Less is More: Fewer Tokens and Blocks Make AIGI Detection Faster and More Generalizable

Haipeng Fang[1,2], Yu Li[1,2,✉], Yixing Lu[1,2], Qiang Sheng[1,2], Juan Cao[1,2], Sheng Tang[1,2]
[1] Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences

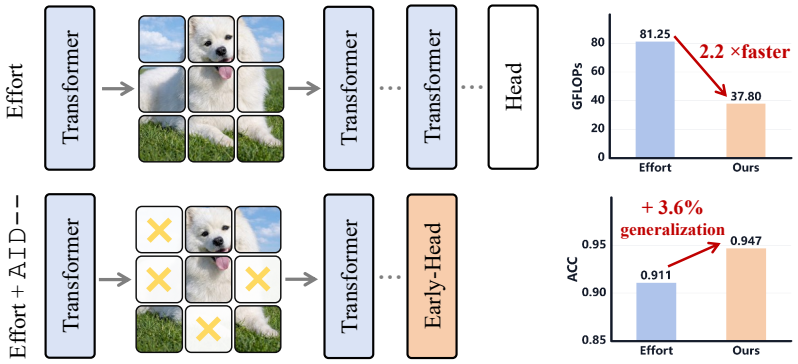

Figure 1: Illustration of AID-- plugged into Effort. *Left:* Effort processes all tokens through the full transformer blocks, while AID-- progressively prunes tokens and adaptively exits at earlier blocks. *Right:* AID-- achieves 2.2× faster inference and +3.6% better generalization on the Genimages.

Abstract

AI-Generated Image (AIGI) Detection is becoming critical as generative models proliferate, yet existing detectors often generalize poorly by relying on narrow, dataset-specific artifacts. Many prior works improve generalization through *additions*, such as introducing extra artifact cues, pretraining models, training constraints, or test-time operations, which may increase inference cost and model complexity. In this paper, we take the opposite route and study generalization through *subtractions* and propose AID--, a plug-and-play framework that improves transformer-based AIGI detectors from both *feature* and *parameter* perspectives. At the feature level, we introduce high-order dependency ranking to identify and prune redundant or potentially misleading tokens for progressive token reduction; and at the parameter level, we design relative-advantage head supervision and a class-asymmetric exit rule to train depth-specific early-exit heads. AID-- can be seamlessly integrated into transformer-based AIGI detectors to improve both efficiency and out-of-distribution generalizability, which is experimentally validated across six benchmarks. For example, on GenImages, integrating AID-- into Effort yields a 2.15× inference speedup and improves generalization accuracy by 3.6%.

1 Introduction

AI-generated images (AIGI) have become increasingly realistic due to the rapid progress in generative models, raising serious concerns about visual misinformation [63], copyright infringement [1], and synthetic identity fraud [22], necessitating AI-generated image detection that distinguishes AI images from non-AI ones in open visual environments. Despite strong in-distribution performance, however, typical detectors often generalize poorly to unseen generators, datasets, or synthesis pipelines. A key

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

reason is that they tend to rely on narrow, dataset-specific artifacts rather than stable and transferable detection cues. Improving out-of-distribution generalization thus remains a central challenge for reliable AIGI detection.

Existing studies mainly address this challenge through *additions*. They introduce extra cues, constraints, modules, or test-time operations to enrich detection signals, with the goal of reducing overfitting to dataset-specific artifacts. Typical examples include frequency-domain artifacts [19, 42, 50], local texture or pixel-level statistics [38, 51, 52], reconstruction or inversion signals [11, 39, 60], data augmentation or adversarial perturbations [31, 71], and stronger pretrained vision or vision-language backbones [12, 41, 72]. While effective, these additive designs often introduce extra computation, additional modules, or heavier inference procedures. More importantly, adding more signals does not necessarily remove dataset-specific shortcuts; it may instead introduce new non-transferable cues. As a result, current generalizable detectors still face a trade-off between generalization and efficiency.

In this paper, we take the opposite line and explore whether AIGI detectors can become both faster and more generalizable ***through subtractions***. Intuitively, the subtraction itself can lead to lower computation overhead and thus faster detection; thus, what to remove decides whether a detector will also be generalizable. Our preliminary experiments reveal that transformer-based AIGI detectors do not always benefit from using all patch tokens or all transformer blocks: Some samples can be reliably classified with fewer tokens or earlier exits, and proper subtraction leads to a clearer AI/non-AI classification boundary. **That is, by identifying and *subtracting* certain tokens or layers, it is possible to improve both generalization and efficiency simultaneously.**

Based on the above analyses, we propose `AID--`, a *subtraction-oriented* plug-and-play framework tailored for transformer-based AIGI detectors. `AID--` improves efficiency and generalization from two complementary perspectives. From the feature perspective, it not only removes generic feature redundancy but also suppresses misleading patch tokens via progressive token reduction, which is guided by high-order dependency ranking that captures both direct and indirect token relationships. From the parameter perspective, it introduces adaptive block early-exit, allowing each sample to stop at a reliable intermediate depth. To make early exits reliable, we further design relative-advantage head supervision for depth-specific exit heads and a class-asymmetric exit rule to avoid premature non-AI decisions. In summary, the contributions of this paper are as follows:

- **Perspective:** We propose a subtraction-based view for AIGI detection, showing that removing redundant or misleading computation can improve both efficiency and generalization.
- **Method:** We propose `AID--`, a tailored plug-and-play framework that improves transformer-based AIGI detectors through progressive token reduction and adaptive block early exit. Specifically, we design the high-order dependency ranking for token reduction and relative-advantage head supervision and a class-asymmetric exit rule for depth-adaptive prediction.
- **Performance:** Extensive experiments demonstrate that `AID--` improves both speed and generalization simultaneously. When applied to Effort, it achieves up to a $2.2\times$ speedup while improving generalization by 3.6% on the GenImages benchmark.

2 Related Works

2.1 AIGI Detection

Early AIGI detection studies mainly improve generalization by adding artifact cues or generation-process traces. Spatial-domain methods study common fingerprints left by CNN-based generators [57]; frequency-based methods expose traces through frequency-domain features or frequency-aware learning [19, 24, 42, 50]; texture-, pixel-statistics-, and gradient-based methods capture global texture distributions, local pixel relations, up-sampling artifacts, or model-induced gradient representations [5, 38, 51, 52]. Besides, methods concerning reconstruction and inversion compare the input with its diffusion-reconstructed version, analyze denoising or inversion consistency, and combine reconstruction errors with spectral or difference-based cues to reveal generation traces [11, 26, 39, 60]. Similar generalization challenges arise in manipulated-region detection [8, 21, 47–49].

Beyond artifact- and reconstruction-based designs, recent AIGI detectors increasingly rely on large-scale pretrained vision or vision-language models to improve cross-generator generalization. Feature-based methods use frozen pretrained representations like CLIP, and train only a nearest-neighbor

classifier, linear head, or lightweight detector on top of them [12, 41, 72]. Adaptation-based methods tailor pretrained features through orthogonal subspace adaptation or information bottleneck learning [67, 69]. Expert- or branch-based methods introduce multiple artifact experts, noise experts, spectral branches, or semantic-artifact decomposition modules to capture complementary detection cues [26, 66, 67]. Training-free or test-time methods use pretrained features with cropping robustness, image perturbations, or consistency-based scoring to avoid detector-specific training [9].

Prior AIGI detectors improve generalization by additive route, which often increases inference cost and may still tie detectors to generator-specific artifacts [12, 19, 41, 57, 60, 66, 67]. Different from them, we take a subtractive route by pruning redundant or misleading tokens and enabling adaptive early exits, which reduces computation while improving cross-generator generalization.

2.2 Transformer Acceleration

Transformer acceleration has been widely studied through efficient attention [10, 58], compact architecture design [36, 59], and model compression [25, 45]. This section focuses on two dynamic inference lines most related to our method: token reduction [3, 33, 43], and early exit [35, 61, 64, 70].

Token reduction. Token reduction accelerates Transformers by reducing the number of tokens processed in intermediate layers. Pruning-based methods like A-ViT [68], SPViT [30], and Cropr[2] progressively remove less informative tokens during inference. Merging- or pooling-based methods shorten the token sequence by combining redundant or similar tokens , as shown in Token Pooling [40], ToMe [3], and ToFu [29]. Reorganization-based methods keep important tokens while reorganizing or compressing less useful ones, including ATS[18], Evo-ViT [65], EViT [33], and FR^{2}ViT [17]. Hybrid and training-adaptive methods further combine different reduction operations or learn reduction ratios, such as DiffRate [6], and METR [34]. Related approaches also extend to diffusion model acceleration [4, 15, 16, 55]. These methods are mostly designed for general vision tasks, while AI-generated image detection relies on subtle and distributed artifacts; therefore, we introduce task-aware token pruning to remove redundant or misleading tokens while preserving cues useful for distinguishing non-AI and AI images.

Early exit. Early exit accelerates Transformers by reducing the number of executed layers. Networks like BranchyNet [53] and Shallow-Deep Networks [28] attach intermediate classifiers to early layers and stop inference for easy samples; methods like BERxiT [62] and COSEE[23] exit once an intermediate prediction signal or confidence is reliable enough; methods like PABEE[70] use consistent predictions across consecutive exits as the stopping signal; and adaptive-depth generation methods like CALM [46] and LayerSkip[14] dynamically allocate different numbers of layers during decoding. For vision Transformers, early-exit designs have been explored by attaching intermediate exits or designing task-specific exit heads, including LGViT [64], EnViT [13], and MET [37]. However, most early-exit methods still rely on a shared confidence threshold for all classes, which is not well matched with AI-generated image detection. Instead, we use class-asymmetric exit thresholds for non-AI and AI images to better control exit reliability.

3 Preliminaries and Motivations

Our method is motivated by a *subtraction-based* view. Instead of improving generalization through *additions*, we ask whether removing redundant or misleading computation can make a detector both faster and more generalizable. We study this question from two perspectives: the *feature* perspective, where patch tokens may contain redundant or misleading responses, and the *parameter* perspective, where late transformer blocks may introduce unnecessary computation or weaken detection cues.

Experiments were conducted based on Effort [67] with a 24-layer ViT-L backbone. We split into four stages, and we analyze two controlled settings: token reduction with different keeping ratios and early exit at different head positions. The evaluation set is constructed by sampling images from multiple generators across UniversalFakeDetection[41], GenImages[73], Chameleon[66], and AIGI-Now[7], with 1K samples selected from each generator. See more details in the supplementary material.

What happens when using fewer tokens or fewer blocks? *Observation 1: AIGI detectors contain clear feature- and parameter-level redundancy.* As shown in Fig. 2(a), reducing the token keeping ratio from 1.0 to 0.75, 0.5, and 0.25 still yields mAcc scores of 83.34, 84.83, 82.09, and 76.79, respectively. Similarly, Fig. 2(b) shows that exiting at Exit 6, Exit 12, and Exit 18 obtains mAcc of

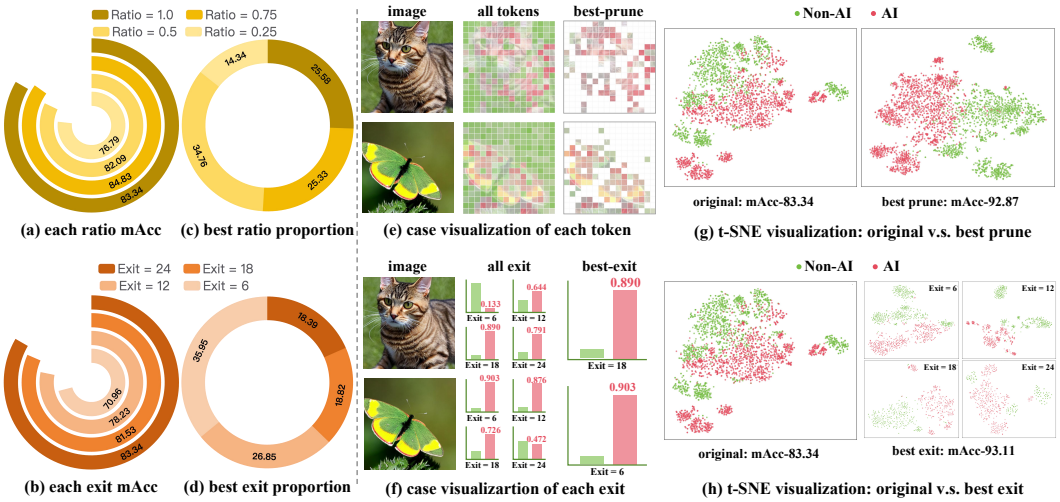

Figure 2: Preliminaries. *(a)-(d):* Token-reduction and early-exit analyses reveal feature- and parameter-level redundancy, with many samples favoring fewer tokens or earlier exits. *(e), (f):* Case visualizations show that redundant tokens and blocks can introduce weak or misleading responses. *(g), (h):* Oracle subtraction yields clearer AI/non-AI separation and higher mAcc, suggesting that removing redundant or misleading computation can improve both efficiency and generalization.

70.96, 78.23, and 81.53, compared with 83.34 at the final Exit 24. These results show that a large fraction of samples can be correctly classified without all tokens or all blocks. *Observation 2: Fewer tokens and layers can yield more reliable predictions sometimes.* We further define the best setting for each sample as the one that produces the most reliable prediction among different token keeping ratios or exit positions. As shown in Fig. 2(c), only 25.58% of samples prefer the full token set, while most achieve their best predictions with fewer tokens. Similarly, Fig. 2(d) shows that only 18.39% of samples prefer the final exit, while most samples are better classified by earlier exits. These results go beyond efficiency, suggesting that some tokens and blocks may weaken generalization.

Why can subtraction improve AIGI detection generalization? *Observation 3: Some patch tokens and late blocks are not merely redundant but misleading for AIGI detection.* As shown in Fig. 2(e), token-level visualization on two AI-generated images shows that many patch tokens produce weak or misleading responses toward the non-AI class, while best pruning keeps tokens concentrated on more discriminative regions. At the block level, Fig. 2(f) shows that different samples prefer different depths. For example, the cat image is best classified at Exit 18, while the butterfly image is best classified at Exit 6. These cases suggest that the benefit of subtraction comes from suppressing harmful local responses and avoiding unreliable late-stage predictions. *Observation 4: Oracle subtraction yields a clearer and more accurate AI/non-AI classification boundary.* We further construct two oracle variants: one selects the best token pruning ratio for each sample, and the other selects the best exit head for each sample. Although these oracle settings are not used during inference, they reveal the potential upper bound of subtracting harmful computation. As shown in Fig. 2(g) and (h), best pruning improves mAcc from 83.34 to 92.87, while best exit improves mAcc from 83.34 to 93.11, with clearer AI/non-AI separation in the t-SNE space.

4 Proposed Method: `AID--`

As illustrated in Fig. 3, `AID--`, a plug-and-play framework for improving the efficiency and generalization of transformer-based AIGI detectors. We divide the transformer network with L blocks into S stages, each containing $k = L/S$ consecutive blocks. At the end of each stage, `AID--` mitigates feature redundancy and generalization degradation from both the *feature* and *parameter* perspectives through progressive token reduction and adaptive block early-exit.

Specifically, adaptive block early-exit estimates whether a sample can be reliably classified at the current stage, while progressive token reduction removes less informative or potentially misleading tokens before subsequent stages. Notably, as suggested by Observations 1–4, this design not only

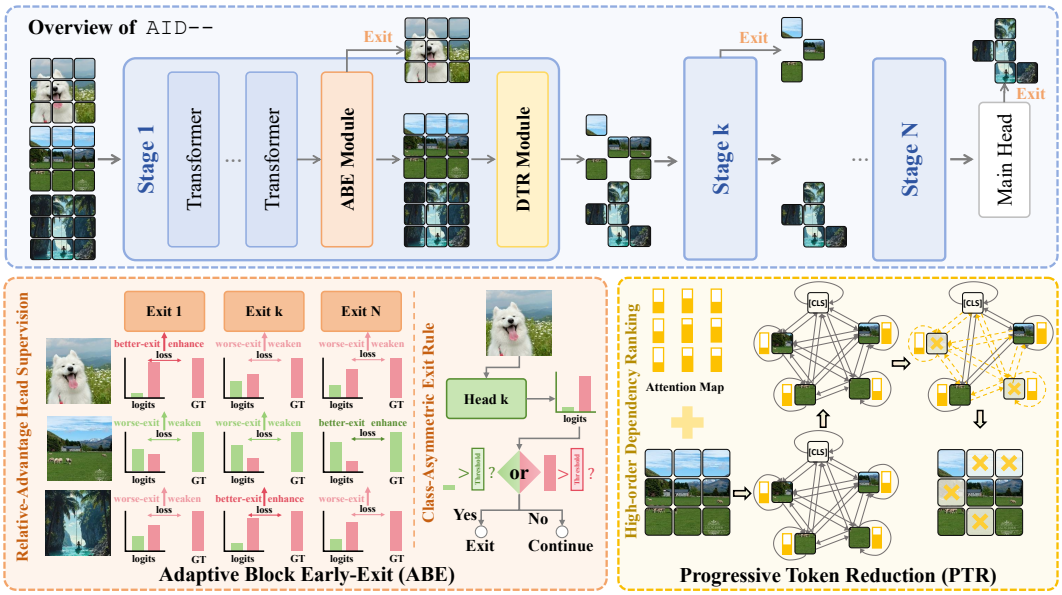

Figure 3: Overview. *Upper:* AID-- performs progressive token reduction and adaptive block early-exit at each stage, mitigating feature redundancy and generalization degradation from both the *feature* and *parameter* perspectives. *Lower left:* Adaptive Block Early-Exit uses relative-advantage head supervision for training and a class-asymmetric exit rule for inference. *Lower right:* Progressive Token Reduction constructs a graph for high-order dependency ranking and token pruning.

reduces token and block computation but also limits the influence of misleading tokens and overly semantic late-stage blocks, leading to more reliable decisions for AIGI detection.

4.1 Progressive Token Reduction

Following the token-level observations in Sec. 3, we progressively remove redundant or misleading patch tokens that may conflict with the image-level decision. Since the [CLS] token from large-scale semantic pre-training has been shown to provide a generalizable representation [41, 67], we adopt the widely used [CLS]-to-patch attention [18, 33, 65] as a training-free token ranking criterion.

However, directly applying this criterion is less reliable in our setting because token reduction is coupled with early-exit prediction. In shallow stages, feature representations are not yet sufficiently stable, and inaccurate token ranking may affect both subsequent feature propagation and early-exit reliability. To address this issue, we introduce High-order Dependency Ranking, which refines attention-based token ranking by modeling both direct and indirect token relationships.

High-order Dependency Ranking. We construct a fully connected, directed, weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where each node in $\mathcal{V}$ corresponds to a token, $\mathcal{E}$ denotes directed edges between tokens, and $\mathcal{A}$ serves as the edge-weight matrix derived from the attention map. Specifically, $\mathcal{A}[i, j]$ denotes the attention from token i to token j. Let c be the index of the [CLS] token.

The standard [CLS]-to-patch attention only measures the direct dependency from the [CLS] token to each patch token:

$$\mathcal{S}^{(1)}[j] = \mathcal{A}[c, j], \quad \mathcal{S}^{(1)} = \mathcal{A}[c, :]. \tag{1}$$

To capture indirect dependencies, we propagate the [CLS] attention over the token graph. The second-order score is computed as:

$$\mathcal{S}^{(2)}[j] = \sum_{u \in \mathcal{V}} \mathcal{A}[c, u]\mathcal{A}[u, j] = \mathcal{S}^{(1)} \cdot \mathcal{A}[:, j], \quad \mathcal{S}^{(2)} = \mathcal{S}^{(1)} \cdot \mathcal{A}. \tag{2}$$

More generally, the m-th order score is recursively defined as

$$\mathcal{S}^{(m)}[j] = \sum_{u \in \mathcal{V}} \mathcal{S}^{(m-1)}[u]\mathcal{A}[u, j] = \mathcal{S}^{(m-1)} \cdot \mathcal{A}[:, j], \quad \mathcal{S}^{(m)} = \mathcal{S}^{(1)}\mathcal{A}^{m-1}. \tag{3}$$

We aggregate direct and indirect dependencies up to the K-th order:

$$\tilde{\mathcal{S}} = \sum_{m=1}^{K} \alpha^{m-1} \mathcal{S}^{(m)} = \sum_{m=1}^{K} \alpha^{m-1} \mathcal{S}^{(1)} \mathcal{A}^{m-1}, \tag{4}$$

where α is a decay factor that reduces the effect of noisy high-order propagation. We set $\alpha = 0.7$ by default. The final pruning score of the j-th patch token is given by $\tilde{\mathcal{S}}[j]$. According to a predefined pruning ratio ρ, tokens with higher scores are retained, while the remaining tokens are removed before entering the next stage. The `[CLS]` token is always retained, and pruning is only applied to patch tokens. For the s-th stage, we set $K_s = S - s + 1$, using a larger propagation order in earlier stages where attention maps are usually less stable.

4.2 Adaptive Block Early-Exit

Motivated by the block-level observations in Sec. 3, deeper blocks do not always contribute positively to AIGI detection. For some samples, late-stage blocks may introduce redundant computation or become overly biased toward high-level semantic cues, making earlier exits more reliable than the final exit. We therefore introduce adaptive block early-exit to reduce unnecessary block computation. Each early-exit head follows the same structure as the original main head.

Standard early-exit strategies are not directly suitable for AIGI detection. During training, they often treat the final exit as a universal teacher, which conflicts with the exit-wise behavior in Observation 2. We therefore propose Relative-Advantage Head Supervision, which assigns sample-adaptive supervision to different exit heads. During inference, standard methods usually apply a shared confidence threshold across classes, while AIGI detection has class-asymmetric risks: unseen AI images are more likely to be misclassified as non-AI images. We thus design a Class-Asymmetric Exit Rule to improve early-exit reliability.

Relative-Advantage Head Supervision. To encourage depth-specific specialization, we adopt two-stage supervision. In the warm-up stage, all exits are trained with uniform ground-truth supervision:

$$\mathcal{L}_{\text{warm}} = \sum_{i=1}^{S} \mathcal{L}_{\text{CE}}(p_i(x), y). \tag{5}$$

After warm-up, we estimate the relative advantage of each exit according to its detached confidence on the ground-truth class:

$$q_i(x, y) = \frac{\text{sg}(p_i(y|x))}{\sum_{j=1}^{S} \text{sg}(p_j(y|x))}, \quad w_i(x, y) = 1 + q_i(x, y) - \frac{1}{S}, \tag{6}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operation. Here, $q_i(x, y)$ measures the sample-wise suitability of the i-th exit relative to other exits. The specialization loss is defined as

$$\mathcal{L}_{\text{spec}} = \sum_{i=1}^{S} w_i(x, y) \mathcal{L}_{\text{CE}}(p_i(x), y). \tag{7}$$

Compared with uniform supervision, exits with $q_i(x, y) > 1/S$ receive stronger supervision, while exits with $q_i(x, y) < 1/S$ receive weaker but still positive supervision. This encourages each exit to focus more on samples that are relatively separable at its own depth, rather than forcing shallow exits to mimic the final exit.

Class-Asymmetric Exit Rule. During inference, we use class-asymmetric thresholds to control early exiting. For the i-th exit, the predicted class and confidence are computed as

$$\hat{y}_i = \arg\max p_i(x), \quad c_i = \max p_i(x). \tag{8}$$

A sample exits at the i-th exit if

$$(\hat{y}_i = \text{non-AI} \ \wedge \ c_i \geq \theta_{\text{non-AI}}) \quad \text{or} \quad (\hat{y}_i = \text{AI} \ \wedge \ c_i \geq \theta_{\text{AI}}). \tag{9}$$

We set $(\theta_{\text{non-AI}}, \theta_{\text{AI}}) = (0.995, 0.90)$ across all backbones and datasets. The higher threshold for non-AI predictions makes early exits more conservative, reducing the risk of prematurely misclassifying unseen AI images as non-AI. In contrast, the lower threshold for AI predictions allows easy AI samples to exit earlier, improving efficiency while maintaining reliable detection.

Table 1: Quantitative comparison on the UniversalFakeDetect benchmark.

Method	GFLOPs	FPS	mAcc	GAN	Deepfakes	Low level	Perceptual loss	Guided	LDM	Glide	Dalle
DeiT-S	4.6 (1.00×)	1263.7 (1.00×)	0.719	**0.890**	0.703	0.570	0.942	0.555	0.560	0.590	0.585
+ Cropr	2.5 (1.85×)	2206.7 (1.75×)	0.714	0.867	**0.769**	**0.590**	0.922	0.555	0.577	0.555	0.620
+ LGViT	3.3 (1.41×)	1895.6 (1.50×)	0.718	0.887	0.695	0.573	**0.945**	0.520	0.567	0.578	**0.635**
+ Cropr + LGViT	2.4 (1.92×)	2274.7 (1.80×)	0.710	0.880	0.601	0.585	0.896	0.575	**0.588**	0.572	0.595
+ AID--	2.3 (1.99×)	2296.8 (1.82×)	**0.727**	**0.890**	0.689	0.588	0.935	**0.590**	**0.588**	**0.595**	0.600
UniFD	81.2 (1.00×)	88.2 (1.00×)	0.814	**0.953**	0.666	0.603	0.657	0.700	0.875	**0.790**	0.868
+ Cropr	40.8 (1.99×)	173.8 (1.97×)	0.802	0.941	0.679	0.610	0.665	0.690	0.850	0.768	0.820
+ LGViT	53.4 (1.52×)	132.1 (1.50×)	0.795	0.945	0.625	0.580	0.640	0.680	0.853	0.760	0.857
+ Cropr + LGViT	34.9 (2.33×)	191.9 (2.18×)	0.797	0.935	0.640	0.590	0.645	0.690	0.857	0.770	0.857
+ AID--	33.9 (2.39×)	210.2 (2.38×)	**0.849**	0.950	**0.753**	**0.613**	**0.831**	**0.795**	**0.895**	0.788	**0.950**
Effort	81.2 (1.00×)	88.0 (1.00×)	0.952	**0.988**	0.878	0.867	**0.985**	0.715	**0.988**	0.973	0.970
+ Cropr	40.8 (1.99×)	175.0 (1.99×)	0.922	0.977	0.825	0.830	0.950	0.680	0.963	0.943	0.865
+ LGViT	53.4 (1.52×)	132.1 (1.50×)	0.909	0.973	0.790	0.810	0.940	0.660	0.950	0.930	0.848
+ Cropr + LGViT	34.9 (2.33×)	191.9 (2.18×)	0.896	0.963	0.750	0.790	0.920	0.640	0.937	0.913	0.888
+ AID--	39.1 (2.08×)	183.2 (2.08×)	**0.953**	0.984	**0.902**	**0.912**	0.918	**0.780**	0.978	**0.980**	**0.980**

Table 2: Quantitative comparison with baseline methods on the GenImages benchmark.

Method	GFLOPs	FPS	mAcc	ADM	BigGAN	GLIDE	Midjourney	SD-v1.4	SD-v1.5	VQDM	Wukong
DeiT-S	4.6 (1.00×)	1210.1 (1.00×)	0.716	0.498	**0.535**	0.581	0.556	0.999	0.998	0.569	0.989
+ Cropr	2.5 (1.84×)	2070.1 (1.71×)	0.721	0.516	0.501	0.601	0.607	0.998	0.998	0.559	0.986
+ LGViT	4.1 (1.12×)	1353.4 (1.12×)	0.720	0.515	0.501	0.580	0.607	**1.000**	**0.999**	0.565	**0.996**
+ Cropr + LGViT	2.4 (1.92×)	1975.3 (1.63×)	0.676	0.507	0.501	0.547	0.533	0.952	0.944	0.527	0.898
+ AID--	2.3 (2.00×)	2107.5 (1.74×)	**0.735**	**0.536**	0.489	**0.646**	**0.684**	0.979	0.979	**0.596**	0.971
UniFD	81.2 (1.00×)	88.1 (1.00×)	0.733	0.552	**0.803**	**0.769**	0.732	0.842	0.840	0.569	0.756
+ Cropr	40.8 (1.99×)	169.4 (1.92×)	0.711	0.558	0.603	0.751	0.742	0.839	0.842	**0.587**	0.768
+ LGViT	45.9 (1.77×)	151.5 (1.72×)	0.688	0.540	0.498	0.702	0.688	0.882	0.877	0.544	0.772
+ Cropr + LGViT	33.4 (2.43×)	213.2 (2.42×)	0.685	**0.561**	0.494	0.648	0.672	0.878	0.876	0.577	0.777
+ AID--	33.8 (2.40×)	207.2 (2.35×)	**0.745**	0.537	0.753	0.718	**0.747**	**0.916**	**0.907**	0.584	**0.802**
Effort	81.2 (1.00×)	87.4 (1.00×)	0.911	0.787	0.776	0.933	0.824	0.998	0.998	0.917	0.974
+ Cropr	40.8 (1.99×)	173.1 (1.98×)	0.918	0.766	0.855	0.938	0.920	0.974	0.978	0.940	0.973
+ LGViT	53.4 (1.52×)	132.1 (1.51×)	0.911	0.712	0.818	0.935	0.903	**1.000**	**0.999**	0.926	0.992
+ Cropr + LGViT	34.9 (2.33×)	191.9 (2.20×)	0.912	0.763	0.772	0.889	0.908	0.998	**0.999**	**0.968**	**0.997**
+ AID--	37.8 (2.15×)	188.0 (2.15×)	**0.947**	**0.801**	**0.922**	**0.974**	**0.941**	0.996	0.996	0.954	0.990

5 Experiments

5.1 Experimental Settings

We evaluate AID-- on four transformer-based AIGI detectors: DeiT-S/16 [54], UniFD [41], Effort [67], and Simplicity Prevails [72]. We follow the original training protocol of each detector and compare AID-- with representative acceleration baselines, including Cropr [2], LGViT [64], and their combination. Experiments are conducted on UniversalFakeDetect [41], GenImages [73], three in-the-wild datasets [32, 66], and Treasure-10, our subset covering ten recent closed-source generators in Treasure [56]. Following common protocols, detectors are trained on ProGAN [27] for UniversalFakeDetect and on SD-v1.4 [44] for other benchmarks. We report mAcc for generalization and GFLOPs, FPS for efficiency. More implementation details and additional acceleration baselines are provided in Appendix A.2.

5.2 Performance Comparisons

Comparison with baselines. We compare AID-- with Cropr and LGViT on UniversalFakeDetect, GenImages, and three in-the-wild datasets. As shown in Table 1, Table 2, and Table 3, Cropr and LGViT can reduce computation, but their generalization is often unstable. In contrast, AID-- achieves a more reliable accuracy-efficiency trade-off across different detectors and benchmarks. For example, on GenImages, Effort+AID-- improves mAcc from 0.911 to 0.947 while reducing GFLOPs by 2.15× and increasing FPS by 2.15×. These results show that AID-- removes redundant computation while preserving useful detection cues.

Trade-off comparison with state-of-the-art methods. Table 4 compares AID-- with state-of-the-art AIGI detectors on Chameleon under different computation budgets. Across all groups, AID-- achieves the best mAcc within its group. In the low-cost group, DeiT-S + AID-- obtains the highest mAcc with the third-lowest GFLOPs. In the medium- and high-cost groups, AID-- achieve both

Table 3: Quantitative comparison on three in-the-wild datasets.

Method	GFLOPs	FPS	mAcc	Chameleon		CommunityAI		SocialRF	
				Non-AI	AI	Non-AI	AI	Non-AI	AI
UniFD	81.2 (1.00×)	88.3 (1.00×)	0.567	0.414	**0.704**	0.474	**0.740**	0.662	0.410
+ Cropr	40.8 (1.99×)	157.0 (1.78×)	0.505	0.404	0.613	0.435	0.528	0.579	0.470
+ LGViT	63.1 (1.29×)	112.3 (1.27×)	0.573	0.636	0.551	0.636	0.489	0.806	0.318
+ Cropr + LGViT	34.0 (2.39×)	206.4 (2.34×)	0.555	**0.844**	0.298	**0.857**	0.263	**0.887**	0.178
+ AID--	39.2 (2.07×)	150.8 (1.71×)	**0.575**	0.454	0.669	0.478	0.689	0.678	**0.480**
Effort	81.3 (1.00×)	88.5 (1.00×)	0.587	0.936	0.328	0.924	0.371	0.670	0.292
+ Cropr	40.8 (1.99×)	168.5 (1.90×)	0.573	0.882	0.347	0.883	0.348	0.571	**0.408**
+ LGViT	53.4 (1.52×)	134.2 (1.52×)	0.588	0.937	0.328	0.925	0.371	0.678	0.291
+ Cropr + LGViT	35.3 (2.30×)	199.7 (2.26×)	0.536	**0.998**	0.034	**0.995**	0.044	**0.995**	0.148
+ AID--	39.2 (2.07×)	179.7 (2.03×)	**0.601**	0.856	**0.527**	0.833	**0.453**	0.572	0.363
SP	1362.7 (1.00×)	53.5 (1.00×)	0.935	0.933	**0.895**	**0.949**	0.946	0.937	**0.948**
+ Cropr	830.9 (1.64×)	86.1 (1.61×)	0.921	0.918	0.878	0.930	0.934	0.925	0.941
+ LGViT	896.5 (1.52×)	79.7 (1.49×)	0.905	0.902	0.867	0.916	0.910	0.903	0.933
+ Cropr + LGViT	744.6 (1.83×)	91.5 (1.71×)	0.903	0.908	0.859	0.912	0.904	0.900	0.934
+ AID--	748.7 (1.82×)	91.5 (1.71×)	**0.942**	**0.964**	0.891	0.943	**0.957**	**0.967**	0.928

Table 4: Trade-off comparisons.

Method	GFLOPs	mAcc	Chameleon	
			Non-AI	AI
NPR	1.7	0.523	**0.999**	0.046
FreqNet	1.9	0.538	0.985	0.090
DeiT-S + AID--	2.3	**0.575**	0.913	**0.237**
CNNSpot	4.1	0.554	0.979	0.128
DeiT-S	4.6	0.531	0.993	0.068
Gram-Net	11.5	0.518	0.992	0.044
SAFE	15.4	0.520	0.993	0.046
LaDeDa	15.9	0.504	0.994	0.015
Effort + AID--	39.3	**0.692**	0.856	0.527
AIDE	66.3	0.574	**0.944**	0.203
UniFD	81.2	0.559	0.414	**0.704**
OMAT	81.2	0.629	0.899	0.359
Effort	81.3	0.632	0.936	0.328
SP + AID--	747.9	**0.927**	**0.964**	0.891
SP	1362.7	0.914	0.933	**0.895**

Table 5: Treasure-10.

Method	GFLOPs	mAcc
Effort	81.3 (1.00×)	0.6942
+ AID--	40.9 (1.99×)	**0.7113**
SP	1362.7 (1.00×)	0.8092
+ AID--	757.1 (1.80×)	**0.8363**

Table 6: Robustness comparison to common perturbations.

Method	GFLOPs	Original	JPEG Compression				Gaussian Blur			
			95	85	75	65	0.5	1.0	1.5	2.0
Effort	81.3 (1.00×)	0.632	0.631	0.630	0.623	0.612	0.628	0.619	0.607	0.594
+ AID--	39.3 (2.07×)	**0.692**	**0.693**	**0.688**	**0.681**	**0.670**	**0.690**	**0.681**	**0.668**	**0.654**
SP	1362.7 (1.00×)	0.914	0.918	0.914	**0.916**	0.891	0.916	0.909	0.897	0.891
+ AID--	748.1 (1.82×)	**0.927**	**0.931**	**0.925**	0.913	**0.903**	**0.928**	**0.922**	**0.908**	**0.899**

the lowest GFLOPs and the highest mAcc. These results show that the proposed subtraction-based design provides a strong accuracy-efficiency trade-off across model scales.

Performance on recent generators. Table 5 reports Treasure-10 results, with per-generator results in Appendix Table 10. AID-- improves Effort's mAcc from 0.6942 to 0.7113 while reducing GFLOPs from 81.3 to 40.9 (1.99×). For SP, the five-seed mean mAcc increases from 0.8092 to 0.8363, while GFLOPs decrease from 1362.7 to 757.1 (1.80×). These results demonstrate improved generalization with less computation even on ten recent closed-source generators in Treasure.

Robustness to Common Perturbations. Table 6 reports robustness under JPEG compression and Gaussian blur. Under the strongest JPEG compression and Gaussian blur, AID-- improves the average accuracy by 5.9% for Effort and 1.0% for SP. These results show that removing redundant computation does not compromise robustness but preserves reliable cues under common post-processing.

5.3 Ablation and Analysis

We conduct ablation studies with Effort on GenImages, covering the main components and their key design choices. More ablation results and analyses are provided in the Appendix C.

Main Ablation. As shown in Table 7, both progressive token reduction (PTR) and adaptive block early-exit (ABE) improve accuracy while reducing computation, showing the effectiveness of feature- and parameter-level subtraction. PTR achieves a 1.98× GFLOPs reduction and improves mAcc by 2.3%, while ABE achieves a 1.49× GFLOPs reduction and improves mAcc by 2.6%. Combining them further reduces GFLOPs by 2.15× and improves mAcc by 3.6%, confirming that PTR and ABE are complementary and jointly provide the best accuracy-efficiency trade-off.

Table 7: Ablation of main components.

Variant	GFLOPs	FPS	mAcc
Effort	81.2 (1.00×)	87.4 (1.00×)	0.911
+ PTR	41.0 (1.98×)	171.5 (1.96×)	0.934
+ ABE	54.6 (1.49×)	130.9 (1.50×)	0.937
+ PTR & ABE	37.8 (2.15×)	188.0 (2.15×)	**0.947**

Effect of high-order dependency ranking. Fig. 4 analyzes progressive token reduction and high-order dependency ranking (HODR). Moderate token reduction improves mAcc, and HODR brings further gains by making token selection more reliable. With HODR, the best mAcc is achieved at a lower keeping ratio than vanilla token reduction, together with an additional 1.1% improvement. This suggests that HODR allows more aggressive pruning while preserving useful detection cues. The visualizations further show that HODR removes more tokens that conflict with the prediction target for both non-AI and AI images. This supports Observation 3 in Sec. 3: when redundant or misleading tokens are better identified, subtraction can improve detection accuracy rather than hurt it.

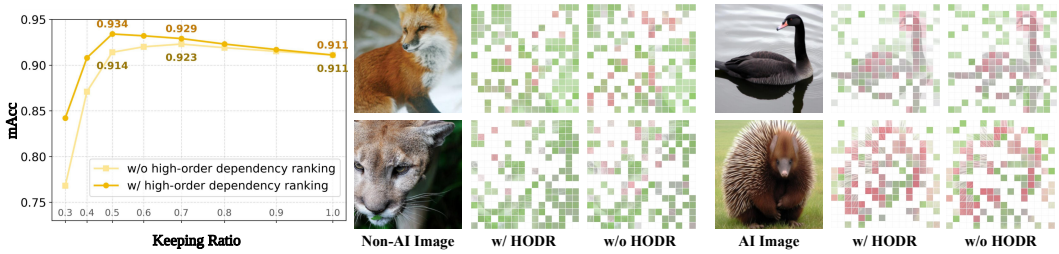

Figure 4: Effect of keeping ratio and high-order dependency ranking in progressive token reduction.

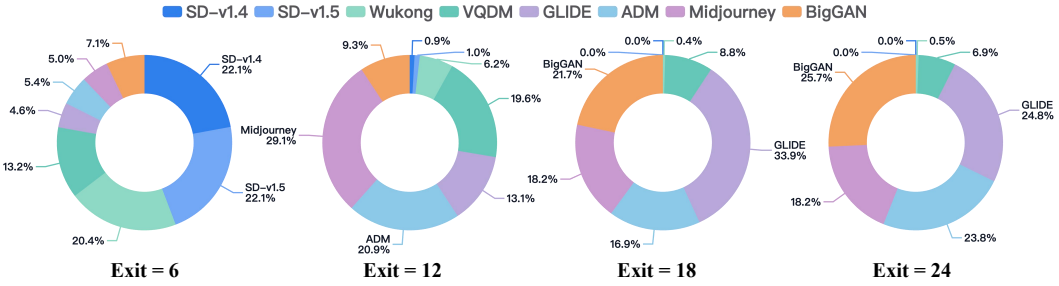

Figure 5: Proportion of pre-exit samples. Colors indicate architectural similarity to SD-v1.4, with the top-2 generators marked. RAHS encourages more specialized decision boundaries across exits.

Table 8: Variants of RAHS.

Variant	GFLOPs	FPS	mAcc
CE	57.2	135.4	0.928
CE + KD	55.7	134.1	0.884
RAHS	54.6	130.9	**0.937**

Effect of relative-advantage head supervision. Table 8 compares different supervision strategies for exit heads. The CE+KD variant leads to worse generalization, suggesting that directly aligning early heads with the final head can weaken their depth-specific roles. In contrast, relative-advantage head supervision (RAHS) achieves the best accuracy-efficiency trade-off by encouraging a clearer division of labor among the four heads. Each head is guided to focus on samples it can handle reliably, leading to more specialized, simpler, and more precise decision boundaries. Fig. 5 further supports this behavior: samples close to SD-v1.4 tend to exit earlier, while samples from more different generators are passed to deeper exits.

Effect of class-asymmetric exit rule. Fig. 6 analyzes the effect of class-asymmetric early-exit thresholds. These thresholds control intermediate exits rather than final classification decisions. With the non-AI threshold fixed to 0.995, varying the AI threshold has only minor effects on both recalls, showing that AI exits are relatively stable. In contrast, with the AI threshold fixed to 0.90, increasing the non-AI threshold brings clear recall gains in the high-threshold region. This suggests that a stricter non-AI threshold helps avoid unreliable shallow exits, especially when unseen AI images are over-confidently predicted as non-AI. We therefore set the non-AI and AI exit thresholds to 0.995 and 0.90, respectively.

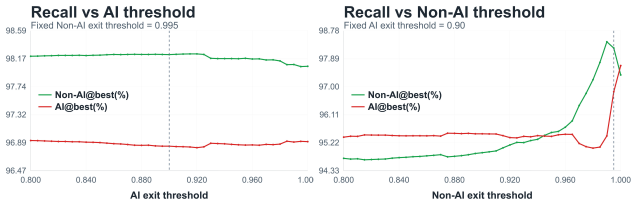

Figure 6: Effect of class-asymmetric exit rule.

6 Conclusion

In this paper, we empirically showed that transformer-based AIGI detectors can be improved from both generalization and efficiency simultaneously. Based on the important observation that not all tokens and not all layers are helpful for detection, we proposed to do subtraction, *i.e.,* removing useless and misleading tokens and allowing early exits from intermediate layers, for faster and more generalizable AIGI detection. Specifically, we designed `AID--`, a tailored plug-and-play framework that improves transformer-based AIGI detectors through progressive token reduction and adaptive block early exit. Experiments on six benchmarks validated the effectiveness and efficiency of `AID--`.

Acknowledgments

This work was supported in part by the Innovation Funding of the Institute of Computing Technology, Chinese Academy of Sciences under Grant No. E561090, and in part by the National Natural Science Foundation of China under Grant No. 62406310.

References

[1] Archer Amon, Zhipeng Yin, Zichong Wang, Avash Palikhe, Tongjia Yu, and Wenbin Zhang. Uncertain boundaries: Multidisciplinary approaches to copyright issues in generative ai. *SIGKDD Explorations Newsletter*, 27:1–12, 2025.

[2] Benjamin Bergner, Christoph Lippert, and Aravindh Mahendran. Token cropr: Faster vits for quite a few tasks. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9740–9750. IEEE, 2025.

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[4] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 4599–4603. IEEE, 2023.

[5] Bar Cavia, Eliahu Horwitz, Tal Reiss, and Yedid Hoshen. Real-time deepfake detection in the real-world. *arXiv preprint arXiv:2406.09398*, 2024.

[6] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate: Differentiable compression rate for efficient vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17164–17174, 2023.

[7] Ruoxin Chen, Jiahui Gao, Kaiqing Lin, Keyue Zhang, Yandan Zhao, Isabel Guan, Taiping Yao, and Shouhong Ding. Task-model alignment: A simple path to generalizable ai-generated image detection. *arXiv preprint arXiv:2512.06746*, 2025.

[8] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14165–14173. IEEE, 2021.

[9] Sungik Choi, Hankook Lee, and Moontae Lee. Training-free detection of ai-generated images via cropping robustness. *arXiv preprint arXiv:2511.14030*, 2025.

[10] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[11] Beilin Chu, Xuan Xu, Xin Wang, Yufei Zhang, Weike You, and Linna Zhou. Fire: Robust detection of diffusion-generated images via frequency-guided reconstruction error. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12830–12839. IEEE, 2025.

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

[13] Yonghao Dong, Qiang He, Penghong Rui, Zhenzhe Zheng, Zhao Li, Feifei Chen, Hai Jin, and Yun Yang. Envit: Enhancing the performance of early-exit vision transformers via exit-aware structured dropout-enabled self-distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 20852–20860, 2026.

[14] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642, 2024.

[15] Haipeng Fang, Yu Li, Fan Tang, Yixing Lu, Juan Cao, and Sheng Tang. Resca: Residual caching for diffusion transformers acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 32957–32966, 2026.

[16] Haipeng Fang, Sheng Tang, Juan Cao, Enshuo Zhang, Fan Tang, and Tong-Yee Lee. Attend to not attended: Structure-then-detail token merging for post-training dit acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18083–18092, 2025.

[17] Haipeng Fang, Ziheng Wu, Xinyi Zou, Jun Huang, Juan Cao, and Sheng Tang. FR^{2}ViT: Finetuning-free token reduction for dense prediction through a refinement-reactivation architecture. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE, 2025.

[18] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European conference on computer vision*, pages 396–414. Springer, 2022.

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

[20] Cheng Gong, Yao Chen, Qiuyang Luo, Ye Lu, Tao Li, Yuzhi Zhang, Yufei Sun, and Le Zhang. Deep feature surgery: Towards accurate and efficient multi-exit networks. In *European Conference on Computer Vision*, pages 435–451. Springer, 2024.

[21] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20606–20615. IEEE, 2023.

[22] Nikhil Gupta. Security risks of generative ai in financial systems: A comprehensive review. *World Journal of Information Systems*, 1(3):17–24, 2024.

[23] Jianing He, Qi Zhang, Hongyun Zhang, Xuanjing Huang, Usman Naseem, and Duoqian Miao. Cosee: consistency-oriented signal-based early exiting via calibrated sample weighting mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24023–24031, 2025.

[24] Yonghyun Jeong, Doyeon Kim, Seungjai Min, Seongho Joe, Youngjune Gwon, and Jongwon Choi. Bihpf: Bilateral high-pass filters for robust deepfake detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 48–57, 2022.

[25] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4163–4174, 2020.

[26] Dimitrios Karageorgiou, Symeon Papadopoulos, Ioannis Kompatsiaris, and Efstratios Gavves. Any-resolution ai-generated image detection by spectral learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18706–18717. IEEE, 2025.

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

[28] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International conference on machine learning*, pages 3301–3310. PMLR, 2019.

[29] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024.

[30] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, Minghai Qin, and Yanzhi Wang. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European conference on computer vision*, pages 620–640. Springer, 2022.

[31] Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Fuli Feng. Improving synthetic image detection towards generalization: An image transformation perspective. In *Proceedings of the 31st ACM SIGKDD conference on knowledge discovery and data mining v. 1*, pages 2405–2414, 2025.

[32] Ziqiang Li, Jiazhen Yan, Ziwen He, Kai Zeng, Weiwei Jiang, Lizhi Xiong, and Zhangjie Fu. Is artificial intelligence generated image detection a solved problem? In *Advances in Neural Information Processing Systems*, 2025.

[33] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Evit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022.

[34] Dongyang Liu, Meina Kan, Shiguang Shan, and Xilin Chen. A simple romance between multi-exit vision transformer and token reduction. In *The Twelfth International Conference on Learning Representations*, 2024.

[35] Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. Fastbert: a self-distilling bert with adaptive inference time. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6035–6044, 2020.

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[37] Zheng Liu, Jinchao Zhu, Nannan Li, and Gao Huang. Multiple-exit tuning: Towards inference-efficient adaptation for vision transformer. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(2):2123–2136, 2026.

[38] Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8060–8069, 2020.

[39] Ruipeng Ma, Jinhao Duan, Fei Kong, Xiaoshuang Shi, and Kaidi Xu. Exposing the fake: Effective diffusion-generated images detection. *arXiv preprint arXiv:2307.06272*, 2023.

[40] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers. *arXiv preprint arXiv:2110.03860*, 2021.

[41] 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.

[42] 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.

[43] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

[44] 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.

[45] Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in neural information processing systems*, 33:20378–20389, 2020.

[46] Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

[47] Zhihao Sun, Haipeng Fang, Juan Cao, Xinying Zhao, and Danding Wang. Rethinking image editing detection in the era of generative ai revolution. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3538–3547, 2024.

[48] Zhihao Sun, Haoran Jiang, Haoran Chen, Yixin Cao, Xipeng Qiu, Zuxuan Wu, and Yu-Gang Jiang. Forgerysleuth: Empowering multimodal large language models for image manipulation detection. *Advances in Neural Information Processing Systems*, 38:153136–153175, 2025.

[49] Zhihao Sun, Haoran Jiang, Danding Wang, Xirong Li, and Juan Cao. Safl-net: Semantic-agnostic feature learning network with auxiliary plugins for image manipulation detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22367–22376. IEEE, 2023.

[50] 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.

[51] 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.

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

[53] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE, 2016.

[54] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[55] Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16080–16089. IEEE, 2024.

[56] Jiaan Wang, Sirui Liu, Yu Li, Kaiyuan Yang, Juan Cao, and Sheng Tang. Fleet: Few shots lead effective AI-generated image detection. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, pages 128302–128326. PMLR, 2026.

[57] 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.

[58] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[59] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[60] 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.

[61] Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. Deebert: Dynamic early exiting for accelerating bert inference. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 2246–2251, 2020.

[62] Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. Berxit: Early exiting for bert with better fine-tuning and extension to regression. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main Volume*, pages 91–104, 2021.

[63] Danni Xu, Shaojing Fan, and Mohan Kankanhalli. Combating misinformation in the era of generative ai models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9291–9298. Association for Computing Machinery, 2023.

[64] Guanyu Xu, Jiawei Hao, Li Shen, Han Hu, Yong Luo, Hui Lin, and Jialie Shen. Lgvit: Dynamic early exiting for accelerating vision transformer. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9103–9114, 2023.

[65] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2964–2972, 2022.

[66] Shilin Yan, Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Weidi Xie. A sanity check for ai-generated image detection. *arXiv preprint arXiv:2406.19435*, 2024.

[67] Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable ai-generated image detection. In *International Conference on Machine Learning*, pages 70268–70288. PMLR, 2025.

[68] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10809–10818, 2022.

[69] Haifeng Zhang, Qinghui He, Xiuli Bi, Weisheng Li, Bo Liu, and Bin Xiao. Towards universal ai-generated image detection by variational information bottleneck network. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23828–23837. IEEE, 2025.

[70] Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341, 2020.

[71] Yue Zhou, Xinan He, Kaiqing Lin, Bing Fan, Feng Ding, and Bin Li. Breaking latent prior bias in detectors for generalizable aigc image detection. In *Advances in Neural Information Processing Systems*, volume 38, pages 30649–30679, 2025.

[72] Yue Zhou, Xinan He, Kaiqing Lin, Bing Fan, Feng Ding, and Bin Li. Simplicity prevails: The emergence of generalizable aigi detection in visual foundation models. *arXiv preprint arXiv:2602.01738*, 2026.

[73] 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 Additional Implementation Details

A.1 Details for Preliminaries and Motivations

This section provides additional details for the preliminary analyses in Sec. 3. All analyses are conducted with Effort [67] using a 24-layer ViT-L backbone. We divide the backbone into four stages, each containing six transformer blocks, and analyze subtraction from two views: token reduction at the feature level and early exit at the block level.

Evaluation set construction. The evaluation set is constructed from multiple generators across UniversalFakeDetection[41], GenImages[73], Chameleon[66], and AIGI-Now[7]. For each generator, we sample up to 1K AI images and 1K non-AI images. If either class contains fewer than 1K images, we use the smaller class size for both classes to form a balanced subset.

Controlled token-reduction setting. For the token-reduction analysis, we train a single 24-layer Effort model that supports four keeping ratios, i.e., 1.0, 0.75, 0.5, and 0.25. Token pruning is applied after each stage with the same keeping ratio. During training, each mini-batch randomly samples one ratio from the candidate set, allowing one model to be evaluated under different pruning budgets. Patch tokens are ranked by the standard CLS-to-patch attention score. At inference time, we fix the keeping ratio and report the corresponding mAcc in Fig. 2(a).

We also compute a sample-wise best ratio for Fig. 2(c). For each sample, we evaluate all candidate ratios and select the one that is most consistent with the ground-truth label. Among correct predictions, we choose the ratio with the highest confidence on the ground-truth class; any correct prediction is preferred over an incorrect one; if all ratios are incorrect, we choose the ratio with the lowest confidence on the wrong class. Fig. 2(c) reports the proportion of samples assigned to each best ratio.

Controlled early-exit setting. For the early-exit analysis, we use the 24-layer Effort and attach a linear exit head after each stage, resulting in Exit 6, Exit 12, Exit 18, and Exit 24. The backbone is frozen, and only the exit heads are trained. All heads are supervised by cross-entropy losses. At inference time, we evaluate each exit independently and report its mAcc in Fig. 2(b).

For Fig. 2(d), we compute a sample-wise best exit using the same ground-truth-aligned rule as above. For each sample, we compare the predictions from all exit heads and assign the sample to the exit with the most reliable ground-truth-aligned prediction. Fig. 2(d) reports the proportion of samples assigned to each best exit. These sample-wise best ratios and exits are only used for analysis, not for the deployable inference pipeline.

Token-level response visualization. For Fig. 2(e), we train an additional Effort model for token-level analysis. The backbone is frozen, while the classification head is trained with randomly selected image tokens for binary classification. This gives a silver-standard token-level response map, indicating whether each patch token tends to support the AI or non-AI class. We then prune tokens by CLS-to-patch attention using the oracle keeping ratio selected above. The visualization shows that some patch tokens produce weak or misleading non-AI responses, while oracle pruning keeps tokens concentrated on more discriminative regions.

Block-level response visualization. For Fig. 2(f), we collect the logits from all exit heads for each sample and select the best-logit using the same ground-truth-aligned rule. The examples show that different samples may prefer different depths, suggesting that later blocks are not always helpful and can weaken predictions for some samples.

Oracle subtraction. The oracle variants in Fig. 2(g) and Fig. 2(h) correspond to Best Prune and Best Exit, respectively. For Best Prune, the original setting uses the full token set with a keeping ratio of 1.0, while Best Prune selects the sample-wise oracle keeping ratio. For Best Exit, the original setting uses the final exit, while Best Exit selects the sample-wise oracle exit. The reported mAcc values are computed under these oracle protocols. In Fig. 2(h), each exit-specific t-SNE only includes samples assigned to that exit by the oracle rule. These oracle results are not used as inference strategies; they estimate the potential gain from removing harmful tokens or avoiding unreliable late blocks.

A.2 Details of Main Experiments

Detector baselines. We evaluate `AID-2` on four transformer-based AIGI detectors: DeiT-S/16[54], UniFD with CLIP ViT-L/14[41], Effort with CLIP ViT-L/14[67], and Simplicity Prevails with

Table 9: Detailed accuracy comparison on the UniversalFakeDetect benchmark.

Method	mAcc	GAN						Deep fakes	Low level		Perceptual loss		Guided	LDM			Glide			Dalle
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN		SITD	SAN	CRN	IMLE		200 steps	200 w/cfg	100 steps	100 27	50 27	100 10	
DeiT-S	0.719	**1.000**	0.909	0.787	0.866	0.857	0.919	0.703	0.640	0.500	0.889	**0.994**	0.555	0.550	**0.570**	0.560	0.595	0.610	0.565	0.585
+ Cropr	0.714	0.999	0.902	0.754	0.813	0.838	0.898	**0.769**	**0.670**	0.510	0.895	0.950	0.555	0.565	0.565	0.600	**0.610**	0.545	0.510	0.620
+ LGViT	0.718	**1.000**	0.905	0.776	**0.867**	**0.858**	0.917	0.695	0.645	0.500	0.899	0.990	0.520	0.550	**0.570**	0.580	0.565	**0.615**	0.555	**0.635**
+ Cropr + LGViT	0.710	**1.000**	0.897	0.810	0.856	0.836	0.882	0.601	0.665	0.505	0.822	0.970	0.575	**0.595**	**0.570**	0.600	0.540	0.570	**0.605**	0.595
+ AID--	**0.727**	0.999	**0.918**	**0.835**	0.833	0.831	**0.922**	0.689	0.660	**0.515**	**0.916**	0.954	**0.590**	0.575	0.550	**0.640**	0.585	**0.615**	0.585	0.600
UniFD	0.814	**1.000**	**0.985**	0.945	0.820	**0.995**	**0.970**	0.666	0.630	0.575	0.595	0.720	0.700	**0.942**	0.738	0.944	0.791	0.799	**0.781**	0.868
+ Cropr	0.802	**1.000**	0.972	0.928	0.801	0.987	0.958	0.679	**0.640**	0.580	0.605	0.725	0.690	0.910	0.725	0.915	0.770	0.775	0.760	0.820
+ LGViT	0.795	**1.000**	0.978	0.932	0.805	0.990	0.962	0.625	0.610	0.550	0.580	0.700	0.680	0.920	0.710	0.930	0.760	0.770	0.750	0.857
+ Cropr + LGViT	0.797	**1.000**	0.970	0.920	0.790	0.980	0.950	0.640	0.620	0.560	0.580	0.710	0.690	0.920	0.720	0.930	0.770	0.780	0.760	0.857
+ AID--	**0.849**	0.999	0.972	**0.969**	**0.898**	0.970	0.890	**0.753**	0.625	**0.600**	**0.745**	**0.917**	**0.795**	0.920	**0.820**	**0.945**	**0.795**	**0.800**	0.770	**0.950**
Effort	0.952	**1.000**	**0.998**	**0.995**	0.942	**0.995**	**1.000**	0.878	**0.920**	0.815	**0.985**	**0.985**	0.715	**0.995**	**0.970**	**1.000**	0.970	0.965	0.985	0.970
+ Cropr	0.922	**1.000**	0.989	0.985	0.910	0.984	0.991	0.825	0.880	0.780	0.950	0.950	0.680	0.970	0.940	0.980	0.940	0.930	0.960	0.865
+ LGViT	0.909	**1.000**	0.985	0.980	0.900	0.980	0.990	0.790	0.860	0.760	0.940	0.940	0.660	0.960	0.920	0.970	0.930	0.910	0.950	0.848
+ Cropr + LGViT	0.896	**1.000**	0.980	0.970	0.880	0.970	0.980	0.750	0.840	0.740	0.920	0.920	0.640	0.950	0.900	0.960	0.910	0.890	0.940	0.888
+ AID--	**0.953**	**1.000**	0.987	0.980	**0.958**	0.983	0.994	**0.902**	0.870	**0.955**	0.918	0.918	**0.780**	0.975	0.960	**1.000**	**0.975**	**0.975**	**0.990**	**0.980**

DINOv3 ViT-7B/16, denoted as SP-DINOv3[72]. These detectors cover compact supervised ViTs, CLIP-based detectors, and large foundation-model-based detectors. DeiT-S/16 is trained end-to-end, while UniFD, Effort, and SP-DINOv3 follow their original frozen-backbone protocols. When integrating AID--, we keep the original learning rate, training epochs, batch size, data processing, and optimizer settings of each detector, and only introduce the proposed token reduction and early-exit modules. For DeiT-S/16, UniFD, and Effort, we divide the transformer into $S = 4$ stages and set the token keeping ratio to 0.5. For SP-DINOv3, we use $S = 8$ stages and a token keeping ratio of 0.85, since its backbone contains more transformer blocks and already provides stronger generalization. The warm-up and specialization stages evenly split the total training epochs. During warm-up, all exits are trained with ground-truth supervision. During specialization, exits are optimized with the proposed relative-advantage supervision.

Acceleration baselines. We compare AID-- with representative token reduction methods, including EViT [33], ATS [18], ToMe [3], METR [34], and Cropr [2], and early-exit methods, including Gong et al. [20], MET [37], LGViT [64], and EnViT [13]. We also evaluate Cropr+LGViT as a direct combination of token reduction and early exit. Based on the full comparison, we use Cropr, LGViT, and Cropr+LGViT as the main acceleration baselines in the main paper, corresponding to token reduction, early exit, and their combination. All acceleration methods are evaluated under the same detector backbones and training data for a fair comparison.

Evaluation benchmarks. We evaluate on UniversalFakeDetect [41], GenImages [73], three in-the-wild datasets, and Treasure-10, our subset covering ten recent closed-source generators in Treasure [56]. For UniversalFakeDetect, we follow UniFD and train on the ProGAN [27] subset. For GenImages, we follow Effort and train on the SD-v1.4 [44] subset. For the in-the-wild datasets and Treasure-10, we use detectors trained on the SD-v1.4 subset of GenImages. The in-the-wild datasets include Chameleon, SocialRF, and CommunityAI [32], which contain online images with diverse resolutions, compression levels, post-processing operations, and unknown generation sources. We use Treasure-10 to evaluate generalization to these recent synthesis models. The selected generators are listed in Table 10. For generalization, we report mean accuracy (mAcc). For efficiency, we report GFLOPs and throughput in images per second (FPS), measured on an NVIDIA A100 GPU with a batch size of 256.

B Additional Detailed Results

Table 9 provides detailed results for the UniversalFakeDetect comparison in Table 1. Table 10 provides detailed results for the Treasure-10 comparison in Table 5.

C Additional Experiments

In this section, we provide additional comparisons with representative transformer acceleration methods, hyperparameter ablations, and statistical analyses of selected key experiments. Unless otherwise specified, all experiments are conducted with Effort as the base detector on GenImages. We follow the same training split, backbone, and evaluation protocol as the main experiments, and report GFLOPs, FPS, and mAcc to compare both efficiency and generalization. For fair comparisons

Table 10: Detailed results on Treasure-10. AI_1–AI_{10} report accuracy on images from each generator, and Non-AI reports accuracy on real images. SP results are reported for seed 42.

Method	AI_1	AI_2	AI_3	AI_4	AI_5	AI_6	AI_7	AI_8	AI_9	AI_{10}	Non-AI
Effort	0.5944	0.0544	0.1358	0.5084	0.8128	0.1290	0.9934	0.3910	0.7058	0.9738	**0.8586**
+ AID--	**0.6602**	**0.0934**	**0.2880**	**0.6482**	**0.8456**	**0.1854**	**0.9946**	**0.5858**	**0.7820**	**0.9804**	0.8162
SP	0.8400	0.5730	0.5980	0.7350	0.7120	0.5310	0.7830	0.2210	0.6540	0.8780	**0.9680**
+ AID--	**0.9012**	**0.6636**	**0.7196**	**0.8710**	**0.8332**	**0.7178**	**0.8982**	**0.3814**	**0.8234**	**0.9496**	0.9081

AI_1–AI_{10} denote GPT-4o Image T2I, Midjourney V7, Doubao Seedream 3.0 T2I, Imagen 4, Nano Banana, Doubao Seedream 4.0, Wan 2.5 T2I Preview, Nano Banana Pro, FLUX.2, and Wan 2.2 T2I Flash, respectively.

Table 11: Additional comparison with SoTA token reduction methods on Effort with GenImages.

Method	GFLOPs	FPS	mAcc	ADM	BigGAN	GLIDE	Midjourney	SD-v1.4	SD-v1.5	VQDM	Wukong
Effort	81.2 (1.00×)	87.4 (1.00×)	0.911	0.787	0.776	0.933	0.824	**0.998**	**0.998**	0.917	0.974
+ EViT	40.8 (1.99×)	172.8 (1.98×)	0.911	0.772	0.824	0.934	0.872	0.989	0.990	0.936	0.970
+ ATS	40.8 (1.99×)	169.6 (1.94×)	0.877	0.734	0.765	0.912	0.801	0.976	0.979	0.896	0.954
+ ToMe	45.2 (1.80×)	158.6 (1.81×)	0.896	0.755	0.803	0.925	0.842	0.985	0.986	0.912	0.961
+ METR	41.0 (1.98×)	166.4 (1.90×)	0.857	0.701	0.746	0.891	0.768	0.965	0.969	0.878	0.936
+ Cropr	40.8 (1.99×)	173.1 (1.98×)	0.918	0.766	0.855	0.938	0.920	0.974	0.978	0.940	0.973
+ ours PTR	41.0 (1.98×)	171.5 (1.96×)	**0.928**	**0.794**	**0.862**	**0.944**	**0.925**	0.986	0.988	**0.949**	**0.976**

Table 12: Additional comparison with SoTA early-exit methods on Effort with GenImages.

Method	GFLOPs	FPS	mAcc	ADM	BigGAN	GLIDE	Midjourney	SD-v1.4	SD-v1.5	VQDM	Wukong
Effort	81.2 (1.00×)	87.4 (1.00×)	0.911	**0.787**	0.776	0.933	0.824	0.998	0.998	0.917	0.974
+ Gong et al.	56.8 (1.43×)	125.6 (1.44×)	0.904	0.724	0.802	0.928	0.882	0.998	0.998	0.914	0.986
+ MET	55.1 (1.47×)	128.7 (1.47×)	0.892	0.690	0.785	0.916	0.866	0.995	0.996	0.899	0.989
+ LGViT	53.4 (1.52×)	132.1 (1.51×)	0.911	0.712	0.818	0.935	0.903	**1.000**	**0.999**	0.926	**0.992**
+ EnViT	54.8 (1.48×)	129.9 (1.49×)	0.864	0.662	0.751	0.889	0.835	0.984	0.986	0.872	0.933
+ ours ABE	54.6 (1.49×)	130.9 (1.50×)	**0.931**	0.771	**0.882**	**0.955**	**0.928**	0.998	0.998	**0.939**	0.977

in Table 11 and Table 12, we retrain all methods, including PTR and ABE, on the same machine with the same random seed.

C.1 Additional Comparison with SoTA Token Reduction

We compare our PTR module in AID-- with EViT [33], ATS [18], ToMe [3], METR [34], and Cropr [2]. We set the keeping ratio to 0.5 for pruning-based methods and tune ToMe's merging ratio r for a comparable computational budget. To preserve generalization, we freeze the backbone by default, training additional modules as required by each method. Cropr is the strongest token-reduction baseline in this comparison and is used in the main paper. Table 11 shows that existing methods reduce GFLOPs and improve throughput but do not consistently improve mAcc, as generic token importance criteria may discard useful forensic cues or retain misleading ones. In contrast, PTR uses progressive token reduction with high-order dependency ranking to better model the relation between patch tokens and the final prediction, achieving a stronger accuracy-efficiency trade-off.

C.2 Additional Comparison with SoTA Early Exit

We compare our ABE module with Gong et al. [20], MET [37], LGViT [64], and EnViT [13]. For these baselines, exit heads are placed at the same stages as AID--, using the same backbone and training data. LGViT is the strongest early-exit baseline in this comparison and is used in the main paper. Table 12 shows that existing methods reduce computation but offer limited generalization gains. Generic early-exit supervision may force shallow heads to imitate the final head despite differences in their visual cues. Our ABE module combines relative-advantage head supervision with a class-asymmetric exit rule, encouraging each exit to specialize in samples it can handle reliably and yielding a stronger accuracy-efficiency trade-off.

Table 13: Hyperparameter ablations on GenImages. † denotes the default setting.

(a) Decay factor

α	0	0.2	0.4	0.6	$0.7^{\dagger}$	0.8	0.9	1.0
mAcc ↑	0.914	0.918	0.924	0.932	**0.934**	0.931	0.931	0.929

(b) Warm-up/specialization split

Training split	$[0, 1]$	$[\frac{1}{4}, \frac{3}{4}]$	$[\frac{1}{2}, \frac{1}{2}]^{\dagger}$	$[\frac{3}{4}, \frac{1}{4}]$	$[1, 0]$
mAcc ↑	—	0.941	**0.947**	0.945	0.928

(c) Number of stages and token keeping ratio

$[S, \rho]$	Baseline	$[2, 0.25]$	$[4, 0.5]^{\dagger}$	$[6, 0.63]$	$[8, 0.7]$
FPS ↑	87.4	194.9	188.0	180.9	173.9
Speedup	1.00×	2.23×	2.15×	2.07×	1.99×
mAcc ↑	0.911	0.932	0.947	**0.949**	0.945

Table 14: Multi-seed results.

Benchmark	Method	mAcc ↑	$t(4)$	p
In-the-Wild	Effort	0.5762 ± 0.0201	—	—
	Effort + `AID--`	$\mathbf{0.5932} \pm 0.0180$	9.815	0.0006
	SP	0.9318 ± 0.0015	—	—
	SP + `AID--`	$\mathbf{0.9410} \pm 0.0023$	7.948	0.0014
Treasure-10	SP	0.8092 ± 0.0038	—	—
	SP + `AID--`	$\mathbf{0.8363} \pm 0.0067$	10.487	0.0005

C.3 Hyperparameter Ablations

We complement the analyses of the token keeping ratio and class-asymmetric exit thresholds in Fig. 4 and Fig. 6 with ablations on the decay factor α, the warm-up/specialization split, and the number of stages S on GenImages. The α ablation uses Effort with PTR only, without ABE; the other two use Effort with the full `AID--` framework. Table 13 summarizes the results.

Decay factor α. The decay factor controls the contribution of higher-order token dependencies, with $\alpha = 0$ retaining only first-order dependencies. As α increases from 0 to 0.7, mAcc improves from 0.914 to 0.934. For $\alpha \in [0.6, 1.0]$, mAcc stays within 0.929–0.934, indicating limited sensitivity in this range. We use $\alpha = 0.7$, which gives the highest mAcc in this ablation.

Warm-up/specialization split. Warm-up establishes classification ability at each exit before relative-advantage supervision encourages specialization. When warm-up occupies one-quarter to three-quarters of the training epochs, mAcc ranges from 0.941 to 0.947. An equal split achieves the highest mAcc of 0.947, while using only warm-up yields 0.928, supporting the benefit of specialization. We therefore allocate half of the training epochs to each phase.

Number of stages S. Since pruning is applied at stage boundaries, we jointly adjust S and the keeping ratio ρ to maintain approximately $2\times$ acceleration. Increasing (S, ρ) from $(2, 0.25)$ to $(4, 0.5)$ improves mAcc from 0.932 to 0.947. With six or eight stages, mAcc remains within 0.945–0.949, while throughput drops from 188.0 FPS at four stages to 180.9 and 173.9 FPS, respectively. These results support $(S, \rho) = (4, 0.5)$ as an accuracy-efficiency trade-off for Effort.

C.4 Multi-Seed Results

For selected key experiments, we use five shared random seeds $\{0, 1, 2, 42, 3407\}$. Table 14 reports mAcc as mean $\pm$ sample standard deviation and two-sided paired t-tests against the corresponding baselines, pairing runs by seed. `AID--` improves mean mAcc in the three evaluated settings, with $p < 0.002$.

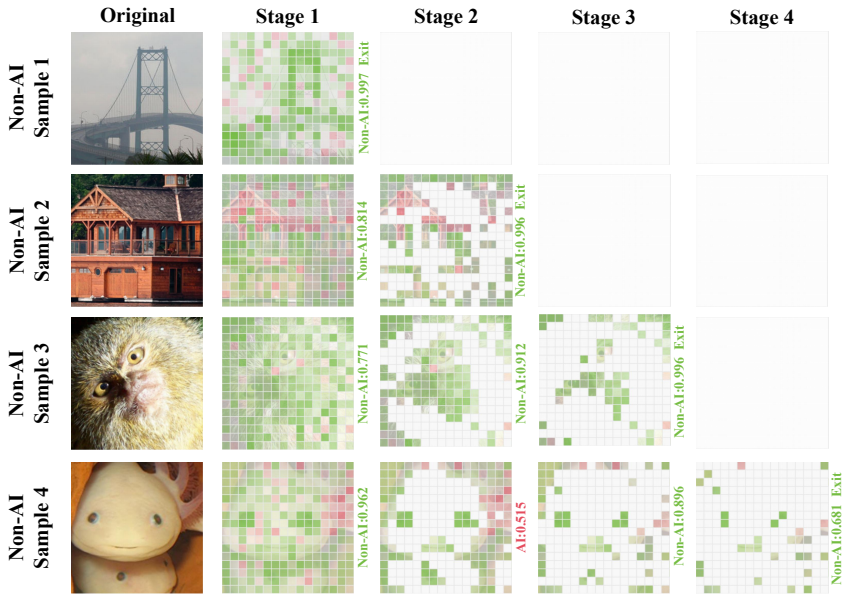

Figure 7: Case visualization of a non-AI image under Effort + AID--.

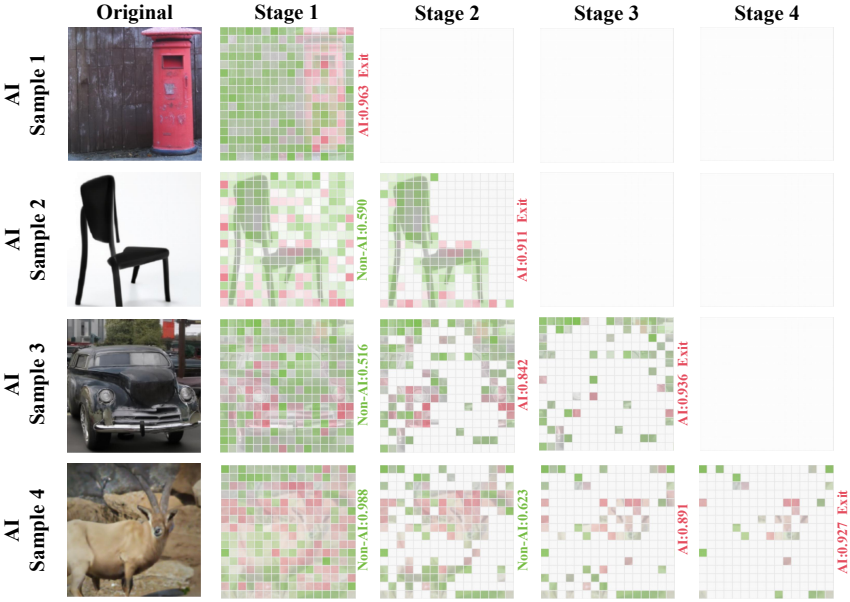

Figure 8: Case visualization of an AI image under Effort + AID--.

D Case Visualization

We provide case visualizations for one non-AI and one AI sample at each exit stage in Fig. 7 and Fig. 8 to exhibit and analyze the inference behavior of AID--. Specifically, we visualize non-AI and AI samples under Progressive Token Reduction (PTR) and Adaptive Block Early-Exit (ABE). For PTR, we show the tokens retained at different stages. For ABE, we report the prediction confidence of each exit and the final exit decision. From these visualizations, we make three observations:

First, the decision cues for AIGI detection seem to be strongly related to image boundaries. In most cases, PTR tends to preserve tokens around object or structure boundaries during progressive pruning, suggesting that boundary-related regions may provide important cues for distinguishing AI and non-AI images.

Second, the difficulty perceived by humans and the difficulty perceived by the model are not always consistent. For example, Non-AI Sample 2 contains rich local details, making it relatively hard for humans to judge without careful inspection. However, the model can already make a reliable prediction at Stage 2. In contrast, AI Samples 2 and 3 contain obvious deformations that violate real-world intuition, and humans can identify them more easily. However, the model still requires later stages to make correct predictions, and both samples are misclassified at Stage 1. This indicates that visual abnormality from a human perspective does not always correspond to shallow-stage separability in the model.

Third, the class-asymmetric exit rule has strong practical value. For AI Sample 4, the Stage-1 exit predicts the image as non-AI with a high confidence of 0.988. With a standard confidence-based exit rule, this sample would likely exit too early and be misclassified. Thanks to the class-asymmetric exit rule, the sample is not allowed to exit easily under a non-AI prediction, and it is finally corrected as AI at later stages. This case shows that using stricter exit criteria for non-AI predictions can reduce risky early exits and improve the reliability of AIGI detection.

E Limitations and Future Work

Architecture scope. Although `AID--` achieves a strong accuracy-efficiency trade-off, our current design is mainly developed for transformer-based AIGI detectors. This is a natural choice because modern AIGI detectors commonly rely on ViT-style backbones such as CLIP and DINO, where patch tokens and stacked blocks provide a direct interface for token reduction and early exit. Nevertheless, extending the same subtraction-based principle to CNN-based or hybrid detectors is still an interesting direction. For example, feature maps in CNNs could be reorganized into spatial units for progressive reduction, and stage-wise convolutional blocks could be equipped with adaptive exits.

Threshold calibration. The exit behavior of `AID--` depends on confidence thresholds. Our class-asymmetric exit rule is designed for the asymmetric risk in AIGI detection and helps prevent risky early exits in practice. We use fixed thresholds of 0.90 for AI and 0.995 for non-AI across all backbones and datasets, which may not be optimal for every deployment scenario. Calibration for specific error preferences remains future work.

Weak cue preservation. Token reduction may face challenges when the discriminative cues are highly local or weak. Some AI-generated images may only contain subtle artifacts after post-processing, compression, or resizing, and overly aggressive pruning may remove useful fine-grained cues. To reduce this risk, our method adopts progressive pruning and high-order dependency ranking instead of one-shot token removal. This design allows the model to preserve more informative tokens across stages while gradually reducing redundant computation. Even so, the trade-off between computation saving and weak cue preservation remains an important direction for future improvement.

F Broader Societal Impacts

This work studies efficient and generalizable AI-generated image detection. Its positive impact is to support the detection of synthetic visual content with lower inference cost and better robustness to unseen generators. This may help practical systems reduce visual misinformation, synthetic media abuse, and copyright-related misuse.

The work may also have negative impacts. Detection systems can be misused for over-aggressive moderation or surveillance. Detection errors may also cause harm, where non-AI images are wrongly labeled as AI-generated or AI-generated images are missed. In addition, stronger detectors may encourage adversaries to design better evasion strategies. Therefore, the proposed method should be used as a supporting tool rather than the only basis for high-stakes decisions.

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 paper's contributions and scope are clearly stated in the abstract and introduction.

   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: Limitations are discussed in the paper.

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 include theoretical results.

   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: All the information needed to reproduce the main experimental results of the paper is provided.

   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: [No]

   Justification: We will publicly release the code upon publication.

   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: All the training and test details necessary to understand the results are specified.

   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: [Yes]

   Justification: For selected key experiments, Appendix C.4 and Table 14 report mean mAcc $\pm$ sample standard deviation over five shared random seeds and two-sided paired t-tests.

   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: We provide sufficient information on the computer resources needed to reproduce 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 conducted in the paper conforms, in every respect, 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: We discuss both potential positive and negative societal impacts of the work performed.

    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 poses no such risks.

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: We use publicly available datasets and models with proper citation, and adhere to their respective licenses.

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: [N/A]

Justification: The paper does not release new assets.

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 experiments or research with human subjects.

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 crowdsourcing experiments or research with human subjects.

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 development in this research does not involve LLMs as any important, original, or non-standard components.

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.