Dissect and Prune: Enhancing Robustness in AI-Generated Image Detection

Anonymous Authors[1]

Abstract

While existing AI-generated image detectors report high performance, we identify that this is largely driven by a critical *prediction asymmetry*: a bias toward the real class that severely limits sensitivity to generated content, especially under standard post-processing operations such as compression and resizing. We hypothesize that this stems from the model's reliance on spurious features, distracting signals that obscure true generative artifacts. To address this, we propose DEAR (Dissect and Prune), which leverages inpainted images to identify and prune these interfering components. Specifically, we find that features strongly aligned to either inpainted or non-inpainted regions are less robust to post-processing. By measuring the alignment between channel activations and inpaint masks, DEAR removes features at both extremes, retaining only those that capture genuine generative artifacts. Experimental results demonstrate that our approach significantly enhances robustness against unseen generators and post-processing, effectively mitigating the prediction asymmetry.

1. Introduction

Recent advances in generative models, particularly diffusion (Ho et al., 2020; Rombach et al., 2022) and flow-based models (Lipman et al., 2022), have enabled the synthesis of near-photorealistic images that are increasingly difficult to distinguish from real photographs. While these technologies offer remarkable generative capabilities, they also raise significant societal concerns, including the proliferation of misinformation through convincing fake imagery, the creation of non-consensual deepfakes, and potential copyright infringement. As AI-generated images (AIGI) become more prevalent and sophisticated, the development of reliable detection methods has become an urgent and critical challenge.

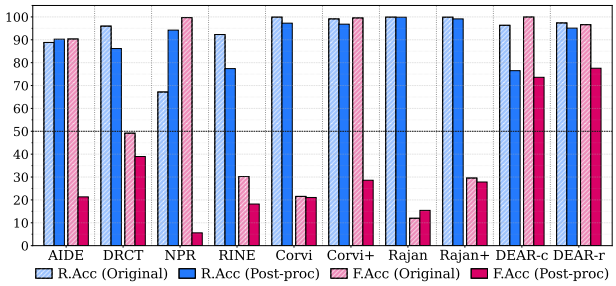

Figure 1. **Prediction Asymmetry in AIGI Detection.** Comparison of real accuracy (R.Acc) and fake accuracy (F.Acc) before and after post processing on FLUX (Labs, 2024) generated images. Existing detectors maintain high R.Acc regardless of post processing, but F.Acc drops dramatically after post processing is applied. This asymmetric degradation reveals that detectors rely on fragile spurious features for fake detection. DEAR mitigates this asymmetry by pruning such features and retaining only robust forensic signals.

To address this challenge, numerous AIGI detectors have been proposed, spanning CNN-based approaches (Wang et al., 2020; Corvi et al., 2023; Tan et al., 2024b; Rajan et al., 2025; Rajan & Lee, 2025), CLIP-based methods (Ojha et al., 2023; Cozzolino et al., 2024; Yan et al., 2025; Tan et al., 2025), Vision Transformer-based methods (Chen et al., 2024a; Guillaro et al., 2025; Chen et al., 2025), and frequency-domain analysis techniques (Tan et al., 2024a). These methods report impressive performance on benchmark datasets, often achieving high accuracy or average precision. Such results have suggested that AIGI detection is approaching a solved problem, with detectors capable of reliably distinguishing AI-generated from real images.

However, upon closer examination of detector performance, we observe a critical phenomenon that we term *prediction asymmetry*: a systematic bias where high overall accuracy is sustained primarily by the near-perfect recognition of real images, while the sensitivity to AI-generated content remains disproportionately low. This imbalance obscures the detector's actual reliability, as high overall metrics hide a failure to identify the very target class they are designed to detect. This behavior becomes markedly more pronounced under common real-world post processing such as JPEG compression and resizing (Li et al., 2025b). Under such con-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ditions, the accuracy on fake images decreases significantly as the detector gravitates toward the real class, becoming a trivial classifier that defaults to predicting real images.

We hypothesize that this prediction asymmetry stems from the detector's reliance on spurious correlations rather than robust forensic traces (Rajan & Lee, 2025). Existing detectors often exploit dataset-specific biases, such as compression artifacts, as indicators of realness, while simultaneously overfitting to fragile shortcuts specific to generated content (Rajan & Lee, 2025; Grommelt et al., 2024; Yan et al., 2025; Kashiani et al., 2025; Ma et al., 2025). This reliance leads to failure even in the original setting; when encountering unseen generators that lack the specific fingerprints seen during training, the detector fails to recognize the generated content (Li et al., 2025b). This vulnerability is further exacerbated by post-processing. Since these non-intrinsic signals are sensitive to perturbations, their degradation diminishes the discriminative cues available to the detector. Consequently, the model exhibits a systematic shift toward predicting the real class.

To address the prediction asymmetry driven by spurious correlations, we aim to identify and mitigate the influence of specific detector features responsible for these non-robust dependencies. To this end, we draw upon the principles of Network Dissection (Bau et al., 2017; 2018; Tousi et al., 2021), a framework designed to interpret internal representations. Successfully dissecting a forensic detector, however, requires a precise ground truth to distinguish between features activating on generated artifacts and those responding to authentic signals. We identify *inpainted images* as the optimal probe for this purpose. Since these images spatially isolate AI-generated content within inpainted regions from the authentic context in non-inpainted backgrounds while offering exact ground truth masks, they provide the necessary reference to rigorously correlate feature activations with pixel provenance.

Building on this insight, we propose DEAR (**D**iss**E**ct **A**nd P**R**une) for robust AIGI detection. Our approach adopts a selective pruning strategy. First, we employ inpainted images as a diagnostic tool, creating a controlled environment where generated pixels coexist with real image contexts. Second, we perform detector dissection by quantifying the divergence in feature activations between the generated (inpainted) regions and the real (non-inpainted) backgrounds. This metric allows us to determine whether a feature is primarily driven by synthetic artifacts or authentic signals. Third, we refine the classifier by pruning features at both extremes of this spectrum, targeting channels strongly aligned with generated content and those biased toward real backgrounds. We find that features at these extremes are highly susceptible to degradation. Consequently, removing these components forces the detector to rely on robust forensic signals, effectively eliminating the spurious representations that degrade performance under perturbations.

Our main contributions are as follows: **(1)** We identify prediction asymmetry as a fundamental limitation of current AIGI detectors, where high overall accuracy conceals a systematic bias toward the real class that severely degrades fake detection under perturbations. **(2)** We propose DEAR, a dissection guided feature pruning mechanism that leverages inpainted images to measure feature alignment and selectively removes spurious features at both extremes, retaining only those that capture genuine generative artifacts. **(3)** We demonstrate that DEAR significantly enhances robustness against unseen generators and post processing operations through extensive experiments, effectively mitigating prediction asymmetry.

2. Preliminaries

To systematically evaluate detector reliability, we analyze their behavior across diverse generators and perturbations, revealing a critical failure mode which we term *Prediction Asymmetry*: a systematic bias where detectors maintain high accuracy on real images while failing to identify AI-generated images. This phenomenon aligns with findings in recent large-scale benchmarks like AIGIBench (Li et al., 2025b), which we analyze along two dimensions: generalization to unseen generators and robustness under post-processing.

2.1. Motivation: Prediction Asymmetry Problem

First, we observe a pronounced generalization gap under unseen-generator shift. As illustrated in Figure 1, Corvi achieves near-perfect R.Acc (99.9%) on original FLUX (Labs, 2024) images but a dismal F.Acc of 21.5%, indicating that detectors default to the real class when specific training fingerprints are absent. Second, this bias amplifies into a **robustness collapse** under post-processing. For instance, NPR's average F.Acc plummets from 95.9% to 12.2% while its R.Acc actually rises from 67.2% to 94.2%, effectively transforming into a trivial classifier that labels all inputs as real once fragile cues are disrupted. This susceptibility exposes a critical practical limitation where aggregate metrics like AUC mask a structural bias toward the real class, inflating reliability assessments in real-world settings.

We attribute this asymmetry to the detector's reliance on *spurious correlations*, which fall into two distinct failure modes. First, detectors frequently exploit dataset-specific biases, such as JPEG compression artifacts or high-frequency details, as spurious indicators of realness (Grommelt et al., 2024; Rajan & Lee, 2025). Consequently, when perturbations introduce similar artifacts to fake images, detectors misclassify them as real. Second, regarding spurious fea-

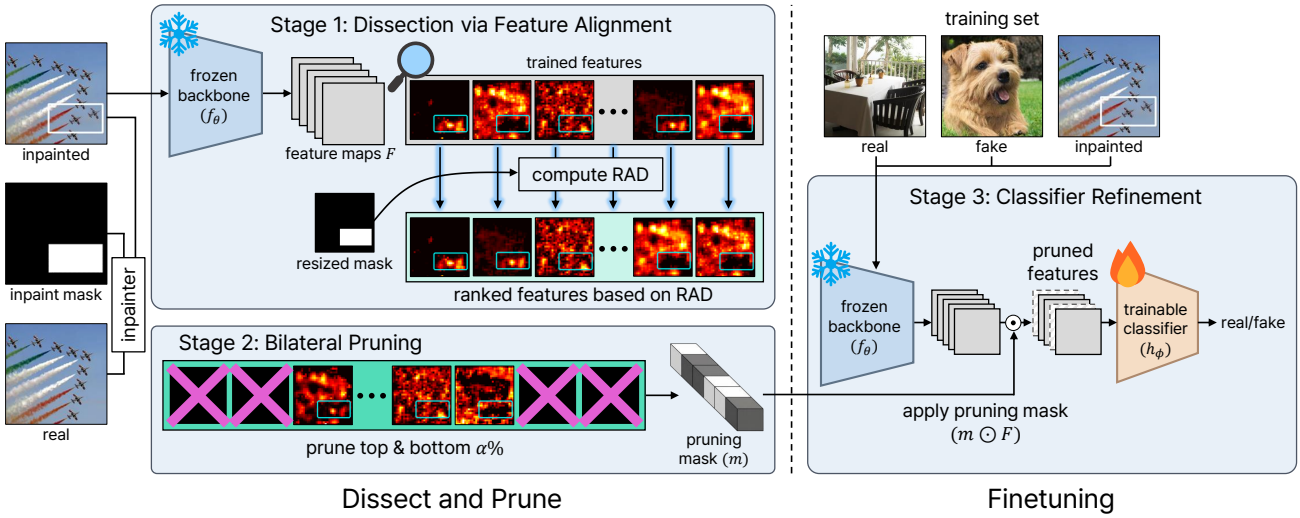

Figure 2. **Overview of Dissect and Prune (DEAR).** Our method operates in three stages: (1) **Dissection via Feature Alignment**, where we use inpainted images and masks as a diagnostic tool to measure how strongly each feature channel aligns with generated or real regions through Regional Activation Discrepancy (RAD); (2) **Bilateral Pruning**, where we identify and remove channels at both extremes of the RAD distribution, as these represent non-intrinsic components relying on fragile shortcuts or spurious realness indicators; and (3) **Classifier Refinement**, where we fine-tune the final linear classifier on the pruned feature space obtained by Dissect and Prune (DEAR) while the backbone remains frozen. (Section 3.2).

tures specific to generated content, evidence suggests that detectors overfit to monotonous and non-robust artifacts, as exemplified by low-rank traces (Yan et al., 2025), spectral biases (Kashiani et al., 2025), and generator-specific cues (Ma et al., 2025). These artifacts serve as fragile shortcuts (Geirhos et al., 2020) that are obliterated by post-processing; once disrupted, the detector defaults to the real class.

The confluence of these factors explains the observed prediction asymmetry and mirrors the concept of a *modality gap* (Li et al., 2025b), where real images possess concentrated and consistent statistics while AI-generated images rely on artifacts characterized by high modality variance. Motivated by these insights into spurious correlations linked to both real and fake classes, we propose DEAR to explicitly dissect and prune features driven by these spurious dependencies, thereby isolating robust forensic features.

2.2. Network Dissection for Feature Analysis

To address the prediction asymmetry problem, it is critical to identify which specific components of the feature backbone are responsible for encoding spurious correlations. Since deep neural networks operate as black boxes, we require a quantitative framework to interpret their internal representations. To this end, we adopt **Network Dissection** (Bau et al., 2017; 2018).

The central premise of this framework is that individual channels within a feature map often emerge as concept-specific units, activating in response to distinct visual attributes. Formally, for a given unit u, its activation map is upsampled to match the image resolution and binarized into a segmentation proposal $\hat{M}_u(\mathbf{x})$ using a statistical threshold. The interpretability of the unit for a concept c is then computed via the Intersection over Union (IoU) score:

$$\mathrm{IoU}_{u,c} = \frac{\sum_{\mathbf{x}\in\mathcal{D}} |\hat{M}_u(\mathbf{x}) \cap M_c(\mathbf{x})|}{\sum_{\mathbf{x}\in\mathcal{D}} |\hat{M}_u(\mathbf{x}) \cup M_c(\mathbf{x})|}, \tag{1}$$

where $M_c(\mathbf{x})$ denotes the ground truth mask for concept c.

While the original framework measures alignment with high-level semantic concepts (e.g., objects, textures), we extend this approach to the problem of AI-generated image detection. In Section 3, we will adapt this formulation to measure feature alignment with both *generative artifacts* (inpainted regions) and *authentic signals* (backgrounds). This approach enables us to profile feature behaviors based on their sensitivity to generated versus authentic content, allowing us to categorize features that strictly activate on fake regions, those tracking real backgrounds, or those exhibiting no distinct regional preference.

3. Dissect and Prune (DEAR)

We present *Dissect and Prune* (DEAR), a feature selection framework designed to address the prediction asymmetry problem in AIGI detection. Figure 2 illustrates an overview of our proposed method. We first introduce a motivating example to highlight the severity of prediction asymmetry in existing detectors. We then describe our approach,

which employs diagnostic data generation via inpainting and network dissection to analyze internal feature alignment. Finally, we demonstrate how this analysis guides the pruning of spurious components and the refinement of the linear classifier to enhance detection robustness.

3.1. Inpainting-Based Diagnostic Data Generation

To dissect detector representations and isolate features responsible for spurious correlations, we require a diagnostic setting that spatially separates generative artifacts from authentic signals. Inpainted images provide precisely this controlled environment. By replacing a masked region of a real image with content synthesized by a generative model, we create samples where pixel provenance varies across defined spatial boundaries. Crucially, this setup allows us to project features onto a quantitative alignment spectrum. By analyzing this spectrum, we can diagnose which subsets of features are the primary drivers of non-robust behaviors, including those strictly adhering to generated regions, those biased toward authentic backgrounds, or those showing no clear regional preference.

We generate our diagnostic dataset using the inpainting variant of Stable Diffusion 1.5 (Rombach et al., 2022). For each real image $\mathbf{x}_{\text{real}}$, we sample a randomly positioned rectangular binary mask $\mathbf{M} \in \{0,1\}^{H\times W}$ to define the region for manipulation. The inpainting model then synthesizes content $\mathbf{x}_{\text{gen}}$ for this masked area while conditioning on the surrounding authentic context. To prevent the detector from exploiting trivial edge discontinuities at the mask boundaries, we apply Gaussian blur to the mask edges before compositing. The final inpainted image is constructed as:

$$\mathbf{x}_{\text{inpaint}} = \mathbf{M} \odot \mathbf{x}_{\text{gen}} + (1 - \mathbf{M}) \odot \mathbf{x}_{\text{real}}, \tag{2}$$

where $\mathbf{x}_{\text{gen}}$ denotes the output from the inpainting model and $\odot$ represents elementwise multiplication. Representative examples are provided in Figure 6 in the Appendix E.

A key advantage of this construction is that the ground truth mask $\mathbf{M}$ is known, which naturally provides a precise spatial reference for internal analysis. This naturally facilitates the application of Network Dissection (Bau et al., 2017; 2018), enabling us to quantitatively measure the alignment between internal feature maps and the generated regions.

3.2. Detector Dissection via Feature Alignment

Given inpainted images where the ground truth mask clearly separates generated and real regions, we now propose a method to quantify how individual feature channels align with these regions. Our approach draws inspiration from Network Dissection (Bau et al., 2017), which identifies interpretable units by measuring overlap between activation maps and semantic concepts. Instead of semantic categories, we measure channel alignment with inpainted regions to expose the reliance on spurious realness indicators and fragile generator shortcuts. By analyzing these alignment characteristics, we identify non-intrinsic components driven by confounding correlations, thereby uncovering the underlying drivers of asymmetric detection.

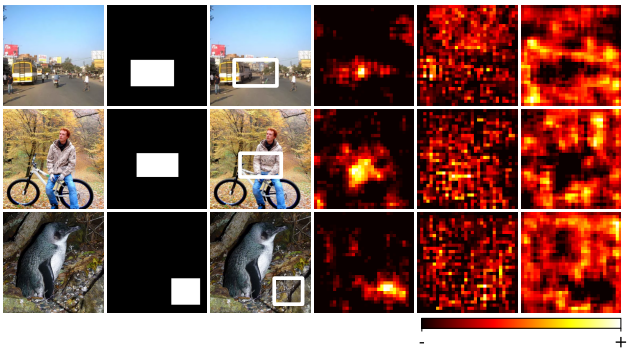

Figure 3. **Feature Alignment Visualization.** From left to right: original real image, inpainting mask, inpainted image, and activation maps from three representative channels. The high RAD channel (fourth column) activates strongly within the inpainted region, the low RAD channel (sixth column) activates predominantly on the background, and the middle RAD channel (fifth column) shows no clear regional preference.

Regional Activation Discrepancy. To quantify the alignment between a feature channel and the inpainted mask, we adopt a density based metric inspired by the Chan Vese segmentation model (Chan & Vese, 2001), which assumes piecewise constant image regions and measures how well a given contour separates distinct intensity levels.

Let $\mathbf{F}_k \in \mathbb{R}^{h\times w}$ denote the activation map of the kth channel from the penultimate convolutional layer, and let $\mathbf{M} \in \{0,1\}^{h\times w}$ be the downsampled inpaint mask. We define the *Regional Activation Discrepancy* (RAD) as:

$$S_k = \mu_{\text{in}}^{(k)} - \mu_{\text{bg}}^{(k)}, \tag{3}$$

where

$$\mu_{\text{in}}^{(k)} = \frac{\sum_{x\in\Omega_{\text{in}}} \mathbf{F}_k(x)}{|\Omega_{\text{in}}|}, \quad \mu_{\text{bg}}^{(k)} = \frac{\sum_{x\in\Omega_{\text{bg}}} \mathbf{F}_k(x)}{|\Omega_{\text{bg}}|}, \tag{4}$$

and $\Omega_{\text{in}} = \{x : \mathbf{M}(x) = 1\}$ denotes the inpainted region and $\Omega_{\text{bg}} = \{x : \mathbf{M}(x) = 0\}$ denotes the background. By normalizing by regional area, RAD remains robust regardless of whether the inpainted region covers a small or large fraction of the image.

Intuitively, a channel with high positive RAD activates strongly within the generated region and weakly in the real background, suggesting sensitivity to generative artifacts. Conversely, a channel with large negative RAD activates preferentially on pixels within real regions.

Analysis. We apply this dissection procedure to two representative detectors: Corvi (Corvi et al., 2023) and Rajan (Rajan et al., 2025). Both employ a ResNet 50 backbone trained on real images from LSUN and COCO. The key difference lies in how training fake images are generated: Corvi uses LDM with text prompts corresponding to real image content, while Rajan generates fake images by passing real images through the VAE encoder decoder of LDM. The latter approach ensures pixel level alignment between real and fake pairs, isolating decoder artifacts as the sole distinguishing factor. We focus on the final convolutional layer before global average pooling, which contains 2048 channels and provides a semantically meaningful representation before classification.

For each detector, we compute RAD values across approximately 5000 inpainted images, averaging per channel scores to obtain a stable ranking. Figure 3 visualizes activation patterns from the Rajan detector at different points along the RAD distribution. Channels with high RAD values exhibit activation maps that closely match the inpainted mask, indicating selective response to generated content. Channels with strongly negative RAD values show the inverse pattern, activating primarily on real regions.

Alignment Predicts Robustness. A natural question arises: do channels that align strongly with generated regions also exhibit greater robustness to post processing perturbations? To investigate this, we measure the sensitivity of each channel to WEBP compression, a particularly relevant perturbation for these detectors. The LSUN images used to train these detectors were originally compressed using WEBP before being saved in PNG format, while the synthetic training images lacked such compression artifacts. This dataset imbalance caused the detectors to spuriously associate WEBP compression patterns with real images (Rajan & Lee, 2025), a phenomenon analogous to the JPEG bias observed in (Grommelt et al., 2024). We quantify channel robustness by computing the mean squared error (MSE) between feature activations before and after applying WEBP compression. Channels whose activations change substantially under this perturbation are deemed less robust.

Figure 4 illustrates the relationship between RAD and robustness, revealing that channels at *both extremes* of the RAD distribution are more susceptible to degradation than those in the middle range. We attribute this trend to the spurious nature of features at these extremes: strongly negative values correspond to dataset-specific signatures like compression artifacts (Rajan & Lee, 2025; Grommelt et al., 2024), while highly positive values overfit to fragile generator shortcuts (Yan et al., 2025; Kashiani et al., 2025; Ma et al., 2025). As both types of signals are easily disrupted by post-processing, their removal enhances stability. In contrast, channels with intermediate RAD values demonstrate greater robustness under perturbation, as they are less reliant on these non-intrinsic artifacts. This suggests that pruning extreme components is an effective strategy to suppress spurious dependencies and enhance overall detector robustness.

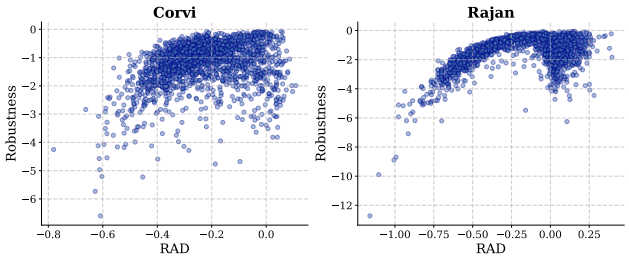

Figure 4. **Relationship between Feature Alignment and Robustness.** Each point represents one of 2048 channels from the final convolutional layer. The vertical axis shows RAD (alignment with inpainted regions) and the horizontal axis shows robustness measured as negative MSE under WEBP compression. Left: Corvi detector. Right: Rajan detector. Channels at the extremes of the RAD distribution exhibit lower robustness, while channels away from these extremes tend to be more robust.

3.3. Feature Pruning and Classifier Refinement

Leveraging the insights from our dissection analysis, we introduce the *Dissect and Prune* strategy. Our analysis indicates that features with extreme alignment scores—whether strongly fake-aligned or strongly real-aligned—are structurally fragile. Therefore, we implement a bilateral pruning mechanism to excise these spurious dependencies.

Bilateral Pruning Strategy. Unlike conventional feature selection methods that strictly prioritize high-activation channels, our analysis (Figure 4) indicates that channels at *both* extremes of the RAD distribution are susceptible to post-processing degradation. To address this, we implement a *bilateral pruning* strategy that removes these non-robust components.

Let $\{S_k\}_{k=1}^{D}$ denote the RAD values computed over the diagnostic dataset. We define a binary pruning mask $\mathbf{m} \in \{0,1\}^D$ to retain only channels within a robust intermediate range. Given lower and upper percentiles α_{low} and α_{high}, we compute thresholds τ_{low} and τ_{high} based on the empirical distribution of S. The mask for the k-th channel is defined as:

$$m_k = \mathbb{1}[\tau_{\text{low}} \leq S_k \leq \tau_{\text{high}}], \tag{5}$$

where $\mathbb{1}[\cdot]$ is the indicator function. This effectively prunes features that are spuriously correlated with specific compression artifacts (negative extreme) or generator-specific noise (positive extreme), retaining only those stable under perturbation.

Classifier Refinement on Pruned Features. We apply this mask to the pre-trained backbone. To prevent the back-

bone from re-learning spurious correlations, we freeze all feature extraction parameters θ. The pruning is applied via element-wise multiplication to the feature tensor $\mathbf{F} \in \mathbb{R}^{D\times h\times w}$ before the global pooling layer:

$$\tilde{\mathbf{F}} = \mathbf{m} \odot \mathbf{F}. \tag{6}$$

With the robust feature subspace defined by $\tilde{\mathbf{F}}$, we reinitialize the weights of the final linear classifier h_ϕ. Crucially, we fine-tune the classifier using a combined dataset comprising both the original training data $\mathcal{D}_{\text{train}}$ and the diagnostic inpaint data $\mathcal{D}_{\text{inpaint}}$. This joint optimization ensures that the classifier adapts to the pruned feature space while learning to distinguish intrinsic generative artifacts from both global images and localized inpainting regions. The complete procedure is formalized in Algorithm 1 in Appendix C.

4. Experiments

In this section, we empirically validate the effectiveness of DEAR. We investigate **(1)** whether it maintains robustness against post-processing artifacts such as compression and resizing, **(2)** whether DEAR generalizes to diverse unseen generative models, and **(3)** whether it effectively alleviates the prediction asymmetry problem. Further implementation details and extended analyses are provided in Appendices C and E.

4.1. Experimental Setup

Datasets. For real images, we use 3,000 images from the Redcaps dataset (Desai et al., 2021), which contains diverse web crawled images with natural variations in content, resolution, and compression. To evaluate generalization across generators, we construct a synthetic test set spanning multiple architectures and generation paradigms. Specifically, we include images from diffusion based models: Stable Diffusion (SD, 3,000 images) (Rombach et al., 2022), FLUX (3,000) (Labs, 2024), Kandinsky (3,100) (Razzhigaev et al., 2023), Playground (3,150) (Li et al., 2024), and PixArt (3,150) (Chen et al., 2024b); latent consistency models: LCM (3,146) (Luo et al., 2023); alternative architectures: Wuerstchen (3,150) (Pernias et al., 2023) and aMUSEd (3,150) (Patil et al., 2024); and commercial systems: Midjourney (3,000) (mid).

To assess performance under realistic deployment conditions, we additionally evaluate on three in the wild benchmarks: Chameleon (6,934 images) (Yan et al., 2024), a curated collection of challenging AI generated images from online art communities; WildRF (1,051) (Cavia et al., 2024), images collected from social media platforms using common hashtags; and LOKI (1,921) (Ye et al., 2025), a comprehensive benchmark spanning diverse generation sources and specialized domains. These benchmarks are particularly valuable as they contain images that have undergone unknown post processing operations during upload and sharing. Further details are provided in Appendix D.1.

Baselines. We compare against a comprehensive set of state of the art detectors spanning multiple paradigms. From CNN based methods, we include Corvi (Corvi et al., 2023), Rajan (Rajan et al., 2025), NPR (Tan et al., 2024b), SAFE (Li et al., 2025a), AIDE (Yan et al., 2024), FerretNet (Liang et al., 2025), and LaDeDa (Cavia et al., 2024). From CLIP based approaches, we evaluate UFD (Ojha et al., 2023), C2P-CLIP (Tan et al., 2025), RINE (Koutlis & Papadopoulos, 2024), and ClipDet (Cozzolino et al., 2024). We also consider Vision Transformer based methods including CoDE (Baraldi et al., 2024) and DRCT (Chen et al., 2024a), as well as training free detectors AEROBLADE (Ricker et al., 2024) and WaRPAD (Choi et al., 2025). Additionally, we compare against Corvi+ and Rajan+ (Rajan & Lee, 2025), which apply the Stay Positive algorithm to retrain only the last layer with non negative weight constraints. Detailed descriptions of each baseline are provided in Appendix D.2.

We apply DEAR to two representative detectors, Corvi and Rajan, and refer to the resulting methods as **DEAR-c** and **DEAR-r**, respectively. Both base detectors employ a ResNet 50 backbone trained on real images from LSUN and COCO, but differ in how synthetic training data is constructed: Corvi uses LDM generated images conditioned on text prompts, while Rajan uses VAE reconstructions of real images to ensure pixel level alignment between training pairs.

4.2. Robustness to Post Processing

A practical detector must maintain reliable performance when images undergo common transformations during storage and transmission. Following the prior evaluation protocol (Rajan & Lee, 2025), we construct a post processed test set by randomly applying compression, resizing, and color jittering to the synthetic images. Table 1 (bottom) summarizes the results. While most baselines suffer noticeable performance degradation under these perturbations, DEAR-c and DEAR-r exhibit substantially improved robustness. The performance gap between DEAR and base detectors becomes especially pronounced on challenging generators like FLUX, where post processing often removes the fragile artifacts that conventional detectors rely upon. This observation aligns with our hypothesis that pruning extreme RAD channels eliminates features vulnerable to such perturbations. Additional ablation studies comparing DEAR against detectors trained from scratch with inpainted data augmentation and analyzing sensitivity to the pruning ratio α are provided in Appendix E (Figures 8 and 9).

Table 1. **Comparison with SoTA detectors.** Methods are grouped by architecture type: CLIP-based, ViT-based (V), Training-free (T-Free), and CNN-based. AUC, R.Acc (real accuracy), and F.Acc (fake accuracy) for each generator. Top: original test images, bottom: post-processed. **Bold** indicates best, underline indicates second best. For training-free methods, we report only threshold-independent AUC ('-') since they lack a calibrated classification threshold.

	Original	**SD**			**MJ**			**KD**			**PG**			**PixArt**			**LCM**			**FLUX**			**Wuerst.**			**aMUSEd**			**Avg**		
	Detectors	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc
CLIP	UFD	66.7	95.1	17.9	54.8	95.1	13.0	70.4	95.1	23.9	72.5	95.1	20.4	73.3	95.1	20.2	72.1	95.1	23.4	21.5	95.1	0.1	93.8	95.1	74.5	95.3	95.1	77.5	68.9	95.1	30.1
	C2P-CLIP	75.3	93.0	40.9	70.8	93.0	18.7	77.9	93.0	30.6	68.3	93.0	7.3	59.6	93.0	2.2	70.5	93.0	17.6	49.9	93.0	8.0	87.1	93.0	52.0	98.8	93.0	98.3	73.1	93.0	30.6
	RINE	**100.0**	**100.0**	**100.0**	99.1	**100.0**	71.5	**100.0**	**100.0**	92.9	98.1	**100.0**	60.4	99.9	**100.0**	87.0	**100.0**	**100.0**	99.4	91.6	**100.0**	30.2	**100.0**	**100.0**	99.8	**100.0**	**100.0**	**100.0**	98.7	**100.0**	82.3
	ClipDet	78.1	85.4	47.4	79.3	85.4	50.9	79.2	85.4	50.8	82.1	85.4	53.0	84.6	85.4	59.8	71.0	85.4	34.7	84.9	85.4	64.9	93.0	85.4	87.0	91.2	85.4	80.5	82.6	85.4	58.8
V	CoDE	87.4	85.5	71.9	81.0	85.5	60.1	68.9	85.5	38.6	86.8	85.5	70.5	93.2	85.5	86.5	85.9	85.5	69.2	62.3	85.5	25.3	70.3	85.5	42.0	74.9	85.5	46.3	79.0	85.5	56.7
T-Free	AEROBLADE	98.9	-	-	99.4	-	-	98.4	-	-	70.9	-	-	94.3	-	-	99.0	-	-	78.3	-	-	81.9	-	-	93.0	-	-	90.5	-	-
	WARPAD	77.2	-	-	74.9	-	-	77.0	-	-	73.2	-	-	66.1	-	-	66.1	-	-	56.1	-	-	92.9	-	-	96.2	-	-	75.5	-	-
CNN	NPR	80.5	67.2	80.4	93.9	67.2	94.2	91.6	67.2	94.5	95.8	67.2	99.4	96.1	67.2	99.9	93.5	67.2	98.4	97.5	67.2	99.7	98.2	67.2	99.1	95.9	67.2	97.3	93.7	67.2	95.9
	SAFE	83.5	97.8	65.0	99.0	97.8	92.2	99.6	97.8	97.9	98.7	97.8	89.3	99.5	97.8	97.7	99.4	97.8	96.6	99.6	97.8	98.2	99.7	97.8	**100.0**	99.6	97.8	98.7	97.6	97.8	92.8
	AIDE	91.4	88.8	78.7	79.3	88.8	55.5	95.1	88.8	87.5	95.0	88.8	87.0	98.1	88.8	96.4	91.5	88.8	77.5	96.1	88.8	90.3	94.4	88.8	87.5	94.0	88.8	82.7	92.8	88.8	82.6
	FerretNet	90.8	95.8	73.0	99.2	95.8	97.1	99.4	95.8	97.5	99.9	95.8	99.8	99.9	95.8	99.9	99.9	95.8	**100.0**	99.8	95.8	98.9	99.7	95.8	99.2	99.8	95.8	99.1	98.7	95.8	96.0
	LaDeDa	90.9	97.8	71.7	97.7	97.8	83.7	98.4	97.8	85.9	97.4	97.8	71.1	97.5	97.8	66.4	99.0	97.8	88.6	99.7	97.8	97.4	99.8	97.8	99.2	99.8	97.8	98.4	97.8	97.8	84.7
	DRCT	98.6	96.0	94.7	96.8	96.0	84.8	99.8	96.0	99.1	90.6	96.0	64.4	75.5	96.0	46.3	96.8	96.0	84.8	82.9	96.0	49.2	97.0	96.0	79.6	94.1	96.0	65.0	92.4	96.0	74.2
	Corvi	100.0	99.9	99.5	99.9	99.9	97.7	**100.0**	99.9	100.0	99.9	99.9	91.3	**100.0**	99.9	**100.0**	100.0	99.9	99.7	87.8	99.9	21.5	**100.0**	99.9	**100.0**	98.9	99.9	68.5	98.5	99.9	86.5
	Corvi+	**100.0**	99.1	99.9	100.0	99.1	99.2	**100.0**	99.1	**100.0**	**100.0**	99.1	**100.0**	**100.0**	99.1	**100.0**	**100.0**	99.1	**100.0**	100.0	99.1	99.6	**100.0**	99.1	**100.0**	**100.0**	99.1	99.9	**100.0**	99.1	99.8
	DEAR-c	100.0	96.3	99.9	**100.0**	96.3	**100.0**	**100.0**	96.3	**100.0**	**100.0**	96.3	**100.0**	**100.0**	96.3	**100.0**	**100.0**	96.3	**100.0**	**100.0**	96.3	**100.0**	**100.0**	96.3	**100.0**	100.0	96.3	99.9	**100.0**	96.3	**100.0**
	Rajan	99.9	99.9	99.2	99.9	99.9	98.0	**100.0**	99.9	99.9	100.0	99.9	95.4	**100.0**	99.9	**100.0**	**100.0**	99.9	99.8	89.2	99.9	12.0	98.3	99.9	45.7	83.9	99.9	24.8	96.8	99.9	75.0
	Rajan+	**100.0**	99.9	**100.0**	100.0	99.9	99.0	**100.0**	99.9	**100.0**	**100.0**	99.9	**100.0**	**100.0**	99.9	**100.0**	**100.0**	99.9	**100.0**	97.4	99.9	29.6	99.2	99.9	71.0	99.9	99.9	93.9	99.6	99.9	88.2
	DEAR-r	99.9	97.4	99.8	99.9	97.4	99.8	**100.0**	97.4	**100.0**	**100.0**	97.4	**100.0**	100.0	97.4	**100.0**	**100.0**	97.4	**100.0**	99.5	97.4	96.6	100.0	97.4	99.9	99.5	97.4	96.6	99.9	97.4	99.2
	Post-proc.	**SD**			**MJ**			**KD**			**PG**			**PixArt**			**LCM**			**FLUX**			**Wuerst.**			**aMUSEd**			**Avg**		
	Detectors	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc
CLIP	UFD	59.2	93.2	13.1	42.9	93.2	8.2	56.4	93.2	14.5	53.3	93.2	7.5	54.5	93.2	9.5	57.2	93.2	12.9	25.0	93.2	0.6	73.7	93.2	33.2	85.3	93.2	50.7	56.4	93.2	16.7
	C2P-CLIP	60.9	96.5	14.3	68.5	96.5	9.7	70.6	96.5	14.0	74.6	96.5	9.6	70.6	96.5	3.2	61.0	96.5	6.2	35.5	96.5	0.6	71.7	96.5	21.1	93.0	96.5	62.5	67.4	96.5	15.7
	RINE	76.0	77.4	61.8	53.3	77.4	28.5	64.8	77.4	41.9	59.7	77.4	37.7	45.3	77.4	19.9	70.3	77.4	48.5	45.7	77.4	18.2	85.2	77.4	78.7	86.4	77.4	80.9	65.2	77.4	46.2
	ClipDet	71.9	76.6	52.8	75.2	76.6	59.4	75.4	76.6	59.6	76.9	76.6	60.6	79.3	76.6	66.2	76.7	76.6	61.2	79.8	76.6	66.7	89.2	76.6	85.2	87.7	76.6	82.3	79.1	76.6	66.0
V	CoDE	82.2	81.0	68.4	68.8	81.0	47.6	64.4	81.0	41.5	69.4	81.0	46.3	76.2	81.0	59.6	67.4	81.0	44.1	63.6	81.0	38.4	70.3	81.0	49.0	70.2	81.0	47.9	70.3	81.0	49.2
T-Free	AEROBLADE	56.1	-	-	78.0	-	-	63.8	-	-	56.7	-	-	58.2	-	-	42.5	-	-	54.6	-	-	56.6	-	-	67.0	-	-	59.3	-	-
	WARPAD	64.5	-	-	59.8	-	-	65.3	-	-	60.1	-	-	57.1	-	-	52.4	-	-	35.0	-	-	82.2	-	-	91.2	-	-	63.1	-	-
CNN	NPR	50.3	94.2	10.0	63.0	94.2	12.6	62.7	94.2	19.8	66.1	94.2	16.3	75.6	94.2	26.2	35.9	94.2	5.2	41.9	94.2	5.6	56.0	94.2	11.6	27.9	94.2	2.1	53.3	94.2	12.2
	SAFE	53.9	99.5	1.4	40.1	99.5	0.7	48.1	99.5	1.4	38.1	99.5	0.3	35.9	99.5	0.7	47.3	99.5	0.0	48.2	99.5	0.5	51.7	99.5	1.0	57.5	99.5	0.6	46.8	99.5	0.7
	AIDE	65.8	90.3	23.4	64.9	90.3	17.2	68.3	90.3	25.5	61.1	90.3	9.9	69.8	90.3	23.2	58.3	90.3	14.8	65.8	90.3	21.3	75.6	90.3	27.7	76.1	90.3	30.5	67.3	90.3	21.5
	FerretNet	56.5	97.7	2.7	44.7	97.7	1.1	59.1	97.7	2.6	64.9	97.7	3.6	73.1	97.7	6.0	49.9	97.7	1.1	58.7	97.7	1.7	65.6	97.7	1.7	60.9	97.7	0.3	59.3	97.7	2.3
	LaDeDa	45.4	99.8	1.5	57.1	99.8	0.2	61.6	99.8	0.9	64.6	99.8	0.0	73.8	99.8	0.1	33.6	99.8	0.0	42.1	99.8	0.1	51.9	99.8	0.2	33.9	99.8	0.0	51.6	99.8	0.3
	DRCT	90.5	86.2	80.6	79.5	86.2	59.4	86.7	86.2	74.2	71.7	86.2	46.6	72.1	86.2	52.2	73.3	86.2	41.4	67.3	86.2	39.0	77.1	86.2	50.6	85.6	86.2	65.7	78.2	86.2	56.6
	Corvi	91.1	97.3	78.4	80.0	97.3	50.2	80.7	97.3	57.0	66.8	97.3	18.5	75.7	97.3	63.4	81.1	97.3	46.0	64.2	97.3	21.1	82.8	97.3	59.3	90.7	97.3	35.8	79.2	97.3	47.7
	Corvi+	95.5	96.8	72.5	85.1	96.8	42.5	89.9	96.8	57.1	88.0	96.8	44.4	93.2	96.8	72.0	91.8	96.8	53.3	84.3	96.8	28.6	91.8	96.8	55.7	85.7	96.8	33.1	89.5	96.8	51.0
	DEAR-c	96.5	76.5	**96.2**	89.0	76.5	84.9	92.4	76.5	89.4	92.9	76.5	91.0	93.8	76.5	91.0	96.1	76.5	96.5	82.4	76.5	73.6	94.8	76.5	**93.3**	93.2	76.5	**96.0**	92.3	76.5	**90.2**
	Rajan	97.7	**99.9**	86.3	94.5	**99.9**	69.9	92.6	**99.9**	65.7	93.8	**99.9**	59.7	93.9	**99.9**	84.5	99.7	**99.9**	93.0	81.2	**99.9**	15.4	87.1	**99.9**	15.6	95.9	**99.9**	33.8	92.9	**99.9**	58.2
	Rajan+	96.1	99.1	83.4	86.1	99.1	64.4	91.6	99.1	71.6	89.3	99.1	68.7	91.4	99.1	79.9	97.3	99.1	84.8	84.7	99.1	27.8	82.1	99.1	17.4	96.6	99.1	63.0	90.6	99.1	62.3
	DEAR-r	**98.6**	95.1	95.7	**95.3**	95.1	**86.0**	**97.3**	95.1	**91.4**	**97.9**	95.1	**92.4**	**97.6**	95.1	**93.2**	**99.8**	95.1	**99.0**	**94.4**	95.1	**77.6**	**95.7**	95.1	82.7	**97.6**	95.1	87.8	**97.1**	95.1	89.5

4.3. Evaluation on the Wild Benchmarks

To assess detector performance under realistic deployment conditions, we evaluate on three challenging benchmarks collected from real world online sources: Chameleon (Yan et al., 2024), WildRF (Cavia et al., 2024), and LOKI (Ye et al., 2025). These datasets contain images that have undergone unknown post processing operations during upload and sharing on social media platforms and art communities, presenting a more stringent test than laboratory controlled settings. As shown in Table 2, DEAR-c and DEAR-r demonstrate strong performance across all three benchmarks, confirming that the robustness gains observed on controlled post processed data transfer to genuinely in the wild scenarios.

4.4. Alleviating Prediction Asymmetry and Improving Calibration

Prediction asymmetry is characterized primarily by low F.Acc, complicating the establishment of a universal decision threshold despite high threshold-independent metrics like AUC. For instance, Table 1 shows that while DRCT achieves an average AUC of 92.4, its fixed-threshold F.Acc drops to 74.2, whereas R.Acc remains near-ceiling (96.0). This divergence indicates a decision rule heavily biased toward the real class, resulting in a substantial miss rate for fake images under standard thresholds.

In contrast, DEAR enhances generalizability by maintaining stable fake score distributions across diverse generative models, thereby mitigating the need for dataset-specific calibration. Figure 5 illustrates this on original test images by comparing score distributions for Playground (4-channel) and FLUX (16-channel). Corvi's fake scores for FLUX shift dramatically below the decision threshold because the detector misinterprets the fine-grained details preserved by FLUX's 16-channel architecture as indicators of authenticity. By pruning these spurious dependencies, DEAR-c aligns score distributions across disparate architectures, ensuring reliable detection with a single threshold.

5. Related Work

Early detection methods focused on GAN generated images, exploiting spectral artifacts from upsampling opera-

Table 2. **Additional results on Chameleon, Loki, and WildRF.** Methods are grouped by architecture type: CLIP-based, ViT-based (V), Training-free (T-Free), and CNN-based. **Bold** indicates best, underline indicates second best. For training-free methods, we report only threshold-independent AUC ('-') since they lack a calibrated classification threshold.

	Original	Chameleon			Loki			WildRF			Avg		
	Detectors	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc
CLIP	UFD	39.6	95.1	2.4	71.4	95.1	32.3	69.2	95.1	34.9	60.1	95.1	23.2
	C2P-CLIP	23.5	93.0	2.9	68.6	93.0	36.3	73.5	93.0	35.7	55.2	93.0	24.9
	RINE	**96.7**	**100.0**	28.5	**96.4**	**100.0**	59.7	**96.7**	**100.0**	46.1	**96.6**	**100.0**	44.7
	ClipDet	64.2	85.4	26.9	76.7	85.4	45.1	82.7	85.4	58.9	74.5	85.4	43.6
V	CoDE	85.8	85.5	69.8	69.9	85.5	47.7	77.3	85.5	55.8	77.7	85.5	57.8
T-Free	AEROBLADE	48.8	-	-	70.1	-	-	73.8	-	-	64.2	-	-
	WARPAD	37.7	-	-	70.3	-	-	77.4	-	-	61.8	-	-
CNN	NPR	18.8	67.2	4.0	74.4	67.2	**65.1**	70.4	67.2	59.7	54.5	67.2	42.9
	SAFE	50.8	97.8	2.8	76.2	97.8	59.8	62.6	97.8	31.6	63.2	97.8	31.4
	AIDE	69.5	88.8	21.7	75.7	88.8	56.5	76.4	88.8	44.9	73.9	88.8	41.1
	FerretNet	49.2	95.8	3.1	78.2	95.8	59.8	69.3	95.8	32.8	65.6	95.8	31.9
	LaDeDa	17.9	97.8	2.7	70.5	97.8	58.0	66.5	97.8	34.3	51.6	97.8	31.6
	DRCT	85.7	96.0	47.5	86.6	96.0	56.6	85.1	96.0	57.6	85.8	96.0	53.9
	Corvi	84.0	99.9	17.8	79.4	99.9	49.4	85.5	99.9	56.4	83.0	99.9	41.2
	Corvi+	88.4	99.1	18.1	85.5	99.1	56.3	81.1	99.1	54.0	85.0	99.1	42.8
	DEAR-c	96.1	96.3	**80.3**	82.4	96.3	63.5	85.7	96.3	**69.0**	88.1	96.3	**70.9**
	Rajan	95.7	99.9	26.2	75.2	99.9	44.0	85.7	99.9	57.6	85.5	99.9	42.6
	Rajan+	91.0	99.9	26.5	81.6	99.9	46.9	83.6	99.9	61.1	85.4	99.9	44.8
	DEAR-r	95.5	97.4	70.0	82.3	97.4	56.8	87.1	97.4	66.1	88.3	97.4	64.3

	Post-proc.	Chameleon			Loki			WildRF			Avg		
	Detectors	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc	AUC	R.Acc	F.Acc
CLIP	UFD	36.6	93.2	2.7	63.9	93.2	24.7	56.1	93.2	20.2	52.2	93.2	15.8
	C2P-CLIP	25.7	96.5	0.6	55.4	96.5	11.6	71.1	96.5	18.9	50.7	96.5	10.4
	RINE	51.5	77.4	24.2	66.3	77.4	44.2	61.0	77.4	38.0	59.6	77.4	35.5
	ClipDet	72.2	76.6	54.2	69.0	76.6	47.0	78.9	76.6	67.8	73.4	76.6	56.4
V	CoDE	69.0	81.0	44.1	65.9	81.0	42.3	70.2	81.0	51.2	68.4	81.0	45.8
T-Free	AEROBLADE	57.9	-	-	49.1	-	-	63.8	-	-	56.9	-	-
	WARPAD	45.6	-	-	63.1	-	-	70.5	-	-	59.7	-	-
CNN	NPR	36.6	94.2	1.0	55.9	94.2	8.3	59.7	94.2	12.8	50.8	94.2	7.4
	SAFE	50.0	99.5	0.2	41.5	99.5	0.6	45.2	99.5	1.0	45.6	99.5	0.6
	AIDE	67.4	90.3	19.4	53.4	90.3	14.7	67.1	90.3	22.7	62.6	90.3	18.9
	FerretNet	47.4	97.7	0.5	47.0	97.7	1.5	54.7	97.7	2.7	49.7	97.7	1.6
	LaDeDa	38.2	99.8	0.1	43.0	99.8	0.8	57.5	99.8	0.7	46.2	99.8	0.5
	DRCT	70.1	86.2	42.3	77.0	86.2	53.8	74.8	86.2	52.6	74.0	86.2	49.6
	Corvi	61.1	97.3	16.4	76.0	97.3	44.2	73.4	97.3	36.9	70.2	97.3	32.5
	Corvi+	**78.7**	96.8	14.1	77.8	96.8	39.6	74.0	96.8	33.0	76.8	96.8	28.9
	DEAR-c	73.5	76.5	**58.8**	79.4	76.5	**70.1**	80.4	76.5	**71.0**	77.7	76.5	**66.6**
	Rajan	71.0	**99.9**	9.4	78.4	**99.9**	41.7	83.6	**99.9**	43.8	77.7	**99.9**	31.6
	Rajan+	68.2	99.1	15.4	**80.8**	99.1	45.9	75.9	99.1	44.6	75.0	99.1	35.3
	DEAR-r	76.7	95.1	40.6	79.8	95.1	56.6	**85.4**	95.1	63.8	**80.6**	95.1	53.7

tions (Tan et al., 2024b) and frequency domain inconsistencies inherent to the generation process. With the emergence of diffusion models (Ho et al., 2020; Rombach et al., 2022), researchers developed specialized detectors targeting diffusion specific signatures. Corvi et al. (Corvi et al., 2023) demonstrated that a ResNet 50 trained on LDM generated images with careful preprocessing can generalize to unseen diffusion architectures. Rajan et al. (Rajan et al., 2025) further improved generalization by constructing aligned training pairs through VAE reconstruction, ensuring that real and synthetic images differ only in decoder artifacts. Beyond CNN based approaches, CLIP based methods (Ojha et al., 2023; Koutlis & Papadopoulos, 2024; Tan et al., 2025; Cozzolino et al., 2024) leverage rich semantic representations learned from web scale data through linear probing or prompt based adaptation. DINOv2 based approaches (Guillaro et al., 2025; Chen et al., 2025) similarly benefit from self supervised pretraining, with B Free (Guillaro et al., 2025) proposing a bias free training paradigm using self conditioned reconstructions. Training free methods such as

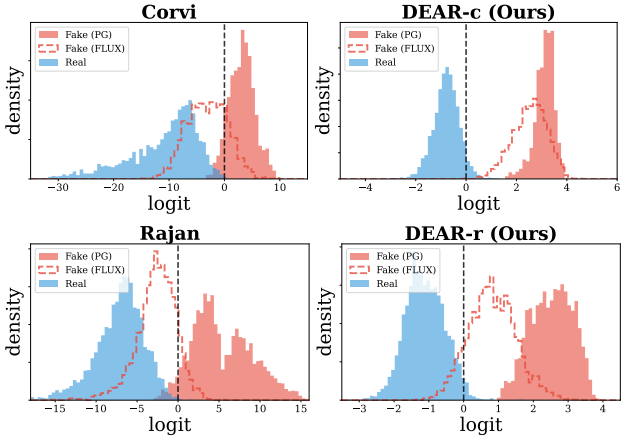

Figure 5. **Score distribution shift.** Baseline detectors (left) exhibit severe distribution shift on unseen generators, while DEAR variants (right) maintain stable fake score distributions above the decision threshold.

AEROBLADE (Ricker et al., 2024) and WaRPAD (Choi et al., 2025) offer deployment flexibility by analyzing reconstruction errors or feature stability without detector specific optimization.

Most relevant is Stay Positive (Rajan & Lee, 2025), which constrains final layer weights to be non negative, effectively ignoring features with negative contributions that correlate with real images. While effective at eliminating real associated spurious features, this approach operates blindly without identifying which channels are problematic. DEAR instead leverages inpainted images to explicitly measure each channel's alignment with generated regions. Crucially, our analysis reveals that features at *both* extremes of the alignment spectrum are structurally fragile. By pruning channels at both extremes, DEAR removes real associated artifacts (e.g., compression signatures) and fake associated artifacts (e.g., generator specific fingerprints), achieving more comprehensive robustness than single sided constraints.

6. Conclusion

We proposed DEAR to address prediction asymmetry, a phenomenon where detectors exhibit high accuracy on real images but experience significant performance drops on generated content. By utilizing inpainted images to quantify feature alignment, our framework identifies and eliminates the spurious channels driving this discrepancy. Our analysis reveals that features at both extremes of the alignment spectrum are susceptible to degradation, and our bilateral pruning strategy effectively mitigates these vulnerabilities while preserving robust forensic signals. By integrating diagnostic dissection with selective refinement, DEAR establishes a principled framework for consistent detection performance across unseen generators and post-processing scenarios.

Impact Statement

This paper advances AI-generated image detection to mitigate societal risks such as misinformation, deepfake misuse, and copyright infringement posed by increasingly powerful generative models. However, insights from this work may also inform the development of generation techniques designed to evade detection. Therefore, we emphasize the need for continuous community efforts to monitor misuse and adapt detection methods to evolving generative technologies.

References

Midjourney. https://www.midjourney.com/.

Baraldi, L., Cocchi, F., Cornia, M., Baraldi, L., Nicolosi, A., and Cucchiara, R. Contrasting deepfakes diffusion via contrastive learning and global-local similarities. In *European Conference on Computer Vision*, pp. 199–216. Springer, 2024.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Bau, D., Zhu, J.-Y., Strobelt, H., Zhou, B., Tenenbaum, J. B., Freeman, W. T., and Torralba, A. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018.

Cavia, B., Horwitz, E., Reiss, T., and Hoshen, Y. Real-time deepfake detection in the real-world. *arXiv preprint arXiv:2406.09398*, 2024.

Chan, T. F. and Vese, L. A. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

Chen, B., Zeng, J., Yang, J., and Yang, R. Drct: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *Forty-first International Conference on Machine Learning*, 2024a.

Chen, J., YU, J., GE, C., Yao, L., Xie, E., Wang, Z., Kwok, J., Luo, P., Lu, H., and Li, Z. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024b.

Chen, R., Xi, J., Yan, Z., Zhang, K.-Y., Wu, S., Xie, J., Chen, X., Xu, L., Guan, I., Yao, T., et al. Dual data alignment makes ai-generated image detector easier generalizable. *arXiv preprint arXiv:2505.14359*, 2025.

Choi, S., Lee, H., and Lee, M. Training-free detection of ai-generated images via cropping robustness. *arXiv preprint arXiv:2511.14030*, 2025.

Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., and Verdoliva, L. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Cozzolino, D., Poggi, G., Corvi, R., Nießner, M., and Verdoliva, L. Raising the bar of ai-generated image detection with clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4356–4366, 2024.

Desai, K., Kaul, G., Aysola, Z., and Johnson, J. Redcaps: Web-curated image-text data created by the people, for the people. *arXiv preprint arXiv:2111.11431*, 2021.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Grommelt, P., Weiss, L., Pfreundt, F.-J., and Keuper, J. Fake or jpeg? revealing common biases in generated image detection datasets. In *ECCV*, 2024.

Guillaro, F., Zingarini, G., Usman, B., Sud, A., Cozzolino, D., and Verdoliva, L. A bias-free training paradigm for more general ai-generated image detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18685–18694, 2025.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Kashiani, H., Talemi, N., et al. Freqdebias: Towards generalizable deepfake detection via consistency-driven frequency debiasing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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

Labs, B. F. Flux. https://github.com/black-forest-labs/flux, 2024.

Li, D., Kamko, A., Akhgari, E., Sabet, A., Xu, L., and Doshi, S. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024.

Li, O., Cai, J., Hao, Y., Jiang, X., Hu, Y., and Feng, F. Improving synthetic image detection towards generalization: An image transformation perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2405–2414, 2025a.

Li, Z., Yan, J., He, Z., Zeng, K., Jiang, W., Xiong, L., and Fu, Z. Is artificial intelligence generated image detection a solved problem? *arXiv preprint arXiv:2505.12335*, 2025b.

Liang, S., Liu, J., Chen, R., and Guan, Q. Ferretnet: Efficient synthetic image detection via local pixel dependencies. *arXiv preprint arXiv:2509.20890*, 2025.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Luo, S., Tan, Y., Huang, L., Li, J., and Zhao, H. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Ma, L., Yan, Z., Xu, J., et al. From specificity to generality: Revisiting generalizable artifacts in detecting face deepfakes. *arXiv preprint arXiv:2504.04827*, 2025.

Ojha, U., Li, Y., and Lee, Y. J. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24480–24489, 2023.

Patil, S., Berman, W., Rombach, R., and von Platen, P. amused: An open muse reproduction. *arXiv preprint arXiv:2401.01808*, 2024.

Pernias, P., Rampas, D., and Aubreville, M. Wuerstchen: Efficient pretraining of text-to-image models. 2023.

Rajan, A. S. and Lee, Y. J. Stay-positive: A case for ignoring real image features in fake image detection. *arXiv preprint arXiv:2502.07778*, 2025.

Rajan, A. S., Ojha, U., Schloesser, J., and Lee, Y. J. Aligned datasets improve detection of latent diffusion-generated images. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=doBkiqESYq.

Razzhigaev, A., Shakhmatov, A., Maltseva, A., Arkhipkin, V., Pavlov, I., Ryabov, I., Kuts, A., Panchenko, A., Kuznetsov, A., and Dimitrov, D. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion. *arXiv preprint arXiv:2310.03502*, 2023.

Ricker, J., Lukovnikov, D., and Fischer, A. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9130–9140, 2024.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Tan, C., Zhao, Y., Wei, S., Gu, G., Liu, P., and Wei, Y. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024a.

Tan, C., Zhao, Y., Wei, S., Gu, G., Liu, P., and Wei, Y. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28130–28139, 2024b.

Tan, C., Tao, R., Liu, H., Gu, G., Wu, B., Zhao, Y., and Wei, Y. C2p-clip: Injecting category common prompt in clip to enhance generalization in deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Tousi, A., Jeong, H., Han, J., Choi, H., and Choi, J. Automatic correction of internal units in generative neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7932–7940, 2021.

Wang, S.-Y., Wang, O., Zhang, R., Owens, A., and Efros, A. A. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8695–8704, 2020.

Yan, S., Li, O., Cai, J., Hao, Y., Jiang, X., Hu, Y., and Xie, W. A sanity check for ai-generated image detection. *arXiv preprint arXiv:2406.19435*, 2024.

Yan, Z., Wang, J., Wang, Z., Jin, P., Zhang, K.-Y., Chen, S., Yao, T., Ding, S., Wu, B., and Yuan, L. Effort: Efficient orthogonal modeling for generalizable ai-generated image detection. In *International Conference on Machine Learning (ICML)*, 2025.

Ye, J., Zhou, B., Huang, Z., Zhang, J., Bai, T., Kang, H., He, J., Lin, H., Wang, Z., Wu, T., et al. Loki: A comprehensive synthetic data detection benchmark using large multimodal models. *ICLR*, 2025.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

A. Problem Formulation and Notations

We address AI-generated image detection as a binary classification problem. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ denote a dataset where $\mathbf{x}_i \in \mathbb{R}^{H\times W\times C}$ represents an image and $y_i \in \{0, 1\}$ represents the label, with $y = 0$ indicating real images and $y = 1$ indicating AI-generated (fake) images. The objective is to train a detector that accurately estimates the probability $p(y = 1|\mathbf{x})$.

We formulate the detector as a composition of two distinct modules: a deep feature backbone $f_\theta(\cdot)$ and a linear classification head $h_\phi(\cdot)$. Specifically, the backbone parameterized by θ maps an input image $\mathbf{x}$ to a high-dimensional feature embedding $\mathbf{z} = f_\theta(\mathbf{x}) \in \mathbb{R}^d$. The classification head parameterized by $\phi = \{\mathbf{w}, b\}$ then projects this embedding to a scalar probability score via a sigmoid activation:

$$p(\mathbf{x}) = h_\phi(\mathbf{z}) = \sigma(\mathbf{w}^\top \mathbf{z} + b), \tag{7}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ denote the weight vector and bias term, respectively. This decomposition allows us to analyze and manipulate the feature extraction (f_θ) and decision-making (h_ϕ) processes independently.

Table 3. **Table of notation**

Notation	Description
Input Space & Detector	
$\mathbf{x}$	Input image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$.
y	Binary label, $y = 0$ for real and $y = 1$ for AI generated (fake).
f_θ	Feature backbone parameterized by θ, mapping image to feature maps.
h_ϕ	Linear classification head parameterized by $\phi = \{\mathbf{w}, b\}$.
$\mathbf{w}, b$	Weight vector and bias of the linear classifier.
$\mathbf{z}$	Feature embedding after global pooling, $\mathbf{z} = \text{Pool}(f_\theta(\mathbf{x})) \in \mathbb{R}^D$.
$\mathcal{D}_{\text{train}}$	Training dataset of real and fake image pairs.
Inpainting & Feature Maps	
$\mathbf{x}_{\text{real}}$	Original real image used for inpainting.
$\mathbf{x}_{\text{gen}}$	AI generated content synthesized by inpainting model.
$\mathbf{x}_{\text{inpaint}}$	Composite inpainted image with both real and generated regions.
$\mathbf{M}$	Binary inpaint mask $\mathbf{M} \in \{0, 1\}^{H\times W}$, where 1 indicates generated region.
$\mathcal{D}_{\text{inpaint}}$	Diagnostic dataset of inpainted images with ground truth masks.
$\mathbf{F}$	Feature tensor from the final conv layer, $\mathbf{F} = f_\theta(\mathbf{x}) \in \mathbb{R}^{D\times h\times w}$.
$\mathbf{F}_k$	Activation map of the kth channel, $\mathbf{F}_k \in \mathbb{R}^{h\times w}$.
D	Total number of channels in the final convolutional layer.
Regional Activation Discrepancy (RAD)	
Ω_{in}	Set of spatial positions in the inpainted (generated) region.
Ω_{bg}	Set of spatial positions in the background (real) region.
$\mu_{\text{in}}^{(k)}$	Mean activation of channel k within the inpainted region.
$\mu_{\text{bg}}^{(k)}$	Mean activation of channel k within the background region.
S_k	RAD value for channel k, defined as $S_k = \mu_{\text{in}}^{(k)} - \mu_{\text{bg}}^{(k)}$.
Bilateral Pruning	
$\alpha_{\text{low}}, \alpha_{\text{high}}$	Lower and upper percentile parameters for pruning.
$\tau_{\text{low}}, \tau_{\text{high}}$	Computed thresholds based on empirical RAD distribution.
$\mathbf{m}$	Binary pruning mask $\mathbf{m} \in \{0, 1\}^D$ indicating retained channels.
$\tilde{\mathbf{F}}$	Pruned feature map after applying mask, $\tilde{\mathbf{F}} = \mathbf{m} \odot \mathbf{F}$.

B. Limitations and Discussion

While DEAR provides a simple yet effective approach for improving detector robustness, it exhibits certain limitations. First, our analysis focuses exclusively on the final convolutional layer before global average pooling, as this layer provides the most semantically meaningful representation for classification. However, lower layers may capture complementary low

Algorithm 1 DEAR: Dissect and Prune

Input: Pretrained detector (f_θ, h_ϕ), Diagnostic data $\mathcal{D}_{\text{inpaint}}$, Train data $\mathcal{D}_{\text{train}}$, percentiles $\alpha_{\text{low}}, \alpha_{\text{high}}$
Output: Robust detector f^*_{DEAR}
`// Stage 1: Dissection (Compute RAD)`
$S \leftarrow \mathbf{0} \in \mathbb{R}^D$
for batch $(\mathbf{x}, \mathbf{M})$ in $\mathcal{D}_{\text{inpaint}}$ **do**
  $\mathbf{F} \leftarrow f_\theta(\mathbf{x})$
  Compute batch RAD using Eq. 3 and accumulate to S
end for
$S \leftarrow S/|\mathcal{D}_{\text{inpaint}}|$
`// Stage 2: Bilateral Pruning`
$\tau_{\text{low}} \leftarrow \text{Percentile}(S, \alpha_{\text{low}})$
$\tau_{\text{high}} \leftarrow \text{Percentile}(S, 100 - \alpha_{\text{high}})$
$\mathbf{m} \leftarrow \mathbb{1}[\tau_{\text{low}} \leq S \leq \tau_{\text{high}}]$
`// Stage 3: Classifier Refinement`
Freeze backbone parameters θ
Reinitialize classifier parameters ϕ
for batch $(\mathbf{x}, y)$ in $\mathcal{D}_{\text{train}}$ **do**
  $\mathbf{F} \leftarrow f_\theta(\mathbf{x})$
  `// Apply pruning mask`
  $\tilde{\mathbf{F}} \leftarrow \mathbf{m} \odot \mathbf{F}$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{BCE}}(h_\phi(\text{Pool}(\tilde{\mathbf{F}})), y)$
  Update ϕ to minimize $\mathcal{L}$
end for
return $(f_\theta, h_\phi, \mathbf{m})$

level artifacts such as pixel level noise patterns or frequency domain irregularities that could further enhance detection. Extending the dissection framework to analyze and selectively prune features across multiple layers remains an important direction for future research. Second, DEAR operates as a post hoc enhancement method that dissects and refines pretrained feature representations. An interesting alternative would be to integrate the alignment based feature selection directly into the training process, enabling the detector to learn robust features from scratch through online dissection. Exploring how to incorporate RAD based supervision during training, rather than applying it retrospectively, represents an intriguing future research direction.

C. Implementation Details

Diagnostic data generation. We construct the diagnostic inpaint dataset using 90,000 images from the LSUN dataset. For each image, we generate a random rectangular mask with area ratio uniformly sampled from $[0.02, 0.2]$ and aspect ratio from $[0.5, 2.0]$. The mask position is randomly placed within the image boundaries. We employ the Stable Diffusion 1.5 inpainting model[1] to synthesize content for the masked regions. To prevent the detector from exploiting trivial edge discontinuities, we apply Gaussian blur to the mask boundaries with a blur factor of 33 before compositing the inpainted content with the original background. All images are processed at 512×512 resolution during inpainting and subsequently resized back to their original dimensions. Figure 6 shows representative examples from the diagnostic dataset.

Dissection and pruning. For RAD computation, we extract feature maps from the final convolutional layer before global average pooling, which contains $D = 2048$ channels for the ResNet 50 backbone. We compute the mean activation difference between inpainted and background regions as specified in Eq. 3, averaging scores across 100 batches (approximately 6,400 samples) from the diagnostic dataset. The bilateral pruning thresholds are determined by percentile parameters α_{low} and α_{high}, which we tune from the set $\{0.05, 0.1, 0.2, 0.3\}$. We apply hard gating, setting the gate value to 1 for retained channels and 0 for pruned channels. A full list of the hyperparameters is reported in Table 4. Figure 7 provides additional examples of

[1] https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-inpainting

the feature alignment visualization across diverse images.

Table 4. **Hyperparameters for DEAR.**

Component	Hyperparameter	Value
Diagnostic Data Generation		
	Source Images	LSUN (90,000 images)
	Mask Area Ratio	$[0.02, 0.2]$
	Mask Aspect Ratio	$[0.5, 2.0]$
	Inpainting Model	Stable Diffusion 1.5
	Mask Blur Factor	33
	Processing Resolution	512×512
Dissection (RAD Computation)		
	Feature Layer	Final conv (layer4)
	Number of Channels (D)	2048
	Score Type	RAD
	Max Scoring Batches	100
Bilateral Pruning		
	$\alpha_{\text{low}}, \alpha_{\text{high}}$	Tuned from $\{0.05, 0.1, 0.2, 0.3\}$
Classifier Refinement		
	Learning Rate	1×10^{-4}
	Optimizer	Adam ($\beta_1 = 0.9$)
	Batch Size (Main / Inpaint)	64 / 64
	Early Stopping Patience	5 epochs
	Early Stopping Delta	0.001

Classifier refinement. After applying the pruning mask, we freeze all backbone parameters and reinitialize the final linear classifier weights to zero. We train the classifier using the Adam optimizer with a learning rate of 1×10^{-4} and $\beta_1 = 0.9$. The training combines both the original training data and the diagnostic inpaint data, with batch sizes of 64 for each. We apply early stopping with a patience of 5 epochs and a minimum improvement threshold of 0.001 in validation accuracy. When validation accuracy stagnates, the learning rate is reduced by a factor of 10, and training terminates when the learning rate falls below 10^{-6}.

Data augmentation. We adopt the data augmentation pipeline established in prior work (Corvi et al., 2023; Wang et al., 2020). During training, we extract random crops of 96×96 pixels from the input images. The augmentation scheme includes random JPEG compression with quality factors sampled from $[30, 100]$, Gaussian blur with sigma in $[0.0, 3.0]$, grayscale conversion, cutout, additive Gaussian noise, and random resized cropping to ensure scale invariance.

Inference. During inference, we do not apply any cropping or resizing to the input images. This is possible because the ResNet 50 backbone employs a spatially adaptive average pooling layer before the final classifier, allowing it to process images of arbitrary resolutions.

Computational resources and runtimes. All experiments were conducted using a single NVIDIA H200 GPU. The approximate execution times for each component are as follows:

- Diagnostic inpaint data generation: 11 hours
- DEAR-c / DEAR-r classifier refinement: approximately 3 hours

These times are per model training instance or data generation run and may vary slightly depending on the specific environment.

D. Experimental Setup Details

D.1. Details of Datasets

We describe the datasets used for training and evaluation in our experiments.

Training Data. Following the established protocol in prior work (Corvi et al., 2023; Rajan et al., 2025), we construct our training set using real images from two widely-used sources: LSUN (Yu et al., 2015) and COCO (Lin et al., 2014). The LSUN dataset provides diverse indoor and outdoor scene categories, while COCO offers images with rich object annotations across various contexts. For synthetic training images, we consider two generation strategies depending on the baseline detector:

- **Corvi-style training**: Fake images are generated by Latent Diffusion Models (LDM) (Rombach et al., 2022) using text prompts that correspond to the semantic content of the real images. This approach maintains content alignment through text-based conditioning.
- **Rajan-style training**: Fake images are produced by passing real images through the VAE encoder-decoder of LDM, creating reconstructions that preserve the exact visual content while introducing only the artifacts from the autoencoder pipeline. This alignment strategy ensures that real and synthetic pairs differ solely in decoder-induced artifacts, eliminating spurious correlations from semantic or resolution mismatches.

Evaluation Data. For evaluation, we use real images from the Redcaps dataset (Desai et al., 2021), which contains diverse web-crawled images with natural variations in content, resolution, and compression. The synthetic evaluation set encompasses images from multiple generative models and in-the-wild benchmarks, organized into the following categories:

Generator-specific evaluation. We evaluate on images from nine different generative models spanning various architectures:

- **Diffusion-based models**: Stable Diffusion (SD) (Rombach et al., 2022), FLUX (Labs, 2024), Kandinsky (Razzhigaev et al., 2023), Playground (Li et al., 2024), and PixArt (Chen et al., 2024b).
- **Latent consistency models**: LCM (Luo et al., 2023), which enables fast sampling through consistency distillation.
- **Alternative architectures**: Wuerstchen (Pernias et al., 2023), which employs a two-stage latent diffusion approach, and aMUSEd (Patil et al., 2024), an autoregressive masked image model.
- **Commercial systems**: Midjourney (mid), a proprietary text-to-image service known for high aesthetic quality.

In-the-wild evaluation. To assess detector performance under realistic deployment conditions, we additionally evaluate on three challenging benchmarks collected from real-world online sources:

- **Chameleon** (Yan et al., 2024): A curated collection of challenging AI-generated images gathered from online AI art communities such as ArtStation, Civitai, and Liblib. These images are specifically selected to be difficult for human perception and represent the current landscape of high-quality synthetic content shared publicly.
- **WildRF** (Cavia et al., 2024): Images collected from social media platforms including Reddit, Facebook, and X (formerly Twitter). Fake images are retrieved using common hashtags such as #aiart, #aigenerated, and #fakephoto, while real images are sourced using tags like #photography and #realphoto. This dataset reflects the types of images that detectors would encounter when deployed for content moderation.
- **LOKI** (Ye et al., 2025): A comprehensive synthetic data detection benchmark originally designed for evaluating large multimodal models across five modalities. We utilize its image subset, which includes diverse generation sources and specialized domains such as satellite imagery and medical images, requiring both perceptual and knowledge-based reasoning for detection.

These in-the-wild benchmarks are particularly valuable because they contain images that have undergone unknown post-processing operations (compression, resizing, filtering) during upload and sharing, presenting a more realistic challenge than laboratory-controlled test sets.

D.2. Baseline Methods

We compare our method against a comprehensive set of state-of-the-art AI-generated image detectors. Below we provide a brief description of each baseline, categorized by their underlying architecture and approach.

CNN-based Detectors.

- **Corvi** (Corvi et al., 2023): A ResNet-50 based detector trained on real images from MSCOCO and LSUN, paired with fake images generated by LDM using text prompts corresponding to the real data. The method preserves low-level forensic traces by removing the downsampling operation in the first convolutional layer and employs aggressive data augmentation to improve robustness.
- **Rajan** (Rajan et al., 2025): Uses the same ResNet-50 architecture and real images as Corvi, but trains on fake images generated by VAE reconstructions of the real images rather than prompt-based generation. This alignment ensures that real and fake training samples differ only in decoder artifacts, reducing spurious correlations from content or resolution mismatches.
- **Corvi+** (Rajan & Lee, 2025): Applies the Stay-Positive algorithm to the Corvi detector. The key insight is that an image should be classified as fake only if it contains artifacts from the generative model, while the absence of such artifacts indicates a real image. The method freezes the backbone of a pre-trained Corvi detector and retrains only the last linear layer with a non-negativity constraint on weights. By clamping weights to stay positive, the detector ignores features associated with real images (which would have negative weights) and focuses exclusively on fake artifacts. This eliminates spurious correlations from post-processing artifacts (e.g., WEBP compression) that the detector may have incorrectly associated with real images.
- **Rajan+** (Rajan & Lee, 2025): Applies the same Stay-Positive algorithm to the Rajan detector. Combined with Rajan's aligned training data (VAE reconstructions), this approach achieves improved robustness to post-processing operations and better generalization to newer generators within the same family (e.g., from LDM to FLUX). The method also enables effective detection of partially inpainted images, where conventional detectors struggle due to their reliance on real image features.
- **NPR** (Tan et al., 2024b): This approach feeds a ResNet-50 classifier with a residual image computed as the difference between the original image and its bilinearly interpolated version. The method exploits artifacts introduced by upsampling operations that are common across various generative architectures.
- **SAFE** (Li et al., 2025a): A lightweight detector (1.44M parameters) with ResNet-50 backbone that addresses training biases through three strategies: (1) replacing downsampling with crop operations to avoid artifact distortion, (2) adding ColorJitter and RandomRotation augmentation to reduce overfitting to color and semantic biases, and (3) employing patch-based random masking to enforce local awareness.
- **AIDE** (Yan et al., 2024): AI-generated Image DEtector with Hybrid Features that leverages multiple experts to simultaneously extract visual artifacts and noise patterns. The method uses a ConvNeXt backbone and introduces the Chameleon dataset containing challenging AI-generated images curated from online sources.
- **FerretNet** (Liang et al., 2025): Extracts local pixel dependency features through median filtering operations that capture neighborhood relationships based on Markov Random Field theory. The method computes residuals between the original image and its median-reconstructed version. The lightweight architecture (1.1M parameters) with depthwise separable and dilated convolutions enables efficient real-time detection.
- **LaDeDa** (Cavia et al., 2024): Locally Aware Deepfake Detection Algorithm that operates on single 9×9 image patches and outputs patch-level deepfake scores, which are then pooled to produce the final image score. The method can be distilled to Tiny-LaDeDa with only 4 convolutional layers for efficient edge deployment.
- **DRCT** (Chen et al., 2024a): Uses a ConvNeXt backbone trained with diffusion reconstruction contrastive learning. The method generates semantically aligned training pairs by reconstructing both real and fake images through DDIM inversion.

CLIP-based Detectors.

- **UFD** (Ojha et al., 2023): Leverages the pre-trained CLIP ViT-L/14 vision encoder with a simple linear probing approach. The frozen CLIP features are shown to contain discriminative information for distinguishing real from generated images across diverse generators.
- **C2P-CLIP** (Tan et al., 2025): Injects category-common prompts into the CLIP framework to enhance generalization in deepfake detection. The method learns prompt embeddings that encode real and fake concepts while keeping the vision encoder frozen.
- **RINE** (Koutlis & Papadopoulos, 2024): Extracts features from intermediate transformer blocks of a frozen CLIP encoder and employs a Trainable Importance Estimator module to learn adaptive weights for aggregating multi-layer representations.
- **ClipDet** (Cozzolino et al., 2024): Systematically explores various CLIP backbones pretrained on different datasets and proposes an ensemble approach that aggregates predictions from multiple CLIP variants to improve robustness across diverse generators.

Vision Transformer-based Detectors.

- **CoDE** (Baraldi et al., 2024): Learns contrastive embeddings by training a ViT encoder to distinguish global and local image similarities. The method combines multiple classification heads including linear, SVM, and k-NN classifiers for robust detection.

Training-free Detectors.

- **AEROBLADE** (Ricker et al., 2024): Detects AI-generated images by measuring the LPIPS reconstruction error when images are passed through multiple LDM autoencoders. Generated images exhibit lower reconstruction error since they were originally produced by similar encoder-decoder pipelines.
- **WaRPAD** (Choi et al., 2025): Measures the sensitivity of DINOv2 features to high-frequency perturbations extracted via Haar wavelet decomposition. The method exploits the observation that real images, trained with RandomResizedCrop augmentation, exhibit higher feature stability under such perturbations compared to AI-generated images.

For fair comparison, we follow the official implementations and pretrained checkpoints released by the authors.

E. Additional Results

We provide extended ablation studies comparing DEAR variants against baseline methods and their extensions. In addition to the baselines described in Appendix D.2, we introduce two additional variants to investigate whether simply augmenting training data with inpainted images can achieve similar benefits to DEAR:

- **Corvi-inpaint**: Trains a ResNet-50 detector from scratch using the standard Corvi training protocol, but augments the training set with our diagnostic inpainted images as additional fake samples.
- **Rajan-inpaint**: Similarly trains from scratch with the Rajan protocol, augmented with inpainted images.

These variants test whether the performance gains of DEAR stem merely from exposure to inpainted data during training, or whether the dissection and pruning mechanism provides orthogonal benefits.

Analysis of results. Figure 8 summarizes the average AUC across all nine generators. Several observations emerge from this comparison. First, DEAR variants consistently match or outperform all baselines on both original and post-processed images, with the performance gap being most pronounced under post-processing. Second, and perhaps more interestingly, Corvi-inpaint and Rajan-inpaint show limited improvement over their base detectors despite being trained with inpainted data. In some cases, such as Rajan-inpaint on post-processed images, performance actually degrades compared to the original Rajan detector. This suggests that naively augmenting training data with inpainted images does not effectively teach the

detector to focus on robust features. In contrast, DEAR's explicit dissection and pruning mechanism successfully identifies and removes the problematic channels, achieving substantial gains without retraining the backbone from scratch.

Figure 9 provides a per-generator breakdown of these results. The detailed view reveals that DEAR's improvements are consistent across generators. We use symmetric pruning with $\alpha_{\text{low}} = \alpha_{\text{high}} = \alpha$ throughout our experiments. The stability across different pruning ratios ($\alpha \in \{0.1, 0.2, 0.3\}$) further demonstrates the robustness of our approach to hyperparameter choices.

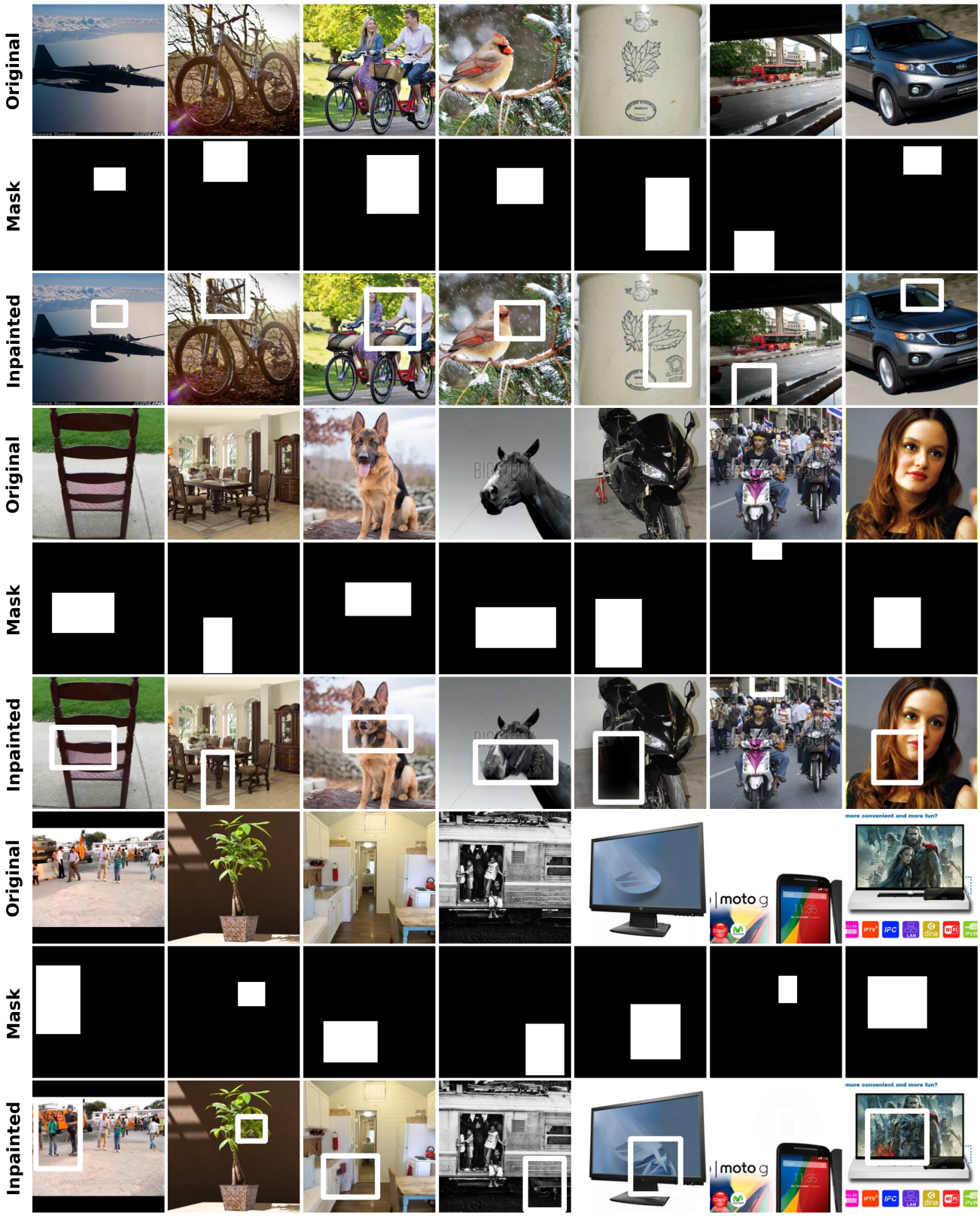

Figure 6. **Examples from the diagnostic inpaint dataset.** Each group of three rows shows: original real images (top), binary inpaint masks (middle), and resulting inpainted images with mask boundaries highlighted in white (bottom).

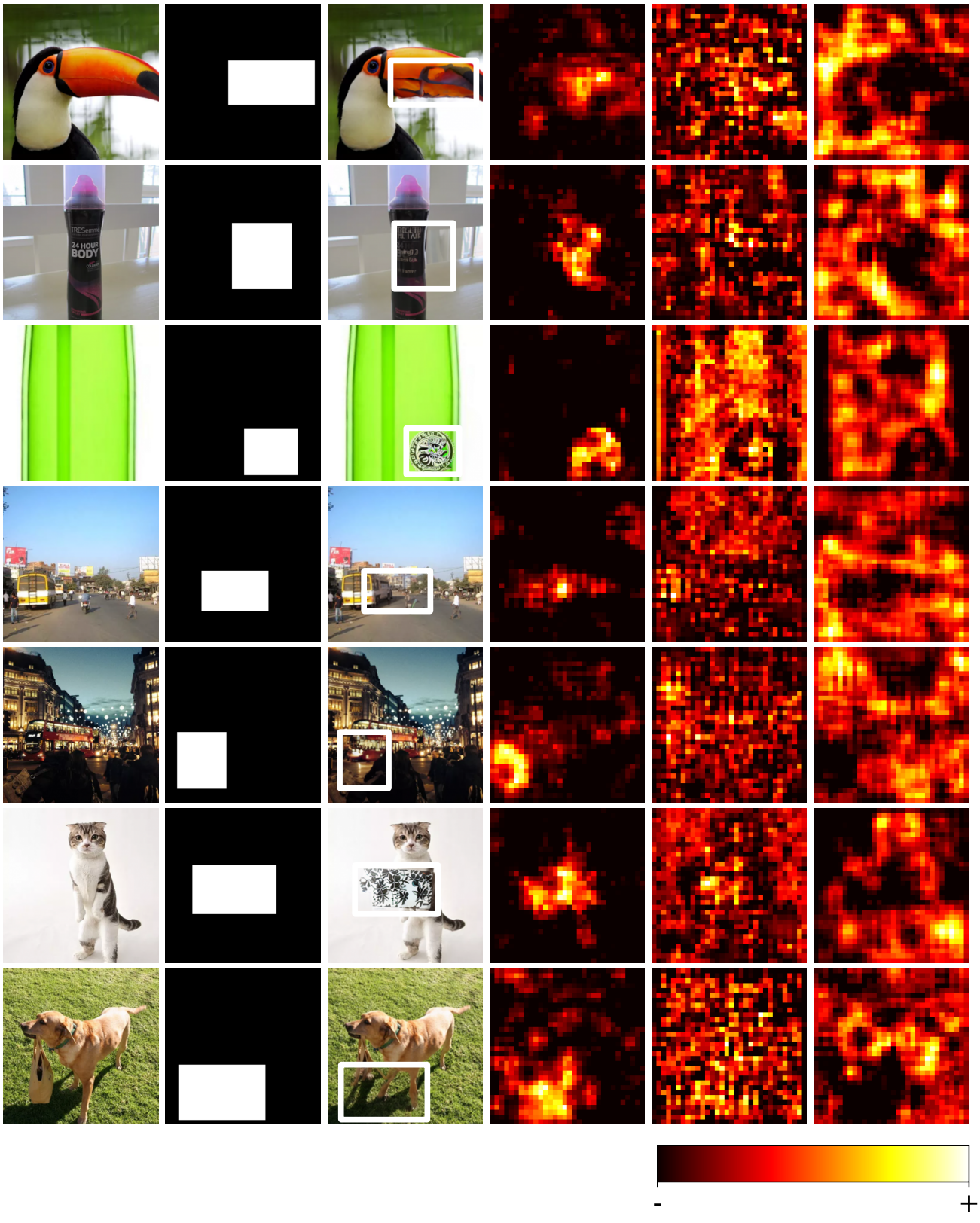

Figure 7. **Extended Feature Alignment Visualization.** Additional examples demonstrating the regional activation patterns across different images. Each row shows (from left to right): original image, inpainting mask, inpainted image, and activation maps from high, middle, and low RAD channels. Consistent with Figure 3, high RAD channels selectively activate on generated regions while low RAD channels respond to real backgrounds.

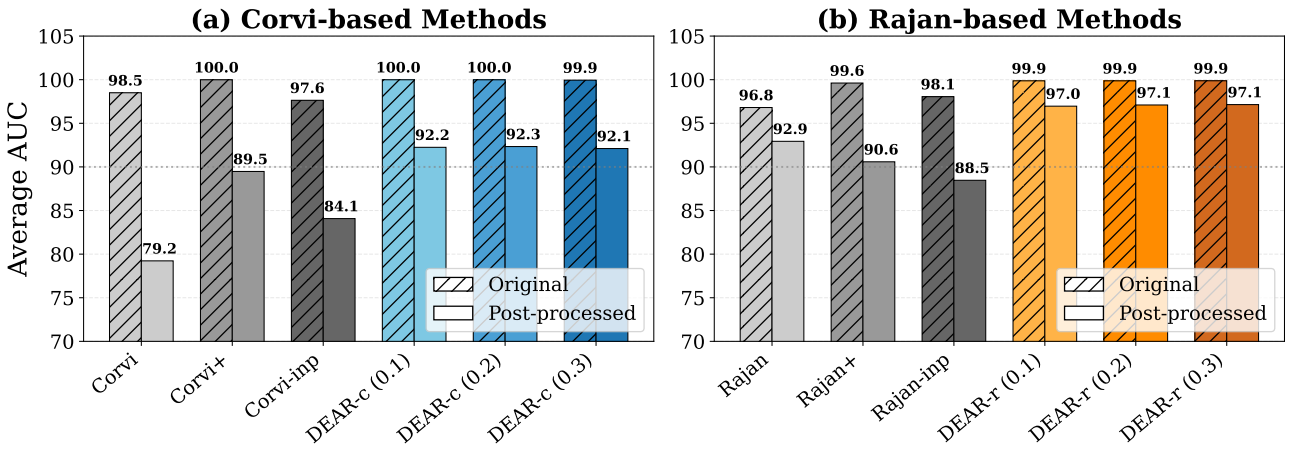

Figure 8. **Average AUC comparison across generators.** We compare baseline detectors (Corvi, Rajan), their Stay-Positive variants (Corvi+, Rajan+), inpaint-trained variants (Corvi-inpaint, Rajan-inpaint), and DEAR with varying pruning ratios ($\alpha \in \{0.1, 0.2, 0.3\}$). Hatched bars indicate performance on original images; solid bars indicate post-processed images. DEAR consistently achieves the highest average AUC, particularly under post-processing where the performance gap is most pronounced.

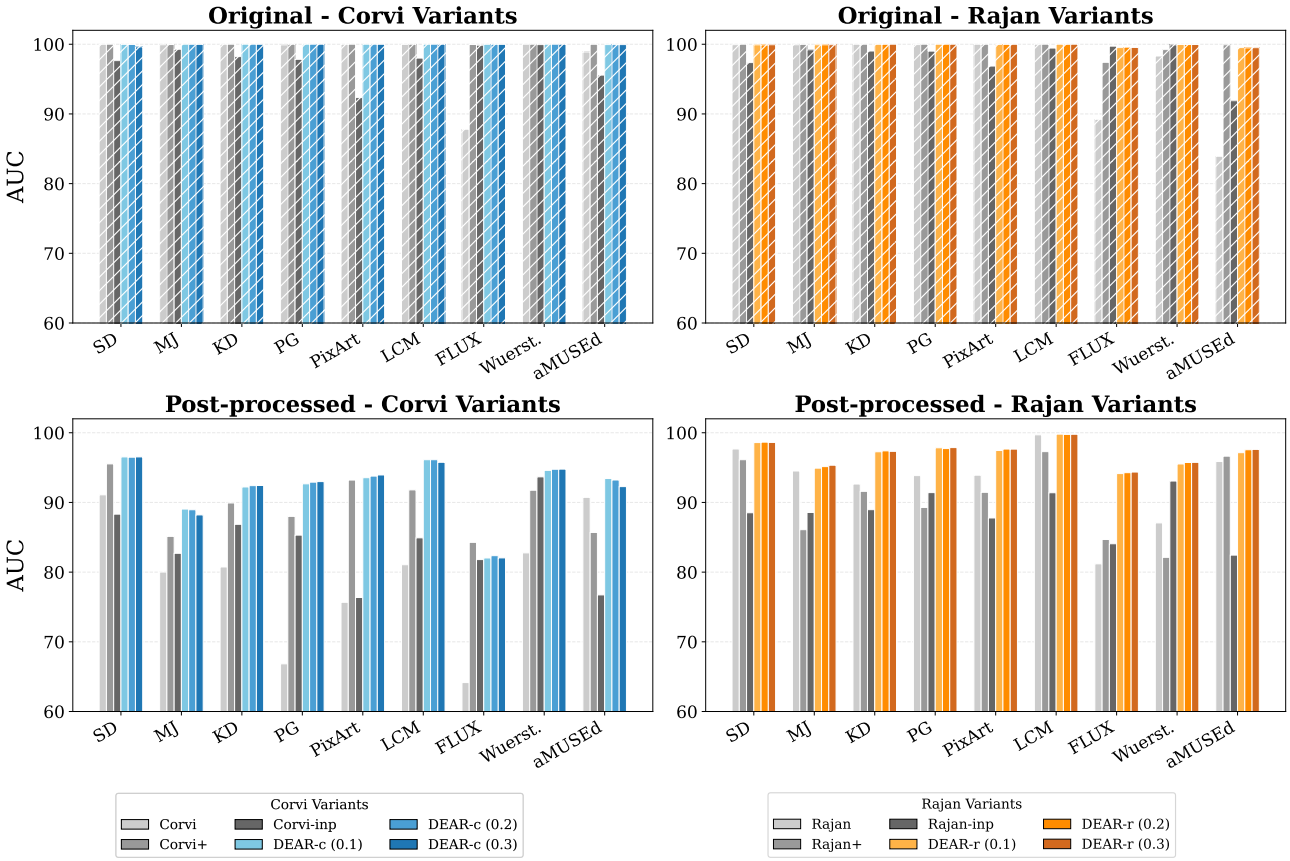

Figure 9. **Per-generator AUC comparison.** Detailed breakdown of detection performance across nine generators (SD, MJ, KD, PG, PixArt, LCM, FLUX, Wuerstchen, aMUSEd). Top row shows results on original images; bottom row shows results on post-processed images. DEAR variants (blue for Corvi-based, orange for Rajan-based) demonstrate superior robustness.