15] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 7

[16] Zihan Liu, Hanyi Wang, Yaoyu Kang, and Shilin Wang. Mixture of low-rank experts for transferable ai-generated image detection. *arXiv preprint arXiv:2404.04883*, 2024. 2

[17] Yunpeng Luo, Junlong Du, Ke Yan, and Shouhong Ding. Lareˆ 2: Latent reconstruction error based method for diffusion-generated image detection. In *CVPR*, 2024. 2

[18] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 1

[19] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, 2023. 2, 5, 6, 7

[20] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, 2020. 6, 7

[21] 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 *ICML*, 2021. 2, 7

[22] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015. 4

[23] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *CVPR*, 2024. 2

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[25] Hao Ruan, Jinliang Lin, Yingxin Lai, Zhiming Luo, and Shaozi Li. Hccm: Hierarchical cross-granularity contrastive and matching learning for natural language-guided drones. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 4524–4533, 2025. 1

[26] Ayush Sarkar, Hanlin Mai, Amitabh Mahapatra, Svetlana Lazebnik, David A Forsyth, and Anand Bhattad. Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. In *CVPR*, pages 28140–28149, 2024. 2

[27] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *ACM CCS*, 2023. 2, 7

[28] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *CVPR*, 2023. 6

[29] 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 *AAAI*, 2024. 6

[30] 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 *CVPR*, pages 28130–28139, 2024. 6

[31] 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 *AAAI*, 2025. 2, 6

[32] Hongsong Wang, Renxi Cheng, Yang Zhang, Chaolei Han, and Jie Gui. Lota: Bit-planes guided ai-generated image detection. In *ICCV*, 2025. 6

[33] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. *CVPR*, pages 8695–8704, 2020. 2, 6, 7

[34] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. Opensdi: Spotting diffusion-generated images in the open world. In *CVPR*, pages 4291–4301, 2025. 2, 6

[35] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. *ICCV*, pages 22445–22455, 2023. 2, 7

[36] Haiwei Wu, Jiantao Zhou, and Shile Zhang. Generalizable synthetic image detection via language-guided contrastive learning. *arXiv preprint arXiv:2305.13800*, 2023. 2

[37] Danni Xu, Shaojing Fan, and Mohan Kankanhalli. Combating misinformation in the era of generative ai models. In *ACM MM*, 2023. 1

[38] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for ai-generated image detection. *ICLR*, 2025. 2, 3, 6

[39] Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable ai-generated image detection. *arXiv preprint arXiv:2411.15633*, 2024. 6

[40] Yongqi Yang, Zhihao Qian, Ye Zhu, Olga Russakovsky, and Yu Wu. D^3: Scaling up deepfake detection by learning from discrepancy. In *CVPR*, 2025. 2

[41] Andrii Yermakov, Jan Cech, and Jiri Matas. Unlocking the hidden potential of clip in generalizable deepfake detection. *arXiv preprint arXiv:2503.19683*, 2025. 2

[42] Kejia Zhang, Keda Tao, Jiasheng Tang, and Huan Wang. Poison as cure: Visual noise for mitigating object hallucinations in lvms. *arXiv preprint arXiv:2501.19164*, 2025. 1

[43] Nan Zhong, Yiran Xu, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Patchcraft: Exploring texture patch for efficient ai-generated image detection. In *CVPRW*, 2023. 2, 3, 6

[44] Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image. *NeurIPS*, 36:77771–77782, 2023. 6