DEDCA: Test-Time Adaptation for Generalized AI-Generated Image Detection

Hao Tan[1,2] **Qimin Zhang**[1] **Lihuang Fang**[3] **Kebing Jin**[4] **Jinghui Qin**[1*]
[1]Guangdong University of Technology
[2]The Hong Kong University of Science and Technology (Guangzhou)
[3]Southern University of Science and Technology
[4]State Key Laboratory of Public Big Data, Guizhou University
tanhao4869@gmail.com, qi-min.zhang@connect.polyu.hk,
12432568@mail.sustech.edu.cn, kbjin@gzu.edu.cn,
scape1989@gmail.com

Abstract

AI-generated image detectors face significant challenges when deployed in real-world environments, particularly when test samples are produced by unseen generators that deviate from the source training distribution. Existing detectors often rely on a single type of evidence, such as semantic representations or low-level generation traces, making them vulnerable when the corresponding cue becomes unreliable. Moreover, adapting detectors to newly emerging generators usually requires target labels or retraining, which is costly and impractical. To address these challenges, we propose Dual-Evidence Disagreement-Constrained Adaptation (DEDCA), a test-time adaptation framework for generalized AI-generated image detection. DEDCA constructs a dual-evidence detector by combining CLIP-based semantic evidence with trace evidence extracted from shuffled entropy and residual statistics, and introduces a reliability-aware gate to perform sample-specific adaptive fusion. Our key idea is to exploit the disagreement between the two evidence streams as an unlabeled adaptation signal, enabling the detector to adjust to target-domain shifts without relying on target labels or blindly trusting a single prediction stream. Experiments show that DEDCA achieves superior performance on the GenImage benchmark by outperforming state-of-the-art AI-generated image detectors and test-time adaptation methods with 92.0% average accuracy.

1 Introduction

Since the rapid progress of generative models [9, 4] has made synthetic images increasingly realistic and accessible, raising growing concerns about misinformation, impersonation, and the credibility of online media [6, 31], AI-generated image detection has become an important problem in visual forensics. Although existing detectors achieve promising results on images generated by known models [28, 45], their performance often drops substantially when facing unseen generators, unseen content distributions, or unknown post-processing operations [33, 1]. Early detectors exploit spatial or frequency artifacts produced by generative models, such as local inconsistencies, spectral traces, and high-frequency residuals [30, 15]. Recent detectors improve generalization by using representations from pretrained large vision-language models [28, 22], model-based reconstruction errors [45, 34, 21], and content-suppressed low-level statistics such as neighboring pixel relationships or local entropy patterns [53, 49]. These studies reveal two major directions for generalized image detection: 1) semantic evidence from strong pretrained visual priors, and 2) trace evidence focusing on local artifact

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

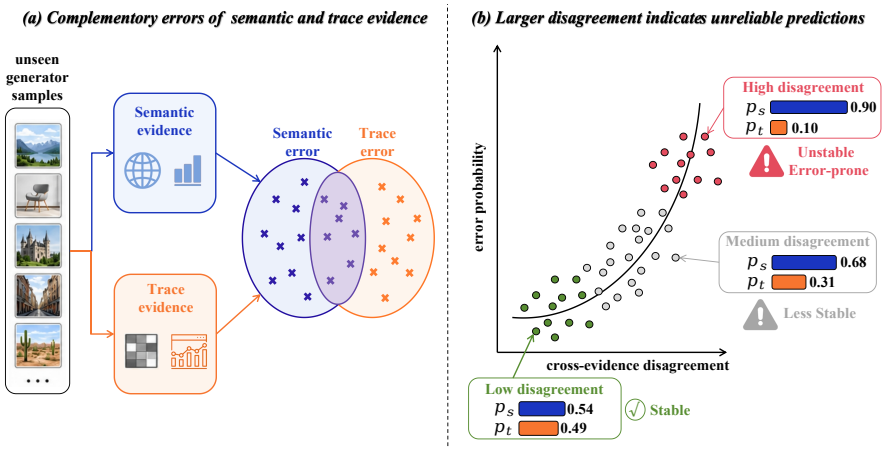


Figure 1: Motivation of dual-evidence disagreement for test-time adaptation. (a) Complementary error patterns between semantic and trace evidence. (b) Cross-evidence disagreement as an indicator of unreliable predictions.

statistics [50]. However, each evidence source has its own limitations. Semantic representations can transfer across content but may be affected by dataset bias and semantic shortcuts, while trace features are closer to the image formation process but can be weakened by generator-specific rendering, compression, resizing, or other post-processing operations [7, 37]. This makes it difficult for a single evidence source to remain reliable under diverse deployment shifts. As illustrated in Fig. 1 (a), we observe that semantic-based and trace-based detectors often make different mistakes on unseen generators and their error sets are only partially overlapped, indicating that the two evidence sources capture complementary cues rather than redundant information, so a robust detector should not rely on a single evidence source, but should explicitly model both semantic-level and trace-level evidence. Furthermore, an effective detector should not only maintain two complementary evidence sources but also estimate how much each source should be trusted for a given input.

Besides, real-world deployment raises another challenge that the target distribution may continuously change [2]. New generators, updated synthesis pipelines, and unknown post-processing operations can appear over time, but collecting newly labeled target samples to retrain is often costly and impractical [39, 12]. Test-time adaptation (TTA) offers a natural solution by adapting a pretrained model with only incoming unlabeled test samples [19], but TTA remains much less explored in AI-generated image detection. Recent work [24] shows the promise of online adaptation for deepfake detection under unknown post-processing and distribution shifts, but its TTA objectives operate on a single prediction stream and rely on entropy minimization or pseudo-label-based updates [41, 27]. However, when the source detector is unreliable on unseen target data, such self-reinforcing objectives may amplify incorrect predictions and lead to confirmation bias or model collapse. Conversely, the dual-evidence design offers a different source of supervision during adaptation. As shown in Fig. 1 (b), samples with larger disagreement between semantic and trace predictions are more likely to be misclassified or assigned unstable predictions. This observation indicates that cross-evidence disagreement can serve as an informative signal for test-time adaptation, enabling the detector to adapt to target shifts without blindly trusting a single prediction stream.

Based on this insight, we propose Dual-Evidence Disagreement-Constrained Adaptation (DEDCA), a test-time adaptation framework for generalized AI-generated image detection. Specifically, DEDCA builds a dual-evidence detector with a semantic branch and a trace branch. The semantic branch uses a frozen CLIP image encoder to extract transferable high-level visual evidence, while the trace branch applies a deterministic Shuffled Entropy-Residual Transform (SERT) to suppress semantic layout and capture generation-related local statistics from entropy maps, high-pass residuals, and gradient magnitudes. The outputs of these two branches are fused by a Decoupled Evidence Reliability Gate (DERG), which decouples their features, models bidirectional evidence interaction, and predicts sample-specific reliability weights for logit fusion. During training, we introduce a cross-evidence disagreement loss to guide dual-evidence learning, encouraging the two branches to produce compatible predictions when both evidence sources are reliable while avoiding a naive collapse into identical behavior. To adapt to a new generator, DEDCA performs Disagreement-Constrained Test-Time Adaptation on unlabeled target samples. Guided by the observation that cross-evidence disagreement

indicates unreliable target predictions, DEDCA reuses this disagreement as an adaptation signal to penalize conflicts between semantic and trace predictions. Extensive experiments on the GenImage [55] benchmark show that DEDCA achieves superior performance by outperforming state-of-the-art (SOTA) AI-generated image detectors and test-time adaptation methods, consistently improving cross-generator robustness and unlabeled target-domain adaptability, especially under substantial shifts among generators. Overall, our contributions can be summarized as follows:

- We propose test-time adaptation for AI-generated image detection from a dual-evidence disagreement perspective, bringing unlabeled target-domain adaptation closer to real-world deployment.
- We propose DEDCA, a dual-evidence detection framework that combines semantic and trace evidence with reliability-aware fusion and exploits cross-evidence disagreement for test-time adaptation.
- Extensive experiments on GenImage show the generalization and adaptability of DEDCA by consistently improving robustness to unseen generators over SOTA detectors and TTA methods.

2 Related Work

Semantic-based Image Detection. Recent AI-generated image detectors increasingly rely on high-level representations from large pretrained models. A representative line of work uses CLIP features as generalizable visual priors, where a frozen vision-language encoder with only a lightweight classifier can achieve strong cross-generator detection performance [28]. Following this paradigm, C2P-CLIP [36] further explores category-common prompts to enhance CLIP-based generalization, while recent multimodal approaches such as SIDA [14] and AIGI-Holmes [54] leverage large multimodal models for detection, localization, and explanation. Although these methods benefit from transferable semantic priors, they may still capture content bias or semantic shortcuts. In contrast, our method treats semantic representations as one evidence source and complements them with trace-level evidence.

Trace-based Image Detection. Another major direction detects AI-generated images by exploiting low-level traces introduced by synthesis pipelines. Early methods focus on spatial artifacts, frequency clues, and high-frequency residuals [30, 44]. Later works improve generalization through reconstruction errors from diffusion or autoencoder [45, 34], neighboring pixel relationships [38], and semantic-suppressed local statistics such as entropy patterns [49]. More recent studies investigate bit-plane cues and bias-reduced training protocols to expose subtle generation artifacts [42, 11]. These methods are closer to the generation process but remain vulnerable to post-processing perturbations. Different from them, our DEDCA treats trace evidence as a complement.

Test-time Adaptation For Image Detection. Test-time adaptation updates a pretrained model using unlabeled target samples, with representative methods based on entropy minimization, augmentation consistency, sample filtering, and stable online adaptation [41, 51, 26, 43, 27, 52]. Despite its relevance to real-world deployment, TTA remains less explored in AI-generated image detection. T2A [24] shows that online adaptation can improve deepfake detectors under unknown post-processing and distribution shifts, while benchmarks such as AIGIBench [18] and MIRAGE [47] further reveal that current detectors still struggle in realistic settings. Unlike the aforementioned single-stream TTA methods, DEDCA exploits cross-evidence disagreement as an unlabeled adaptation signal, reducing dependence on pseudo-labels or entropy from one prediction stream.

3 Methodology

The overall pipeline is illustrated in Fig. 2. DEDCA detects AI-generated images by jointly modeling semantic evidence and trace evidence. Given an input image, DEDCA first extracts high-level visual representations with a frozen CLIP image encoder, while a parallel trace branch applies a deterministic Shuffled Entropy-Residual Transform (SERT) to suppress global semantic layout and expose local artificial statistics. The output of these two evidence branches are then passed to a Decoupled Evidence Reliability Gate (DERG), which estimates their sample-specific reliability and fuses their logits accordingly. Besides, DEDCA uses cross-evidence disagreement in both training

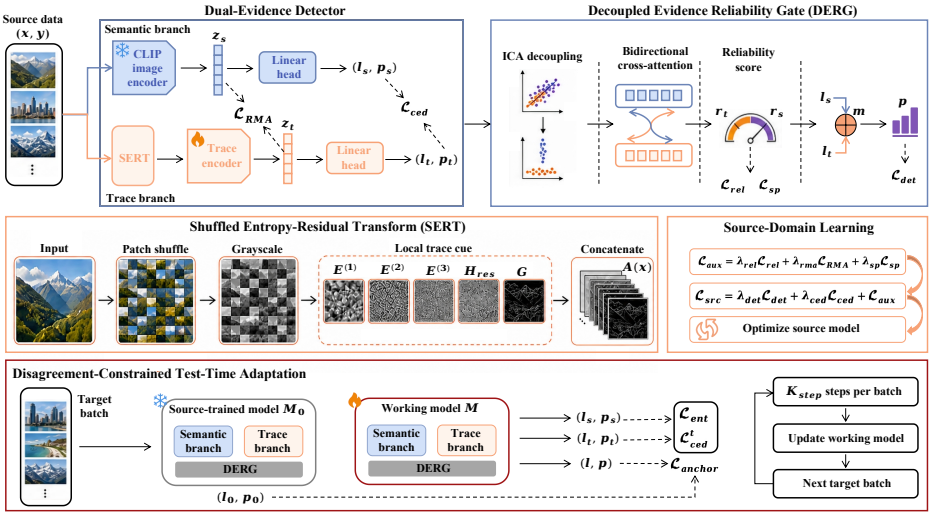


Figure 2: Overall framework of DEDCA. The training stage learns a dual-evidence detector with SERT-based trace modeling and DERG-based reliability fusion, while the test-time stage adapts the detector on unlabeled target batches through disagreement-constrained optimization.

and test-time adaptation. During training, this disagreement regularizes semantic and trace evidence to remain compatible when both are reliable. During test-time adaptation, it serves as an unlabeled constraint for adapting the detector to target-domain shifts.

3.1 Dual-Evidence Detector

Given an input image $x \in \mathbb{R}^{H\times W\times 3}$, DEDCA processes it through two parallel evidence branches to produce features (z_s, z_t) and logits (ℓ_s, ℓ_t), which are subsequently used by DERG for reliability-aware fusion. At the semantic branch, a frozen CLIP [32] image encoder f_s and a trainable lightweight binary semantic classification head are deployed to produce required features as follows:

$$z_s = f_s(x), \quad \ell_s = w_s^T z_s + b_s, \quad p_s = \sigma(\ell_s), \tag{1}$$

where $z_s \in \mathbb{R}^{d_s}$ is the semantic feature, ℓ_s is the semantic logit, p_s is the corresponding fake probability, and w_s and b_s denote the weight and bias of the binary semantic classification head.

At the trace branch, the SERT is adopted to remove the original semantic layout and extract local statistical maps from the transformed image. Specifically, the image is divided into non-overlapping patches and rearranged by a fixed global permutation Π, producing a shuffled image $x_{shuf} = \Pi(x)$. The same permutation is used at all stages, including training, testing, and adaptation, so the transform is deterministic rather than an additional data augmentation. This operation can break the global object and scene layout while preserving local pixel statistics that may contain generation traces. After shuffling, SERT converts the image into a grayscale map $R \in [0,1]^{H\times W}$. Then, we quantize each grayscale intensity into one of B discrete bins as $q(u,v) = min(\lfloor B \cdot R(u,v) \rfloor, B-1) \in \{0, ..., B-1\}$. For each spatial location (u,v), a local window $\mathcal{N}(u,v)$ is used to estimate the histogram of quantized intensities. Subsequently, for each window, the local entropy $E(u,v)$ is defined as $E(u,v) = -\sum_{k=0}^{B-1} p_k(u,v) log(p_k(u,v)+\epsilon)$, where $p_k(u,v) = \frac{1}{|\mathcal{N}(u,v)|}\sum_{(a,b)\in\mathcal{N}(u,v)} \mathbb{I}[q(a,b) = k]$ and $p_k(u,v)$ denote the empirical probability of the k-th bin in this window. This value describes the randomness of local intensity distributions. Since it depends on the distribution of quantized values rather than explicit object structures, it provides a content-suppressed cue for capturing local differences between real and generated images. To cover trace patterns at different spatial granularities, SERT computes entropy maps at the original resolution and on two average-pooled grayscale maps. We denote the resulting maps as $E^{(1)}$, $E^{(2)}$, and $E^{(4)}$, where the superscript indicates the downsampling factor before entropy computation. In particular, $E^{(1)}$ is computed from the original grayscale map, while $E^{(2)}$ and $E^{(4)}$ are computed after $2\times$ and $4\times$ average pooling and then bilinearly upsampled back to $H \times W$. Besides, SERT also extracts two residual cues from the shuffled grayscale image: a high-pass residual map obtained by a Laplacian operator [5] and

a gradient magnitude map obtained from horizontal and vertical Scharr responses [20]. All these maps are normalized per sample. The final trace representation is the channel-wise concatenation of these maps as: $A(x) = Concat[E^{(1)}, E^{(2)}, E^{(4)}, H_{res}, G]$, where $A(x) \in \mathbb{R}^{C_a \times H \times W}$, H_{res} denotes the high-pass residual map, and G denotes the gradient magnitude map. $A(x)$ are then fed into a convolutional trace encoder f_t, which is a shallow convolutional stem followed by a ResNet-18 backbone. The encoder output is passed to a linear classification head to produce the required features. This can be defined as follows:

$$z_t = f_t(A(x)), \quad \ell_t = w_t^T z_t + b_t, \quad p_t = \sigma(\ell_t), \tag{2}$$

where $z_t \in \mathbb{R}^{d_t}$ is the trace feature, ℓ_t is the trace logit, and w_t and b_t are the weight and bias of the trace classification head.

3.2 Decoupled Evidence Reliability Gate

The semantic branch and trace branch provide complementary evidence, but their reliability is not fixed across samples. For instance, semantic evidence may be more stable when low-level traces are weakened by post-processing, whereas trace evidence may be more informative when semantic representations are affected by content bias. Therefore, DEDCA introduces a Decoupled Evidence Reliability Gate (DERG) to estimate the reliability of each branch from its feature-level interaction. Specifically, DERG first applies ICA-based [17] decoupling to decouple the semantic and trace features for reducing the influence of entangled feature correlations before reliability estimation as follows: $\tilde{z}_s = L2Norm((z_s - \mu_s)U_s), \quad \tilde{z}_t = L2Norm((z_t - \mu_t)U_t)$, where U_s and U_t are ICA unmixing matrices, and μ_s and μ_t are corresponding feature means. Then, the decoupled features $\tilde{z}_s$ and $\tilde{z}_t$ are converted into token sequences for cross-evidence interaction. Specifically, $\tilde{z}_s$ and $\tilde{z}_t$ are separately projected and reshaped into T tokens with dimension d, denoted as Z_s, $Z_t \in \mathbb{R}^{T \times d}$. Subsequently, DERG performs bidirectional cross-attention where semantic tokens query trace tokens to obtain trace-conditioned semantic context, while trace tokens query semantic tokens to obtain semantic-conditioned trace context. This procedure can be formulated as $C_s = softmax(\frac{Z_s W_q (Z_t W_k)^T}{\sqrt{d}}) Z_t W_v$ and $C_t = softmax(\frac{Z_t W'_q (Z_s W'_k)^T}{\sqrt{d}}) Z_s W'_v$, where C_s and C_t are the contextualized semantic and trace tokens, respectively. W_q, W_k, W_v, W'_q, W'_k, and W'_v are learnable projection matrices for query, key, and value transformations in the two cross-attention directions.

To estimate reliability, DERG summarizes both the original tokens and their cross-evidence contexts by average pooling. For the semantic stream, the pooled representation $[Mean(Z_s); Mean(C_s)]$ describes semantic evidence together with the trace information it attends to. Similarly, $[Mean(Z_t); Mean(C_t)]$ describes trace evidence under a semantic context. The reliability scores for two branches can be defined as $r_s = softplus(W_s^{rel}[Mean(Z_s); Mean(C_s)] + b_s^{rel})$ and $r_t = softplus(W_t^{rel}[Mean(Z_t); Mean(C_t)] + b_t^{rel})$, where r_s, $r_t > 0$ indicate the estimated reliability scores of semantic and trace evidence, W_s^{rel}, b_s^{rel} and W_t^{rel}, b_t^{rel} are learnable parameters of the semantic and trace reliability scoring layers, respectively. The fused logit ℓ is then obtained by weighting the two branch logits as follows:

$$\ell = m\ell_s + (1 - m)\ell_t, \quad p = \sigma(\ell), \tag{3}$$

where $m = \frac{r_s}{r_s + r_t + \epsilon}$, p is the fused prediction which is used as the main detection output and also serves as the basis for the training and adaptation objectives. In this way, DERG performs adaptive sample-specific evidence selection rather than static ensembling. DERG can increase the contribution of semantic evidence when trace cues are unreliable, or rely more on trace evidence when local statistical patterns provide stronger detection signals.

3.3 Training Objective

DEDCA is pre-trained on labeled source-domain data before test-time adaptation with two goals: 1) Each branch should learn discriminative evidence for distinguishing authentic and AI-generated images from its own feature space, and 2) The model should learn how the two evidence branches interact and coordinate due to their possibly unreliable predictions. Therefore, cross-evidence disagreement is introduced as an explicit training signal for measuring evidence conflict, allowing DEDCA to model the prediction reliability of the two branches rather than treating their predictions independently.

Given a data pair $\{(x, y)\}$, the detector produces the semantic logit ℓ_s, trace logit ℓ_t, and fused logit ℓ. The basic supervised detection loss is defined over all 3 logits as follows:

$$\mathcal{L}_{det} = BCE(\ell_s, y) + BCE(\ell_t, y) + BCE(\ell, y), \tag{4}$$

where $BCE(\cdot, \cdot)$ denotes the binary cross-entropy loss. The branch-level terms ensure that both evidence streams remain individually discriminative, while the fused term directly optimizes the final prediction used at inference. Besides, we also introduce cross-evidence disagreement regularization to guide the interaction between the two branches via their branch-level fake probabilities p_s and p_t, which encourages compatible predictions when both evidence sources are considered reliable, while preserving their complementary behavior when the gate detects an imbalance in reliability. It is expressed as follows:

$$\mathcal{L}_{ced} = wJSD(p_s, p_t)^{\gamma}, \tag{5}$$

where $JSD(p_s, p_t) = \frac{1}{2}KL(p_s||\bar{p}) + \frac{1}{2}KL(p_t||\bar{p})$, $\bar{p} = \frac{p_s+p_t}{2}$, $\bar{p}$ denotes the mean Bernoulli prediction of the two branches, and $KL(\cdot||\cdot)$ is the Kullback–Leibler divergence. Since a direct minimization of this disagreement would force the two branches to behave similarly for all samples, which is undesirable because one evidence stream can be unreliable under certain visual conditions, the disagreement $JSD(p_s, p_t)$ is weighted by the gate confidence. The weight is defined as $w = stopgrad(4m(1-m))$, where m is the semantic reliability weight predicted by DERG, and the $Stopgrad(\cdot)$ prevents the disagreement loss from trivially changing the gate weight m to reduce the penalty. This weight becomes large when DERG assigns comparable reliability to both branches, and becomes small when the gate clearly favors one branch.

Furthermore, an auxiliary calibration loss is deployed to stabilize reliability estimation and improve open-set generalization as follows:

$$\mathcal{L}_{aux} = \lambda_{rel}\mathcal{L}_{rel} + \lambda_{rma}\mathcal{L}_{RMA} + \lambda_{sp}\mathcal{L}_{sp}, \tag{6}$$

where $\mathcal{L}_{rel} = MSE(m, stopgrad(m^*))$ is the reliability supervision item, $MSE(\cdot)$ denotes the mean squared error, $m^* = \frac{exp(-e_s)}{exp(-e_s)+exp(-e_t)}$, which is the soft semantic reliability target from the semantic and trace detection errors $e_s = BCE(\ell_s, y)$ and $e_t = BCE(\ell_t, y)$ of each sample. m^* is detached from the computation graph so that it can serve as a fixed sample-wise supervision signal for the gate, encouraging the gate to assign a higher weight to the branch that better explains the current labeled sample. $\mathcal{L}_{RMA} = \mathbb{I}(y=0)||u-u_r||_2^2 + \mathbb{I}(y=1)(max\{0, \delta - ||u-u_r||_2\})^2$ is the Reliability-Guided Real-Manifold Anchoring we introduced to reduce the dependence on a closed set of source fake generators. It provides an additional realness reference for the fused representation, which is useful when target fake images deviate from the source fake distribution. Real samples are encouraged to stay close to an EMA-based real image prototype μ_r, while fake samples are pushed outside a margin δ. u is defined as $u = L2Norm(mu_s + (1-m)u_t)$, where $u_s = L2Norm(g_s(z_s))$, $u_t = L2Norm(g_t(z_t))$, g_s and g_t are two learnable projection heads for semantic and trace features. $\mathcal{L}_{sp} = ||W_s^{rel}||_1 + ||W_t^{rel}||_1$ is a lightweight sparsity regularization to the reliability scoring layers, where $||\cdot||_1$ denotes the element-wise L1 norm. Overall, the full training loss is defined as follows:

$$\mathcal{L}_{src} = \lambda_{det}\mathcal{L}_{det} + \lambda_{ced}\mathcal{L}_{ced} + \mathcal{L}_{aux}, \tag{7}$$

where λ_* represents the loss weight hyperparameter.

3.4 Disagreement-Constrained Test-Time Adaptation

DEDCA deploys a disagreement-constrained test-time adaptation to handle the new images or domains from unseen generators or undergo unknown post-processing. Let M_0 denote the well-trained source model and M denote the working model initialized from M_0. During test-time adaptation, M_0 is kept fixed as a source anchor, while M is updated sequentially on incoming target images. To avoid unstable updates, DEDCA only updates DERG, the biases of the semantic and trace classification heads, and the affine parameters of BatchNorm layers in the trace branch. This design makes the detector adjust evidence fusion and feature normalization flexibly while preserving the learned representations at the training stage.

Specifically, given an unlabeled input x, the working model produces branch-level probabilities p_s, p_t and p_s, the reliability weight m, and the fused prediction p. We first use entropy minimization on the fused prediction:

$$\mathcal{L}_{ent} = -plog(p+\epsilon) - (1-p)log(1-p+\epsilon), \tag{8}$$

where ϵ is a small constant for numerical stability. Then, DEDCA exploits cross-evidence disagreement as a direct unlabeled signal for target adaptation:

$$\mathcal{L}_{ced}^{t} = wJSD(p_s, p_t)^{\gamma}, \tag{9}$$

where $w = stopgrad(4m(1-m))$. This design follows the observation that when two evidence streams trained from different cues produce conflicting predictions on a target sample, the conflict itself indicates unreliable evidence coordination under the current target distribution. $\mathcal{L}_{ced}^{t}$ penalizes conflicts between semantic and trace predictions when both evidence sources are considered reliable, thereby adapting the reliability gate and branch calibration to the target distribution.

However, unconstrained online adaptation can drift away from the source detector. Thus, we use the fixed source model M_0 to provide stable anchors and deploy an anchoring loss to constrain the shift from the source detector as follows:

$$\mathcal{L}_{anchor} = \lambda_{gate}\mathcal{L}_{gate} + \lambda_{drift}\mathcal{L}_{drift}, \tag{10}$$

where $\mathcal{L}_{gate} = ||\theta_g - \theta_g^0||_2^2$ and $\mathcal{L}_{drift} = a(\ell - \ell^0)^2$. θ_g and θ_g^0 denote the DERG parameters of the working model and the source model. $\mathcal{L}_{gate}$ aims to keep the DERG parameters close to their source-trained values. $\mathcal{L}_{drift}$ aims to constrain the fused logit with a confidence-aware source anchor. For each target sample, M_0 produces a source logit ℓ^0 and probability p^0. The anchor weight a can be defined as $a = stopgrad[(2|p^0 - 0.5|)^{\beta}]$, where β controls how strongly source confidence affects the constraint. For the sample on which the source model is more confident, the a will be assigned a larger value. Overall, the final test-time adaptation objective can be defined as follows:

$$\mathcal{L}_{tta} = \lambda_{ent}\mathcal{L}_{ent} + \lambda_{ced}^{t}\mathcal{L}_{ced}^{t} + \mathcal{L}_{anchor}, \tag{11}$$

For each incoming target batch, DEDCA first computes the source predictions using M_0, then updates the working model M for K_{step} gradient steps by minimizing $\mathcal{L}_{tta}$. The updated working model produces the final predictions for the same batch and is carried forward to the next batch.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. Following recent studies on generalized AI-generated image detection, we evaluate our method on the GenImage [55] benchmark under the leave-one-generator setting. GenImage is a large-scale dataset containing real images from ImageNet and synthetic images produced by representative generative models, including Midjourney [55], Stable Diffusion V1.4 [35], Stable Diffusion V1.5 [35], Wukong [55], ADM [8], GLIDE [25], VQDM [10], and BigGAN [3]. In each run, one generator is held out as the unseen target domain, while the remaining generators are used as source domains for training. This protocol evaluates whether a detector can benefit from diverse source data without being confused by cross-generator interference, and whether it can still generalize to a new generator that is absent during source-domain learning. During test-time adaptation, the held-out target images are provided only as unlabeled test batches. Target labels are never used for adaptation. The working model M is continuously updated along the target stream and is not reset after each batch. All compared TTA methods are evaluated under a batch-transductive setting with a batch size of 64 and 10 adaptation steps per batch. This setting reflects a realistic deployment scenario where a detector trained on known generators encounters images from a newly emerging generator without target annotations. We report accuracy (ACC) and average precision (AP) as the evaluation metrics.

Implementation details. We implement our method in PyTorch and conduct all experiments on a single *NVIDIA RTX 3090* GPU with 24GB memory. Input images are resized to 256×256, while the CLIP branch follows the standard ViT-B/16 preprocessing with 224×224 inputs. The model is first warmed up for $K_{warm} = 5$ epochs, where DERG is disabled and the semantic and trace branches are optimized with their own binary detection head, after which the FastICA transformations are fitted on source-domain features. Joint source-domain training is then performed for 50 epochs. We set λ_{det}, λ_{ced}, λ_{rel}, λ_{RMA}, andλ_{sp} to 0.5, 10, 10, 10, and 0.1, respectively. During test-time adaptation, we update only the DERG parameters, the biases of the semantic and trace classification heads, and the affine parameters of BatchNorm layers in the trace branch. The CLIP encoder, trace convolutional weights, projection heads, and ICA transformations are kept frozen, while BatchNorm

Table 1: Leave-one-generator comparison on GenImage in terms of ACC (%). The best result is highlighted in **bold**, and the second-best result is underlined.

Method	Test Subset								Avg. Acc
	SD1.4	ADM	BigGAN	GLIDE	Midj.	SD1.5	VQDM	Wukong	
FSD (zero-shot)	88.0	74.1	62.1	93.9	75.1	88.0	69.1	88.0	79.8
FSD (10-shot)	88.8	79.2	82.2	97.1	80.9	88.8	76.2	88.8	85.2
MLEP	88.6	74.5	84.8	95.4	76.4	93.0	75.4	91.3	84.9
LARE^2	96.2	**82.5**	60.1	96.9	72.4	90.2	83.5	90.9	84.1
AIDE	95.4	72.8	89.7	95.1	82.6	94.8	71.9	92.1	86.8
MoLD	98.5	77.1	96.8	98.7	81.5	97.4	**84.9**	96.0	91.4
PeTTA	91.3	81.4	84.2	91.0	**90.2**	92.3	70.3	90.3	86.4
PeTTA-DEDCA	93.9	71.9	85.5	94.8	74.9	94.8	71.4	89.1	84.5
T2A	60.1	58.5	50.2	96.8	64.2	64.9	53.7	63.5	64.0
T2A-MLEP	85.9	59.7	79.0	81.1	68.3	82.6	59.3	79.9	74.5
T2A-DEDCA	95.9	76.3	96.1	96.6	76.1	96.7	69.1	96.7	87.9
ReservoirTTA-MLEP	78.1	71.3	61.2	79.8	73.4	81.1	57.9	75.9	72.3
ReservoirTTA-DEDCA	84.4	73.9	86.9	86.2	73.0	84.9	68.5	92.4	81.3
DEDCA (w/o TTA)	97.9	75.7	95.1	97.7	80.2	98.2	73.9	96.2	89.4
DEDCA (Ours)	**99.3**	80.2	**98.5**	**99.1**	82.2	**99.1**	78.7	**98.7**	**92.0**

Table 3: Ablation study of the source-domain detector under the leave-ADM setting. All variants are evaluated without test-time adaptation.

Setting	ACC	AP
Evidence source		
Semantic only	$66.8_{\downarrow 11.8}$	$78.6_{\downarrow 9.1}$
Trace only	$69.9_{\downarrow 7.7}$	$82.2_{\downarrow 5.0}$
Fixed avg. fusion	$73.0_{\downarrow 3.6}$	$84.9_{\downarrow 1.8}$
SERT design		
w/o patch shuffle	$74.4_{\downarrow 1.7}$	$85.1_{\downarrow 1.6}$
w/o multi-scale entropy	$71.7_{\downarrow 5.3}$	$85.4_{\downarrow 1.3}$
DERG design		
w/o ICA decoupling	$74.2_{\downarrow 2.0}$	$84.5_{\downarrow 2.3}$
w/o bi-cross-attn.	$72.8_{\downarrow 3.8}$	$83.7_{\downarrow 3.2}$
Source objectives		
w/o $\mathcal{L}_{ced}$	$70.4_{\downarrow 7.0}$	$79.3_{\downarrow 8.3}$
w/o $\mathcal{L}_{rel}$	$74.8_{\downarrow 1.2}$	$83.8_{\downarrow 3.1}$
w/o $\mathcal{L}_{\mathrm{RMA}}$	$73.6_{\downarrow 2.8}$	$81.7_{\downarrow 5.5}$
DEDCA full	75.7	86.5

Table 2: Cross-generator analysis of DEDCA on GenImage in terms of ACC/AP (%).

Excluded	Model	Test Subset								Avg. ACC/AP
		SD1.4	ADM	BigGAN	GLIDE	Midj.	SD1.5	VQDM	Wukong	
SD1.4	DEDCA (w/o TTA)	97.9/99.8	99.0/99.9	99.3/100	98.5/99.9	88.9/98.3	98.0/99.8	98.9/99.9	96.4/99.6	97.1/99.7
	DEDCA	**99.3/100**	99.4/100	99.6/100	99.1/100	92.8/99.2	99.4/100	99.4/100	98.6/99.9	98.5/99.9
ADM	DEDCA (w/o TTA)	97.5/99.4	75.7/86.5	97.4/99.8	97.1/99.9	95.8/99.2	97.4/99.9	97.3/99.5	97.5/99.8	94.5/98.0
	DEDCA	99.0/99.9	**80.2/90.5**	99.0/100	98.6/100	97.6/99.7	99.2/100	99.0/99.9	99.4/100	96.5/98.8
BigGAN	DEDCA (w/o TTA)	98.9/99.9	99.1/99.9	95.1/99.4	98.6/99.9	94.6/99.3	98.8/99.9	98.6/99.9	97.9/99.8	97.7/99.8
	DEDCA	99.5/100	99.4/100	**98.5/99.9**	99.1/100	95.9/99.6	99.4/100	99.4/100	98.7/99.9	98.7/99.9
GLIDE	DEDCA (w/o TTA)	98.4/99.9	98.4/99.9	98.6/100	97.7/99.8	95.0/99.1	98.4/99.9	98.4/99.9	98.1/99.8	97.9/99.8
	DEDCA	99.5/100	99.3/99.9	99.4/100	**99.1/99.9**	96.1/99.5	99.4/100	99.4/100	99.1/99.9	98.9/99.9
Midjourney	DEDCA (w/o TTA)	98.0/99.7	98.1/99.9	98.5/100	97.7/99.8	80.2/94.1	98.1/99.7	93.1/98.6	97.8/99.6	95.2/98.9
	DEDCA	99.3/99.9	99.4/100	99.4/100	98.9/99.9	**82.2/96.8**	99.2/99.9	98.0/99.8	99.0/99.8	96.9/99.5
SD1.5	DEDCA (w/o TTA)	98.1/99.8	99.1/99.9	99.3/100	98.7/99.1	95.2/99.4	98.2/99.8	99.0/99.9	97.3/99.8	98.1/99.7
	DEDCA	99.4/100	99.4/100	99.6/100	99.0/99.9	96.4/99.6	**99.1/99.9**	99.6/100	98.7/99.9	98.9/99.9
VQDM	DEDCA (w/o TTA)	98.8/99.9	98.3/99.9	99.1/100	98.2/99.8	90.1/98.9	98.7/99.9	73.9/85.8	97.2/99.6	94.3/98.0
	DEDCA	99.5/100	99.4/100	99.7/100	99.1/99.9	93.9/99.5	99.6/100	**78.7/89.6**	98.3/99.9	96.0/98.6
Wukong	DEDCA (w/o TTA)	98.8/99.9	99.1/100	99.1/100	98.7/99.9	96.2/99.5	99.0/99.9	99.1/99.9	96.2/99.5	98.3/99.8
	DEDCA	99.4/99.9	99.5/100	99.6/100	99.2/100	97.1/99.7	99.6/100	99.4/100	**98.7/99.7**	99.1/99.9

Table 4: Ablation study of the test-time adaptation stage under the leave-ADM setting. All variants start from the same source-trained DEDCA checkpoint and adapt on unlabeled target batches.

Setting	ACC	AP
DEDCA w/o TTA	$75.7_{\downarrow 5.6}$	$86.5_{\downarrow 4.4}$
$\mathcal{L}_{ent}$ only	$76.1_{\downarrow 5.1}$	$86.9_{\downarrow 4.0}$
w/o $\mathcal{L}_{ent}$	$79.0_{\downarrow 1.5}$	$89.7_{\downarrow 0.9}$
w/o $\mathcal{L}^t_{ced}$	$77.1_{\downarrow 3.9}$	$87.2_{\downarrow 3.6}$
w/o $\mathcal{L}_{gate}$	$79.7_{\downarrow 0.6}$	$89.6_{\downarrow 1.0}$
w/o $\mathcal{L}_{drift}$	$78.9_{\downarrow 1.6}$	$89.0_{\downarrow 1.7}$
w/o $\mathcal{L}_{anchor}$	$77.9_{\downarrow 2.9}$	$87.8_{\downarrow 3.0}$
Update all params	$78.3_{\downarrow 2.4}$	$89.8_{\downarrow 0.8}$
DEDCA full	80.2	90.5

layers use fixed source-domain running statistics throughout adaptation. We deploy Adam [16] with a learning rate of $1e^{-3}$ and perform $K_{step} = 10$ gradient steps for each incoming target batch. We set λ_{ent}, λ^t_{ced}, λ_{gate}, and λ_{drift} to 0.5, 10, 0.2, and 0.2, respectively. We set $\gamma = 2$, $\delta = 1$, $\beta = 2$, and $B = 32$. SERT divides each 256×256 image into non-overlapping 8×8-pixel patches and uses one fixed permutation generated with seed 42 across training, validation, testing, TTA, and all leave-one-generator folds. Local entropy is computed using a 4×4 window with stride 1 at downsampling factors of 1, 2, and 4. Entropy, Laplacian, and Scharr responses are computed after reassembling the shuffled patches, retaining internal seams and applying reflect padding only at the outer image boundary. The semantic and trace features are both 512-dimensional and are projected into $T = 16$ tokens of dimension $d = 32$.

Baselines. We compare the DEDCA with the multiple state-of-the-art AI-generated image detection methods and TTA methods, including **FSD** [46], **AIDE** [48] **MLEP** [49], **LARE2** [23], **MoLD** [29], **PeTTA** [13], **T2A** [24], and **ReservoirTTA** [40].

4.2 Main Results

Table 1 shows the main experimental results. DEDCA (w/o TTA) denotes our detector without test-time adaptation, while DEDCA denotes our detector with the proposed disagreement-constrained test-time adaptation. For baselines written as TTA–detector, such as T2A-DEDCA or ReservoirTTA-MLEP, we apply the corresponding TTA algorithm to the source-only version of the detector. As shown in Table 1, DEDCA achieves the best average ACC of $92.02 \pm 0.68\%$ over five runs (mean $\pm$ standard deviation), compared with $91.41 \pm 0.94\%$ for MoLD and $89.39 \pm 0.52\%$ for DEDCA (w/o TTA). Both improvements are statistically significant under two-sided paired t-tests ($p = 0.021$ and

$p < 0.001$, respectively). This gain indicates that the proposed adaptation stage provides consistent benefits beyond the source-trained dual-evidence detector. DEDCA obtains the best results on five out of eight target generators, including SD1.4 (99.3%), BigGAN (98.5%), GLIDE (99.1%), SD1.5 (99.1%), and Wukong (98.7%). Although LARE^2 and MoLD perform better on ADM and VQDM, respectively, their performance is less balanced across different generators, e.g., LARE^2 drops to 60.1% on BigGAN and MoLD reaches only 77.1% on ADM. Conversely, DEDCA maintains strong results across both diffusion- and GAN-generated subsets, indicating stronger robustness under diverse generator shifts. Besides, those generic TTA baselines, PeTTA and ReservoirTTA, do not consistently improve their base detectors, indicating that general-purpose adaptation strategies may be unstable for AI-generated image detection. T2A, which is a task-specific TTA for AI-generated image detection, reaches 87.9% when applied to DEDCA, but it still remains 4.1% below DEDCA on average, showing the superiority of using cross-evidence disagreement as the adaptation signal.

The effectiveness of DEDCA can be further understood from its disagreement-driven adaptation behavior. When the semantic and trace branches conflict on an unseen target sample, the conflict indicates that at least one evidence stream is not well calibrated. DEDCA uses this conflict as an explicit optimization signal. If semantic evidence remains confident but trace evidence is weak, the cross-evidence disagreement loss encourages the adaptable trace-side parameters to match the new trace statistics. Conversely, if the trace evidence is informative but the semantic response is less reliable, DEDCA does not aggressively adapt the semantic encoder, since the frozen CLIP branch is intended to preserve a stable visual prior, and instead adjusts the reliability gate to reduce the semantic weight and rely more on the trace branch. In harder cases where one evidence source cannot be effectively corrected, the gate still provides a fallback by suppressing the unreliable branch. Therefore, DEDCA can improve target-domain performance by turning cross-evidence conflicts into a label-free calibration signal for both trace-side adjustment and reliability-aware fusion, rather than blindly enforcing self-training labels.

Furthermore, we also provide a detailed cross-generator analysis in Table 2. For each excluded generator, we evaluate DEDCA (w/o TTA) and DEDCA on all generator subsets. It can be observed that DEDCA maintains high ACC/AP on most generator subsets even when different generators are excluded from training, indicating that under our dual-evidence design, heterogeneous source generators do not introduce severe mutual interference.

4.3 Ablation Study

We conduct ablation studies under the leave-ADM setting, where ADM is held out as the unseen target generator. This setting is challenging enough to reveal the contribution of each component, while still allowing a clear comparison between the source-domain detector and the test-time adaptation stage. All results are reported in terms of ACC and AP. In Table 3, Table 4, and the following ablation analysis, performance drops are reported as relative percentages with respect to the corresponding full model. Table 3 shows the ablation results of the source-domain detector. Compared with full DEDCA, using only the semantic or trace branch causes clear performance drops, with ACC decreasing by 11.8% and 7.7%, respectively, confirming that the two evidence sources are complementary. Directly averaging the two branches is also inferior to DERG, with a 3.6% ACC drop, indicating that adaptive reliability estimation is more effective than static fusion. Among the architectural components, removing multi-scale entropy maps and bidirectional cross-attention leads to relatively larger ACC drops of 5.3% and 3.8%, showing the importance of multi-granularity trace modeling and cross-evidence interaction. For the training objectives, removing $\mathcal{L}_{ced}$ results in the largest degradation, with ACC/AP decreasing by 7.0%/8.3%, which indicates that the disagreement constraint is crucial for aligning semantic and trace evidence while preserving their complementary roles.

Table 4 further validates the effectiveness of the disagreement-constrained test-time adaptation. DEDCA improves over DEDCA without TTA by 5.6% ACC and 4.4% AP, demonstrating that unlabeled target adaptation brings consistent gains. In contrast, using $\mathcal{L}ent$ alone only achieves limited improvement and remains 5.1%/4.0% lower than the full DEDCA, indicating that confidence minimization alone is insufficient. Removing $\mathcal{L}^t ced$ causes a substantial drop of 3.9% ACC and 3.6% AP, confirming that target-domain cross-evidence disagreement provides the main adaptation signal. Overall, all the above results show the effectiveness of our proposed DEDCA.

5 Conclusion

In this work, we propose Dual-Evidence Disagreement-Constrained Adaptation (DEDCA), a test-time adaptation framework for generalized AI-generated image detection under unseen generators. DEDCA builds a dual-evidence detector by combining CLIP-based semantic evidence with trace evidence extracted by the Shuffled Entropy-Residual Transform (SERT). To effectively coordinate the two evidence streams, we introduce the Decoupled Evidence Reliability Gate (DERG), which estimates sample-specific reliability and performs adaptive logit fusion. Beyond source-domain learning, DEDCA further performs Disagreement-Constrained Test-Time Adaptation, where cross-evidence disagreement is used as an unlabeled adaptation signal to adjust the detector on incoming target batches without target labels. Experiments on GenImage demonstrate that DEDCA achieves stronger cross-generator generalization than existing AI-generated image detectors and TTA baselines, showing the effectiveness of exploiting complementary semantic and trace evidence and disagreement-based adaptation for generalized detection.

6 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62206314, Science and Technology Projects in Guangzhou under Grant No. 2024A04J4388.

References

[1] Reza Babaei, Samuel Cheng, Rui Duan, and Shangqing Zhao. Generative artificial intelligence and the evolving challenge of deepfake detection: A systematic analysis. *Journal of Sensor and Actuator Networks*, 14(1):17, 2025.

[2] Spiros Baxevanakis, Giorgos Kordopatis-Zilos, Panagiotis Galopoulos, Lazaros Apostolidis, Killian Levacher, Ipek Baris Schlicht, Denis Teyssou, Ioannis Kompatsiaris, and Symeon Papadopoulos. The mever deepfake detection service: Lessons learnt from developing and deploying in the wild. In *Proceedings of the 1st International Workshop on Multimedia AI against Disinformation*, pages 59–68, 2022.

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

[4] Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion models. *IEEE transactions on knowledge and data engineering*, 36(7):2814–2830, 2024.

[5] Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, 2024.

[6] 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*, pages 103–120. Springer, 2020.

[7] 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)*, pages 1–5. IEEE, 2023.

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

[9] M Durgadevi et al. Generative adversarial network (gan): A general review on different variants of gan and applications. In *2021 6th international conference on communication and electronics systems (ICCES)*, pages 1–8. IEEE, 2021.

[10] 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*, pages 10696–10706, 2022.

[11] Fabrizio Guillaro, Giada Zingarini, Ben Usman, Avneesh Sud, Davide Cozzolino, and Luisa Verdoliva. A bias-free training paradigm for more general ai-generated image detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18685–18694, 2025.

[12] Febryan Hidayat, Khasnan Hanif, and Maratus Sholikhah Miftakul Janah. The use of midjourney in design buying and selling business from the perspective of ibn ashur's maqashid sharia. *Indonesian Journal of Islamic Economic Law*, 2(1):15–28, 2025.

[13] Trung-Hieu Hoang, Duc M Vo, et al. Persistent test-time adaptation in recurring testing scenarios. *Advances in Neural Information Processing Systems*, 37:123402–123442, 2024.

[14] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28831–28841, 2025.

[15] Taehoon Kim, Jongwook Choi, Yonghyun Jeong, Haeun Noh, Jaejun Yoo, Seungryul Baek, and Jongwon Choi. Beyond spatial frequency: Pixel-wise temporal frequency-based deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11198–11207, 2025.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Sangmin S Lee, Kiwon Lee, and Guiyeom Kang. Eeg artifact removal by bayesian deep learning & ica. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 932–935. IEEE, 2020.

[18] Ziqiang Li, Jiazhen Yan, Ziwen He, Kai Zeng, Weiwei Jiang, Lizhi Xiong, and Zhangjie Fu. Is artificial intelligence generated image detection a solved problem? *arXiv preprint arXiv:2505.12335*, 2025.

[19] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

[20] Hanyang Lin, Yongzhao Zhan, Shiqin Liu, Xiao Ke, and Yuzhong Chen. A deep learning based bank card detection and recognition method in complex scenes: A deep learning based bank card detection and recognition method in complex scenes. *Applied Intelligence*, 52(13):15259–15277, 2022.

[21] Huanqi Lin, Jinghui Qin, Xiaoqi Wu, Tianshui Chen, and Zhijing Yang. Revisiting dire: towards universal ai-generated image detection. *Neural Networks*, page 108084, 2025.

[22] 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*, pages 10770–10780, 2024.

[23] Yunpeng Luo, Junlong Du, Ke Yan, and Shouhong Ding. Lareˆ 2: Latent reconstruction error based method for diffusion-generated image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17006–17015, 2024.

[24] Hong-Hanh Nguyen-Le, Van-Tuan Tran, Dinh-Thuc Nguyen, and Nhien-An Le-Khac. Think twice before adaptation: Improving adaptability of deepfake detection via online test-time adaptation. *arXiv preprint arXiv:2505.18787*, 2025.

[25] 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.

[26] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

[27] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

[28] 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*, pages 24480–24489, 2023.

[29] NaHyeon Park, Kunhee Kim, Junsuk Choe, and Hyunjung Shim. Rethinking the use of vision transformers for ai-generated image detection. *arXiv preprint arXiv:2512.04969*, 2025.

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

[31] Jinghui Qin, Jialin Chen, Tianshui Chen, and Zhijing Yang. Multimodal feature disentangle-fusion network for detecting and grounding multi-modal media manipulation. *Expert Systems with Applications*, page 132076, 2026.

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

[33] Md Shohel Rana, Mohammad Nur Nobi, Beddhu Murali, and Andrew H Sung. Deepfake detection: A systematic literature review. *IEEE access*, 10:25494–25513, 2022.

[34] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9130–9140, 2024.

[35] 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*, pages 10684–10695, 2022.

[36] 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, pages 7184–7192, 2025.

[37] 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 on Artificial Intelligence*, volume 38, pages 5052–5060, 2024.

[38] 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*, pages 28130–28139, 2024.

[39] Francesco Tassone, Luca Maiano, and Irene Amerini. Continuous fake media detection: adapting deepfake detectors to new generative techniques. *Computer Vision and Image Understanding*, 249:104143, 2024.

[40] Guillaume Vray, Devavrat Tomar, Xufeng Gao, Jean-Philippe Thiran, Evan Shelhamer, and Behzad Bozorgtabar. Reservoirtta: Prolonged test-time adaptation for evolving and recurring domains. *arXiv preprint arXiv:2505.14511*, 2025.

[41] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[42] Hongsong Wang, Renxi Cheng, Yang Zhang, Chaolei Han, and Jie Gui. Lota: Bit-planes guided ai-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17246–17255, 2025.

[43] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7201–7211, 2022.

[44] 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*, pages 8695–8704, 2020.

[45] 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*, pages 22445–22455, 2023.

[46] Shiyu Wu, Jing Liu, Jing Li, and Yequan Wang. Few-shot learner generalizes across ai-generated image detection. *arXiv preprint arXiv:2501.08763*, 2025.

[47] Cheng Xia, Manxi Lin, Jiexiang Tan, Xiaoxiong Du, Yang Qiu, Junjun Zheng, Xiangheng Kong, Yuning Jiang, and Bo Zheng. Mirage: Towards ai-generated image detection in the wild. *arXiv preprint arXiv:2508.13223*, 2025.

[48] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for ai-generated image detection. In *International Conference on Learning Representations*, volume 2025, pages 70702–70720, 2025.

[49] Lin Yuan, Xiaowan Li, Yan Zhang, Jiawei Zhang, Hongbo Li, and Xinbo Gao. Mlep: Multi-granularity local entropy patterns for universal ai-generated image detection. *arXiv preprint arXiv:2504.13726*, 2025.

[50] Hongji Zhang, Hao Tan, Jinghui Qin, and Zhijing Yang. Mff-net: Multi-view feature fusion network for ai-generated image detection. *Expert Systems with Applications*, page 134394, 2026.

[51] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

[52] Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. Come: Test-time adaption by conservatively minimizing entropy. *arXiv preprint arXiv:2410.10894*, 2024.

[53] Chenming Zhou, Jiaan Wang, Yu Li, Lei Li, Juan Cao, and Sheng Tang. Beyond semantic features: Pixel-level mapping for generalized ai-generated image detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 36101–36109, 2026.

[54] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18746–18758, 2025.

[55] 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. *Advances in neural information processing systems*, 36:77771–77782, 2023.

A A Brief Introduction to Baselines

We compare the DEDCA with the multiple state-of-the-art AI-generated image detection methods and TTA methods, including **FSD** [46], **AIDE** [48], **MLEP** [49], **LARE2** [23], **MoLD** [29], **PeTTA** [13], **T2A** [24], and **ReservoirTTA** [40]. The brief introduction of each baseline is listed as follows:

- **FSD** [46]: It formulates generalized AI-generated image detection as a few-shot classification problem and learns a prototypical metric space, where images from the same generator are encouraged to stay close to each other.
- **AIDE** [48]: It combines CLIP-based semantic features with low-level artifact features extracted from high- and low-frequency patches selected by DCT scoring, and concatenates these features for image detection.
- **MLEP** [49]: It constructs multi-granularity local entropy patterns from shuffled image patches at multiple scales, aiming to capture source-invariant local statistical cues while disrupting semantic content.
- **LARE2** [23]: It detects diffusion-generated images by extracting latent reconstruction error features and refining image features with reconstruction-error guidance.
- **MoLD** [29]: It rethinks CLIP-ViT features for AI-generated image detection and dynamically integrates features from multiple ViT layers with a gating mechanism, rather than relying only on the final-layer representation.
- **PeTTA** [13]: It targets persistent test-time adaptation in recurring testing scenarios and adjusts the adaptation strategy when the model tends to diverge, balancing target adaptation with collapse prevention.
- **T2A** [24]: It is an online TTA method for deepfake detection that avoids relying solely on entropy minimization by introducing uncertainty-aware negative learning, uncertain sample prioritization, and gradient masking.
- **ReservoirTTA** [40]: It addresses prolonged TTA under evolving and recurring domains by maintaining a reservoir of domain-specialized models and routing incoming samples to suitable specialists through online domain clustering.

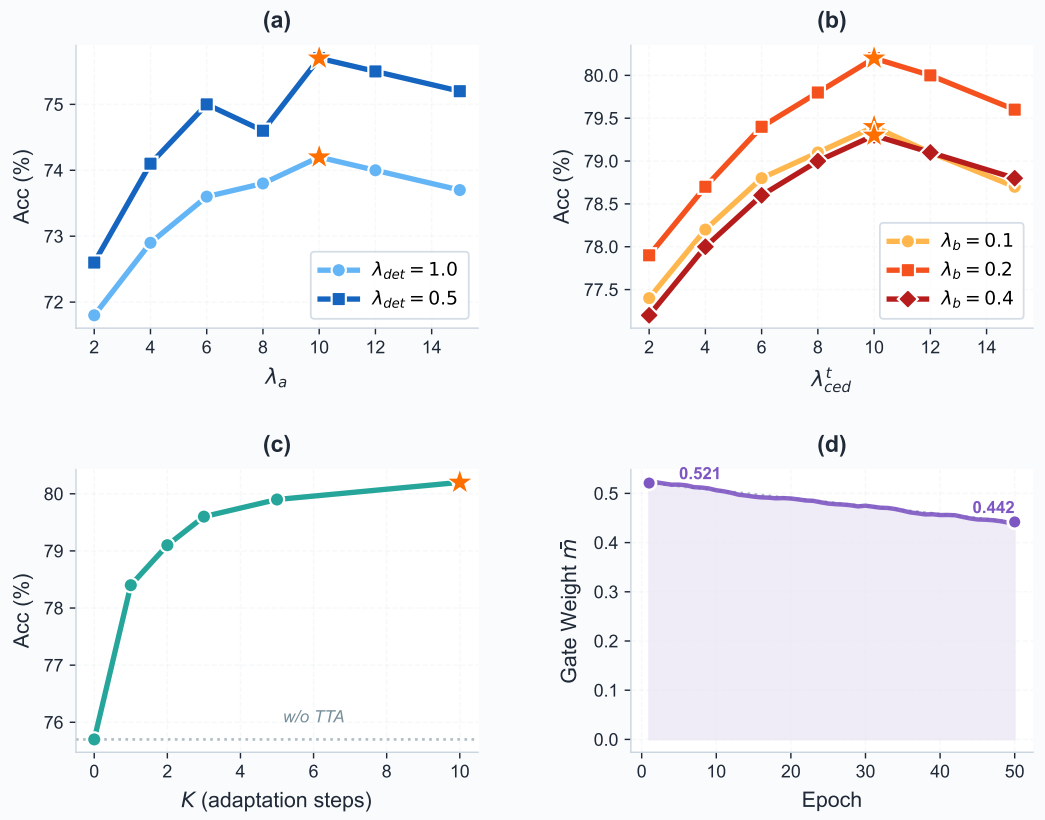


Figure 3: Parameter and gate analysis of DEDCA under the leave-ADM setting. (a) Effect of source-domain loss balance, where $\lambda_{ced} = \lambda_{rel} = \lambda_{\text{RMA}} = \lambda_a$ and $\lambda_{sp} = 0.1$. (b) Effect of test-time adaptation loss balance, where $\lambda_{ent} = 0.5$ and $\lambda_{gate} = \lambda_{drift} = \lambda_b$. (c) Effect of the number of adaptation steps K_{step}. (d) Evolution of the average gate weight $\bar{m}$ during source-domain training.

B Hyperparameter Analysis

Figure 3 (a) studies the source-domain loss balance. We fix $\lambda_{sp} = 0.1$ and set $\lambda_{ced} = \lambda_{rel} = \lambda_{\text{RMA}} = \lambda_a$. It can be observed that reducing λ_{det} from 1.0 to 0.5 consistently improves performance under different λ_a values, suggesting that overly emphasizing the supervised detection objective may bias the model toward source-generator patterns. When $\lambda_{det} = 0.5$, the ACC increases from 72.6% to 75.7% as λ_a grows from 2 to 10, while larger values such as 12 and 15 bring no further gain. This indicates that stronger evidence regularization is beneficial for generalized detection, but an excessively large weight may slightly over-constrain the source-domain learning. Figure 3 (b) analyzes the TTA loss balance, where $\lambda_{ent} = 0.5$ is fixed and $\lambda_{gate} = \lambda_{drift} = \lambda_b$. The best result is achieved when $\lambda_b = 0.2$ and $\lambda^t_{ced} = 10$, reaching 80.2% ACC. In contrast, weaker anchoring with $\lambda_b = 0.1$ and stronger anchoring with $\lambda_b = 0.4$ both lead to lower performance, showing that test-time adaptation requires a proper balance between target-domain disagreement optimization and source-model preservation.

Figure 3 (c) shows the effects of the number of adaptation steps. Compared with the non-adapted baseline, using only one adaptation step improves ACC from 75.7% to 78.4%, indicating that DEDCA can quickly exploit unlabeled target data. The performance finally increases to 80.2% at $K = 10$. Figure 3 (d) visualizes the evolution of the epoch-averaged reliability weight $\bar{m}$ during source-domain training. Since m denotes the fusion weight of the semantic branch and $1 - m$ corresponds to the trace branch, the decrease from 0.521 to 0.442 indicates that DERG gradually shifts the fused prediction from a slight preference for semantic evidence to a stronger reliance on trace evidence. This observation provides an additional insight: semantic cues may be more useful for establishing a

transferable representation at the early stage, whereas generation-related trace cues become more reliable after sufficient cross-evidence optimization.

C Robustness Evaluation

To evaluate robustness under realistic image transformations, we apply common post-processing operations to the held-out test sets and report the average results over all eight leave-one-generator settings. Specifically, the post-processing operations include color saturation adjustment with factor 2.0, color contrast adjustment with factor 0.6, Gaussian blur with $\sigma = 1.0$, resizing with scale factor 0.75, and JPEG compression with quality factor 75. These operations are applied to both real and fake samples.

Table 5: Robustness evaluation under common post-processing operations. Δ denotes the absolute ACC drop compared with the clean average performance of each method.

Method	Color Saturation			Color Contrast			Gaussian Blur			Resize			JPEG			Average		
	ACC	AP	Δ	ACC	AP	Δ	ACC	AP	Δ	ACC	AP	Δ	ACC	AP	Δ	ACC	AP	Δ
MoLD	89.0	96.8	2.4	88.1	96.0	**3.3**	86.7	94.8	**4.7**	87.2	95.2	**4.2**	86.9	95.5	4.5	87.6	95.7	**3.8**
T2A-DEDCA	83.9	94.0	4.0	82.5	92.8	5.4	80.6	91.2	7.3	81.0	91.6	6.9	81.7	92.1	6.2	81.9	92.3	6.0
ReservoirTTA-DEDCA	77.5	90.6	3.8	77.0	89.2	4.3	75.1	87.5	6.2	74.9	88.1	6.4	76.4	88.6	4.9	76.2	88.8	5.1
DEDCA (w/o TTA)	85.2	96.1	4.2	83.4	94.8	6.0	81.0	92.6	8.4	81.8	93.2	7.6	82.5	93.9	6.9	82.8	94.1	6.6
DEDCA	**90.4**	**98.0**	**1.6**	**88.6**	**97.0**	3.4	**86.8**	**95.9**	5.2	**86.7**	**95.6**	5.3	**88.0**	**96.7**	**4.0**	**88.1**	**96.6**	3.9

Table 5 reports the robustness evaluation results. DEDCA achieves the best average ACC/AP of 88.1%/96.6% across all post-processing operations, outperforming DEDCA (w/o TTA) by 5.3% ACC and 2.5% AP. It also reduces the average ACC drop from 6.6 to 3.9, showing that the proposed adaptation can effectively compensate for distribution shifts introduced by post-processing. DEDCA achieves the highest ACC of 86.8% under Gaussian blur and remains competitive under resizing with an ACC of 86.7%. MoLD shows competitive robustness in terms of ACC drop, which is reasonable since its CLIP-based multi-layer representation is less directly tied to low-level trace statistics. However, its absolute average ACC/AP remains lower than DEDCA. Compared with generic TTA baselines such as T2A-DEDCA and ReservoirTTA-DEDCA, DEDCA consistently achieves stronger performance, indicating that disagreement-constrained adaptation is more suitable for robust generalized AI-generated image detection.

Table 6: Computational cost comparison. Infer Time denotes the non-TTA inference time. Params@TTA denotes the number of parameters updated during test-time adaptation.

Method	Total Params	Params@TTA	Infer Time	TTA Time
MLEP	1.40M	–	1.42	–
LARE^2	104.31M	–	0.85	–
MoLD	303.71M	–	10.72	–
Xception	20.81M	–	1.57	–
T2A (Xception)	20.81M	54.5K	1.57	11.70
PeTTA (ResNet-50)	25.56M	53.1K	0.65	3.63
ReservoirTTA (ResNet-50)	25.56M	53.1K	0.65	3.02
DEDCA (w/o TTA)	98.07M	–	2.84	–
DEDCA (Ours)	98.07M	540.3K	2.84	4.54

D Computational Efficiency

We further evaluate the computational efficiency of DEDCA to examine the additional cost introduced by the dual-evidence detector and the disagreement-constrained test-time adaptation. Table 6 reports the model size, the number of parameters updated during test-time adaptation, the non-TTA inference time, and the adaptation time. All measurements are conducted on a single *NVIDIA RTX 3090* GPU. It can be observed that DEDCA achieves a favorable balance between detection performance and computational cost. Compared with MoLD, DEDCA uses fewer parameters and reduces the total test-time cost from 10.72 ms to 7.38 ms even when its adaptation time is included, while improving

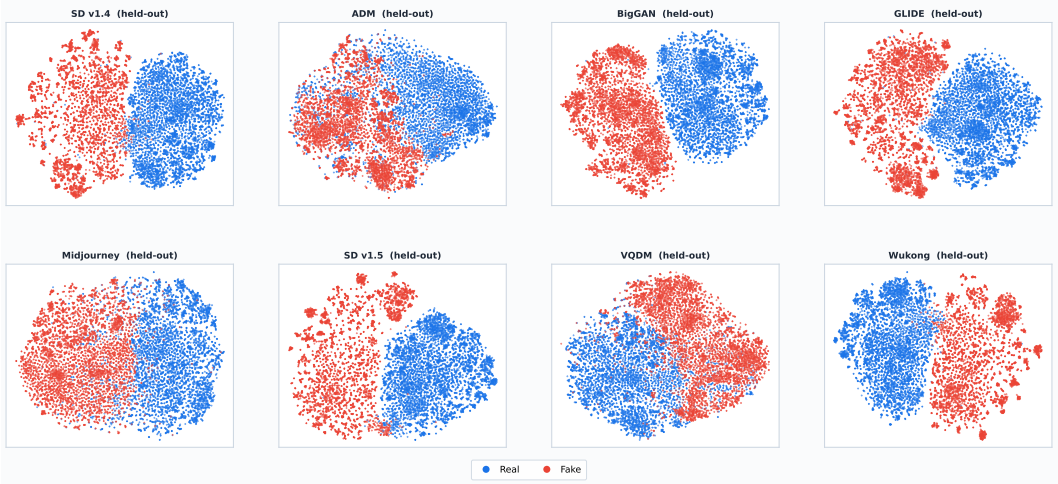


Figure 4: t-SNE visualization of real and fake feature distributions produced by DEDCA (w/o TTA). Each subplot corresponds to one held-out generator, where blue and red points denote real and fake samples, respectively.

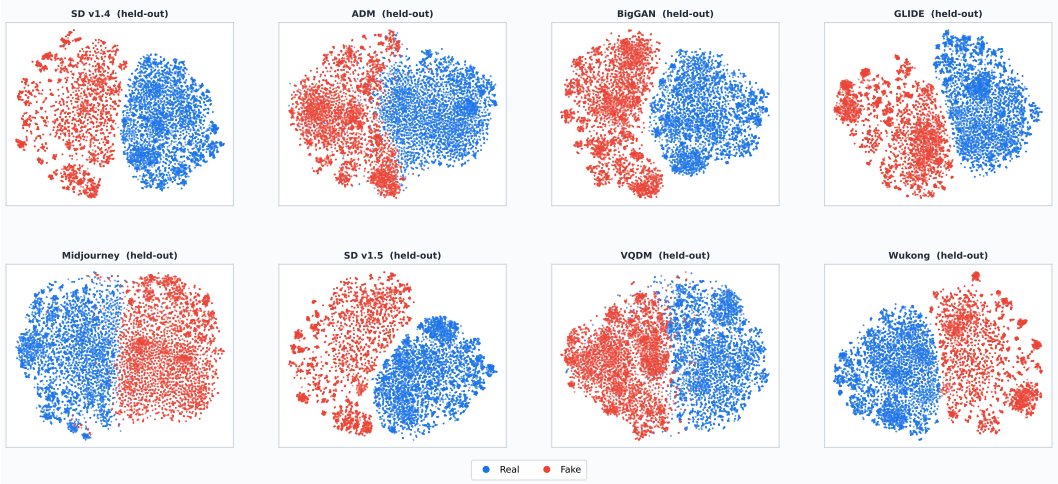


Figure 5: t-SNE visualization of real and fake feature distributions produced by DEDCA. Each subplot corresponds to one held-out generator, where blue and red points denote real and fake samples, respectively.

the average ACC from 91.4% to 92.0%, corresponding to a relative gain of about 0.7%. Compared with MLEP, DEDCA introduces a larger model and higher runtime due to its dual-evidence design, but improves the average ACC from 84.9% to 92.0%, giving a relative gain of about 8.4%. Compared with LARE^2, DEDCA has slightly fewer parameters, although its inference time is higher, and improves the average ACC from 84.1% to 92.0%, corresponding to a relative gain of about 9.4%. During test-time adaptation, DEDCA updates 540.3K parameters, including only a restricted subset of the model, while keeping the main feature extractors frozen. Although this number is larger than that of the compared TTA methods, its TTA time remains moderate at 4.54 ms and is lower than that of T2A. Compared with PeTTA, DEDCA only increases the TTA time from 3.63 ms to 4.54 ms, while improving the average ACC from 86.4% to 92.0%, giving a relative gain of about 6.5%. These results indicate that DEDCA's dual-evidence mechanism obtains stronger detection performance without excessively increasing the computational burden.

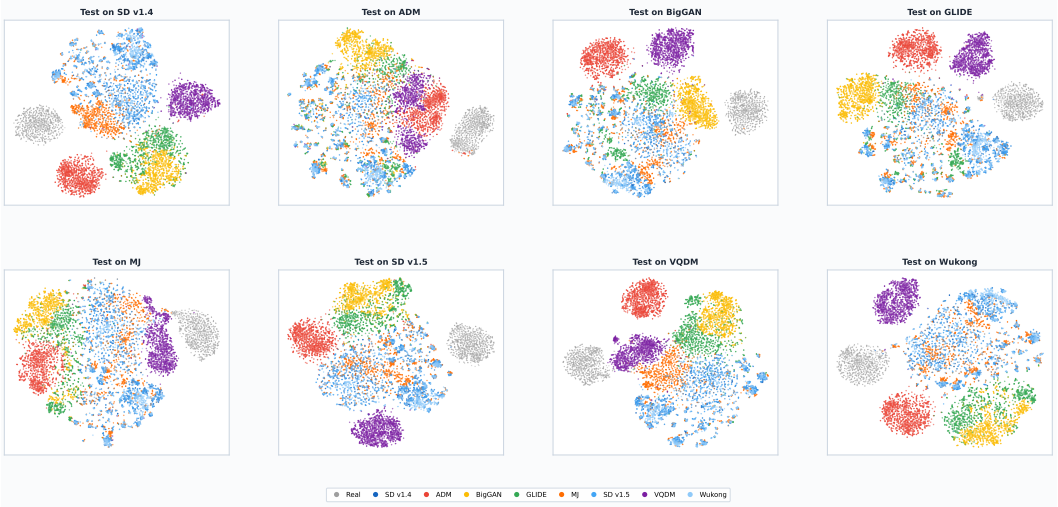


Figure 6: Multi-generator t-SNE visualization of the feature space produced by DEDCA without test-time adaptation.

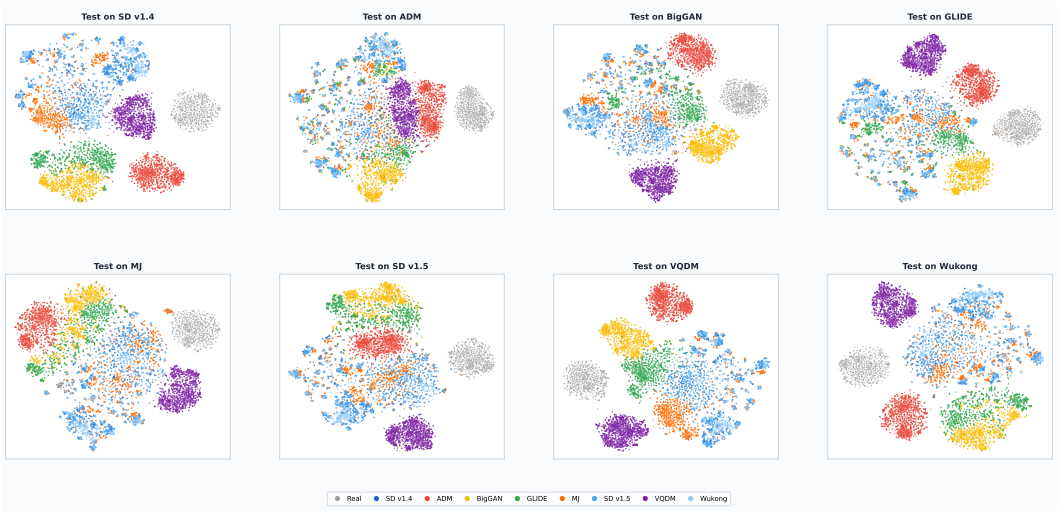


Figure 7: Multi-generator t-SNE visualization of the feature space produced by DEDCA with test-time adaptation.

E Visualization Analysis

To further evaluate the effectiveness of the DEDCA, we visualize the feature spaces of DEDCA (w/o TTA) and DEDCA using t-SNE. The feature distributions of real and fake samples from the held-out generator under different leave-one settings are shown in Figure 4 and Figure 5. It can be observed that the two categories form clear separations in the feature space, indicating that DEDCA can learn discriminative real-fake representations even when facing generators unseen during training. Compared with DEDCA (w/o TTA), the DEDCA further produces more compact intra-class distributions and clearer real-fake margins, showing that test-time adaptation improves the target-domain feature structure.

Furthermore, Figure 6 and Figure 7 visualize all eight generators in the same feature space for DEDCA (w/o TTA) and DEDCA. In both cases, real samples generally form a distinguishable cluster, while fake samples exhibit meaningful generator-related structures rather than collapsing into a single mixed distribution. ADM and VQDM tend to occupy relatively independent regions, and BigGAN and GLIDE also appear as recognizable groups. SD v1.4, SD v1.5, and Wukong are

naturally located in adjacent regions, since they share closely related diffusion-based generation characteristics. Midjourney is also close to this group, which may be attributed to its similar photorealistic rendering style and diffusion-like local trace patterns. After test-time adaptation, these generator-related structures become more organized while maintaining better separation from real samples. This observation suggests that DEDCA does not merely learn a binary real-fake boundary, but also implicitly preserves generator-family-level trace variations in the learned representation.

F Current Limitations and Future Work

While DEDCA demonstrates strong cross-generator generalization and effective unlabeled test-time adaptation, several directions remain worth further exploration. First, our experiments mainly follow the leave-one-generator protocol on GenImage, which provides a controlled evaluation of unseen-generator robustness. Future work could further validate the method in more open-world scenarios involving mixed generators, continuously updated synthesis pipelines, user editing, screenshots, and platform-specific image transformations. Further evaluation could also extend the current batch-transductive protocol to randomized target-stream orders and strictly prequential predict-then-adapt settings. Second, although SERT provides useful content-suppressed trace evidence, studying its behavior under more diverse post-processing and editing operations would offer a more comprehensive understanding of practical robustness. Third, DEDCA currently performs online adaptation with a restricted set of trainable parameters, and future work may explore richer reliability modeling, memory-assisted adaptation, and lightweight dual-evidence architectures to further improve long-term stability and deployment efficiency.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's main contributions, including the dual-evidence detector, reliability-aware fusion, disagreement-constrained test-time adaptation, and improved performance on GenImage. These claims are supported by the methodology and the experimental results reported in the main paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper discusses practical constraints through robustness and computational efficiency analyses, including the effects of post-processing operations, adaptation cost, and the limited subset of parameters updated during test-time adaptation. These analyses clarify the scope and deployment trade-offs of the proposed method. We also mention our limitations and future work in Appendix F.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present theoretical results, formal theorems, or proofs. The mathematical formulations are used to define the model architecture, training losses, and test-time adaptation objectives.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the dataset protocol, leave-one-generator split, evaluation metrics, preprocessing details, optimization settings, adaptation procedure, and all major hyperparameters needed to reproduce the main results in Section 4.1. The model components and losses are also described in sufficient detail in Section 3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The experiments are based on the publicly available GenImage benchmark, and the implementation details needed to reproduce the results are described in the paper. We will provide data, code, and instructions for the final paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not

including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the data splits, leave-one-generator evaluation protocol, unlabeled target adaptation setting, metrics, optimizer, learning rate, training epochs, adaptation steps, loss weights, and updated parameter subsets in Section 4.1. These details are sufficient to understand how the reported results are obtained.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The reported results are averaged over five independent runs to reduce randomness, but the current manuscript does not include error bars, confidence intervals, or formal significance tests. We omit them for space reasons and focus on consistent comparisons across generators, ablations, robustness settings, and efficiency analyses.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper states that experiments are conducted on a single NVIDIA RTX 3090 GPU with 24GB memory and reports model size, updated parameters, inference time, and adaptation time in Appendix D. These details provide the main compute requirements and test-time cost needed to reproduce and assess the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is conducted on existing image detection benchmarks and pretrained models, without human-subject experiments or unsafe data collection. The goal is to improve AI-generated image detection and visual forensics, which is consistent with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper motivates the positive societal impact of AI-generated image detection for addressing misinformation, impersonation, and media credibility. It also acknowledges that detector errors may affect real-world deployment reliability, motivating robustness evaluations under unseen generators and common post-processing operations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, scraped dataset, or other asset with substantial misuse risk. The proposed system is a detection method evaluated on existing benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits the existing datasets, pretrained models, and baseline methods used in the experiments through citations. The experiments rely on publicly available research assets, and their intended academic use is consistent with the cited resources and their terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces a new model implementation for DEDCA and provides documentation for its architecture, training procedure, test-time adaptation protocol, and evaluation setting. We will release our codebase along with detailed README files. The release does not include source code copied from external libraries; all code provided is our own.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects. Therefore, participant instructions, screenshots, and compensation details are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve human-subject research or crowdsourcing. As a result, IRB approval or equivalent institutional review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method does not use LLMs as an important, original, or non-standard component of the research. Any possible use of language models for writing, editing, or formatting would not affect the methodology, scientific rigor, or originality of the work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.