MICRO-DEFECTS EXPOSE MACRO-FAKES: DETECTING AI-GENERATED IMAGES VIA LOCAL DISTRIBUTIONAL SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent generative models can produce images that appear highly realistic, raising challenges in distinguishing real and AI-generated images. Yet existing detectors based on pre-trained feature extractors tend to over-rely on global semantics, limiting sensitivity to the critical micro-defects. In this work, we propose *Micro-Defects expose Macro-Fakes* (MDMF), a local distribution-aware detection framework that amplifies micro-scale statistical irregularities into macro-level distributional discrepancies. To avoid localized forensic cues being diluted by plain aggregation, we introduce a learnable *Patch Forensic Signature* that projects semantic patch embeddings into a compact forensic latent space. We then use *Maximum Mean Discrepancy* (MMD) to quantify distributional discrepancies between generated and real images. Our theory-grounded analysis shows that patch-wise modeling yields provably larger discrepancies when localized forensic signals are present in generated images, enabling more reliable separation from real images. Extensive experiments demonstrate strong cross-generator performance across multiple benchmarks, validating the effectiveness of the distributional detection framework.

1 INTRODUCTION

Deep generative models have made rapid advances in recent years (Ho et al., 2020; Saharia et al., 2022; Podell et al., 2023), with diffusion-based architectures enabling the synthesis of highly realistic images from natural language descriptions. Such advances now power widely used platforms, including Stable Diffusion (Rombach et al., 2022), DALL·E (Ramesh et al., 2022), and Midjourney. While this progress has accelerated the creation of high-quality content, it also raises significant concerns regarding misinformation (Zhou et al., 2023), deepfakes (Heidari et al., 2024), and digital forgery (Somepalli et al., 2023). As modern generative models improve in visual fidelity, reliably distinguishing AI-generated images from natural ones becomes challenging and essential, motivating increasing interest in AI-generated image detection (Zhu et al., 2023b; Chen et al., 2024a).

Previous studies have achieved promising progress by exploiting artifacts left by generative processes (Wang et al., 2023; Chen et al., 2024a; Ojha et al., 2023; Zhang et al., 2025b). Most approaches adopt an image-level paradigm and treat detection as global classification, either learning discriminative features with supervision (Chen et al., 2024a; Liu et al., 2024) or measuring deviations in frozen representation spaces (Ojha et al., 2023; He et al., 2024). However, as modern diffusion models increasingly leave *sparse* and *localized* forensic traces (Wang et al., 2024a; 2025b), detectors built upon pre-trained representations can over-rely on global semantics, which reduces sensitivity to the micro-scale defects that are most diagnostic of generation. Several recent works have explored patch modeling to capture finer-grained cues (Zhong et al., 2023; Liu et al., 2024; Choi et al., 2025). Nevertheless, when localized evidence is still summarized by plain aggregation, subtle forensic cues can remain diluted and the decision may continue to be driven by semantics rather than generation-induced irregularities. This naturally motivates a fundamental research question:

> *Can we learn representations that amplify micro-scale statistical irregularities into stable macro-level distributional discrepancies for AI-generated image detection?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

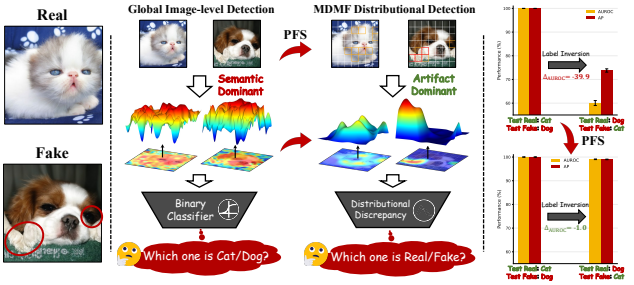


Figure 1: **Intuition behind *Patch Forensic Signature* (PFS).** ***Left:*** a real cat and a generated dog with plausible localized irregularities (highlighted). ***Middle:*** global image-level detection aggregates a **semantic-dominant** representation, inadvertently reducing real/fake detection into semantic recognition (e.g., "cat vs. dog"). PFS maps patch-wise representations into an **artifact-dominant** forensic space, making subtle generation-induced statistical deviations more salient. MDMF thus leverages their distributional discrepancy to answer "real vs. fake". ***Right:*** Label inversion stress test. Global detection suffers a sharp performance drop under inverted labels, whereas PFS remains stable, suggesting a shift from semantics to artifacts (see Section 2.2).

In this paper, we propose a distributional detection perspective grounded in localized forensic evidence. Concretely, instead of representing an image with a single global feature vector, we decompose it into local regions and analyze the statistics of their features. This perspective is well matched to modern generators, whose artifacts often manifest as *localized* statistical shifts that are easily suppressed by uniform aggregation into global representations. To operationalize this idea, we introduce the ***Patch Forensic Signature*** **(PFS)**, a learnable patch-level representation tailored for forensic analysis. PFS reparameterizes semantic patch embeddings into a dedicated forensic space that deemphasizes semantic content while preserving and amplifying subtle statistical irregularities introduced by the generative process (as shown in Figure 1 and Section 2.2).

Based on the Patch Forensic Signature, we propose ***Micro-Defects expose Macro-Fakes*** **(MDMF)**, a distributional detection framework that transforms sparse, localized forensic artifacts into reliable image-level signals. Specifically, MDMF employs *Maximum Mean Discrepancy* (MMD) (Gretton et al., 2012; Liu et al., 2020a) to quantify distributional discrepancy between patch-level PFS representations of test images and those of reference real images. Under the stated sparse-defect and regularity assumptions, the theoretical analysis characterizes how patch-wise modeling can amplify localized shifts relative to global pooling and how MMD can separate the modeled real and generated populations. This analysis provides a principled explanation for why aggregating localized evidence at the distribution level leads to reliable separation, even when individual artifacts are weak.

We conduct extensive experiments to evaluate the effectiveness and generalization of MDMF. Our evaluation covers widely used benchmarks: ImageNet (Deng et al., 2009), LSUN-Bedroom (Yu et al., 2015), and GenImage (Zhu et al., 2023b). Across them, MDMF consistently achieves strong detection performance, demonstrating robustness to diverse generative architectures and training paradigms. To further stress-test the method, we conduct case studies on OpenSora-generated videos (Zheng et al., 2024), where many existing detectors degrade substantially while MDMF identifies stable forensic signals. We summarize our contributions as follows:

- We introduce a new perspective for AI-generated image detection, modeling images as collections of localized visual evidence to study subtle local statistical deviations in generated images. (Section 2.2)
- We propose the ***Patch Forensic Signature*** **(PFS)**, a learnable forensic representation that reparameterizes semantic embeddings into a latent space designed to suppress semantic invariances while preserving and amplifying generative artifacts. (Section 2.3)
- We develop ***Micro-Defects expose Macro-Fakes*** **(MDMF)**, a distributional detection framework that aggregates localized forensic evidence through MMD, with theoretical separation guarantees under the stated modeling assumptions. Experiments across diverse benchmarks show the effectiveness and generalization of MDMF. (Sections 2.4 and 3)

2 Micro-Defects Expose Macro-Fakes.

Preliminary. Let $\mathbb{P}$ denote the distribution of real images defined on an image space $\mathcal{X} \subset \mathbb{R}^{H\times W\times C}$, where H, W, and C denote the image height, width, and number of channels. Given i.i.d. samples $S_{\mathbb{P}} = \{x_n\}_{n=1}^{N}$ drawn from $\mathbb{P}$, the goal of AI-generated image detection is to determine whether a test image $\tilde{y}$ originates from $\mathbb{P}$ or from an alternative distribution $\mathbb{Q}$ introduced by generative models.

2.1 MOTIVATION

Recent advances in generative modeling have substantially reduced perceptually salient artifacts. As a result, some high-fidelity generated images can differ from real images through localized deviations that may be weak in global representations (Wang et al., 2024a; 2025b). We refer to this regime as *Local Distributional Shifts*. Most existing approaches adopt an image-level paradigm and cast detection as global classification (Ojha et al., 2023; Chen et al., 2024a; Tan et al., 2024). For detectors built on recognition-oriented pretrained features, global pooling can preserve semantic correlations that are unrelated to the real/fake distinction.

We conceptually and empirically analyze this limitation. *Conceptually*, Figure 2(a) provides a mechanistic view where semantic content and generation artifacts jointly contribute to an image. A global detector may compress local variation into a representation that emphasizes object semantics. This creates an opportunity for semantic shortcuts when training labels correlate with object category. *Empirically*, we validate this semantic bias using a label inversion toy experiment in Figure 1. We train a global image-level real/fake classifier on a confounded split with *real cats* and *generated dogs*, and evaluate it on the inverted split with *real dogs* and *generated cats*. The DINOv2-based global classifier in this controlled experiment exhibits a sharp performance drop under label inversion, indicating reliance on the category-label correlation in this setting.

To mitigate the semantic dominance, we seek a representation that weakens the influence of global semantics while retaining artifact-related cues. A natural step is to decompose an image into local patches and operate on the resulting patch representations. As illustrated in Figure 2(b), the patch-wise formulation avoids collapsing the image into a globally pooled feature, which weakens the semantic shortcut that can confound real/fake prediction under global aggregation. However, generative artifact patterns are diverse and difficult to model explicitly, and patch embeddings from standard visual backbones are still influenced by semantics. This motivates us to learn a patch-wise representation that suppresses semantic dominance while preserving statistical deviations from the generation.

2.2 THE PATCH FORENSIC SIGNATURE

We introduce the ***Patch Forensic Signature*** **(PFS)**, a learnable representation that reparameterizes semantic patch embeddings into a dedicated forensic space. At a high level, PFS suppresses semantic variation and accentuates generation-induced statistical deviations, yielding signatures that align more closely with artifact-driven evidence. We next formalize PFS by first defining the extracted patch signature field and then specifying the learnable projection.

Patch Signature Field. Let $x \in \mathbb{R}^{H\times W\times C}$ be an input image. We leverage a pre-trained self-supervised vision backbone (e.g., DINOv2 (Oquab et al., 2024)) to decompose the image into a grid of K non-overlapping patch tokens:

$$\mathbf{E}(x) = \{\mathbf{e}_i(x) \in \mathbb{R}^D\}_{i=1}^K, \tag{1}$$

where D is the embedding dimension. While patch-wise modeling weakens semantic shortcuts under global aggregation, $\mathbf{e}_i(x)$ remains largely semantics-oriented, and thus generative statistical cues are still not salient in this space. We then introduce a learnable reparameterization into the *forensic space*, defined as a compact latent space where semantic variation is deemphasized while patch-wise statistical deviations become more separable under the detection objective.

Definition 2.1. (**Patch Forensic Signature (PFS)**.) *Given a patch embedding $\mathbf{e}_i(x)$, we define a learnable projection function $\phi_\theta : \mathbb{R}^D \to \mathbb{R}^d$, parameterized by a lightweight Multilayer Perceptron (MLP), to map semantic embeddings into a compact forensic latent space. We refer to the mapped representation as the Patch Forensic Signature (PFS):*

$$\mathbf{z}_i(x) = \phi_\theta(\mathbf{e}_i(x)) \in \mathbb{R}^d. \tag{2}$$

Consequently, the image x is represented by a set of signature vectors $\mathbf{Z}_\theta(x) = [\mathbf{z}_1(x), \ldots, \mathbf{z}_K(x)]^\top \in \mathbb{R}^{K\times d}$. Our later experiments and analysis will show that, under a suitable learning objective (e.g., Eq. 5), this mapping plays a central role by learning to reparameterize patch-level representations into a dedicated forensic space that deemphasizes semantic variation while preserving and amplifying subtle statistical irregularities introduced by the generative process.

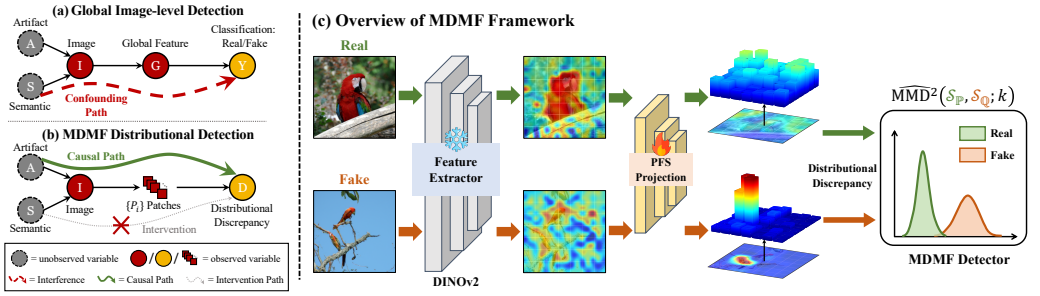


Figure 2: **Motivation and Overview of the MDMF framework.** ***(a)*** Global image-level detection compresses an image into a single feature for real/fake classification, where semantic factors can dominate the decision through a confounding path. ***(b)*** MDMF instead operates on patches and bases its prediction on distributional discrepancy, suppressing semantic interference and aligning the decision with artifact-related signals. ***(c)*** Given real and generated images, a frozen DINOv2 extracts patch representations, which are mapped by the PFS into a forensic space. MDMF then measures the discrepancy between PFS distributions to produce the final score.

2.3 Exploring PFS for Detecting AI-Generated Images

PFS provides patch-wise signatures that emphasize artifact-related statistical cues, yet the resulting evidence remains spatially sparse even in the PFS space. A plain image-level pooling over PFS signatures can still average out these localized cues, making reliable detection difficult for highly realistic generations. This motivates a distributional perspective, where we compare the *distributions* of patch signatures between real and generated images to emphasize subtle statistical irregularities. To operationalize this idea, we adopt the kernel two-sample testing framework via *Maximum Mean Discrepancy* (MMD) (Gretton et al., 2012). MMD quantifies distributional discrepancy through kernel mean embeddings in a *reproducing kernel Hilbert space* (RKHS), where small but systematic deviations across local observations can accumulate into a stable image-level signal (Liu et al., 2020a). Building on these, we establish the ***Micro-Defects expose Macro-Fakes*** **(MDMF)** framework, which transforms sparse patch-level forensic cues into reliable detection scores, as shown in Figure 2 (c).

MMD Formulation. Consider two arbitrary sets of images $\mathcal{S}_{\mathbb{P}} = \{x_n\}_{n=1}^{N} \sim \mathbb{P}$ and $\mathcal{S}_{\mathbb{Q}} = \{y_m\}_{m=1}^{N} \sim \mathbb{Q}$. To measure the distance between distributions $\mathbb{P}$ and $\mathbb{Q}$, we employ an unbiased U-statistic estimator for the squared MMD,

$$\widehat{\mathrm{MMD}}_u^2(\mathcal{S}_{\mathbb{P}}, \mathcal{S}_{\mathbb{Q}}; k) := \frac{1}{N(N-1)} \sum_{i \neq j} H_{ij}, \tag{3}$$

where k denotes the kernel of an RKHS and $H_{ij} := k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(y_i, x_j)$. The corresponding $\widehat{\mathrm{MMD}}_b^2 := \frac{1}{N^2} \sum_{ij} H_{ij}$ is the squared MMD between the empirical distributions of $\mathcal{S}_{\mathbb{P}}$ and $\mathcal{S}_{\mathbb{Q}}$ (Liu et al., 2020a). According to the null hypothesis testing framework (Gretton et al., 2012), under the null hypothesis $\mathfrak{H}_0 : \mathbb{P} = \mathbb{Q}$, $\widehat{\mathrm{MMD}}_u^2(\cdot)$ should be close to zero, while strictly positive under the alternative hypothesis $\mathfrak{H}_1 : \mathbb{P} \neq \mathbb{Q}$. Leveraging this, we design the following optimization and detection protocols.

Optimization Protocol. We construct $\mathcal{S}_{\mathbb{P}}^{tr}$ by aggregating real training images and $\mathcal{S}_{\mathbb{Q}}^{tr}$ from generated training images. We ideally expect to correctly reject $\mathfrak{H}_0$ and derive $\mathfrak{H}_1 : \mathbb{P} \neq \mathbb{Q}$, i.e., $\mathcal{S}_{\mathbb{P}}^{tr}$ and $\mathcal{S}_{\mathbb{Q}}^{tr}$ come from different distributions. To enhance discriminative power, we utilize a deep Gaussian kernel (Liu et al., 2020a) with bandwidth γ for MMD:

$$k_\omega(x, y) = \exp\left(-\frac{\|\mathbf{Z}_\theta(x) - \mathbf{Z}_\theta(y)\|_2^2}{2\gamma^2}\right), \tag{4}$$

Simply maximizing $\widehat{\mathrm{MMD}}_u^2$ can be problematic if the variance of the statistic also increases, leading to unstable gradients. Following the test power maximization principle (Gretton et al., 2012), we optimize the parameters $\omega = \{\theta, \gamma\}$, namely the projection weights in ϕ_θ and kernel bandwidth, to maximize the **regularized test power criterion** with variance $\hat{\sigma}_{H_1}^2 = \frac{4}{N^3} \sum_{i=1}^{N} \left(\sum_{j=1}^{N} H_{ij}\right)^2 -$

$\frac{4}{N^4}\left(\sum_{i=1}^{N}\sum_{j=1}^{N} H_{ij}\right)^2$,

$$\max_{\omega} \; J_\lambda(\mathcal{S}_{\mathbb{P}}^{tr}, \mathcal{S}_{\mathbb{Q}}^{tr}; k_\omega) = \frac{\widehat{\mathrm{MMD}}_u^2(\mathcal{S}_{\mathbb{P}}^{tr}, \mathcal{S}_{\mathbb{Q}}^{tr}; k_\omega)}{\sqrt{\hat{\sigma}_{H_1}^2 + \lambda}}, \tag{5}$$

Detection Protocol. With the learned parameters ω^*, we apply MMD with the biased estimator to detect individual test images by quantifying their PFS distributional deviation from a set of reference images, following recent works (Zhang et al., 2024b; 2025a) that demonstrate MMD's effectiveness in single-sample detection. Given a set of reference images $\{x_r\}_{r=1}^R$ and a test $\tilde{y}$, we compute the MDMF score:

$$S_{\text{MDMF}}(\tilde{y}) = \widehat{\mathrm{MMD}}_b^2(\mathcal{S}_{\text{ref}}, \{\tilde{y}\}; k_{\omega^*}) = \frac{1}{R^2}\sum_{r,r'=1}^{R} \cdot k_{\omega^*}(x^{(r)}, x^{(r')}) + k_{\omega^*}(\tilde{y}, \tilde{y}) - \frac{2}{R}\sum_{r=1}^{R} k_{\omega^*}(x^{(r)}, \tilde{y}). \tag{6}$$

Hence, we can formalize the detection model $f(\cdot)$ to determine whether a given input $\tilde{y}$ is generated:

$$f(\tilde{y}) = \begin{cases} \text{Generated}, & \text{if } S_{\text{MDMF}}(\tilde{y}) > \tau, \\ \text{Real}, & \text{otherwise}, \end{cases} \tag{7}$$

Although our method detects generated images by measuring distributional discrepancies via MMD, its effectiveness fundamentally relies on PFS extracting artifact-sensitive patch evidence that is often weakened in global representations (see theoretical analysis in Section 2.4 and empirical analysis in Section 3).

2.4 Theoretical Analysis

In this section, we provide theoretical justification for MDMF's detection mechanism. First, we show that PFS amplifies sparse localized deviations that tend to be diluted in global image-level detection (Propositions 2.4 and 2.5). Second, we establish that MMD on PFS converts this amplified shift into reliable real/fake separation (Proposition 2.6 and Theorem 2.7). We first introduce the assumptions.

Assumption 2.2. Real images $\{x_n\}_{n=1}^N$ are i.i.d. sampled from distribution $\mathbb{P}$, and generated images $\{y_m\}_{m=1}^N$ are i.i.d. sampled from distribution $\mathbb{Q}$. Given any real or generated image, we extract K non-overlapping patch embeddings $\{\mathbf{e}_i\}_{i=1}^K \subset \mathbb{R}^D$ using a fixed pre-trained encoder (e.g., DINOv2). Each patch embedding follows a σ_e-sub-Gaussian distribution (Wainwright, 2019) in $\mathbb{R}^D$.

Assumption 2.3 (Sparse Defect Model). For a generated image y, we assume each patch embedding:

$$\mathbf{e}_i(y) = \mathbf{u}_i + a_i s_i \boldsymbol{\mu}_{\text{defect}}, \tag{8}$$

where $\mathbf{u}_i \sim \mathcal{SG}(\mathbf{0}, \sigma_e^2 \mathbf{I}_D)$, $a_i \sim \text{Bernoulli}(\rho)$ indicates whether the patch is defective, and $s_i \in \{+1, -1\}$ is an independent Rademacher variable with $\mathcal{P}(s_i = +1) = \mathcal{P}(s_i = -1) = 1/2$. Hence $\mathbb{E}[\mathbf{e}_i(y)] = \mathbf{0}$ but defective patches elevate second-order energy. For real images, $\mathbf{e}_i(x) = \mathbf{u}_i$.

Assumption 2.2 follows common practice in representation analysis works (Wang et al., 2024b; Zhang et al., 2025a; 2024a), while Assumption 2.3 aligns with sparse-artifact observations in generated images (Wang et al., 2024a; 2025b). Under these assumptions, we then establish that PFS amplifies localized defects into a detectable distributional shift.

Proposition 2.4. *Assume ϕ_θ is twice differentiable at $\mathbf{0}$ with Hessian $\mathbf{H}_\phi(\mathbf{0}) \in \mathbb{R}^{d\times D\times D}$. Let $\Delta_{\text{PFS}} := \mathbb{E}_{\mathbb{Q}}[\phi_\theta(\mathbf{e}_i(y))] - \mathbb{E}_{\mathbb{P}}[\phi_\theta(\mathbf{e}_i(x))]$. Then the leading-order PFS mean shift satisfies*

$$\Delta_{\text{PFS}} \;\approx\; \frac{\rho}{2}\,\mathcal{Q}(\boldsymbol{\mu}_{\text{defect}}), \tag{9}$$

where $\mathcal{Q}(\boldsymbol{\mu}) \in \mathbb{R}^d$ denotes the Hessian-induced quadratic form of ϕ_θ evaluated along direction $\boldsymbol{\mu}$, i.e., $\big[\mathcal{Q}(\boldsymbol{\mu})\big]_\ell = \boldsymbol{\mu}^\top \nabla^2 \phi_{\theta,\ell}(\mathbf{0})\boldsymbol{\mu}$, for $\ell = 1, \ldots, d$. If $\mathcal{Q}(\boldsymbol{\mu}_{\text{defect}}) \neq 0$, $\|\Delta_{\text{PFS}}\|_2 > 0$ for any $\rho > 0$.

Proposition 2.5. *Under Assumption 2.3 and Proposition 2.4, we define the global-pooled leading order shift as $\Delta_{\text{global}} := \mathbb{E}_{\mathbb{Q}}[\phi_\theta(\bar{\mathbf{e}}(y))] - \mathbb{E}_{\mathbb{P}}[\phi_\theta(\bar{\mathbf{e}}(x))]$, where $\bar{\mathbf{e}}(x) = \frac{1}{K}\sum_{i=1}^K \mathbf{e}_i(x)$. Then the leading-order shifts satisfy:*

$$\|\Delta_{\text{PFS}}\|_2 \approx K\,\|\Delta_{\text{global}}\|_2 > \|\Delta_{\text{global}}\|_2, \tag{10}$$

Notably, Proposition 2.5 does not imply unbounded gains as the number of patches increases. The patch-granularity experiment in Section 3.3 shows a non-monotonic trend over the evaluated resolutions; Proposition 2.5 does not assert that increasing the number of patches always improves performance. We quantify how the amplified PFS shift manifests as a measurable population MMD gap between $\mathbb{P}$ and $\mathbb{Q}$.

Proposition 2.6. *Let $k_\omega(\cdot,\cdot)$ be a Gaussian kernel where ω denotes the set of projection weights θ and kernel bandwidth γ. Under Proposition 2.4 and a Gaussian surrogate in PFS space, the population $\widehat{\mathrm{MMD}}^2$ between $\mathbb{P}$ and $\mathbb{Q}$ satisfies:*

$$\widehat{\mathrm{MMD}}^2(\mathbb{P},\mathbb{Q};k_\omega) = 2\left(\frac{\gamma^2}{\gamma^2+2\sigma_z^2}\right)^{\frac{Kd}{2}}\left[1-\exp\left(-\frac{K\|\boldsymbol{\Delta}_{\mathrm{PFS}}\|_2^2}{2(\gamma^2+2\sigma_z^2)}\right)\right], \tag{11}$$

where σ_z^2 denotes the isotropic proxy variance of the Gaussian surrogate in PFS space. $\widehat{\mathrm{MMD}}^2(\mathbb{P},\mathbb{Q};k_\omega)$ is strictly positive for $\|\boldsymbol{\Delta}_{\mathrm{PFS}}\|_2 > 0$ and is monotonically increasing.

Building on Proposition 2.6, we derive the finite-sample concentration guarantees for detection.

Theorem 2.7. *Let $S_r = \{x_i\}_{i=1}^M \overset{i.i.d}{\sim} \mathbb{P}$ be a reference set of real images and $S_t = \{y_j\}_{j=1}^N$ be test images, let $\lambda = \gamma^2 + 2\sigma_z^2$. For any $\delta \in (0,1)$, with probability at least $1-\delta$, the following holds:*

(Case I: Real test image). *If $S_t \overset{i.i.d}{\sim} \mathbb{P}$,*

$$\widehat{\mathrm{MMD}}_u^2(S_r,S_t) \le \underbrace{C_1(\sigma_z,\gamma)\sqrt{\left(\frac{1}{M}+\frac{1}{N}\right)\log\frac{2}{\delta}}}_{\textit{Finite-sample fluctuation}}. \tag{12}$$

(Case II: Generated test image). *If $S_t \overset{i.i.d}{\sim} \mathbb{Q}$,*

$$\widehat{\mathrm{MMD}}_u^2(S_r,S_t) \ge \underbrace{2\left(\frac{\gamma^2}{\lambda}\right)^{\frac{Kd}{2}}\left[1-\exp\left(-\frac{K\|\boldsymbol{\Delta}_{\mathrm{PFS}}\|_2^2}{2\lambda}\right)\right]}_{\textit{Artifact-induced signature shift}} - \underbrace{C_2(\sigma_z,\gamma)\sqrt{\left(\frac{1}{M}+\frac{1}{N}\right)\log\frac{2}{\delta}}}_{\textit{Finite-sample error}}. \tag{13}$$

Interpretation. Theorem 2.7 establishes that the empirical MMD concentrates around its population value with deviation scaling as $O(\sqrt{1/M+1/N})$. For real test images, the population MMD vanishes and values reflect only finite-sample fluctuations. For generated images, Proposition 2.6 guarantees a positive gap scaling with $\|\boldsymbol{\Delta}_{\mathrm{PFS}}\|_2^2$. When this separation dominates, real images yield smaller MMD scores than generated ones, justifying reliable detection for AI-generated images.

3 Experiments

3.1 Experimental Setup

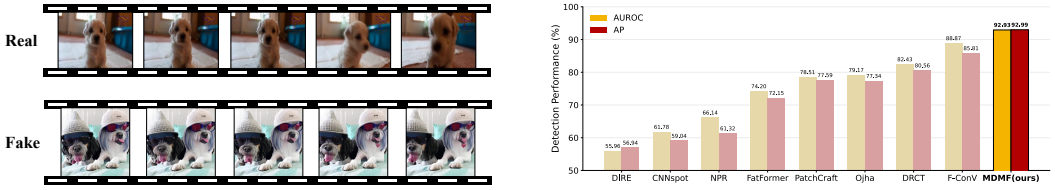
We provide detailed experimental setups in Appendix D.

Datasets. Following previous works (Wang et al., 2020; Zhang et al., 2025b), we evaluate our MDMF on the following benchmarks: **ImageNet** (Deng et al., 2009), **LSUN-Bedroom** (Yu et al., 2015), and **GenImage** (Zhu et al., 2023b). To further assess generalization to generators beyond image benchmarks, we additionally conduct a case study on videos generated by **OpenSora** (Zheng et al., 2024). Specifically, we sample 3,275 generated videos and extract 10 frames per video, resulting in 32,750 frames and treat them as generated images. For real data, we sample the same number of natural videos and frames on MSR-VTT (Xu et al., 2016).

Baselines and Evaluation Metrics. We compare our MDMF with the following training-based detection baselines in the main experiments: CNNspot (Wang et al., 2020), Ojha (Ojha et al., 2023),

Table 1: Detection performance (%) on ImageNet benchmark. Bold and underlined values indicate the best and second-best results in each column.

Methods	Venue	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average	
		AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
CNNspot (Wang et al., 2020)	CVPR'20	62.25	63.13	63.28	62.27	63.16	64.81	62.85	61.16	85.71	84.93	74.85	71.45	68.41	68.67	61.83	62.91	60.98	61.69	67.04	66.78
Ojha (Ojha et al., 2023)	CVPR'23	83.37	82.95	79.60	78.15	80.35	79.71	82.93	81.72	93.07	92.77	87.45	84.88	85.36	83.15	85.19	84.22	90.82	90.71	85.35	84.25
DIRE (Wang et al., 2023)	ICCV'23	51.82	50.29	53.14	52.96	52.83	51.84	54.67	55.10	51.62	50.83	50.70	50.27	50.95	51.36	55.95	54.83	52.58	52.10	52.70	52.18
NPR (Tan et al., 2024)	CVPR'24	85.68	80.86	84.34	79.79	91.98	86.96	86.15	81.26	89.73	84.46	82.21	78.20	84.13	78.73	80.21	73.21	89.61	84.15	86.00	80.84
PatchCraft (Zhong et al., 2023)	—	81.83	79.65	70.88	69.36	68.47	65.19	75.38	73.29	99.85	99.26	98.55	97.91	96.33	96.25	91.28	91.47	92.56	92.17	86.13	84.95
DRCT (Chen et al., 2024a)	ICML'24	90.26	90.07	85.74	83.85	90.24	89.88	88.27	89.06	95.87	94.99	86.89	86.12	89.11	88.39	92.38	92.41	94.44	94.47	90.36	89.92
FatFormer (Liu et al., 2024)	CVPR'24	91.77	90.36	83.58	83.17	92.58	92.06	86.93	85.14	98.76	98.47	97.65	98.02	97.64	97.57	96.55	95.96	97.65	97.27	93.68	93.11
LOTA (Wang et al., 2025a)	ICCV'25	66.84	65.73	67.18	66.61	80.68	88.33	74.85	84.49	77.95	78.06	78.96	87.92	73.55	83.99	67.41	76.16	82.34	90.19	74.42	80.16
C2P-CLIP (Tan et al., 2025)	AAAI'25	72.12	77.88	69.07	75.10	90.06	95.72	48.68	74.04	99.84	99.88	85.82	94.19	94.39	97.69	82.27	91.60	98.97	99.60	82.36	89.52
SAFE (Li et al., 2025)	KDD'25	65.51	59.52	64.78	59.12	91.41	94.36	87.42	92.64	93.07	92.11	90.80	94.57	90.11	93.85	88.84	92.28	94.41	96.53	85.15	86.11
AIDE (Yan et al., 2025a)	ICLR'25	90.32	90.96	86.96	88.08	90.44	94.95	78.77	87.97	99.62	99.65	96.46	98.26	97.62	98.85	98.19	99.10	99.50	99.75	93.10	95.29
Effort (Yan et al., 2025b)	ICML'25	88.28	89.96	83.74	85.89	94.15	97.30	84.14	92.47	**99.96**	**99.96**	94.46	97.56	94.24	97.52	95.52	97.93	**99.84**	**99.93**	92.70	95.39
F-ConV (Zhang et al., 2025b)	NeurIPS'25	**92.74**	91.65	88.51	87.67	88.87	88.47	85.94	84.88	98.94	98.98	98.14	98.72	98.52	98.38	96.79	96.33	95.52	95.38	93.77	93.38
MDMF	—	92.56	**93.57**	**88.86**	**90.16**	**94.63**	**97.35**	**88.89**	**94.48**	99.93	99.94	**98.99**	**99.52**	**98.76**	**99.41**	**98.84**	**99.46**	99.40	99.72	**95.65**	**97.07**



(a) Examples of *Real* and *Fake* videos

(b) Detection Performance on OpenSora

Figure 3: Example visualizations and performance comparison on OpenSora.

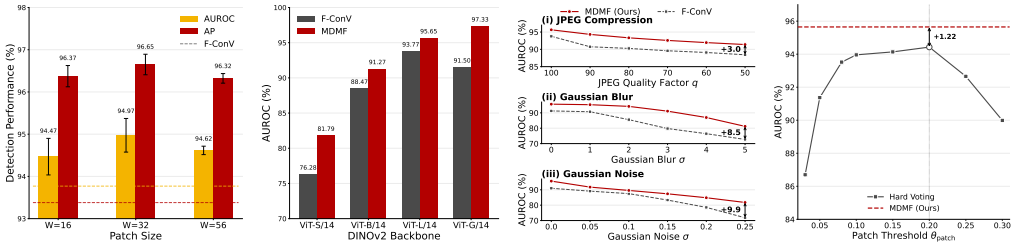
DIRE (Wang et al., 2023), PatchCraft (Zhong et al., 2023), NPR (Tan et al., 2024), DRCT (Chen et al., 2024a), FatFormer (Liu et al., 2024), LOTA (Wang et al., 2025a), C2P-CLIP (Tan et al., 2025), SAFE (Li et al., 2025), AIDE (Yan et al., 2025a), Effort (Yan et al., 2025b), and F-ConV (Zhang et al., 2025b). Following (Zhang et al., 2025b), we adopt these metrics: ① average precision (AP); ② area under the receiver operating characteristic curve (AUROC); ③ classification accuracy (ACC).

Implementation Details. Following previous studies (Ojha et al., 2023; Liu et al., 2024), we apply random cropping and random horizontal flipping at training, while center cropping at testing, both with no other augmentations. To balance detection performance and efficiency, we adopt DINOv2 ViT-L/14 (Oquab et al., 2024) to extract patch embeddings and pool the patch size to $W = 32$ for PFS computation in main experiments. The projection ϕ_θ and kernel bandwidth γ are jointly trained during optimization.

3.2 Main Results

Detection performance comparison with baselines. Table 1 reports detection performance on the ImageNet benchmark across nine generative models spanning diffusion, GANs, and transformers. MDMF demonstrates consistently strong performance across all evaluated generators, indicating robust generalization under diverse generative mechanisms. Notably, MDMF shows particularly strong performance on recent diffusion-based models, which are known to produce highly realistic images with sparse and localized artifacts that challenge existing detectors. These results validate that our PFS distributional modeling effectively captures the subtle, localized forensic signals characteristic of modern generative paradigms. Beyond diffusion models, MDMF also maintains competitive performance on earlier generative paradigms. This consistent behavior further demonstrates that MDMF effectively captures generator-agnostic forensic artifacts and amplifies micro-scale defects into robust macro-level detection signals across both emerging and conventional generative models.

Case Study on OpenSora-Generated Content. We further evaluate MDMF on a challenging case study using frames sampled from OpenSora (Zheng et al., 2024), a recent video generation model that is not seen during training. Figure 3(a) shows the advanced diffusion-generated videos with strong temporal consistency introduced by OpenSora, resulting in frames that are globally coherent and largely free of perceptual artifacts. As illustrated in Figure 3(b), while several competitive baselines exhibit notable performance degradation, MDMF still maintains robust detection performance on OpenSora-generated frames. This contrast indicates that the distributional modeling of MDMF captures localized forensic signatures that persist even under substantial domain shifts, enabling effective generalization to emerging video generation paradigms that are unseen during training.



(a) Diff. Patch Size (b) Diff. Backbone (c) Diff. Perturbation (d) Comp. Patch Voting

Figure 4: **Further analysis.** (a) Sensitivity to patch size W; (b) Robustness to DINOv2 backbone variants; (c) Robustness to post-processing perturbations; (d) Comparison with patch-level hard voting under varying θ_{patch}.

3.3 Ablation and Further Analysis

We provide detailed results and discussions in Appendix E.

Ablation of core components in MDMF. Table 2 analyzes the contribution of PFS modeling and MMD optimization in MDMF. For variants without MMD, we train binary classifiers with a standard BCE loss. In particular, the PFS w/o MMD variant uses a lightweight attention head to pool patch-wise PFS into an image-level score (details in Appendix D). Notably, even without MMD optimization, attention-aggregated PFS still achieves competitive performance, indicating that the forensic reparameterization already suppresses semantic dominance and highlights generation-related cues. In contrast, the effect of MMD is dependent on the underlying representation. When applied to global pooling, MMD fails to yield performance gains, whereas combining with PFS leads to a clear improvement. This shows that MMD serves as a complementary amplifier when patch-wise forensic evidence is preserved in the PFS space, rather than when it is diluted by the semantics-dominant global representations.

Table 2: **Ablation of key components on ImageNet.** Variants without MMD are trained with a BCE objective, while PFS modeling without MMD adds a lightweight attention head for aggregation.

PFS Modeling	MMD Optimization	Average	
		AUROC (↑)	AP (↑)
Global Pooling			
✗	✗	90.14	93.33
✗	✓	86.53	92.18
PFS Modeling			
✓	✗	93.22	95.34
✓	✓	**95.65**	**97.07**

Effect of patch granularity. To evaluate the effect of patch granularity, we vary the patch size W while keeping other settings fixed. Figure 4(a) shows a non-monotonic dependence on W. Performance varies modestly across the practical range of patch sizes and peaks at an intermediate granularity. This is an empirical observation over the evaluated patch sizes. Coarse partitions ($W = 56$) can lose spatial resolution, while fine partitions ($W = 16$) can weaken the evidence within each patch and increase estimation variance. This indicates that PFS benefits from an intermediate granularity that balances localized sensitivity with reliable estimation.

Robustness to the encoder architecture. To evaluate sensitivity to the feature extractor, we instantiate MDMF with multiple DINOv2 backbone variants and compare it with F-ConV under the same setting. As shown in Figure 4(b), MDMF consistently achieves higher detection performance across all evaluated encoders, demonstrating robustness to the underlying backbone choice. MDMF maintains an advantage with smaller backbones (e.g., ViT-S/14), and continues to improve as the backbone is scaled up, whereas F-ConV exhibits non-monotonic behavior and degrades on the largest encoder (e.g., ViT-G/14). This contrast suggests that the proposed PFS representation provides robust forensic signals that transfer effectively across encoder scales, leading to more stable behavior.

Robustness to common post-processing perturbations. We further evaluate MDMF and F-ConV under JPEG compression, Gaussian blur, and Gaussian noise on ImageNet. As shown in Figure 4(c), MDMF maintains higher AUROC across all severity levels, with markedly gentler degradation than F-ConV (-4.3 vs. -5.3 for JPEG, -14.4 vs. -18.6 for blur, -13.9 vs. -19.2 for noise), and the gap widens as severity grows ($+3.0/+8.5/+9.9$ AUROC at the most severe levels). The result is consistent with the benefit of using patch-level signatures and distributional scoring, although this comparison does not isolate their individual contributions.

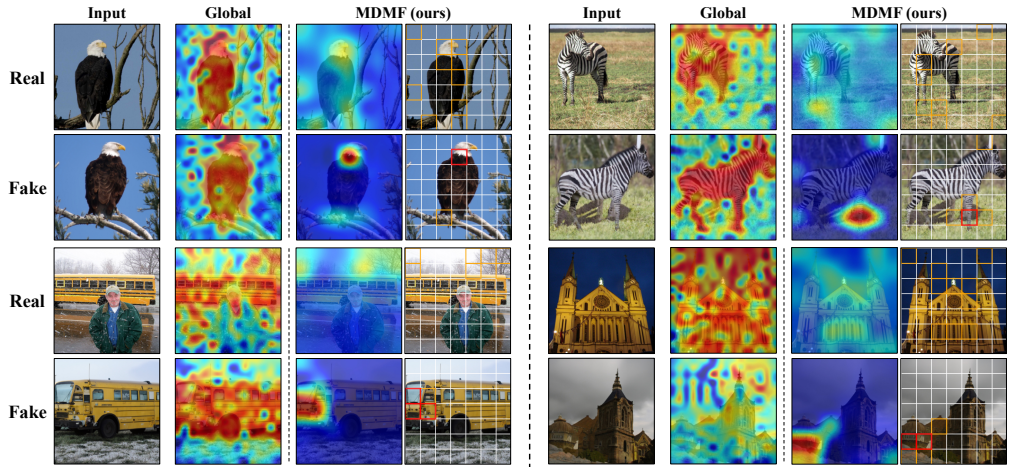


Figure 5: **Qualitative visualization of localized forensic evidence.** We compare representative real images and category-matched generated images using the model-specific heatmaps described in Appendix D.3; warmer colors indicate stronger local responses. Global-pooling baseline primarily highlights semantically salient regions with similar patterns for real and generated samples, whereas MDMF shows localized responses on generated images and diffuse activations on real images, consistent with a different spatial response pattern.

Comparison with patch-level hard voting. A natural alternative to MDMF's distributional aggregation is to classify each patch independently and aggregate via hard voting on the resulting fake-patch ratio. Both pipelines use the same frozen DINOv2 backbone, patch grid (49 patches at $W{=}32$), and training images. The voting baseline trains a linear per-patch classifier with BCE, whereas MDMF trains the PFS head with an MMD objective and compares patch distributions at inference. Voting requires a per-patch cutoff θ_{patch} in addition to the image-level decision threshold τ. As shown in Figure 4(d), its average AUROC ranges from 86.70 to 94.43 across the tested cutoffs, and its best setting trails MDMF by 1.22 points. This comparison favors the complete PFS/MMD pipeline under the shared data and backbone setting; it does not isolate the effect of aggregation alone.

Qualitative visualization of localized forensic cues. To better understand how MDMF detects highly realistic samples, Figure 5 shows representative real images and category-matched ADM images with model-specific heatmaps; Appendix D.3 describes how each map is computed. First, we can observe that the generated images exhibit strong semantic coherence and high visual fidelity. Consistently, the global pooling visualization primarily highlights semantically salient regions, such as object boundaries and high-contrast textures, indicating similar patterns for real and generated images and limited sensitivity to sparse local artifacts. In contrast, MDMF produces more localized responses on generated images and assigns higher activation to regions that contain subtle generative irregularities, while producing more diffuse patterns on real images. This reflects a pronounced distributional discrepancy between real and generated samples in the PFS space, which provides a strong basis for MMD to produce a stable detection signal. This also aligns with our theoretical and quantitative analysis, suggesting that MDMF can surface localized evidence that is suppressed by semantic-dominated global features.

4 Conclusion

In this paper, we present a distributional perspective for AI-generated image detection by modeling an image as a collection of localized visual evidence. Building on this view, we introduce ***Patch Forensic Signature*** **(PFS)**, a learnable forensic representation that reparameterizes semantic embeddings into a latent space to suppress semantic invariances while amplifying generative artifacts. We further propose ***Micro-Defects expose Macro-Fakes*** **(MDMF)**, which measures distributional discrepancy over PFS via MMD to aggregate localized evidence into stable image-level detection signals, and we provide theoretical analysis that establishes the advantage of PFS and the separation between real and generated images. Extensive experiments on multiple benchmarks with detailed ablations and analyses demonstrate the effectiveness and generalization of MDMF.

AI USE STATEMENT

The authors led the research, developed the core MDMF method, conducted the primary experiments, and wrote most of the manuscript. Generative AI tools were used in a supporting role to brainstorm and refine methodological and experimental choices, assist with mathematical formulation and checking derivations, help with data processing and analysis code, organize and interpret experimental records, and draft or edit portions of the manuscript. The authors critically reviewed and revised AI-assisted material against the implementation and recorded results, made the final scientific and editorial decisions, and take responsibility for the entire submission, including any AI-assisted text, claims, code, or artifacts.

ETHICS STATEMENT

MDMF is intended to provide one signal when assessing whether an image is AI-generated. A detector score alone cannot establish the origin of an image or the intent of its creator. As our perturbation tests and false-positive examples show, errors may arise when image style, processing, or reference-set composition differs from the evaluated setting. Incorrectly labeling a real image as synthetic can harm creators and other affected people, so consequential decisions should include human review, corroborating provenance evidence, and a way to contest errors. Public descriptions of detection methods may also help adversaries develop evasions. We report the evaluated settings and limitations to support careful interpretation and use of the results.

REPRODUCIBILITY STATEMENT

The datasets and their preprocessing are described in Appendix D.1, the training and test protocols, reference-set construction, model configuration, environment, and hyperparameters in Appendix D.3, and the algorithms and theoretical proofs in Appendices A and B. We train on ProGAN for the ImageNet and LSUN-Bedroom evaluations and on SD v1.4 for GenImage and the OpenSora case study, as detailed in the appendix. Our training and evaluation code, together with pretrained checkpoints where applicable, are included in the submitted supplementary materials.

REFERENCES

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

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

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

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

Haoxing Chen, Yan Hong, Zizheng Huang, Zhuoer Xu, Zhangxuan Gu, Yaohui Li, Jun Lan, Huijia Zhu, Jianfu Zhang, Weiqiang Wang, et al. Demamba: Ai-generated video detection on million-scale genvideo benchmark. *arXiv preprint arXiv:2405.19707*, 2024b.

Ruoxin Chen, Junwei Xi, Zhiyuan Yan, Ke-Yue Zhang, Shuang Wu, Jingyi Xie, Xu Chen, Lei Xu, Isabel Guan, Taiping Yao, and Shouhong Ding. Dual data alignment makes ai-generated image detector easier generalizable. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

Jiaxi Gu, Xiaojun Meng, Guansong Lu, Lu Hou, Niu Minzhe, Xiaodan Liang, Lewei Yao, Runhui Huang, Wei Zhang, Xin Jiang, et al. Wukong: A 100 million large-scale chinese cross-modal pre-training benchmark. *Advances in Neural Information Processing Systems*, 35:26418–26431, 2022.

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

Zhiyuan He, Pin-Yu Chen, and Tsung-Yi Ho. Rigid: A training-free and model-agnostic framework for robust ai-generated image detection. *arXiv preprint arXiv:2405.20112*, 2024.

Arash Heidari, Nima Jafari Navimipour, Hasan Dag, and Mehmet Unal. Deepfake detection using deep learning methods: A systematic and comprehensive review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 14(2):e1520, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

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

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

Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J Sutherland. Learning deep kernels for non-parametric two-sample tests. In *International conference on machine learning*, pp. 6316–6326. PMLR, 2020a.

Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10770–10780, 2024.

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*, pp. 10012–10022, 2021.

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*, pp. 8060–8069, 2020b.

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

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

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*, pp. 86–103. Springer, 2020.

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

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6048–6058, 2023.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28130–28139, 2024.

Chuangchuang Tan, Renshuai Tao, Huan Liu, Guanghua Gu, Baoyuan Wu, Yao Zhao, and Yunchao Wei. C2p-clip: Injecting category common prompt in clip to enhance generalization in deepfake detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7184–7192, 2025.

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*, pp. 10347–10357. PMLR, 2021.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

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

Kaihong Wang, Lingzhi Zhang, and Jianming Zhang. Detecting human artifacts from text-to-image models. *arXiv preprint arXiv:2411.13842*, 2024a.

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

Yiming Wang, Pei Zhang, Baosong Yang, Derek Wong, Zhuosheng Zhang, and Rui Wang. Embedding trajectory for out-of-distribution detection in mathematical reasoning. *Advances in Neural Information Processing Systems*, 37:42965–42999, 2024b.

Yiyang Wang, Xi Chen, Xiaogang Xu, Sihui Ji, Yu Liu, Yujun Shen, and Hengshuang Zhao. Diffdoctor: Diagnosing image diffusion models before treating. *arXiv preprint arXiv:2501.12382*, 2025b.

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

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296, 2016.

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

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70268–70288, 2025b.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Boxuan Zhang, Jianing Zhu, Zengmao Wang, Tongliang Liu, Bo Du, and Bo Han. What if the input is expanded in ood detection? *Advances in Neural Information Processing Systems*, 37:21289–21329, 2024a.

Shuhai Zhang, Yiliao Song, Jiahao Yang, Yuanqing Li, Bo Han, and Mingkui Tan. Detecting machine-generated texts by multi-population aware optimization for maximum mean discrepancy. *arXiv preprint arXiv:2402.16041*, 2024b.

Shuhai Zhang, ZiHao Lian, Jiahao Yang, Daiyuan Li, Guoxuan Pang, Feng Liu, Bo Han, Shutao Li, and Mingkui Tan. Physics-driven spatiotemporal modeling for ai-generated video detection. *arXiv preprint arXiv:2510.08073*, 2025a.

Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, pp. 1–6. IEEE, 2019.

Yonggang Zhang, Jun Nie, Xinmei Tian, Mingming Gong, Kun Zhang, and Bo Han. Detecting generated images by fitting natural image distributions. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Nan Zhong, Yiran Xu, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Patchcraft: Exploring texture patch for efficient ai-generated image detection. *arXiv preprint arXiv:2311.12397*, 2023.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pp. 1–20, 2023.

Mingjian Zhu, Hanting Chen, Mouxiao Huang, Wei Li, Hailin Hu, Jie Hu, and Yunhe Wang. Gendet: Towards good generalizations for ai-generated image detection. *arXiv preprint arXiv:2312.08880*, 2023a.

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, 2023b.

APPENDIX FOR MDMF

The appendix is organized as follows. In Appendix B, we provide the complete proofs of all theoretical results stated in Section 2.4, together with the underlying assumptions and supporting lemmas. In Appendix C, we provide a more detailed discussion of related works on generative image modeling and AI-generated image detection. In Appendix D, we describe additional experimental setups, including dataset details, evaluation metrics, and implementation details. In Appendix E, we report additional experimental results, including evaluations on extra benchmarks (LSUN-Bedroom, GenImage), threshold-optimized ACC results, the full ablation tables, the PFS representation diagnostics, the patch-granularity, encoder-architecture, reference-size, post-processing-robustness, and patch-level hard-voting analyses, a B-Free/DDA complementarity analysis, and a failure-case analysis on borderline real images. In Appendix F, we present extended qualitative heatmaps across multiple generators. In Appendix G, we discuss reference selection and atypical real-image scores, and in Appendix H, we discuss the broader impact of our work.

A ALGORITHMS

We summarize the training and testing pipelines of MDMF in Algorithm 1 and 2, respectively.

Algorithm 1 Training MDMF

Input: Training real images $\mathcal{S}_{\mathbb{P}}^{tr}$, generated images $\mathcal{S}_{\mathbb{Q}}^{tr}$; projection head ϕ_θ; deep kernel k_ω; regularization λ; learning rate η
Initialize $\omega \leftarrow \{\theta_0, \gamma_0\}$
for $t = 1, 2, \ldots, T$ **do**
  Sample mini-batches $\{x_b\}_{b=1}^{B} \sim \mathcal{S}_{\mathbb{P}}^{tr}$ and $\{y_b\}_{b=1}^{B} \sim \mathcal{S}_{\mathbb{Q}}^{tr}$
  Form PFS vectors $\mathbf{Z}_\theta(x_b) \leftarrow [\mathbf{z}_1(x_b), \ldots, \mathbf{z}_K(x_b)]^\top$, $\mathbf{Z}_\theta(y_b) \leftarrow [\mathbf{z}_1(y_b), \ldots, \mathbf{z}_K(y_b)]^\top$
  Compute unbiased MMD $M(\omega) \leftarrow \widehat{\mathrm{MMD}}_u^2(\mathcal{S}_{\mathbb{P}}^{tr}, \mathcal{S}_{\mathbb{Q}}^{tr}; k_\omega)$ using Eqn. 3 and estimate variance $\hat{\sigma}_{H_1}^2$
  Optimize test-power objective $J_\lambda(\omega) \leftarrow \frac{M(\omega)}{\sqrt{\hat{\sigma}_{H_1}^2 + \lambda}}$ using Eqn. 5, $\omega \leftarrow \omega + \eta \nabla_{\mathrm{Adam}} J_\lambda(\omega)$
end for
Output: Trained projection head ϕ_{θ^*} and kernel k_{ω^*}

Algorithm 2 Detecting Images with MDMF

Input: Reference real images $\mathcal{S}_{\mathbb{P}}^{re}$; test images $\mathcal{S}^{te}$; trained ϕ_{θ^*}; kernel k_{ω^*}; threshold τ
Build reference PFS vector $\mathbf{Z}_\theta(x)$ from $x \sim \mathcal{S}_{\mathbb{P}}^{re}$
for $\tilde{y} \in \mathcal{S}^{te}$ **do**
  $\mathbf{Z}_\theta(\tilde{y}) \leftarrow [\mathbf{z}_1(\tilde{y}), \ldots, \mathbf{z}_K(\tilde{y})]^\top$
  $S_{\mathrm{MDMF}}(\tilde{y}) \leftarrow \widehat{\mathrm{MMD}}_b^2(\mathcal{S}_{\mathbb{P}}^{re}, \{\tilde{y}\}; k_{\omega^*})$ using Eqn. 6
  $f(\tilde{y}) \leftarrow \mathbb{I}(S_{\mathrm{MDMF}}(\tilde{y}) > \tau)$ using Eqn. 7
end for
Output: Predictions $\{f(\tilde{y})\}$

B THEORETICAL ANALYSIS

B.1 PRELIMINARIES AND MODELING ASSUMPTIONS

This section provides the probabilistic tools and regularity conditions used in our proofs. We formalize (i) sub-Gaussian patch embeddings, (ii) weak spatial dependence across patches (used only when relating patchwise and global pooling), and (iii) second-order regularity of the PFS mapping.

Sub-Gaussian random vectors. A random vector $X \in \mathbb{R}^m$ is called *σ-sub-Gaussian* if for all unit vectors $u \in \mathbb{S}^{m-1}$ and all $t \in \mathbb{R}$,

$$\mathbb{E}\Big[\exp\big(t\,u^\top(X - \mathbb{E}[X])\big)\Big] \le \exp\left(\frac{\sigma^2 t^2}{2}\right). \tag{14}$$

We denote by $\mathcal{SG}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$ a σ-sub-Gaussian distribution with mean $\boldsymbol{\mu}$ and isotropic proxy covariance $\sigma^2\mathbf{I}$.

Extracted patch embeddings. Real images are i.i.d. from $\mathbb{P}$ and generated images are i.i.d. from $\mathbb{Q}$. Given an image, we extract K non-overlapping patch embeddings $\{\mathbf{e}_i\}_{i=1}^K \subset \mathbb{R}^D$ from a fixed pre-trained encoder (e.g., DINOv2). Throughout the analysis, we assume each patch embedding is σ_e-sub-Gaussian:

$$\mathbf{e}_i(x) \sim \mathcal{SG}(\mathbf{0}, \sigma_e^2\mathbf{I}_D), \qquad \mathbf{e}_i(y) \text{ follows the sparse-defect model in Assumption 2.3.}$$

Weak spatial dependence across patches. Within one image, patch embeddings may exhibit spatial correlation. To quantify this, we model $\{\mathbf{e}_i\}_{i=1}^K$ as an α-mixing sequence. Let $\mathcal{F}_1^i$ be the σ-algebra generated by $\{\mathbf{e}_1, \ldots, \mathbf{e}_i\}$ and $\mathcal{F}_{i+\ell}^K$ generated by $\{\mathbf{e}_{i+\ell}, \ldots, \mathbf{e}_K\}$. The α-mixing coefficient is

$$\alpha(\ell) := \sup_i \sup_{A \in \mathcal{F}_1^i,\, B \in \mathcal{F}_{i+\ell}^K} \big|\mathbb{P}(A \cap B) - \mathbb{P}(A)\mathbb{P}(B)\big|. \tag{15}$$

We assume exponential mixing:

$$\alpha(\ell) \le C_\alpha e^{-c_\alpha \ell} \quad \text{for some constants } C_\alpha, c_\alpha > 0. \tag{16}$$

This assumption is only used to control covariance shrinkage after global pooling.

Effective sample size. Define an *effective* patch count

$$\frac{1}{K_{\text{eff}}} := \frac{1}{K} + \frac{2}{K^2} \sum_{\ell=1}^{K-1} (K - \ell)\, \beta(\ell), \tag{17}$$

where $\beta(\ell)$ upper-bounds cross-patch covariance contribution at lag ℓ (e.g., $\beta(\ell) \propto \alpha(\ell)^\eta$ for some $\eta \in (0, 1]$ under standard mixing-to-covariance bounds). Under exponential mixing (16), $\sum_{\ell \ge 1} \beta(\ell) < \infty$ and thus $K_{\text{eff}} = \Theta(K)$ (i.e., it scales linearly with K up to constants).

PFS mapping and second-order regularity. Let $\phi_\theta : \mathbb{R}^D \to \mathbb{R}^d$ be the learnable patch forensic signature (PFS) mapping, and write $\phi_\theta = (\phi_{\theta,1}, \ldots, \phi_{\theta,d})$.

Assumption B.1 (Locally Smooth PFS Mapping (Second-order)). There exist constants $L, M, R > 0$ and a neighborhood $\mathcal{E} \subset \mathbb{R}^D$ containing the typical support mass of both real and generated patch embeddings such that for all $\mathbf{e} \in \mathcal{E}$,

$$\|J_\phi(\mathbf{e})\|_{\text{op}} \le L, \qquad \|\nabla^2 \phi_{\theta,\ell}(\mathbf{e})\|_{\text{op}} \le M \;\text{ for all } \ell = 1, \ldots, d, \tag{18}$$

and the second-order Taylor remainder satisfies, for each ℓ,

$$\big|R_\ell(\mathbf{e})\big| \le \frac{R}{6}\|\mathbf{e}\|_2^3, \tag{19}$$

where $R_\ell(\mathbf{e})$ is the remainder term in the expansion of $\phi_{\theta,\ell}(\mathbf{e})$ around $\mathbf{0}$.

Remark B.2. Assumption B.1 is mild when ϕ_θ is implemented with smooth activations (e.g., GELU or $\tanh$) and embeddings are ℓ_2-normalized, which effectively restricts $\mathbf{e}$ to a compact region. The Hessian bound ensures the second-order term is controlled, and (19) formalizes that higher-order terms are negligible in the leading-order analysis.

B.2 Proof of Proposition 2.4

Proof. We prove the exact decomposition (9) by a second-order Taylor expansion.

Recall the definition of Δ_{PFS}:

$$\Delta_{\text{PFS}} := \mathbb{E}_{\mathbb{Q}}[\phi_\theta(\mathbf{e}_i(y))] - \mathbb{E}_{\mathbb{P}}[\phi_\theta(\mathbf{e}_i(x))]\,.$$

Under Assumption 2.3, we have

$$\mathbf{e}_i(x) = \mathbf{u}_i, \qquad \mathbf{e}_i(y) = \mathbf{u}_i + a_i s_i \boldsymbol{\mu}_{\text{defect}},$$

where $\mathbf{u}_i \sim \mathcal{SG}(\mathbf{0}, \sigma_e^2 \mathbf{I}_D)$, $a_i \sim \text{Bernoulli}(\rho)$, and s_i is Rademacher independent of $\mathbf{u}_i, a_i$.

First, we compute the mean of generated patch embeddings:

$$\begin{aligned}\mathbb{E}[\mathbf{e}_i(y)] &= \mathbb{E}[\mathbf{u}_i] + \mathbb{E}[a_i s_i]\, \boldsymbol{\mu}_{\text{defect}} \\ &= \mathbf{0} + \mathbb{E}[a_i]\mathbb{E}[s_i]\, \boldsymbol{\mu}_{\text{defect}} \qquad \text{(by independence of } a_i \text{ and } s_i) \\ &= \rho \cdot 0 \cdot \boldsymbol{\mu}_{\text{defect}} = \mathbf{0},\end{aligned}$$

and similarly $\mathbb{E}[\mathbf{e}_i(x)] = \mathbb{E}[\mathbf{u}_i] = \mathbf{0}$. Hence any leading-order shift cannot arise from the linear (Jacobian) term.

Compute Second-order Taylor expansion of ϕ_θ. Let $\phi_\theta = (\phi_{\theta,1}, \ldots, \phi_{\theta,d})$. For each output coordinate $\ell \in \{1, \ldots, d\}$, since ϕ_θ is twice differentiable at $\mathbf{0}$, a second-order Taylor expansion around $\mathbf{0}$ yields

$$\phi_{\theta,\ell}(\mathbf{e}) = \phi_{\theta,\ell}(\mathbf{0}) + \nabla\phi_{\theta,\ell}(\mathbf{0})^\top \mathbf{e} + \frac{1}{2}\mathbf{e}^\top \nabla^2 \phi_{\theta,\ell}(\mathbf{0})\mathbf{e} + R_\ell(\mathbf{e}), \tag{20}$$

where the remainder $R_\ell(\mathbf{e}) = o(\|\mathbf{e}\|_2^2)$ as $\|\mathbf{e}\|_2 \to 0$.

Apply (20) to $\mathbf{e} = \mathbf{e}_i(y)$ and $\mathbf{e} = \mathbf{e}_i(x)$ and take expectations:

$$\mathbb{E}_{\mathbb{Q}}\big[\phi_{\theta,\ell}(\mathbf{e}_i(y))\big] = \phi_{\theta,\ell}(\mathbf{0}) + \nabla\phi_{\theta,\ell}(\mathbf{0})^\top \mathbb{E}[\mathbf{e}_i(y)] + \frac{1}{2}\mathbb{E}\big[\mathbf{e}_i(y)^\top \nabla^2 \phi_{\theta,\ell}(\mathbf{0})\mathbf{e}_i(y)\big] + \mathbb{E}[R_\ell(\mathbf{e}_i(y))], \tag{21}$$

$$\mathbb{E}_{\mathbb{P}}\big[\phi_{\theta,\ell}(\mathbf{e}_i(x))\big] = \phi_{\theta,\ell}(\mathbf{0}) + \nabla\phi_{\theta,\ell}(\mathbf{0})^\top \mathbb{E}[\mathbf{e}_i(x)] + \frac{1}{2}\mathbb{E}\big[\mathbf{e}_i(x)^\top \nabla^2 \phi_{\theta,\ell}(\mathbf{0})\mathbf{e}_i(x)\big] + \mathbb{E}[R_\ell(\mathbf{e}_i(x))]. \tag{22}$$

Subtracting (22) from (21), the constant terms cancel. Since $\mathbb{E}[\mathbf{e}_i(y)] = \mathbb{E}[\mathbf{e}_i(x)] = \mathbf{0}$, so the linear terms also vanish. Therefore,

$$\Delta_{\text{PFS},\ell} := \mathbb{E}_{\mathbb{Q}}\big[\phi_{\theta,\ell}(\mathbf{e}_i(y))\big] - \mathbb{E}_{\mathbb{P}}\big[\phi_{\theta,\ell}(\mathbf{e}_i(x))\big] = \frac{1}{2}\Big(\mathbb{E}[\mathbf{e}_i(y)^\top H_\ell \mathbf{e}_i(y)] - \mathbb{E}[\mathbf{e}_i(x)^\top H_\ell \mathbf{e}_i(x)]\Big) + \epsilon_\ell, \tag{23}$$

where we write $H_\ell := \nabla^2\phi_{\theta,\ell}(\mathbf{0})$ and group the remainder difference into

$$\epsilon_\ell := \mathbb{E}[R_\ell(\mathbf{e}_i(y))] - \mathbb{E}[R_\ell(\mathbf{e}_i(x))], \quad \text{which is higher order.}$$

We now compute the difference $\mathbb{E}[\mathbf{e}^\top H_\ell \mathbf{e}]$ under x and y.

Under the real distribution, $\mathbf{e}_i(x) = \mathbf{u}_i$:

$$\mathbb{E}[\mathbf{e}_i(x)^\top H_\ell \mathbf{e}_i(x)] = \mathbb{E}[\mathbf{u}_i^\top H_\ell \mathbf{u}_i].$$

Under the generated distribution, $\mathbf{e}_i(y) = \mathbf{u}_i + a_i s_i \boldsymbol{\mu}_{\text{defect}}$:

$$\begin{aligned}\mathbb{E}[\mathbf{e}_i(y)^\top H_\ell \mathbf{e}_i(y)] &= \mathbb{E}\Big[(\mathbf{u}_i + a_i s_i \boldsymbol{\mu}_{\text{defect}})^\top H_\ell (\mathbf{u}_i + a_i s_i \boldsymbol{\mu}_{\text{defect}})\Big] \\ &= \mathbb{E}[\mathbf{u}_i^\top H_\ell \mathbf{u}_i] + 2\,\mathbb{E}[a_i s_i]\,\mathbb{E}[\boldsymbol{\mu}_{\text{defect}}^\top H_\ell \mathbf{u}_i] + \mathbb{E}[(a_i s_i)^2]\,\boldsymbol{\mu}_{\text{defect}}^\top H_\ell \boldsymbol{\mu}_{\text{defect}}.\end{aligned} \tag{24}$$

Here we used bilinearity of the quadratic expansion and independence to separate expectations in the cross term, so we have

$$\mathbb{E}[a_i s_i] = \mathbb{E}[a_i]\mathbb{E}[s_i] = \rho \cdot 0 = 0,$$

so the cross term in (24) vanishes. Moreover, since $s_i^2 = 1$ and $a_i \in \{0, 1\}$, we have $(a_i s_i)^2 = a_i$ and therefore

$$\mathbb{E}[(a_i s_i)^2] = \mathbb{E}[a_i] = \rho.$$

Plugging these into (24) yields

$$\begin{aligned}\mathbb{E}[\mathbf{e}_i(y)^\top H_\ell \mathbf{e}_i(y)] &= \mathbb{E}[\mathbf{u}_i^\top H_\ell \mathbf{u}_i] + \rho\, \boldsymbol{\mu}_{\text{defect}}^\top H_\ell \boldsymbol{\mu}_{\text{defect}} \\ &= \mathbb{E}[\mathbf{e}_i(x)^\top H_\ell \mathbf{e}_i(x)] + \rho\, \boldsymbol{\mu}_{\text{defect}}^\top H_\ell \boldsymbol{\mu}_{\text{defect}}\end{aligned}$$

Hence,

$$\mathbb{E}[\mathbf{e}_i(y)^\top H_\ell \mathbf{e}_i(y)] - \mathbb{E}[\mathbf{e}_i(x)^\top H_\ell \mathbf{e}_i(x)] = \rho\, \boldsymbol{\mu}_{\text{defect}}^\top H_\ell \boldsymbol{\mu}_{\text{defect}}. \tag{25}$$

Substituting (25) into (23) gives, for every coordinate ℓ,

$$\Delta_{\text{PFS},\ell} = \frac{\rho}{2}\, \boldsymbol{\mu}_{\text{defect}}^\top \nabla^2 \phi_{\theta,\ell}(\mathbf{0}) \boldsymbol{\mu}_{\text{defect}} + \epsilon_\ell.$$

Stacking the coordinates proves (9). The reverse triangle inequality yields $\|\Delta_{\text{PFS}}\|_2 \geq \frac{\rho}{2}\|\mathcal{Q}(\boldsymbol{\mu}_{\text{defect}})\|_2 - \|\boldsymbol{\epsilon}\|_2$, proving the stated sufficient condition for a nonzero shift. □

B.3 Proof of Proposition 2.5

Proof. We show that global pooling dilutes the same second-order defect signature by a factor $1/K$, hence $\|\Delta_{\text{PFS}}\|_2 \approx K\|\Delta_{\text{global}}\|_2$ at leading order.

Recall the definition of the global pooled embeddings:

$$\bar{\mathbf{e}}(x) = \frac{1}{K}\sum_{i=1}^{K} \mathbf{e}_i(x), \qquad \bar{\mathbf{e}}(y) = \frac{1}{K}\sum_{i=1}^{K} \mathbf{e}_i(y).$$

Under Assumption 2.3,

$$\mathbf{e}_i(x) = \mathbf{u}_i, \qquad \mathbf{e}_i(y) = \mathbf{u}_i + a_i s_i \boldsymbol{\mu}_{\text{defect}}.$$

Similar to the proof of Proposition 2.4, if we simply adopt the linearity of expectation:

$$\mathbb{E}[\bar{\mathbf{e}}(x)] = \frac{1}{K}\sum_{i=1}^{K} \mathbb{E}[\mathbf{u}_i] = \mathbf{0}, \qquad \mathbb{E}[\bar{\mathbf{e}}(y)] = \frac{1}{K}\sum_{i=1}^{K} \mathbb{E}[\mathbf{u}_i + a_i s_i \boldsymbol{\mu}_{\text{defect}}] = \mathbf{0}.$$

Thus, as in Proposition 2.4, the leading-order shift arises from second-order terms. For each coordinate ℓ, we apply the same second-order expansion at $\mathbf{0}$:

$$\phi_{\theta,\ell}(\bar{\mathbf{e}}) = \phi_{\theta,\ell}(\mathbf{0}) + \nabla\phi_{\theta,\ell}(\mathbf{0})^\top \bar{\mathbf{e}} + \frac{1}{2}\bar{\mathbf{e}}^\top H_\ell \bar{\mathbf{e}} + R_\ell(\bar{\mathbf{e}}), \quad H_\ell = \nabla^2\phi_{\theta,\ell}(\mathbf{0}).$$

According to Appendix B.2, taking expectations and subtracting between y and x, the constant and linear terms cancel since $\mathbb{E}[\bar{\mathbf{e}}(y)] = \mathbb{E}[\bar{\mathbf{e}}(x)] = \mathbf{0}$:

$$\Delta_{\text{global},\ell} := \mathbb{E}_{\mathbb{Q}}\big[\phi_{\theta,\ell}(\bar{\mathbf{e}}(y))\big] - \mathbb{E}_{\mathbb{P}}\big[\phi_{\theta,\ell}(\bar{\mathbf{e}}(x))\big] = \frac{1}{2}\Big(\mathbb{E}[\bar{\mathbf{e}}(y)^\top H_\ell \bar{\mathbf{e}}(y)] - \mathbb{E}[\bar{\mathbf{e}}(x)^\top H_\ell \bar{\mathbf{e}}(x)]\Big) + \tilde{\epsilon}_\ell, \tag{26}$$

where $\tilde{\epsilon}_\ell$ collects higher-order remainder differences.

Then we **reduce the pooled quadratic forms to covariances**. Since $\mathbb{E}[\bar{\mathbf{e}}(x)] = \mathbb{E}[\bar{\mathbf{e}}(y)] = \mathbf{0}$, we use $\mathbb{E}[\mathbf{v}^\top A\mathbf{v}] = \operatorname{tr}(A\,\mathrm{Cov}(\mathbf{v}))$ to write

$$\mathbb{E}[\bar{\mathbf{e}}(y)^\top H_\ell \bar{\mathbf{e}}(y)] - \mathbb{E}[\bar{\mathbf{e}}(x)^\top H_\ell \bar{\mathbf{e}}(x)] = \operatorname{tr}\Big(H_\ell\big(\mathrm{Cov}(\bar{\mathbf{e}}(y)) - \mathrm{Cov}(\bar{\mathbf{e}}(x))\big)\Big). \tag{27}$$

We expand the covariance of the pooled embedding, since $\bar{\mathbf{e}} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{e}_i.$, then

$$\begin{aligned}
\mathrm{Cov}(\bar{\mathbf{e}}) &= \mathbb{E}\Big[\big(\bar{\mathbf{e}} - \mathbb{E}[\bar{\mathbf{e}}]\big)\big(\bar{\mathbf{e}} - \mathbb{E}[\bar{\mathbf{e}}]\big)^\top\Big] \\
&= \mathbb{E}\left[\Big(\frac{1}{K}\sum_{i=1}^{K}(\mathbf{e}_i - \mathbb{E}[\mathbf{e}_i])\Big)\Big(\frac{1}{K}\sum_{j=1}^{K}(\mathbf{e}_j - \mathbb{E}[\mathbf{e}_j])\Big)^\top\right] \\
&= \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}\mathbb{E}\big[(\mathbf{e}_i - \mathbb{E}[\mathbf{e}_i])(\mathbf{e}_j - \mathbb{E}[\mathbf{e}_j])^\top\big] \\
&= \frac{1}{K^2}\sum_{i=1}^{K}\mathrm{Cov}(\mathbf{e}_i) + \frac{1}{K^2}\sum_{i\neq j}\mathrm{Cov}(\mathbf{e}_i, \mathbf{e}_j) \\
&= \frac{1}{K^2}\sum_{i=1}^{K}\mathrm{Cov}(\mathbf{e}_i) + \frac{1}{K^2}\sum_{1\le i<j\le K}\Big(\mathrm{Cov}(\mathbf{e}_i, \mathbf{e}_j) + \mathrm{Cov}(\mathbf{e}_i, \mathbf{e}_j)^\top\Big) \\
&= \frac{1}{K^2}\sum_{i=1}^{K}\mathrm{Cov}(\mathbf{e}_i) + \frac{2}{K^2}\sum_{1\le i<j\le K}\mathrm{Sym}\big(\mathrm{Cov}(\mathbf{e}_i, \mathbf{e}_j)\big),
\end{aligned} \tag{28}$$

where $\mathrm{Sym}(A) := (A + A^\top)/2$. Noting that only the symmetric part of $\mathrm{Cov}(\bar{\mathbf{e}})$ contributes to the second-order expansion, and it only appears through quadratic forms and trace operators, so we replace each $\mathrm{Cov}(\mathbf{e}_i, \mathbf{e}_j)$ by its symmetric part $\mathrm{Sym}(\mathrm{Cov}(\mathbf{e}_i, \mathbf{e}_j))$, yielding a factor of 2 when summing over $i < j$. Using (28) for both x and y, we can write the pooled covariance difference exactly as

$$\begin{aligned}
\mathrm{Cov}(\bar{\mathbf{e}}(y)) - \mathrm{Cov}(\bar{\mathbf{e}}(x)) = \frac{1}{K^2}\sum_{i=1}^{K}\Big(\mathrm{Cov}(\mathbf{e}_i(y)) - \mathrm{Cov}(\mathbf{e}_i(x))\Big) \\
+ \frac{2}{K^2}\sum_{1\le i<j\le K}\mathrm{Sym}\Big(\mathrm{Cov}(\mathbf{e}_i(y), \mathbf{e}_j(y)) - \mathrm{Cov}(\mathbf{e}_i(x), \mathbf{e}_j(x))\Big).
\end{aligned} \tag{29}$$

Recall $e_i(x) = \mathbf{u}_i$ and $e_i(y) = \mathbf{u}_i + d_i\boldsymbol{\mu}_{\mathrm{defect}}$ with $d_i = a_i s_i$. Under the stated assumptions $\mathbb{E}[d_i] = 0$ and the independence between $\{d_i\}$ and $\{\mathbf{u}_i\}$, all mixed cross-terms vanish. Based on Assumption 2.3, to complete this proof, we further assume that the signed defect indicators $\{d_i\}_{i=1}^{K}$ exhibit at most weak spatial dependence, i.e., $|\mathrm{Cov}(d_i, d_{i+\ell})| \le \rho\,\beta(\ell)$ with $\sum_{\ell\ge1}\beta(\ell) < \infty$. We thus obtain for any $i \neq j$:

$$\mathrm{Cov}(\mathbf{e}_i(y), \mathbf{e}_j(y)) - \mathrm{Cov}(\mathbf{e}_i(x), \mathbf{e}_j(x)) = \mathrm{Cov}(d_i, d_j)\,\boldsymbol{\mu}_{\mathrm{defect}}\boldsymbol{\mu}_{\mathrm{defect}}^\top. \tag{30}$$

Similarly, for the diagonal term we have (cf. Proposition 2.4)

$$\mathrm{Cov}(\mathbf{e}_i(y)) - \mathrm{Cov}(\mathbf{e}_i(x)) = \mathrm{Var}(d_i)\,\boldsymbol{\mu}_{\mathrm{defect}}\boldsymbol{\mu}_{\mathrm{defect}}^\top = \rho\,\boldsymbol{\mu}_{\mathrm{defect}}\boldsymbol{\mu}_{\mathrm{defect}}^\top. \tag{31}$$

Plugging (30)–(31) into (29) yields

$$\mathrm{Cov}(\bar{\mathbf{e}}(y)) - \mathrm{Cov}(\bar{\mathbf{e}}(x)) = \frac{\rho}{K}\,\boldsymbol{\mu}_{\mathrm{defect}}\boldsymbol{\mu}_{\mathrm{defect}}^\top + \frac{2}{K^2}\sum_{1\le i<j\le K}\mathrm{Cov}(d_i, d_j)\,\boldsymbol{\mu}_{\mathrm{defect}}\boldsymbol{\mu}_{\mathrm{defect}}^\top. \tag{32}$$

Under the mixing decay $|\mathrm{Cov}(d_i, d_{i+\ell})| \le \rho\,\beta(\ell)$ and stationarity, the off-diagonal sum is bounded by

$$\sum_{1\le i<j\le K}|\mathrm{Cov}(d_i, d_j)| \le \sum_{\ell=1}^{K-1}(K-\ell)\rho\,\beta(\ell),$$

and therefore, in operator norm,

$$\big\|\mathrm{Cov}(\bar{\mathbf{e}}(y)) - \mathrm{Cov}(\bar{\mathbf{e}}(x))\big\|_{\mathrm{op}} \le \left(\frac{\rho}{K} + \frac{2\rho}{K^2}\sum_{\ell=1}^{K-1}(K-\ell)\beta(\ell)\right)\big\|\boldsymbol{\mu}_{\mathrm{defect}}\boldsymbol{\mu}_{\mathrm{defect}}^\top\big\|_{\mathrm{op}} = \frac{\rho}{K_{\mathrm{eff}}}\,\|\boldsymbol{\mu}_{\mathrm{defect}}\|_2^2, \tag{33}$$

where K_{eff} is defined in (17). In particular, if $\{d_i\}$ is independent across patches, then $\mathrm{Cov}(d_i, d_j) = 0$ for $i \neq j$ and the bound is tight with $K_{\text{eff}} = K$. Moreover, under exponential mixing $\sum_{\ell \geq 1} \beta(\ell) < \infty$, we have $K_{\text{eff}} = \Theta(K)$.

Repeating the same second-order Taylor argument as in Proposition 2.4 with $\mathbf{e}$ replaced by $\bar{\mathbf{e}}$ yields, for each coordinate ℓ,

$$\Delta_{\text{global},\ell} \approx \frac{1}{2} \boldsymbol{\mu}_{\text{defect}}^{\top} \nabla^2 \phi_{\theta,\ell}(\mathbf{0}) \boldsymbol{\mu}_{\text{defect}} \cdot \frac{\rho}{K_{\text{eff}}} = \frac{\rho}{2K_{\text{eff}}} \left[\mathcal{Q}(\boldsymbol{\mu}_{\text{defect}})\right]_\ell.$$

Stacking $\ell = 1, \ldots, d$ gives

$$\Delta_{\text{global}} \approx \frac{\rho}{2K_{\text{eff}}} \mathcal{Q}(\boldsymbol{\mu}_{\text{defect}}).$$

Under the Proposition 2.4, $\Delta_{\text{PFS}} \approx \frac{\rho}{2} \mathcal{Q}(\boldsymbol{\mu}_{\text{defect}})$, hence

$$\Delta_{\text{global}} \approx \frac{1}{K_{\text{eff}}} \Delta_{\text{PFS}}.$$

Therefore,

$$\|\Delta_{\text{PFS}}\|_2 \approx K_{\text{eff}} \|\Delta_{\text{global}}\|_2.$$

Under exponential mixing, $K_{\text{eff}} = \Theta(K)$, hence the patch-level shift dominates the global-pooled shift by a factor linear in K up to constants, consistent with (10):

$$\|\Delta_{\text{PFS}}\|_2 \approx K \, \|\Delta_{\text{global}}\|_2 > \|\Delta_{\text{global}}\|_2, \tag{34}$$

□

B.4 Proof of Proposition 2.6

On the Gaussian surrogate in PFS space. The closed-form calculation below assumes equal, isotropic Gaussian covariance for the two vectorized PFS fields. This is an additional surrogate model; the sparse-defect assumption and sub-Gaussian patch embeddings do not imply it. In particular, a sparse defect can change covariance. The positivity and monotonicity in $\|\Delta_{\text{PFS}}\|_2$ proved below hold within this fixed-covariance surrogate. For general PFS distributions, a Gaussian RBF kernel gives positive MMD when their feature-field laws differ, but a larger mean difference alone need not increase MMD.

Proof. We prove (11), and the positivity and monotonicity claims in (11).

Recall the definition of our gaussian deep kernel in 4:

$$k_\omega(x, y) = \exp\left(-\frac{\|\mathbf{Z}_\theta(x) - \mathbf{Z}_\theta(y)\|_2^2}{2\gamma^2}\right) =: \; k_\gamma\big(\mathbf{Z}_\theta(x), \mathbf{Z}_\theta(y)\big),$$

where $\mathbf{Z}_\theta(x) \in \mathbb{R}^{K \times d}$ is the Patch Signature Field. Here $\|\cdot\|_2$ denotes the entry-wise Euclidean norm of the field (i.e., the ℓ_2 norm after flattening), which coincides with the Frobenius norm on $\mathbb{R}^{K \times d}$. Thus k_γ is a Gaussian RBF kernel on the ambient Euclidean space $\mathbb{R}^{K \times d}$ (equivalently $\mathbb{R}^{Kd}$).

Define the (random) feature fields induced by $\mathbf{Z}_\theta$:

$$\mathbf{X} := \mathbf{Z}_\theta(x), \quad x \sim \mathbb{P}, \qquad \mathbf{Y} := \mathbf{Z}_\theta(y), \quad y \sim \mathbb{Q},$$

and similarly $\mathbf{X}' := \mathbf{Z}_\theta(x')$ for an independent copy $x' \sim \mathbb{P}$ and $\mathbf{Y}' := \mathbf{Z}_\theta(y')$ for an independent copy $y' \sim \mathbb{Q}$. With this notation, the population MMD under k_ω admits the standard expansion

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega) = \mathbb{E}\big[k_\gamma(\mathbf{X}, \mathbf{X}')\big] + \mathbb{E}\big[k_\gamma(\mathbf{Y}, \mathbf{Y}')\big] - 2\,\mathbb{E}\big[k_\gamma(\mathbf{X}, \mathbf{Y})\big], \tag{35}$$

where $\mathbf{X}, \mathbf{X}', \mathbf{Y}, \mathbf{Y}' \in \mathbb{R}^{K \times d}$ and $k_\gamma(A, B) = \exp\Big(-\frac{\|A-B\|_2^2}{2\gamma^2}\Big)$.

To obtain a closed-form expression under the Gaussian kernel k_γ on $\mathbb{R}^{K \times d}$, we adopt the following Gaussian surrogate for the feature fields:

$$\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma_z^2 \mathbf{I}_{Kd}), \qquad \mathbf{Y} \sim \mathcal{N}(\boldsymbol{\Delta}_{\mathbf{Z}}, \sigma_z^2 \mathbf{I}_{Kd}),$$

and similarly for independent copies $\mathbf{X}', \mathbf{Y}'$. Here $\boldsymbol{\Delta}_{\mathbf{Z}} := \mathbb{E}[\mathbf{Z}_\theta(y)] - \mathbb{E}[\mathbf{Z}_\theta(x)] \in \mathbb{R}^{K\times d}$ denotes the mean shift of the Patch Signature Field, and $\mathbf{I}_{Kd}$ denotes isotropic covariance under the entry-wise Euclidean structure of $\mathbb{R}^{K\times d}$. In particular, if each patch undergoes the same PFS mean shift $\boldsymbol{\Delta}_{\mathrm{PFS}} \in \mathbb{R}^d$, then $\boldsymbol{\Delta}_{\mathbf{Z}} = \mathbf{1}_K \boldsymbol{\Delta}_{\mathrm{PFS}}^\top$ and hence $\|\boldsymbol{\Delta}_{\mathbf{Z}}\|_2^2 = K\|\boldsymbol{\Delta}_{\mathrm{PFS}}\|_2^2$.

Here σ_z^2 is a covariance parameter of the stated Gaussian surrogate, not a bound inferred from the sub-Gaussian proxy σ_e. Under this surrogate, the expectations in (35) reduce to Gaussian integrals in $\mathbb{R}^{Kd}$.

Step 1: Compute the self-terms $\mathbb{E}[k_\gamma(\mathbf{X}, \mathbf{X}')]$ and $\mathbb{E}[k_\gamma(\mathbf{Y}, \mathbf{Y}')]$. Let $\mathbf{X}, \mathbf{X}' \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \sigma_z^2 \mathbf{I}_{Kd})$ and define $\boldsymbol{\delta} := \mathbf{X} - \mathbf{X}'$. Then $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, 2\sigma_z^2 \mathbf{I}_{Kd})$ and

$$\begin{aligned}\mathbb{E}\big[k_\gamma(\mathbf{X}, \mathbf{X}')\big] &= \mathbb{E}_{\boldsymbol{\delta}} \exp\left(-\frac{\|\boldsymbol{\delta}\|_2^2}{2\gamma^2}\right) \\ &= \int_{\mathbb{R}^{Kd}} \exp\left(-\frac{\|\boldsymbol{\delta}\|_2^2}{2\gamma^2}\right) (2\pi)^{-Kd/2} (2\sigma_z^2)^{-Kd/2} \exp\left(-\frac{\|\boldsymbol{\delta}\|_2^2}{4\sigma_z^2}\right) d\boldsymbol{\delta} \\ &= (2\pi)^{-Kd/2} (2\sigma_z^2)^{-Kd/2} \int_{\mathbb{R}^{Kd}} \exp\left(-\|\boldsymbol{\delta}\|_2^2 \left(\frac{1}{2\gamma^2} + \frac{1}{4\sigma_z^2}\right)\right) d\boldsymbol{\delta}. \end{aligned} \tag{36}$$

Define

$$A := \left(\frac{1}{\gamma^2} + \frac{1}{2\sigma_z^2}\right) \mathbf{I}_{Kd},$$

so that

$$\|\boldsymbol{\delta}\|_2^2 \left(\frac{1}{2\gamma^2} + \frac{1}{4\sigma_z^2}\right) = \frac{1}{2}\, \boldsymbol{\delta}^\top A \boldsymbol{\delta}.$$

Hence

$$\mathbb{E}\big[k_\gamma(\mathbf{X}, \mathbf{X}')\big] = (2\pi)^{-Kd/2} (2\sigma_z^2)^{-Kd/2} \int_{\mathbb{R}^{Kd}} \exp\left(-\frac{1}{2}\, \boldsymbol{\delta}^\top A \boldsymbol{\delta}\right) d\boldsymbol{\delta}. \tag{37}$$

Using the Gaussian integral identity

$$\int_{\mathbb{R}^{Kd}} \exp\big(-\tfrac{1}{2}\, u^\top A u\big)\, du = (2\pi)^{Kd/2} |A|^{-1/2},$$

we obtain

$$\mathbb{E}\big[k_\gamma(\mathbf{X}, \mathbf{X}')\big] = (2\sigma_z^2)^{-Kd/2}\, |A|^{-1/2}. \tag{38}$$

Since $A = c\mathbf{I}_{Kd}$ with

$$c = \frac{1}{\gamma^2} + \frac{1}{2\sigma_z^2} = \frac{\gamma^2 + 2\sigma_z^2}{2\sigma_z^2\gamma^2},$$

we have $|A| = c^{Kd}$ and $|A|^{-1/2} = c^{-Kd/2}$, yielding

$$\mathbb{E}\big[k_\gamma(\mathbf{X}, \mathbf{X}')\big] = (2\sigma_z^2)^{-Kd/2} \left(\frac{2\sigma_z^2\gamma^2}{\gamma^2 + 2\sigma_z^2}\right)^{Kd/2} = \left(\frac{\gamma^2}{\gamma^2 + 2\sigma_z^2}\right)^{Kd/2}.$$

Therefore,

$$\mathbb{E}\big[k_\gamma(\mathbf{X}, \mathbf{X}')\big] = \left(\frac{\gamma^2}{\gamma^2 + 2\sigma_z^2}\right)^{\frac{Kd}{2}}. \tag{39}$$

Let $\mathbf{Y}, \mathbf{Y}' \overset{i.i.d.}{\sim} \mathcal{N}(\boldsymbol{\Delta}_{\mathbf{Z}}, \sigma_z^2 \mathbf{I}_{Kd})$ and define $\boldsymbol{\delta}' := \mathbf{Y} - \mathbf{Y}'$. Then $\boldsymbol{\delta}' \sim \mathcal{N}(\mathbf{0}, 2\sigma_z^2 \mathbf{I}_{Kd})$ (the means cancel), so the same calculation as Step 1 yields

$$\mathbb{E}\big[k_\gamma(\mathbf{Y}, \mathbf{Y}')\big] = \left(\frac{\gamma^2}{\gamma^2 + 2\sigma_z^2}\right)^{\frac{Kd}{2}}. \tag{40}$$

Step 2: Compute the cross-term $\mathbb{E}[k_\gamma(\mathbf{X},\mathbf{Y})]$. Let $\mathbf{X}\sim\mathcal{N}(\mathbf{0},\sigma_z^2\mathbf{I}_{Kd})$ and $\mathbf{Y}\sim\mathcal{N}(\boldsymbol{\Delta}_\mathbf{Z},\sigma_z^2\mathbf{I}_{Kd})$ be independent. Define $\boldsymbol{\eta}:=\mathbf{X}-\mathbf{Y}$. Then $\boldsymbol{\eta}\sim\mathcal{N}(-\boldsymbol{\Delta}_\mathbf{Z},2\sigma_z^2\mathbf{I}_{Kd})$ and

$$\begin{aligned}\mathbb{E}\big[k_\gamma(\mathbf{X},\mathbf{Y})\big] &= \mathbb{E}_{\boldsymbol{\eta}}\exp\!\left(-\frac{\|\boldsymbol{\eta}\|_2^2}{2\gamma^2}\right)\\ &= \int_{\mathbb{R}^{Kd}}\exp\!\left(-\frac{\|\boldsymbol{\eta}\|_2^2}{2\gamma^2}\right)(2\pi)^{-Kd/2}(2\sigma_z^2)^{-Kd/2}\exp\!\left(-\frac{\|\boldsymbol{\eta}+\boldsymbol{\Delta}\|_2^2}{4\sigma_z^2}\right)d\boldsymbol{\eta}, \qquad (41)\end{aligned}$$

where $\boldsymbol{\Delta}:=\boldsymbol{\Delta}_\mathbf{Z}$ for brevity. Expanding $\|\boldsymbol{\eta}+\boldsymbol{\Delta}\|_2^2=\|\boldsymbol{\eta}\|_2^2+2\boldsymbol{\eta}^\top\boldsymbol{\Delta}+\|\boldsymbol{\Delta}\|_2^2$, the exponent becomes

$$\begin{aligned}-\frac{\|\boldsymbol{\eta}\|_2^2}{2\gamma^2}-\frac{\|\boldsymbol{\eta}+\boldsymbol{\Delta}\|_2^2}{4\sigma_z^2} &= -\left(\frac{1}{2\gamma^2}+\frac{1}{4\sigma_z^2}\right)\|\boldsymbol{\eta}\|_2^2\\ &\quad -\frac{1}{2\sigma_z^2}\boldsymbol{\eta}^\top\boldsymbol{\Delta}-\frac{\|\boldsymbol{\Delta}\|_2^2}{4\sigma_z^2}. \qquad (42)\end{aligned}$$

As in Step 1, set

$$A:=\left(\frac{1}{\gamma^2}+\frac{1}{2\sigma_z^2}\right)\mathbf{I}_{Kd},\qquad b:=\frac{1}{2\sigma_z^2}\boldsymbol{\Delta},$$

so that

$$-\left(\frac{1}{2\gamma^2}+\frac{1}{4\sigma_z^2}\right)\|\boldsymbol{\eta}\|_2^2-\frac{1}{2\sigma_z^2}\boldsymbol{\eta}^\top\boldsymbol{\Delta}=-\frac{1}{2}\,\boldsymbol{\eta}^\top A\boldsymbol{\eta}-b^\top\boldsymbol{\eta}.$$

Plugging into (41) gives

$$\mathbb{E}\big[k_\gamma(\mathbf{X},\mathbf{Y})\big]=(2\pi)^{-Kd/2}(2\sigma_z^2)^{-Kd/2}\exp\!\left(-\frac{\|\boldsymbol{\Delta}\|_2^2}{4\sigma_z^2}\right)\int_{\mathbb{R}^{Kd}}\exp\!\left(-\frac{1}{2}\,\boldsymbol{\eta}^\top A\boldsymbol{\eta}-b^\top\boldsymbol{\eta}\right)d\boldsymbol{\eta}. \qquad (43)$$

Using

$$\frac{1}{2}\,\boldsymbol{\eta}^\top A\boldsymbol{\eta}+b^\top\boldsymbol{\eta}=\frac{1}{2}(\boldsymbol{\eta}+A^{-1}b)^\top A(\boldsymbol{\eta}+A^{-1}b)-\frac{1}{2}b^\top A^{-1}b,$$

we obtain

$$\begin{aligned}\int_{\mathbb{R}^{Kd}}\exp\!\left(-\frac{1}{2}\,\boldsymbol{\eta}^\top A\boldsymbol{\eta}-b^\top\boldsymbol{\eta}\right)d\boldsymbol{\eta} &= \exp\!\left(\frac{1}{2}b^\top A^{-1}b\right)\int_{\mathbb{R}^{Kd}}\exp\!\left(-\frac{1}{2}(\boldsymbol{\eta}+A^{-1}b)^\top A(\boldsymbol{\eta}+A^{-1}b)\right)d\boldsymbol{\eta}\\ &= \exp\!\left(\frac{1}{2}b^\top A^{-1}b\right)\cdot(2\pi)^{Kd/2}|A|^{-1/2}. \qquad (44)\end{aligned}$$

Substituting (44) into (43) yields

$$\mathbb{E}\big[k_\gamma(\mathbf{X},\mathbf{Y})\big]=(2\sigma_z^2)^{-Kd/2}|A|^{-1/2}\exp\!\left(-\frac{\|\boldsymbol{\Delta}\|_2^2}{4\sigma_z^2}+\frac{1}{2}b^\top A^{-1}b\right). \qquad (45)$$

As before, $A=c\mathbf{I}_{Kd}$ with

$$c=\frac{\gamma^2+2\sigma_z^2}{2\sigma_z^2\gamma^2},$$

so

$$(2\sigma_z^2)^{-Kd/2}|A|^{-1/2}=(2\sigma_z^2)^{-Kd/2}c^{-Kd/2}=\left(\frac{\gamma^2}{\gamma^2+2\sigma_z^2}\right)^{Kd/2}.$$

Since $A^{-1}=\frac{1}{c}\mathbf{I}_{Kd}=\frac{2\sigma_z^2\gamma^2}{\gamma^2+2\sigma_z^2}\mathbf{I}_{Kd}$ and $b=\frac{1}{2\sigma_z^2}\boldsymbol{\Delta}$, we have

$$\begin{aligned}b^\top A^{-1}b &= \left(\frac{1}{2\sigma_z^2}\boldsymbol{\Delta}\right)^\top\left(\frac{2\sigma_z^2\gamma^2}{\gamma^2+2\sigma_z^2}\mathbf{I}_{Kd}\right)\left(\frac{1}{2\sigma_z^2}\boldsymbol{\Delta}\right)\\ &= \frac{\gamma^2}{2\sigma_z^2(\gamma^2+2\sigma_z^2)}\|\boldsymbol{\Delta}\|_2^2. \qquad (46)\end{aligned}$$

Therefore

$$-\frac{\|\mathbf{\Delta}\|_2^2}{4\sigma_z^2}+\frac{1}{2}b^\top A^{-1}b=-\frac{\|\mathbf{\Delta}\|_2^2}{4\sigma_z^2}+\frac{\gamma^2}{4\sigma_z^2(\gamma^2+2\sigma_z^2)}\|\mathbf{\Delta}\|_2^2$$
$$=-\frac{\|\mathbf{\Delta}\|_2^2}{2(\gamma^2+2\sigma_z^2)}. \tag{47}$$

Combining the prefactor and exponent yields

$$\mathbb{E}\big[k_\gamma(\mathbf{X},\mathbf{Y})\big]=\left(\frac{\gamma^2}{\gamma^2+2\sigma_z^2}\right)^{\frac{Kd}{2}}\exp\!\left(-\frac{\|\mathbf{\Delta_Z}\|_2^2}{2(\gamma^2+2\sigma_z^2)}\right). \tag{48}$$

Combine the three terms. Substituting (39), (40), and (48) into (35) gives

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q};k_\omega)=2\left(\frac{\gamma^2}{\gamma^2+2\sigma_z^2}\right)^{\frac{Kd}{2}}\left[1-\exp\!\left(-\frac{\|\mathbf{\Delta_Z}\|_2^2}{2(\gamma^2+2\sigma_z^2)}\right)\right],$$

Since $\mathbf{\Delta_Z}=\mathbf{1}_K\mathbf{\Delta}_{\mathrm{PFS}}^\top$, we have $\|\mathbf{\Delta_Z}\|_2^2=K\|\mathbf{\Delta}_{\mathrm{PFS}}\|_2^2$. Hence,

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q};k_\omega)=2\left(\frac{\gamma^2}{\gamma^2+2\sigma_z^2}\right)^{\frac{Kd}{2}}\left[1-\exp\!\left(-\frac{K\|\mathbf{\Delta}_{\mathrm{PFS}}\|_2^2}{2(\gamma^2+2\sigma_z^2)}\right)\right],$$

which proves (11).

Positivity and monotonicity in $\|\mathbf{\Delta}_{\mathrm{PFS}}\|_2$. Let $a:=\left(\frac{\gamma^2}{\gamma^2+2\sigma_z^2}\right)^{\frac{Kd}{2}}>0$ and $t:=\|\mathbf{\Delta}_{\mathrm{PFS}}\|_2\geq 0$. Then

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q};k_\omega)=2a\left(1-e^{-Kt^2/(2(\gamma^2+2\sigma_z^2))}\right).$$

If $t>0$, then $e^{-Kt^2/(2(\gamma^2+2\sigma_z^2))}\in(0,1)$ and hence $\mathrm{MMD}^2>0$. Moreover,

$$\frac{d}{dt}\left(1-e^{-Kt^2/(2(\gamma^2+2\sigma_z^2))}\right)=e^{-Kt^2/(2(\gamma^2+2\sigma_z^2))}\cdot\frac{Kt}{\gamma^2+2\sigma_z^2}\geq 0,$$

with strict inequality for $t>0$. Hence, positivity and monotonicity are both proved. □

B.5 Proof of Theorem 2.7

Lemma B.3 (Transformation of exponential concentration inequality in (Gretton et al., 2012)). *Let $\widehat{\mathrm{MMD}}_u^2$ denote the unbiased U-statistic estimator and MMD^2 be the population quantity. By Theorem 7 of Gretton et al. (2012), there exists an absolute constant $c>0$ such that for any $\varepsilon>0$,*

$$\Pr\!\left(\left|\widehat{\mathrm{MMD}}_u^2-\mathrm{MMD}^2\right|>\varepsilon\right)\leq 2\exp\!\left(-c\,\varepsilon^2\frac{MN}{M+N}\right). \tag{49}$$

Equivalently, for any $\delta\in(0,1)$, with probability at least $1-\delta$,

$$\left|\widehat{\mathrm{MMD}}_u^2-\mathrm{MMD}^2\right|\leq C\sqrt{\left(\frac{1}{M}+\frac{1}{N}\right)\log\frac{2}{\delta}}, \tag{50}$$

where $C=\frac{1}{\sqrt{c}}$ is an absolute constant (depending only on the kernel bound).

Proof. We show the transformation (49) ⇒ (50) step by step.

We choose ε by setting the right-hand side of (49) equal to δ:

$$2\exp\!\left(-c\,\varepsilon^2\frac{MN}{M+N}\right)=\delta.$$

Divide both sides by 2 and take $\log$:

$$-c\,\varepsilon^2 \frac{MN}{M+N} = \log\frac{\delta}{2} = -\log\frac{2}{\delta}.$$

Multiply by $-\frac{M+N}{cMN}$:

$$\varepsilon^2 = \frac{M+N}{cMN}\log\frac{2}{\delta} = \frac{1}{c}\left(\frac{1}{M}+\frac{1}{N}\right)\log\frac{2}{\delta}.$$

Let $C := \frac{1}{\sqrt{c}}$. Then

$$\varepsilon = C\sqrt{\left(\frac{1}{M}+\frac{1}{N}\right)\log\frac{2}{\delta}}.$$

Plugging this choice of ε back into (49) gives

$$\Pr\left(\left|\widehat{\mathrm{MMD}}_u^2 - \mathrm{MMD}^2\right| > C\sqrt{\left(\frac{1}{M}+\frac{1}{N}\right)\log\frac{2}{\delta}}\right) \le \delta,$$

which is exactly (50). □

Theorem 2.7. *Let $S_r = \{x_i\}_{i=1}^M \overset{i.i.d}{\sim} \mathbb{P}$ and $S_t = \{y_j\}_{j=1}^N$ be test-image set. Let $\lambda = \gamma^2 + 2\sigma_z^2$. For any $\delta \in (0,1)$, with probability at least $1-\delta$, the bounds* (12) *and* (13) *hold.*

Proof. By Lemma B.3, for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\left|\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) - \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega)\right| \le C\sqrt{\left(\frac{1}{M}+\frac{1}{N}\right)\log\frac{2}{\delta}}. \tag{51}$$

We will use C_1 and C_2 to allow different absolute constants in the two cases.

Case I: Real test set ($S_t \overset{i.i.d}{\sim} \mathbb{P}$). If $S_r \sim \mathbb{P}$ and $S_t \sim \mathbb{P}$, then the two distributions are identical, hence

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{P}; k_\omega) = 0.$$

$$\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) = \widehat{\mathrm{MMD}}_u^2 - \mathrm{MMD}^2(\mathbb{P}, \mathbb{P}; k_\omega).$$

Therefore, on the event (51),

$$\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) \le \left|\widehat{\mathrm{MMD}}_u^2 - \mathrm{MMD}^2(\mathbb{P}, \mathbb{P}; k_\omega)\right| \le C_1\sqrt{\left(\frac{1}{M}+\frac{1}{N}\right)\log\frac{2}{\delta}},$$

which is exactly (12).

Case II: Generated test set ($S_t \overset{i.i.d}{\sim} \mathbb{Q}$). Define

$$A := \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega), \qquad B := \widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) - \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega).$$

Then by construction,

$$\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) = A + B. \tag{52}$$

Since $B \ge -|B|$ always holds; adding A to both sides gives the elementary inequality

$$A + B \;\ge\; A - |B|.$$

Applying this to (52) yields

$$\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) \ge \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega) - \left|\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) - \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega)\right|. \tag{53}$$

On the event (51), we have

$$\left|\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) - \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega)\right| \le C_2\sqrt{\left(\frac{1}{M} + \frac{1}{N}\right)\log\frac{2}{\delta}}. \tag{54}$$

By Proposition 2.6 (population MMD under PFS mean shift) and letting $\lambda = \gamma^2 + 2\sigma_z^2$, we have

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega) = 2\left(\frac{\gamma^2}{\lambda}\right)^{\frac{Kd}{2}}\left[1 - \exp\left(-\frac{K\|\boldsymbol{\Delta}_{\mathrm{PFS}}\|_2^2}{2\lambda}\right)\right]. \tag{55}$$

Substitute (54) and (55) into (53). With probability at least $1-\delta$,

$$\begin{aligned}\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) \ge\ & 2\left(\frac{\gamma^2}{\lambda}\right)^{\frac{Kd}{2}}\left[1 - \exp\left(-\frac{K\|\boldsymbol{\Delta}_{\mathrm{PFS}}\|_2^2}{2\lambda}\right)\right] \\ & - C_2\sqrt{\left(\frac{1}{M} + \frac{1}{N}\right)\log\frac{2}{\delta}},\end{aligned} \tag{56}$$

which is exactly (13). This completes Case II and the proof. □

Corollary B.4 (Separation of empirical MMD for real vs. generated images)**.** *Let $S_r = \{x_i\}_{i=1}^M \overset{i.i.d.}{\sim} \mathbb{P}$ be a reference set of real images, and let S_t be a test-image set. Consider the empirical statistic $\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega)$ defined with the deep kernel k_ω.*

Fix any $\delta \in (0,1)$ and define

$$\varepsilon_{M,N}(\delta) := C\sqrt{\left(\frac{1}{M} + \frac{1}{N}\right)\log\frac{2}{\delta}},$$

where C is the constant appearing in Lemma B.3. Then, with probability at least $1-2\delta$, the two statements in Theorem 2.7 hold simultaneously. Consequently, whenever

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega) \ >\ 2\,\varepsilon_{M,N}(\delta), \tag{57}$$

the empirical ordering

$$\widehat{\mathrm{MMD}}_u^2(S_r, S_t^{(\mathbb{Q})}; k_\omega) \ >\ \widehat{\mathrm{MMD}}_u^2(S_r, S_t^{(\mathbb{P})}; k_\omega)$$

holds with probability at least $1-2\delta$.

Proof. We combine the high-probability bounds established in Theorem 2.7 for the real and generated cases.

By Theorem 2.7, for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) \le \varepsilon_{M,N}(\delta) \quad \text{if } S_t \sim \mathbb{P},$$

and with probability at least $1-\delta$,

$$\widehat{\mathrm{MMD}}_u^2(S_r, S_t; k_\omega) \ge \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega) - \varepsilon_{M,N}(\delta) \quad \text{if } S_t \sim \mathbb{Q}.$$

The two deviation events above each fail with probability at most δ. By the union bound, both inequalities hold simultaneously with probability at least $1-2\delta$.

On this event, we have

$$\begin{aligned}&\widehat{\mathrm{MMD}}_u^2(S_r, S_t^{(\mathbb{Q})}; k_\omega) - \widehat{\mathrm{MMD}}_u^2(S_r, S_t^{(\mathbb{P})}; k_\omega) \\ &\quad \ge \big(\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega) - \varepsilon_{M,N}(\delta)\big) - \varepsilon_{M,N}(\delta) = \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega) - 2\varepsilon_{M,N}(\delta).\end{aligned}$$

Therefore, if condition (57) holds, the right-hand side is strictly positive, which yields the claimed empirical ordering.

Interpretation. Since $\varepsilon_{M,N}(\delta) = \mathcal{O}\Big(\sqrt{(1/M + 1/N)\log(1/\delta)}\Big)$, for any fixed population gap $\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\omega) > 0$, the separation condition (57) is satisfied once the sample sizes M, N are sufficiently large. □

Proposition B.5 (Conditional ordering for the single-image score)**.** *Fix the learned Gaussian kernel k_ω independently of the reference bank $S_{\mathrm{ref}} = \{x_i\}_{i=1}^R \overset{\text{i.i.d.}}{\sim} \mathbb{P}$. For two fixed test images y_r, y_g independent of this bank, define the population similarity margin*

$$m(y_r, y_g) := \mathbb{E}_{X\sim\mathbb{P}}\big[k_\omega(X, y_r) - k_\omega(X, y_g)\big].$$

If $m(y_r, y_g) > 0$, the biased single-image score in Eq. 6 obeys

$$\Pr_{S_{\mathrm{ref}}}[S_{\mathrm{MDMF}}(y_g) > S_{\mathrm{MDMF}}(y_r)] \geq 1 - \exp\left(-\frac{R\, m(y_r, y_g)^2}{2}\right).$$

If $Y_r \sim \mathbb{P}$ and $Y_g \sim \mathbb{Q}$ are mutually independent and independent of the bank, and $\Pr[m(Y_r, Y_g) \geq m_0] \geq 1 - \eta$ for some $m_0 > 0$, then their score ordering holds with probability at least $1 - \eta - \exp(-Rm_0^2/2)$ over the test pair and reference bank.

Proof. The reference self-term and $k_\omega(y, y) = 1$ in Eq. 6 cancel when comparing two test images. Thus

$$S_{\mathrm{MDMF}}(y_g) - S_{\mathrm{MDMF}}(y_r) = \frac{2}{R}\sum_{i=1}^{R}\big[k_\omega(x_i, y_r) - k_\omega(x_i, y_g)\big].$$

Conditional on y_r, y_g, the summands are independent, lie in $[-1, 1]$, and have mean $m(y_r, y_g)$. Hoeffding's inequality bounds the probability that their average is nonpositive by $\exp(-Rm(y_r, y_g)^2/2)$. For random test images, condition on the event $m(Y_r, Y_g) \geq m_0$ and use a union bound with its failure probability η. □

This result concerns the actual single-image scoring rule. Its population similarity-margin condition is separate from the positive two-population MMD condition in Proposition 2.6; the latter alone does not imply per-image score ordering.

C Detailed Related Work

C.1 Generative Models for Image Generation

Early image generation methods, including GANs (Brock, 2018; Karras et al., 2019), VAEs (Kingma & Welling, 2013; Sohn et al., 2015), have established the foundation of modern generative models but often exhibited visible artifacts. Diffusion models have since become the dominant paradigm, achieving strong fidelity (Ho et al., 2020; Saharia et al., 2022). Representative diffusion families include DDPM (Ho et al., 2020; Nichol & Dhariwal, 2021), ADM (Dhariwal & Nichol, 2021), LDM (Rombach et al., 2022), SDXL (Podell et al., 2023), and DiT (Peebles & Xie, 2023). These advances have also enabled widely deployed text-to-image systems like GLIDE (Nichol et al., 2021), Wukong (Gu et al., 2022), and Midjourney. Recent video-generation systems such as Sora (Brooks et al., 2024) and OpenSora (Zheng et al., 2024) further raise generation quality, which can produce individual frames as challenging synthetic images. As generative models evolve, artifacts become *weak* and *sparse* (Wang et al., 2024a; 2025b), which motivates us to amplify localized distributional deviations.

C.2 AI-Generated Image Detection

The rapid improvement of generative models has created an urgent demand for reliable AI-generated image detection. Early detectors mainly train image-level binary classifiers, as exemplified by CNNSpot (Wang et al., 2020). To better generalize across unseen generators, Ojha (Ojha et al., 2023) trains detectors in CLIP space for transfer, DIRE (Wang et al., 2023) uses diffusion reconstruction error as a detection feature. DRCT (Chen et al., 2024a) learns from diffusion reconstructions and contrastive hard samples to enhance robustness, F-ConV (Zhang et al., 2025b) exploits manifold

geometry with flow-based extrusion. Motivated by the increasing sparsity of generative artifacts, some methods shift to patch-level evidence. PatchCraft (Zhong et al., 2023) enhances texture traces via smash and reconstruction, FatFormer (Liu et al., 2024) adapts CLIP features with a forgery-aware transformer. A complementary line of work trains patch-level classifiers that score individual patches independently, as in the approach of Chai et al. (2020). Independent patch scoring requires a patch threshold as well as an image-level decision threshold, as we examine in Figure 4(d). MDMF instead learns patch forensic signatures and measures their *distributional discrepancy* via MMD, with finite-sample concentration guarantees (Theorem 2.7).

D Additional Experiment Setups

D.1 Details of Datasets

D.1.1 Details of Image Benchmarks

ImageNet (Deng et al., 2009). We use the ImageNet real images and their corresponding synthetic counterparts released in the DGM-Eval repository.[1] All images are provided following Stein et al. (2023), and are stored at a resolution of 256×256. The set of generators includes ADM, ADMG, BigGAN, DiT-XL/2, GigaGAN, LDM, StyleGAN-XL, RQ-Transformer, and Mask-GIT.

LSUN-Bedroom (Yu et al., 2015). Real and generated samples for LSUN-Bedroom are also taken from the same DGM-Eval release.[2] The dataset (Stein et al. (2023)) provides images at 256×256; during preprocessing, we apply random cropping to obtain 224×224 inputs. The generated images are produced by ADM, DDPM, iDDPM, StyleGAN, Diffusion-Projected GAN, Projected GAN, and Unleashing Transformers.

GenImage (Zhu et al., 2023b). We additionally adopt GenImage, which is publicly available at:[3] According to Zhu et al. (2023b), the real images are sourced from ImageNet, while the image resolutions vary across subsets. The generative sources covered by GenImage include Midjourney, SD v1.4, SD v1.5, ADM, GLIDE, Wukong, VQDM, and BigGAN.

D.1.2 Details of Video Case Study

OpenSora (Zheng et al., 2024). Recent progress in video generation has substantially improved the realism of synthetic videos, raising new concerns about the trustworthiness of digital media. Since the proprietary model behind Sora[4] is not publicly accessible, we instead employ Open-Sora, an open-source high-fidelity video generation framework with fully released code and model weights as a practical stress test for evaluating generalization. Concretely, we randomly sample 3,275 videos from the OpenSora subset in the GenVideo (Chen et al., 2024b) benchmark. Each video contains 10 frames, yielding 32,750 frames in total, which we treat as the *OpenSora-generated video dataset*. For preprocessing, we follow the same pipeline as in the image benchmarks and apply random cropping to obtain 224×224 inputs.

MSR-VTT (Xu et al., 2016). As natural video data, we use MSR-VTT, a large-scale web video benchmark with diverse content and comprehensive categories, widely adopted for video understanding and video-to-text tasks. For preprocessing, we randomly sample 3,275 videos from MSR-VTT, and then randomly select 10 frames per video, resulting in 32,750 frames in total as the *real* set. We follow the same pipeline as the image benchmarks and apply random cropping to obtain 224×224 inputs.

D.2 Details on Evaluation Metrics

AI-generated image detection is inherently a binary classification task. Let $\mathrm{TP}(t), \mathrm{TN}(t), \mathrm{FP}(t), \mathrm{FN}(t)$ denote the numbers of true positives, true negatives, false positives, and false negatives when thresholding the detector score at t. Accordingly, the true positive rate

[1] https://github.com/layer6ai-labs/dgm-eval

[2] https://github.com/layer6ai-labs/dgm-eval

[3] https://github.com/GenImage-Dataset/GenImage

[4] https://openai.com/index/sora/

(TPR) and false positive rate (FPR) are

$$\mathrm{TPR}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FN}(t)}, \qquad \mathrm{FPR}(t) = \frac{\mathrm{FP}(t)}{\mathrm{FP}(t) + \mathrm{TN}(t)}.$$

The area under the receiver operating characteristic curve (AUROC). AUROC summarizes the detector's ranking quality by measuring how well positives are separated from negatives across *all* possible thresholds. Formally, it is the area under the ROC curve obtained by plotting $\mathrm{TPR}(t)$ against $\mathrm{FPR}(t)$ as t varies:

$$\mathrm{AUROC} = \int_0^1 \mathrm{TPR}\big(\mathrm{FPR}^{-1}(u)\big)\, du,$$

where larger AUROC indicates better overall discriminability independent of a specific operating point.

The average precision (AP). Average Precision evaluates precision–recall trade-offs by aggregating precision over different recall levels, and is commonly used when the positive class may be relatively rare. Let $\mathrm{Precision}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t)+\mathrm{FP}(t)}$ and $\mathrm{Recall}(t) = \mathrm{TPR}(t)$. AP is defined as the area under the precision–recall (PR) curve:

$$\mathrm{AP} = \int_0^1 \mathrm{Precision}(r)\, dr,$$

where $\mathrm{Precision}(r)$ denotes precision as a function of recall r along the PR curve.

The classification accuracy (ACC). Accuracy reports the fraction of correctly classified samples at a chosen threshold, counting both true positives and true negatives:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}.$$

Unlike AUROC/AP which integrate over thresholds, ACC depends on the selected decision threshold. Following (Ojha et al., 2023), we adopt an automatic thresholding strategy during testing: the decision threshold is chosen to best separate real and AI-generated samples according to the detector scores, i.e., selecting the operating point that yields the strongest class separation on the evaluation split. However, we do not treat ACC as a primary metric for comparison because its value can vary noticeably with the thresholding protocol and the underlying data characteristics (e.g., class prior, domain shift, or how representative a small validation set is if used for calibration). In contrast, AUROC and AP summarize performance across all possible thresholds, making them more robust and better aligned with practical deployment scenarios where the preferred operating point may differ.

D.3 Details of Implementations

Main Experiments. Following prior works (Ojha et al., 2023; Liu et al., 2024), we use random cropping and random horizontal flipping during training, and apply center cropping at test time, without additional augmentations. For the main experiments, we adopt DINOv2 ViT-L/14 (Oquab et al., 2024) as the feature backbone to extract patch embeddings. To balance detection accuracy and efficiency, we pool the patch embeddings with a patch size of $W = 32$ for computing Patch Forensic Signatures (PFS). The PFS projection head ϕ_θ maps each pooled patch embedding (dimension 1024) to a bounded scalar score, using a lightweight feed-forward projection with hidden dimension 256, dropout 0.3, and a final $\tanh$ activation. For training data, we follow the common cross-dataset protocol: ProGAN is used to train models evaluated on ImageNet and LSUN-Bedroom, while SD v1.4 is used as the training set for the GenImage benchmark and our OpenSora case study. For reference data at test time, we use 3k real references that are strictly disjoint from the test split: for ImageNet and GenImage, we sample 3 images per ImageNet training class (3k in total); for LSUN-Bedroom, we randomly sample 3k LSUN real images from a split disjoint from testing; and for the OpenSora case study, we sample 175 MSR-VTT real videos disjoint from testing and extract 10 frames per video (1,750 real frames) as references. During training, we jointly optimize the projection parameters θ and the kernel bandwidth γ. We train for 25 epochs with batch size 256 using AdamW (learning rate 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay 0.01), and initialize the scale parameter with $\sigma = 1.0$. All experiments are conducted on a server with an NVIDIA H200 GPU using Python 3.10.19 and PyTorch 2.9.1.

Threshold selection in practice. The image-level decision threshold τ in Algorithm 2 can be set in deployment in either of two ways. (i) *Validation-based selection*: a small held-out validation set with both real and AI-generated samples is used to choose τ as the operating point that maximizes the desired criterion (e.g., F1, accuracy, or a target FPR). (ii) *Real-only calibration*: choose τ as an upper quantile of scores on held-out real images disjoint from the reference bank. This sets an empirical false-positive rate on the calibration set without requiring target-generator images; transfer to future data depends on how representative those real images are. In all reported AUROC and AP results we sweep τ over the full score range, while ACC is computed under the per-evaluation optimal threshold following (Ojha et al., 2023), so that no single τ choice is implicitly tied to a specific generator.

Figure 1. We compare a *global image-level* baseline and our *PFS-based* pipeline under the label-inversion stress test. ***For data configuration,*** we build the toy benchmark from the ProGAN dataset by selecting the *cat* and *dog* categories from each split. During training, we assign all real samples to cats and all fake samples to dogs, using 18,000 images per class (18,000 real-cat and 18,000 fake-dog). The validation set follows the same configuration with 200 images per class. For testing, we consider two settings. (i) *Matched-label test:* the same label configuration as training, with 200 real-cat and 200 fake-dog images. (ii) *Label-inversion test:* we swap the category composition while keeping the real/fake labels fixed, i.e., 200 real-dog and 200 fake-cat images, to stress-test whether a detector relies on semantic category cues versus generation artifacts. ***For model configuration,*** we use DINOv2 ViT-L/14 as the frozen feature extractor and take the `[CLS]` token as the image representation for global image-level detector. On top of it, we train a lightweight two-layer classification head (hidden dimension 256 with dropout 0.3) to predict a single logit, optimized with binary cross-entropy (BCE) loss. For PFS pipeline, we follow the same patch-wise setup as in the main experiments: the image is partitioned into patches and each patch embedding is mapped into the PFS space via the same projection head as in our main method. To obtain an image-level decision, we additionally learn a lightweight attention (scoring) head with the same hidden dimension and dropout, which outputs a scalar weight/logit for each patch and aggregates patch-level logits into a final image-level logit. The entire model is trained with BCE loss under the same label setting as the Global baseline.

Table 2. For variants *w/o MMD*, we adopt the same model configurations as in the toy experiment (Figure 1): a global baseline that classifies from the DINOv2 `[CLS]` token, and a PFS-based model that aggregates patch-wise PFS scores via a lightweight attention head, both trained with a BCE objective. For *Global + MMD*, we replace the BCE objective with an MMD-based optimization on the *image-level* predictions: we take the global image logit for each sample and compute a one-dimensional, pairwise MMD within each mini-batch between real and AI-generated sets, using it as the training signal. Finally, to further isolate the benefit of *PFS modeling* beyond a particular aggregator, we additionally evaluate alternative patch-level aggregation schemes (e.g., mean/max/top-k pooling) in Appendix E.3, demonstrating that PFS consistently outperforms global pooling under different aggregation choices.

Figure 4. For qualitative localization, we adopt a Grad-CAM-style visualization on the MDMF detector. Given an input image resized to 224×224, we extract DINOv2 ViT-L/14 patch tokens and pool them to a $W{=}32$ patch grid. We then compute patch logits in the learned PFS space, and obtain patch-wise saliency by backpropagating the mean patch logit to the pooled patch embeddings. The final patch importance is computed by combining the patch logit with the gradient magnitude, followed by normalization and resizing to the image resolution for overlay visualization. For the global baseline, we visualize an attention map derived from normalized DINOv2 patch-token magnitudes to provide a comparable heatmap.

D.4 Notation Summary

For quick reference, Table 3 summarizes the symbols used throughout the paper. We group them by where they first appear: patch tokenization and PFS modeling (Section 2.3), the distributional discrepancy and detection rule (Section 2.3 and Algorithm 2), and the theoretical analysis (Section 2.4 and Appendix B).

Table 3: Notation summary. Each symbol is also defined upon first use in the corresponding section.

Symbol	Description
Patch tokenization and PFS modeling.	
$\mathbb{P}, \mathbb{Q}$	Distributions of real and AI-generated images, respectively.
$\mathbf{x}$	A single image (real or AI-generated) sampled from $\mathbb{P}$ or $\mathbb{Q}$.
K	Number of (non-overlapping) patches extracted per image.
W	Spatial side length of each patch (we use W=32 in the main experiments).
$\mathbf{e}_i \in \mathbb{R}^D$	The i-th patch embedding produced by the frozen DINOv2 ViT-L/14 backbone.
ϕ_θ	Learnable Patch Forensic Signature (PFS) projection $\phi_\theta : \mathbb{R}^D \to \mathbb{R}^d$.
$\mathbf{z}_i = \phi_\theta(\mathbf{e}_i)$	The i-th PFS feature derived from $\mathbf{e}_i$.
MMD-based detection.	
k_ω	Characteristic kernel parameterized by ω (Gaussian RBF with bandwidth γ).
$\mathrm{MMD}^2(\cdot, \cdot; k_\omega)$	(Squared) Maximum Mean Discrepancy between two distributions in the PFS space.
$\mathcal{S}^{re}_{\mathbb{P}}$	Reference set of R real images (with R swept from 1k to 10k in App. E.7).
$\mathcal{S}^{te}$	Test set of images on which detection is performed.
τ	Image-level decision threshold applied to the MMD-based score.
Theoretical analysis.	
σ_e	Sub-Gaussian proxy for patch embeddings (Assumption 2.2).
$\boldsymbol{\mu}_{\mathrm{defect}}$	Signed defect direction in the sparse-defect model; the embedding mean shift is zero (Assumption 2.3).
$\mathcal{Q}(\cdot)$	Hessian-induced quadratic form of ϕ_θ (Proposition 2.4).
η	Exponent used when relating mixing coefficients to covariance bounds in Appendix B.1.

E Additional Experimental Results

E.1 Results on Additional Benchmarks

LSUN-Bedroom. As shown in Table 4, our method maintains consistently strong performance across diverse generators, covering both diffusion-based models and GAN variants, indicating good cross-model generalization. In particular, we achieve the best average AUROC on LSUN-Bedroom, while keeping the average AP highly competitive, suggesting that our detection evidence transfers reliably beyond the training distribution.

Table 4: Detection performance (%) on LSUN-Bedroom. Bold numbers are superior results. We mainly compare training-based methods.

Methods	Models														Average	
	ADM		DDPM		iDDPM		Diffusion GAN		Projected GAN		StyleGAN		Unleashing Transformer			
	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
CNNspot	64.83	64.24	79.04	80.58	76.95	76.28	88.45	87.19	90.80	89.94	95.17	94.94	93.42	93.11	84.09	83.75
Ojha	71.26	70.95	79.26	78.27	74.80	73.46	84.56	82.91	82.00	78.42	81.22	78.08	83.58	83.48	79.53	77.94
DIRE	57.19	56.85	61.91	61.35	59.82	58.29	53.18	53.48	55.35	54.93	57.66	56.90	67.92	68.33	59.00	58.59
NPR	75.43	72.60	91.42	90.89	89.49	88.25	76.17	74.19	75.07	74.59	68.82	63.53	84.39	83.67	80.11	78.25
F-ConV	76.59	74.40	93.53	92.16	88.90	86.85	98.10	98.03	97.93	97.81	91.63	90.16	97.31	96.91	92.00	**90.91**
MDMF	74.67	65.21	93.05	90.08	89.10	84.52	99.85	99.74	99.91	99.84	97.96	96.89	99.10	98.59	**93.38**	90.70

GenImage. Table 5 further demonstrates that our method generalizes well to GenImage, which contains heterogeneous sources ranging from proprietary engines (e.g., Midjourney) to various diffusion and GAN models, achieving the best average accuracy among compared methods. We present the results of some baselines reported in (Zhu et al., 2023b), including DeiT-S (Touvron et al., 2021), Swin-T (Liu et al., 2021), Spec (Zhang et al., 2019), F3Net (Qian et al., 2020), GramNet (Liu

et al., 2020b), and GenDet (Zhu et al., 2023a). Overall, the strong average performance across such diverse generative sources highlights the robustness of our approach under real-world distribution shifts.

Table 5: AI-generated image detection performance (ACC, %) on GenImage.

	Models								
Methods	Midjourney	SD V1.4	SD V1.5	ADM	GLIDE	Wukong	VQDM	BigGAN	Average
	Training Methods								
ResNet-50	54.9	99.9	99.7	53.5	61.9	98.2	56.6	52.0	72.1
DeiT-S	55.6	99.9	99.8	49.8	58.1	98.9	56.9	53.5	71.6
Swin-T	62.1	99.9	99.8	49.8	67.6	99.1	62.3	57.6	74.8
CNNspot	52.8	96.3	95.9	50.1	39.8	78.6	53.4	46.8	64.2
Spec	52.0	99.4	99.2	49.7	49.8	94.8	55.6	49.8	68.8
F3Net	50.1	99.9	99.9	49.9	50.00	99.9	49.9	49.9	68.7
GramNet	54.2	99.2	99.1	50.3	54.6	98.9	50.8	51.7	69.9
DIRE	60.2	99.9	99.8	50.9	55.0	99.2	50.1	50.2	70.7
Ojha	73.2	84.2	84.0	55.2	76.9	75.6	56.9	80.3	73.3
NPR	81.0	98.2	97.9	76.9	89.8	96.9	84.1	84.2	88.6
FatFormer	92.7	100.0	99.9	75.9	88.0	99.9	98.8	55.8	88.9
GenDet	89.6	96.1	96.1	58.0	78.4	92.8	66.5	75.0	81.6
DRCT	91.5	95.0	94.4	79.4	89.1	94.6	90.0	81.6	89.4
F-ConV	89.3	98.8	98.5	74.9	89.3	95.6	86.7	87.6	90.1
MDMF	83.5	99.4	99.2	79.4	92.4	97.6	89.7	86.6	**91.0**

E.2 ACC Results on the ImageNet Benchmark

To complement the threshold-free AUROC and AP metrics reported in the main Table 1, we additionally report threshold-optimized accuracy (ACC) on the ImageNet benchmark, following the protocol of Ojha et al. (Ojha et al., 2023), where the decision threshold is independently selected per method to maximize accuracy on each evaluation. We focus on the 5 recent 2025 baselines (LOTA, C2P-CLIP, SAFE, AIDE, Effort) that we reproduced under our unified evaluation protocol, since these are the most directly comparable to MDMF and were the primary baselines highlighted in our 2025-baseline comparison. Under this per-evaluation oracle threshold protocol, MDMF achieves the highest mean ACC (91.07), exceeding AIDE by 3.16 points, and leads on 7 of 9 generators. These ACC values describe the best operating point on each evaluation set; they do not estimate accuracy with a threshold fixed before testing.

Table 6: Detection accuracy (ACC, %) on the ImageNet benchmark. Threshold is selected per method on each evaluation to maximize accuracy, following (Ojha et al., 2023). Bold numbers are superior results.

	Models									
Methods	ADM	ADMG	LDM	DiT	BigGAN	GigaGAN	StyleGAN-XL	RQ-Transformer	Mask-GIT	Average (↑)
LOTA (Wang et al., 2025a)	61.80	61.77	76.87	72.87	68.77	76.32	75.42	72.74	77.12	71.52
C2P-CLIP (Tan et al., 2025)	68.92	66.84	82.87	66.67	99.16	80.24	88.72	73.46	98.02	80.54
SAFE (Li et al., 2025)	63.28	62.67	85.75	80.96	85.96	84.39	84.13	83.71	88.88	79.97
AIDE (Yan et al., 2025a)	82.63	79.33	83.07	74.53	97.38	90.72	92.67	93.78	97.06	87.91
Effort (Yan et al., 2025b)	80.35	75.71	86.48	76.14	**99.25**	87.64	87.39	88.75	**98.82**	86.72
MDMF	**85.81**	**81.53**	**88.16**	**81.81**	99.08	**95.75**	**95.23**	**95.41**	96.82	**91.07**

E.3 Full Results of Ablation Study for Core Components in MDMF

Table 7 reports the complete ablation results for the core components of MDMF on ImageNet, including global baselines trained with BCE or MMD, as well as PFS-based variants. Beyond the default attention head aggregation (i.e., PFS-Attn-BCE), we additionally evaluate several alternative ways of aggregating patch logits in the PFS space (mean, max, top-k) to isolate the effect of PFS modeling and aggregation choice.

Two observations consistently emerge and align with our motivation. First, replacing global image-level pooling with PFS-based patch evidence yields a clear improvement across generators, indicating that the cues for AI-generated images are better captured as localized, artifact-sensitive signals rather than a single semantic-dominant representation. This supports the view that modeling an image

as a collection of patch-wise forensic evidence provides a stronger and more transferable basis for real/fake detection than relying on global features.

Second, while simple PFS-space aggregations (mean/max/top-k) already outperform the global baselines, the best performance is achieved when PFS is further coupled with MMD optimization (i.e., our MDMF). This suggests that the key is not merely the pooling operator, but explicitly learning and comparing the *distributional discrepancy* of patch-level signatures, which amplifies subtle, localized defects into a reliable macro-level detection signal.

Table 7: Detailed detection performance (%) for ablation study. Bold numbers are superior results.

Methods	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average	
	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
Global-BCE	86.89	88.01	82.35	83.34	86.53	92.80	80.15	89.00	98.21	98.34	94.07	97.01	93.70	96.81	94.89	97.48	94.46	97.20	90.14	93.33
Global-MMD	83.08	87.01	75.53	80.51	79.60	90.42	72.22	86.23	98.16	98.59	92.22	96.55	91.96	96.42	93.66	97.26	92.36	96.65	86.53	92.18
PFS-Mean-BCE	86.54	87.27	83.53	84.17	90.98	94.85	84.04	90.61	99.39	99.39	97.99	98.74	97.04	98.32	97.63	98.58	98.63	99.07	92.86	94.56
PFS-Max-BCE	83.26	84.82	80.07	81.35	88.89	94.41	81.26	89.92	99.82	99.84	96.27	98.24	95.06	97.65	96.30	98.27	97.98	99.06	90.99	93.73
PFS-Top-5-BCE	86.08	87.19	82.92	83.84	90.99	95.39	83.49	91.14	99.81	99.83	97.19	98.64	95.99	98.09	96.98	98.55	98.38	99.23	92.42	94.66
PFS-Attn-BCE	87.09	88.73	84.11	85.54	91.47	95.76	85.08	92.18	99.90	99.91	97.81	98.97	97.06	98.61	97.69	98.93	98.72	99.41	93.22	95.34
MDMF	**92.56**	**93.57**	**88.86**	**90.16**	**94.63**	**97.35**	**88.89**	**94.48**	**99.93**	**99.94**	**98.99**	**99.52**	**98.76**	**99.41**	**98.84**	**99.46**	**99.40**	**99.72**	**95.65**	**97.07**

E.4 Additional Diagnostics of the Learned Patch Representation

Linear probes. We examine what information is linearly accessible from the trained MDMF head before its final scalar projection. On 4,000 images sampled from the 20-class ProGAN validation split, we sample eight patches per image and compare normalized DINOv2 patch features, a 128-dimensional PCA projection, and the 128-dimensional penultimate activation of the MDMF head. PCA is fitted on the probe-training portion only. Logistic-regression probes use a fixed 70/30 split by image, so patches from one image cannot appear in both probe training and testing. Object-category and real/fake labels are inherited from each image; the latter are not annotations of individual patch artifacts. The penultimate MDMF representation has lower category-probe accuracy than either DINOv2 control, while its real/fake probe accuracy remains similar. This is evidence of reduced linear category decodability in this representation on the sampled split; it does not establish that semantic information is absent or that individual patches have verified artifacts.

Generic deep-kernel featurizer. We compare MDMF with a patch-level deep-kernel variant that retains the same frozen DINOv2 patch features, four-class ProGAN training split, variance-normalized MMD training objective, and ImageNet evaluation protocol. The variant uses the 128-dimensional output of the same MLP trunk as its learned kernel features, rather than MDMF's scalar PFS output; its kernel bandwidth is initialized by a median heuristic to match the vector feature scale. Table 9 reports average results across the nine ImageNet generators. The global-pooling MMD result is already included in Table 7. MDMF is higher by 3.78 AUROC and 2.30 AP points in this comparison. The two training runs differ in output representation and bandwidth initialization, so this result supports the complete scalar-PFS configuration under the evaluated protocol rather than isolating one design choice.

E.5 Full Results for the Effect of Patch Granularity

Table 10 reports the full results of the patch-granularity study corresponding to Figure 4(a), where we vary the pooled patch size $W \in \{16, 32, 56\}$ and repeat each setting with five random seeds. Consistent with Figure 4(a), the results exhibit a non-monotonic dependence on W, with intermediate granularity (e.g., $W = 32$) offering the best overall trade-off. The multi-seed breakdown further suggests that overly fine partitions can introduce higher variability in the estimated distributional discrepancy, whereas overly coarse partitions may miss sparse localized artifacts, reinforcing that effective PFS modeling requires a balanced spatial granularity.

E.6 Full Results for the Robustness to Encoder Architecture

Table 11 provides the full per-generator results corresponding to Figure 4(b), comparing MDMF against the training-based baseline F-ConV under different DINOv2 encoder variants (ViT-S/14, ViT-B/14, ViT-L/14, and ViT-G/14), reporting AUROC/AP and their averages. Across all backbones, MDMF consistently improves over F-ConV, indicating that our gains are not tied to a specific feature

Table 8: Linear-probe accuracy (%) on sampled ProGAN validation patches. The category task has 20 classes (chance: 5%). Real/fake labels are image-level labels assigned to sampled patches.

Patch representation	Dimension	Category	Real/fake
DINOv2 features	1024	83.52	97.50
PCA of DINOv2 features	128	79.98	96.52
MDMF head, penultimate layer	128	46.65	97.88

Table 9: Average ImageNet detection performance (%) with patch-level deep-kernel features.

Method	Avg. AUROC	Avg. AP
Generic vector featurizer (128D)	91.87	94.77
MDMF, scalar PFS (1D)	95.65	97.07

extractor and transfer reliably across encoder architectures and scales. Notably, MDMF exhibits stable scaling behavior as the backbone grows, suggesting that PFS-based local forensic cues and distributional comparison provide a more backbone-agnostic detection signal than global image-level modeling, which can be more sensitive to semantic representations and thus less stable under architecture changes.

E.7 Impact of Reference Size and Runtime Analysis

Table 12 reports the detailed detection performance and runtime on the ImageNet benchmark under different reference set sizes R used in the *test-time* MMD scoring (Eq. 6). Importantly, R is *independent of training*: it only controls the number of reference images used during inference when computing the MDMF score for each test image. The reference images are sampled from the ImageNet training split, strictly disjoint from the test split, to evaluate how R affects detection stability and deployment cost. Within each reported seed, changing R from 1k to 10k has little effect on AUROC/AP. Across seeds, however, average AUROC at R=3k ranges from 94.55 to 95.65, so reference-set composition has a larger observed effect than size in this sweep. From an efficiency perspective, although each test image must be compared against R references, in practice we precompute and cache the PFS embeddings of the reference set, so the one-off reference encoding cost is amortized and negligible at inference time. The remaining computation reduces to GPU-efficient matrix operations whose arithmetic cost scales linearly with R, but is typically fast relative to feature extraction and is thus weakly reflected in end-to-end inference time, which is dominated by backbone forward passes and instantaneous hardware load. Based on this trade-off, we adopt $R = 3\text{k}$ in our main experiments to balance efficiency with stable performance.

Reference-domain check. To separate reference size from semantic-domain matching, we keep the trained ImageNet evaluation model and test lists fixed and compare two equally sized real-reference banks. The ImageNet-training bank of 1k images gives average AUROC/AP of $95.61/97.05$, while a bank of 1k LSUN-Bedroom real images gives $95.56/97.03$. The 3k ImageNet bank used in the main result gives $95.65/97.07$. This single cross-domain comparison shows little change for the tested bedroom reference bank; it does not establish insensitivity to arbitrary reference domains.

E.8 Full Results for the Robustness to Common Post-Processing Perturbations

Tables 13–15 report the per-generator detection performance corresponding to Figure 4(c), where we evaluate MDMF and F-ConV under three families of post-processing perturbations applied at test time on the ImageNet benchmark: JPEG compression (quality factor $q \in \{100, 90, 80, 70, 60, 50\}$), Gaussian blur (kernel standard deviation $\sigma \in \{0, 1, 2, 3, 4, 5\}$), and additive Gaussian noise (standard deviation $\sigma \in \{0, 0.05, 0.10, 0.15, 0.20, 0.25\}$). Each perturbation is applied to all real and generated test images, while the reference set used in MDMF's MMD scoring is kept clean to mirror a deployment scenario in which the detector is exposed to corrupted test inputs against an in-domain reference of unmodified real images. All other settings (DINOv2 ViT-L/14 backbone, W=32 patch granularity, R=3k reference size) are identical to the main results. Across all three perturbation

Table 10: Detailed detection performance (%) for patch granularity.

Patch Size	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average	
	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
$W = 16$	89.27	91.12	84.45	86.86	92.55	96.46	85.16	92.79	99.93	99.94	98.39	99.27	97.96	99.08	98.27	99.22	98.99	99.56	93.89	96.03
	89.85	91.50	85.41	87.43	92.61	96.47	85.83	93.02	99.92	99.93	98.53	99.33	98.01	99.09	98.35	99.25	99.08	99.59	94.18	96.18
	90.56	92.20	86.07	88.25	93.21	96.78	86.37	93.41	99.92	99.93	98.62	99.37	98.22	99.19	98.42	99.28	99.07	99.58	94.50	96.44
	90.77	92.23	86.26	88.21	93.33	96.81	86.69	93.50	99.93	99.93	98.69	99.39	98.19	99.17	98.45	99.29	99.18	99.63	94.61	96.46
	91.77	92.88	87.67	89.12	94.10	97.11	87.74	93.94	99.94	99.94	98.84	99.45	98.48	99.29	98.69	99.39	99.28	99.67	95.17	96.76
$W = 32$	90.56	92.08	86.14	88.06	93.40	96.80	86.38	93.32	99.93	99.94	98.61	99.36	98.18	99.17	98.58	99.35	99.15	99.62	94.55	96.41
	90.71	92.18	86.39	88.23	93.56	96.86	86.88	93.53	99.94	99.94	98.67	99.38	98.26	99.20	98.53	99.33	99.18	99.62	94.68	96.47
	90.88	92.24	86.73	88.38	93.80	96.97	87.11	93.62	99.94	99.94	98.71	99.40	98.34	99.23	98.59	99.35	99.21	99.64	94.81	96.53
	91.63	92.81	87.75	89.24	93.95	97.06	87.91	94.02	99.93	99.93	98.83	99.44	98.61	99.34	98.75	99.42	99.32	99.68	95.18	96.77
	92.56	93.57	88.86	90.16	94.63	97.35	88.89	94.48	99.93	99.94	98.99	99.52	98.76	99.41	98.84	99.46	99.40	99.72	95.65	97.07
$W = 56$	90.33	91.75	86.11	87.90	93.14	96.25	86.37	92.95	99.93	99.93	98.66	99.37	98.19	99.16	98.52	99.25	99.17	99.61	94.46	96.19
	90.47	92.02	86.25	88.19	93.28	96.60	86.51	93.27	99.93	99.93	98.66	99.30	98.19	99.09	98.52	99.24	99.20	99.64	94.56	96.35
	90.62	92.27	86.40	88.44	93.43	96.85	86.66	93.52	99.93	99.93	98.69	99.40	98.22	99.19	98.55	99.34	99.20	99.64	94.63	96.51
	90.74	91.71	86.55	87.81	93.48	96.34	86.79	92.79	99.93	99.93	98.66	99.37	98.22	99.16	98.55	99.25	99.20	99.64	94.68	96.22
	90.82	91.97	86.60	88.14	93.63	96.55	86.86	93.22	99.93	99.93	98.69	99.37	98.22	99.16	98.55	99.28	99.22	99.66	94.75	96.34

Table 11: Detailed detection performance (%) for different encoder architectures. Bold numbers are superior results.

Backbone	Methods	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average	
		AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
ViT-S/14	F-ConV	76.97	80.06	70.98	71.46	71.73	71.66	67.51	68.48	86.07	87.25	79.30	80.47	78.71	78.76	78.76	79.64	76.54	77.36	76.28	77.24
	MDMF	**78.68**	**79.90**	**73.46**	**73.94**	**79.29**	**88.35**	**71.95**	**83.38**	**94.69**	**95.42**	**86.07**	**92.60**	**81.23**	**89.44**	**84.11**	**91.46**	**86.66**	**93.19**	**81.79**	**87.52**
ViT-B/14	F-ConV	86.66	86.93	81.16	82.44	85.01	85.36	78.55	79.53	96.07	96.26	90.37	90.42	93.87	94.49	92.41	93.32	92.18	92.18	88.47	88.99
	MDMF	**86.96**	**88.15**	**82.68**	**83.83**	**88.65**	**94.13**	**82.10**	**90.46**	**99.57**	**99.61**	**95.65**	**97.89**	**94.49**	**97.26**	**94.71**	**97.44**	**96.64**	**98.43**	**91.27**	**94.13**
ViT-L/14	F-ConV	**92.74**	91.65	88.51	87.67	88.87	88.47	85.94	84.88	98.94	98.98	98.14	98.72	98.52	98.38	96.79	96.33	95.52	95.38	93.77	93.38
	MDMF	92.56	**93.57**	**88.86**	**90.16**	**94.63**	**97.35**	**88.89**	**94.48**	**99.93**	**99.94**	**98.99**	**99.52**	**98.76**	**99.41**	**98.84**	**99.46**	**99.40**	**99.72**	**95.65**	**97.07**
ViT-G/14	F-ConV	90.90	92.51	85.75	87.70	87.49	89.17	82.49	84.59	96.59	97.08	95.49	96.04	96.38	96.70	93.96	94.97	94.49	95.34	91.50	92.68
	MDMF	**95.64**	**96.26**	**93.20**	**94.10**	**96.65**	**98.39**	**92.38**	**96.30**	**99.99**	**99.99**	**99.55**	**99.79**	**99.39**	**99.72**	**99.43**	**99.74**	**99.73**	**99.88**	**97.33**	**98.24**

families, MDMF consistently retains higher AUROC and AP than F-ConV at every severity level and on every individual generator, and the gap typically widens as severity grows (e.g., on Mask-GIT, MDMF/F-ConV move from $99.40/93.42$ AUROC at $\sigma_{\text{blur}}{=}0$ to $80.74/69.87$ at $\sigma_{\text{blur}}{=}5$, and from $99.40/93.97$ at $\sigma_{\text{noise}}{=}0$ to $78.97/66.94$ at $\sigma_{\text{noise}}{=}0.25$). Notably, the GAN-style generators (BigGAN, GigaGAN, StyleGAN-XL) and the AR generators (RQ-Transformer, Mask-GIT) are remarkably stable under JPEG, with MDMF's AUROC remaining above 92% even at $q{=}50$, whereas the diffusion generators (ADM, ADMG, LDM, DiT-XL/2) show larger drops under blur and noise. The measurements establish relative robustness under these test perturbations; they do not identify which image frequencies or model components cause the difference.

E.9 Full Results for the Comparison with Patch-Level Hard Voting

Setting and motivation. Hard voting is the most natural alternative to MDMF's distributional aggregation: instead of estimating a population-level discrepancy, classify each patch independently and combine the resulting per-patch decisions into an image-level prediction. Both methods use the same frozen DINOv2 ViT-L/14 backbone, the same patch tokenization ($K{=}49$ at $W{=}32$), and the same 4-class ProGAN training images. The voting variant trains a single linear per-patch classifier with BCE; MDMF instead uses a learned PFS mapping, an MMD-based training objective, and distributional scoring. Thus, this comparison evaluates the two complete patch-based pipelines under matched data and features, rather than changing only the aggregation rule.

A central feature of hard voting that distinguishes it from MDMF is its dependence on *two* coupled thresholds at test time. (i) A per-patch sigmoid cutoff $\theta_{\text{patch}} \in [0, 1]$, which converts each per-patch fake probability into a hard 0/1 decision. (ii) An image-level decision threshold τ applied to the resulting fake-patch ratio $r(I) = \frac{1}{K}\sum_k \mathbb{1}[\sigma(z_k) > \theta_{\text{patch}}]$. The detector flags the image as fake when $r(I) > \tau$. In contrast, MDMF requires only the single image-level threshold τ: per-patch decisions are never materialized, and forensic evidence is aggregated continuously through the MMD score over PFS embeddings. The presence of the additional θ_{patch} in the voting pipeline introduces a non-trivial deployment burden, because a cutoff must be selected using validation data and the observed best cutoff varies across generators; extreme choices can suppress useful variation in the fake-patch ratio.

To isolate the AUROC induced purely by the choice of θ_{patch} (and not by a particular τ), we follow the threshold-free protocol used throughout this paper: for each θ_{patch}, we treat the fake-patch ratio

Table 12: Detailed detection performance (%) and runtime for different reference sizes.

Seed	Ref Size	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average		Runtime
		AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)	
Seed = 0	$R = 1$k	92.47	93.54	88.74	90.11	94.60	97.34	88.81	94.46	99.93	99.94	98.99	99.52	98.75	99.41	98.83	99.46	99.40	99.72	95.61	97.05	718s
	$R = 3$k	92.56	93.57	88.86	90.16	94.63	97.35	88.89	94.48	99.93	99.94	98.99	99.52	98.76	99.41	98.84	99.46	99.40	99.72	95.65	97.07	701s
	$R = 5$k	92.53	93.56	88.82	90.14	94.62	97.35	88.86	94.47	99.93	99.94	98.99	99.52	98.76	99.41	98.84	99.46	99.40	99.72	95.64	97.06	776s
	$R = 7$k	92.56	93.57	88.85	90.16	94.63	97.35	88.89	94.48	99.93	99.94	98.99	99.52	98.76	99.41	98.84	99.46	99.40	99.72	95.65	97.07	616s
	$R = 10$k	92.56	93.57	88.85	90.15	94.63	97.35	88.90	94.48	99.93	99.94	98.99	99.52	98.76	99.41	98.84	99.46	99.40	99.72	95.65	97.07	766s
Seed = 42	$R = 1$k	90.51	92.06	86.09	88.04	93.39	96.80	86.36	93.32	99.93	99.94	98.61	99.36	98.18	99.17	98.58	99.35	99.15	99.62	94.53	96.40	705s
	$R = 3$k	90.56	92.08	86.14	88.06	93.40	96.80	86.38	93.32	99.93	99.94	98.61	99.36	98.18	99.17	98.58	99.35	99.15	99.62	94.55	96.41	734s
	$R = 5$k	90.55	92.07	86.13	88.05	93.39	96.80	86.38	93.32	99.93	99.94	98.61	99.36	98.18	99.17	98.58	99.35	99.15	99.62	94.55	96.41	744s
	$R = 7$k	90.58	92.09	86.17	88.07	93.41	96.81	86.39	93.33	99.93	99.94	98.61	99.36	98.18	99.17	98.59	99.35	99.16	99.62	94.56	96.41	775s
	$R = 10$k	90.55	92.08	86.14	88.05	93.40	96.80	86.37	93.32	99.93	99.94	98.61	99.36	98.18	99.17	98.58	99.35	99.15	99.62	94.55	96.41	784s
Seed = 123	$R = 1$k	90.64	92.15	86.31	88.19	93.53	96.86	86.83	93.51	99.94	99.94	98.66	99.38	98.26	99.20	98.53	99.32	99.17	99.62	94.65	96.46	804s
	$R = 3$k	90.71	92.18	86.39	88.23	93.56	96.86	86.88	93.53	99.94	99.94	98.67	99.38	98.26	99.20	98.53	99.33	99.18	99.62	94.68	96.47	765s
	$R = 5$k	90.70	92.18	86.38	88.22	93.54	96.86	86.86	93.53	99.94	99.94	98.67	99.38	98.26	99.20	98.53	99.33	99.18	99.62	94.67	96.47	820s
	$R = 7$k	90.72	92.18	86.41	88.23	93.56	96.86	86.88	93.53	99.94	99.94	98.67	99.38	98.26	99.20	98.53	99.33	99.18	99.62	94.68	96.48	825s
	$R = 10$k	90.70	92.18	86.38	88.21	93.55	96.86	86.86	93.53	99.94	99.94	98.67	99.38	98.26	99.20	98.53	99.33	99.18	99.62	94.67	96.47	828s
Seed = 456	$R = 1$k	90.86	92.23	86.72	88.37	93.79	96.97	87.11	93.62	99.94	99.94	98.71	99.40	98.34	99.23	98.59	99.35	99.21	99.63	94.81	96.53	809s
	$R = 3$k	90.88	92.24	86.73	88.38	93.80	96.97	87.11	93.62	99.94	99.94	98.71	99.40	98.34	99.23	98.59	99.35	99.21	99.64	94.81	96.53	845s
	$R = 5$k	90.89	92.25	86.74	88.38	93.80	96.97	87.11	93.63	99.94	99.94	98.71	99.40	98.34	99.23	98.59	99.35	99.21	99.64	94.82	96.53	846s
	$R = 7$k	90.88	92.24	86.73	88.37	93.80	96.97	87.11	93.62	99.94	99.94	98.71	99.40	98.34	99.23	98.59	99.35	99.21	99.64	94.81	96.53	824s
	$R = 10$k	90.88	92.24	86.73	88.37	93.80	96.97	87.11	93.62	99.94	99.94	98.71	99.40	98.34	99.23	98.59	99.35	99.21	99.64	94.81	96.53	995s
Seed = 789	$R = 1$k	91.80	92.92	87.99	89.40	94.08	97.11	88.14	94.11	99.93	99.93	98.86	99.46	98.64	99.36	98.77	99.43	99.33	99.68	95.28	96.82	827s
	$R = 3$k	91.63	92.81	87.75	89.24	93.95	97.06	87.91	94.02	99.93	99.93	98.83	99.44	98.61	99.34	98.75	99.42	99.32	99.68	95.18	96.77	878s
	$R = 5$k	91.63	92.82	87.76	89.25	93.96	97.07	87.91	94.02	99.93	99.93	98.83	99.44	98.61	99.34	98.75	99.42	99.32	99.68	95.18	96.77	775s
	$R = 7$k	91.60	92.79	87.71	89.21	93.93	97.05	87.87	94.00	99.93	99.93	98.82	99.44	98.60	99.34	98.75	99.42	99.31	99.68	95.17	96.76	833s
	$R = 10$k	91.60	92.79	87.71	89.21	93.93	97.05	87.86	94.00	99.93	99.93	98.82	99.44	98.60	99.34	98.75	99.42	99.31	99.68	95.17	96.76	925s

Table 13: Detailed detection performance (%) under JPEG compression on ImageNet. Bold numbers are superior results between MDMF and F-ConV at each severity level.

Quality q	Methods	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average	
		AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
$q = 100$	F-ConV	**92.74**	91.65	88.51	87.67	88.87	88.47	85.94	84.88	98.94	98.98	98.14	98.72	98.52	98.38	96.79	96.33	95.52	95.38	93.77	93.38
	MDMF	92.56	**93.57**	**88.86**	**90.16**	**94.63**	**97.35**	**88.89**	**94.48**	**99.93**	**99.94**	**98.99**	**99.52**	**98.76**	**99.41**	**98.84**	**99.46**	**99.40**	**99.72**	**95.65**	**97.07**
$q = 90$	F-ConV	91.59	93.23	86.36	87.95	86.29	87.91	81.49	82.91	96.05	96.58	93.45	94.22	94.90	95.45	95.27	96.04	91.34	92.73	90.75	91.89
	MDMF	**91.76**	92.87	**87.63**	**89.02**	**91.30**	**95.62**	**86.64**	**93.26**	**99.75**	**99.78**	**98.12**	**99.09**	**98.36**	**99.22**	**97.27**	**98.71**	**98.00**	**99.05**	**94.32**	**96.29**
$q = 80$	F-ConV	90.54	91.91	86.04	87.66	83.24	84.32	82.30	83.60	95.66	96.28	93.73	94.68	95.32	95.69	94.68	95.36	90.92	91.73	90.27	91.25
	MDMF	**91.20**	**92.32**	**86.90**	**88.24**	**89.34**	**94.52**	**85.24**	**92.45**	**99.43**	**99.52**	**97.33**	**98.70**	**98.11**	**99.09**	**96.15**	**98.16**	**96.25**	**98.20**	**93.33**	**95.69**
$q = 70$	F-ConV	90.82	92.17	85.95	87.25	83.22	84.50	79.97	81.87	95.67	96.03	93.30	94.26	94.75	95.56	93.36	94.28	89.24	90.65	89.59	90.73
	MDMF	**90.94**	92.05	**86.46**	**87.80**	**88.09**	**93.82**	**84.13**	**91.81**	**98.98**	**99.14**	**96.60**	**98.34**	**97.88**	**98.98**	**95.41**	**97.80**	**94.69**	**97.43**	**92.58**	**95.24**
$q = 60$	F-ConV	89.93	91.55	83.89	84.35	83.66	85.39	80.06	81.89	94.71	95.40	92.50	93.48	95.02	95.66	93.22	94.19	88.81	90.07	89.09	90.22
	MDMF	**90.74**	**91.82**	**86.06**	**87.36**	**87.19**	**93.28**	**83.23**	**91.27**	**98.47**	**98.72**	**95.97**	**98.02**	**97.63**	**98.86**	**94.90**	**97.55**	**93.33**	**96.75**	**91.95**	**94.85**
$q = 50$	F-ConV	89.94	91.10	85.05	87.05	82.02	83.12	77.22	79.41	94.65	95.34	91.65	92.79	94.16	94.74	93.02	94.36	88.20	89.19	88.43	89.68
	MDMF	**90.52**	**91.58**	**85.81**	87.03	**86.35**	**92.77**	**82.36**	**90.73**	**97.94**	**98.28**	**95.46**	**97.75**	**97.40**	**98.74**	**94.45**	**97.31**	**92.20**	**96.15**	**91.39**	**94.48**

$r(I)$ as a continuous score and sweep τ to obtain a ROC curve, then report AUROC and AP. This is the same metric used for MDMF (with the MMD score replacing $r(I)$), and it gives every voting configuration its best image-level operating point. Figure 4(d) shows the resulting voting AUROC as θ_{patch} is swept; here we provide the per-generator full results.

Table 16 reports the per-generator results across the eight patch thresholds spanning the regime explored in Figure 4(d), from an aggressive low cutoff (θ_{patch}=0.03) at which nearly every patch is flagged fake to a stringent high cutoff (θ_{patch}=0.30) past the AUROC peak. Four observations emerge. First, MDMF outperforms hard voting on *every* generator under *every* θ_{patch} choice in both AUROC and AP. Even at the voting peak (θ_{patch}=0.20), MDMF still leads on each individual generator (e.g., +1.42 on ADM, +2.36 on ADMG, +2.12 on LDM, +3.02 on DiT-XL/2), although the best voting cutoff differs by generator. Second, the voting AUROC traces a clear single-peaked profile in θ_{patch}: rising from 86.70 at θ_{patch}=0.03 through 91.37 (0.05), 93.52 (0.08), 93.96 (0.10), 94.14 (0.15), peaking at 94.43 (0.20), and then decaying to 92.66 (0.25) and 89.99 (0.30). The full sweep shown in Figure 4(d) confirms this single-peak shape and the absence of a flat plateau, indicating that voting performance is genuinely sensitive to the choice of θ_{patch} rather than robust within a wide tolerance band. Third, the diffusion generators (ADM, ADMG, LDM, DiT-XL/2) drive most of this volatility. Across the eight thresholds, their per-generator AUROC swings by 7–11 points (e.g., DiT-XL/2 moves from 82.22 at θ_{patch}=0.03 to 86.68 at 0.15 and back down to 77.24 at 0.30, while ADMG ranges from 76.86 at 0.30 to 87.43 at 0.08), whereas the GAN/AR generators (BigGAN, GigaGAN, StyleGAN-XL, RQ-Transformer, Mask-GIT) stay above 94% AUROC across all but the most extreme low thresholds. These results show that thresholding patch scores changes the ranking of diffusion images substantially in this experiment. MDMF's MMD score uses the continuous patch signatures without a per-patch decision boundary. Fourth, even the best voting configuration (θ_{patch}=0.20, AUROC 94.43) trails MDMF by +1.22 AUROC on average, and the gap is most pronounced precisely on the four diffusion generators (collectively a +2.23 AUROC advantage at the voting peak, growing to +10.48 at θ_{patch}=0.30). Together, the sweep shows an

Table 14: Detailed detection performance (%) under Gaussian blur on ImageNet. Bold numbers are superior results between MDMF and F-ConV at each severity level.

Blur σ	Methods	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average	
		AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
$\sigma = 0$	F-ConV	91.07	92.32	85.23	86.23	86.17	87.42	80.90	82.89	96.94	97.14	94.89	95.23	97.05	97.08	95.59	95.51	93.42	94.07	91.25	91.99
	MDMF	**92.56**	**93.57**	**88.86**	**90.16**	**94.63**	**97.35**	**88.89**	**94.48**	**99.93**	**99.94**	**98.99**	**99.52**	**98.76**	**99.41**	**98.84**	**99.46**	**99.40**	**99.72**	**95.65**	**97.07**
$\sigma = 1$	F-ConV	90.02	91.04	86.12	87.68	85.61	86.77	81.86	83.19	95.51	96.05	93.94	94.73	96.05	96.23	94.53	95.13	92.81	93.77	90.72	91.62
	MDMF	**92.64**	**93.54**	**88.93**	**90.05**	**93.66**	**96.83**	**87.94**	**93.89**	**99.87**	**99.89**	**98.84**	**99.44**	**98.69**	**99.37**	**98.48**	**99.28**	**98.68**	**99.37**	**95.30**	**96.85**
$\sigma = 2$	F-ConV	86.44	87.09	81.20	81.82	78.52	79.15	77.44	76.69	90.06	90.68	89.70	89.83	91.41	91.75	90.93	91.29	84.54	84.97	85.58	85.92
	MDMF	**92.68**	**93.26**	**88.69**	**89.36**	**91.50**	**95.53**	**85.77**	**92.45**	**99.50**	**99.55**	**98.03**	**99.00**	**98.09**	**99.05**	**97.44**	**98.73**	**96.15**	**98.07**	**94.21**	**96.11**
$\sigma = 3$	F-ConV	83.06	83.24	75.60	75.46	74.35	74.47	67.65	67.69	83.97	85.43	82.80	83.64	86.21	85.72	86.45	87.02	78.00	78.02	79.79	80.08
	MDMF	**90.52**	**90.84**	**85.64**	**85.72**	**87.22**	**92.73**	**80.55**	**88.87**	**97.40**	**97.50**	**95.66**	**97.64**	**96.20**	**97.96**	**95.18**	**97.44**	**91.52**	**95.37**	**91.10**	**93.79**
$\sigma = 4$	F-ConV	80.93	80.71	71.61	70.83	69.31	68.70	64.71	64.47	81.52	81.33	78.99	78.71	83.70	81.84	82.18	82.01	75.10	75.63	76.45	76.03
	MDMF	**87.49**	**87.34**	**81.44**	**80.74**	**82.02**	**89.00**	**75.11**	**84.90**	**93.76**	**93.64**	**92.06**	**95.38**	**92.98**	**95.94**	**91.00**	**94.80**	**86.88**	**92.22**	**86.97**	**90.44**
$\sigma = 5$	F-ConV	77.20	76.48	70.00	67.85	67.24	65.14	62.93	61.22	76.86	75.80	74.56	74.12	79.25	76.64	76.36	74.26	69.87	67.56	72.70	71.01
	MDMF	**83.08**	**82.17**	**75.84**	**74.15**	**75.57**	**84.15**	**68.73**	**80.17**	**87.81**	**86.87**	**86.42**	**91.58**	**88.14**	**92.66**	**84.58**	**90.36**	**80.74**	**87.79**	**81.21**	**85.55**

Table 15: Detailed detection performance (%) under additive Gaussian noise on ImageNet. Bold numbers are superior results between MDMF and F-ConV at each severity level.

Noise σ	Methods	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average	
		AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
$\sigma = 0$	F-ConV	91.76	93.00	84.37	85.94	85.67	86.82	81.43	83.13	96.74	97.01	94.10	94.92	96.08	96.53	94.88	95.47	93.97	94.55	91.00	91.93
	MDMF	**92.56**	**93.57**	**88.86**	**90.16**	**94.63**	**97.35**	**88.89**	**94.48**	**99.93**	**99.94**	**98.99**	**99.52**	**98.76**	**99.41**	**98.84**	**99.46**	**99.40**	**99.72**	**95.65**	**97.07**
$\sigma = 0.05$	F-ConV	90.55	91.89	84.97	86.72	84.19	85.60	78.30	79.90	94.12	94.40	91.99	93.00	95.04	95.38	94.53	95.05	88.43	89.47	89.12	90.16
	MDMF	**90.79**	**91.94**	**86.12**	**87.49**	**87.42**	**93.38**	**82.97**	**91.09**	**98.10**	**98.38**	**95.29**	**97.65**	**97.22**	**98.65**	**95.10**	**97.60**	**92.71**	**96.35**	**91.75**	**94.73**
$\sigma = 0.10$	F-ConV	89.85	91.20	83.72	84.05	79.79	81.11	76.87	77.49	92.57	93.39	90.76	91.39	93.85	94.01	92.10	92.48	88.08	89.23	87.51	88.26
	MDMF	**90.20**	91.20	**85.02**	**86.16**	**84.80**	**91.69**	**79.48**	**88.82**	**95.53**	**96.12**	**92.85**	**96.31**	**95.77**	**97.85**	**93.34**	**96.64**	**89.20**	**94.36**	**89.58**	**93.24**
$\sigma = 0.15$	F-ConV	88.30	88.56	80.01	79.89	77.07	77.95	68.20	69.34	89.03	88.40	85.67	85.62	90.44	89.85	89.52	88.97	81.68	81.65	83.32	83.36
	MDMF	**89.29**	**90.08**	**83.57**	**84.29**	**82.40**	**90.01**	**76.35**	**86.62**	**93.06**	**93.75**	**90.30**	**94.77**	**93.97**	**96.79**	**91.42**	**95.48**	**86.15**	**92.44**	**87.39**	**91.58**
$\sigma = 0.20$	F-ConV	86.16	85.11	77.80	77.34	71.74	71.14	63.43	62.87	84.32	83.99	78.39	77.08	85.05	83.32	85.35	85.05	74.77	74.21	78.56	77.79
	MDMF	**88.06**	**88.64**	**81.63**	**81.98**	**79.71**	**88.08**	**73.33**	**84.39**	**90.25**	**90.83**	**87.23**	**92.83**	**91.66**	**95.37**	**88.91**	**93.89**	**82.77**	**90.22**	**84.84**	**89.58**
$\sigma = 0.25$	F-ConV	80.00	78.91	71.90	69.59	65.82	65.02	60.61	60.19	75.65	74.29	71.32	69.67	77.21	75.38	76.94	77.04	66.94	65.49	71.82	70.62
	MDMF	**86.37**	**86.48**	**79.12**	**78.72**	**76.52**	**85.61**	**70.09**	**81.90**	**86.76**	**86.84**	**83.61**	**90.32**	**88.75**	**93.38**	**85.65**	**91.61**	**78.97**	**87.47**	**81.76**	**86.93**

advantage for the complete MDMF pipeline over this linear-head voting baseline and avoids selecting an additional patch cutoff. Because the heads and training objectives also differ, the gap cannot be attributed to MMD aggregation alone.

E.10 Complementarity with B-Free and DDA

We compare B-Free (Guillaro et al., 2025) and DDA (Chen et al., 2025) with variants that use their released frozen backbones for patch features and train MDMF's PFS/MMD head on the four-class ProGAN split. All variants are evaluated on the same ImageNet image lists. Table 17 reports average AUROC and AP.

Adding MDMF increases average AUROC/AP for both methods. The original checkpoints use their respective aligned-data training recipes, so these results illustrate complementarity rather than a matched-training-data head ablation.

E.11 Failure Case Analysis: Borderline Real Images

Setting. Although MDMF reaches strong detection performance across the evaluated benchmarks, no detector is perfect. To better understand where its residual error budget actually goes, we sort all 50,000 ImageNet validation real images by the MDMF score and inspect the highest-scoring (most "fake-looking") tail. Figure 6 reports four representative cases together with their MDMF heatmaps and per-patch grids. In each case the image is genuinely real (i.e., not produced by any generative model), yet MDMF nonetheless assigns it a high fake-side score.

Visual observations. These high-scoring real images include a macaw photographed against a bright, noisy background; a painted cat; a soft-focus bear photograph; and a monochrome bus photograph. The displayed patch responses are concentrated in different regions of these images. These examples illustrate that visually unusual real content can receive high fake-side scores. The heatmaps are qualitative and do not establish which image property caused each score or whether the highlighted region is an artifact in the statistical sense.

Table 16: Detailed detection performance (%) of patch-level hard voting at the eight patch thresholds $\theta_{\text{patch}} \in \{0.03, 0.05, 0.08, 0.10, 0.15, 0.20, 0.25, 0.30\}$ swept in Figure 4(d), compared with MDMF on the ImageNet benchmark. Bold numbers indicate the best result on each generator across all configurations.

Method	ADM		ADMG		LDM		DiT		BigGAN		GigaGAN		StyleGAN XL		RQ-Transformer		Mask GIT		Average	
	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC (↑)	AP (↑)
Voting ($\theta_{\text{patch}} = 0.03$)	85.35	78.69	83.01	76.67	86.24	88.49	82.22	86.32	88.99	81.95	88.63	89.86	88.43	89.84	88.61	89.86	88.80	89.96	86.70	85.74
Voting ($\theta_{\text{patch}} = 0.05$)	88.22	86.82	84.06	83.82	90.42	93.46	83.30	90.67	95.85	92.34	95.10	95.56	94.79	95.43	95.12	95.59	95.43	95.76	91.37	92.16
Voting ($\theta_{\text{patch}} = 0.08$)	90.94	89.71	87.43	86.21	92.24	94.64	86.69	91.62	97.71	96.50	96.64	97.11	96.25	96.93	96.68	97.16	97.08	97.36	93.52	94.14
Voting ($\theta_{\text{patch}} = 0.10$)	90.96	90.70	86.36	86.04	92.27	95.15	85.61	91.07	98.99	98.71	97.83	98.70	97.43	98.50	97.88	98.75	98.30	98.95	93.96	95.17
Voting ($\theta_{\text{patch}} = 0.15$)	91.35	91.34	87.41	87.32	92.65	95.41	86.68	92.04	98.81	98.67	97.56	98.16	97.13	97.95	97.61	98.20	98.06	98.42	94.14	95.28
Voting ($\theta_{\text{patch}} = 0.20$)	91.14	91.21	86.50	86.45	92.51	95.62	85.87	91.58	99.89	99.84	98.45	99.11	97.97	98.85	98.51	99.16	99.04	99.39	94.43	95.69
Voting ($\theta_{\text{patch}} = 0.25$)	88.17	88.19	82.33	82.23	89.66	93.56	82.12	88.70	99.89	99.86	97.87	98.71	97.23	98.34	97.98	98.78	98.69	99.21	92.66	94.18
Voting ($\theta_{\text{patch}} = 0.30$)	83.35	83.36	76.86	76.74	85.59	90.67	77.24	85.15	99.83	99.81	96.61	97.84	95.70	97.28	96.81	97.98	97.92	98.63	89.99	91.94
MDMF (Ours)	**92.56**	**93.57**	**88.86**	**90.16**	**94.63**	**97.35**	**88.89**	**94.48**	**99.93**	**99.94**	**98.99**	**99.52**	**98.76**	**99.41**	**98.84**	**99.46**	**99.40**	**99.72**	**95.65**	**97.07**

Table 17: Average ImageNet detection performance (%). The combined variants use each method's frozen backbone with a newly trained MDMF head.

Method	Avg. AUROC	Avg. AP
B-Free (original head)	96.28	97.13
B-Free + MDMF	**98.25**	**98.45**
DDA (original head)	96.36	97.77
DDA + MDMF	**99.18**	**99.34**

Interpretation. A real image with a high fake-side score is a false positive at operating thresholds below that score. This limitation matters when real test images differ in style or acquisition conditions from the reference bank. We therefore treat the examples as diagnostic cases, rather than evidence that MDMF has identified a specific cause of each error.

F Detailed Visualizations

In the main paper, we present qualitative visualizations on ADM (Figure 5/ 7). Here we further report results on images generated by ADMG, LDM, and DiT (Figure 8–10). Across these generators, we observe a consistent trend: the global pooling baseline mainly attends to semantically salient regions (e.g., object contours and high-contrast textures) with similar patterns on real and generated samples, suggesting limited sensitivity to sparse, localized artifacts. In contrast, MDMF produces more localized activations on generated images and comparatively diffuse responses on real images, indicating that patch-wise PFS evidence induces a stronger distributional discrepancy that can be leveraged for robust detection. Overall, these cross-model visualizations support the generalization of MDMF and corroborate our claim that localized forensic cues are suppressed by semantic-dominant global features but become salient under PFS-based distributional modeling.

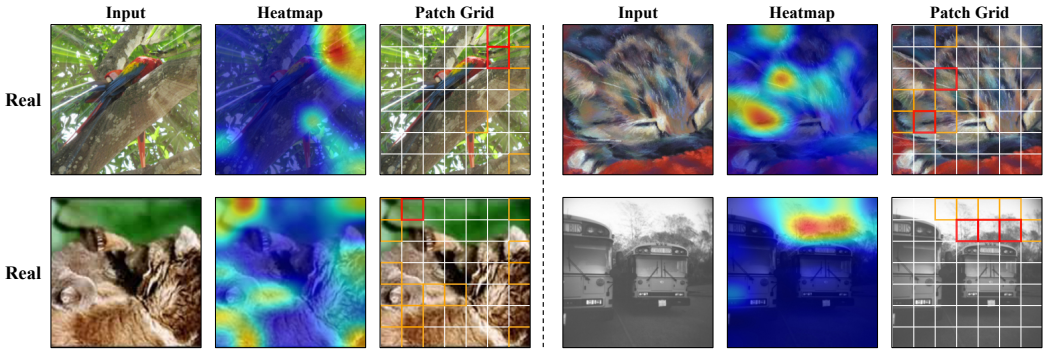


Figure 6: Failure cases on borderline real images from the ImageNet validation set. For each example we show the input, the MDMF heatmap, and the per-patch grid (red/orange = patches with the highest MDMF score). Warmer colors indicate higher predicted likelihood of being fake.

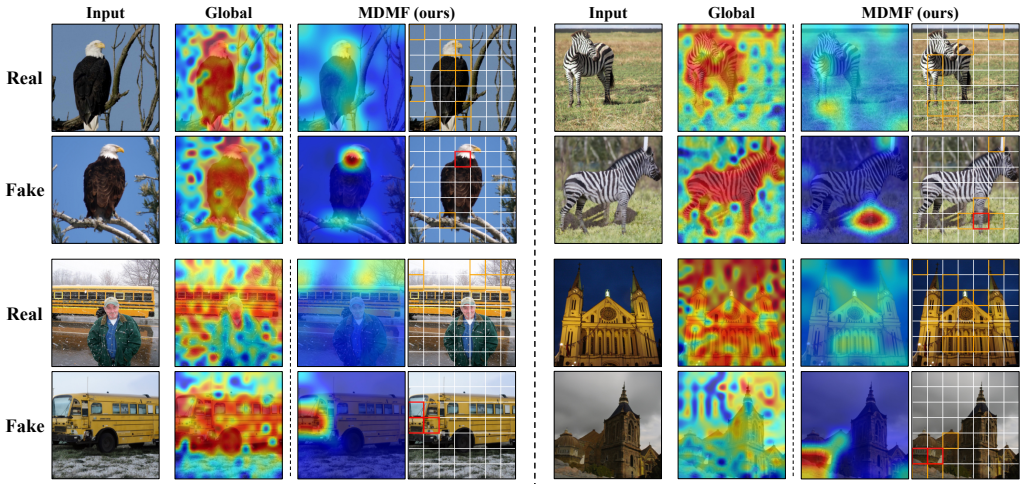


Figure 7: **Qualitative visualization on ADM.** We compare real images and category-matched generated images, visualizing the responses of a global pooling baseline versus MDMF. Warmer colors indicate higher predicted likelihood of being fake.

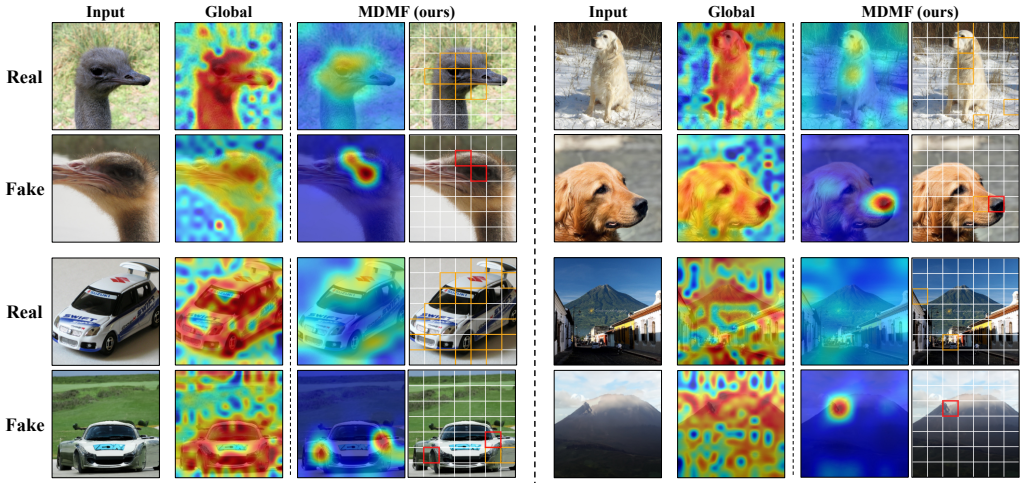


Figure 8: **Additional qualitative visualization on ADMG.** We compare real images and category-matched generated images, visualizing the responses of a global pooling baseline versus MDMF. Warmer colors indicate higher predicted likelihood of being fake.

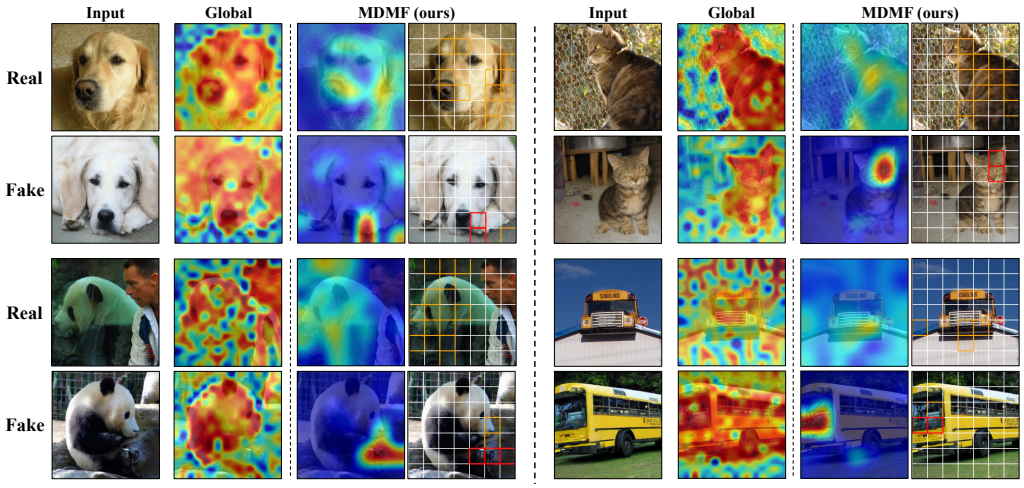


Figure 9: **Additional qualitative visualization on LDM.** We compare real images and category-matched generated images, visualizing the responses of a global pooling baseline versus MDMF. Warmer colors indicate higher predicted likelihood of being fake.

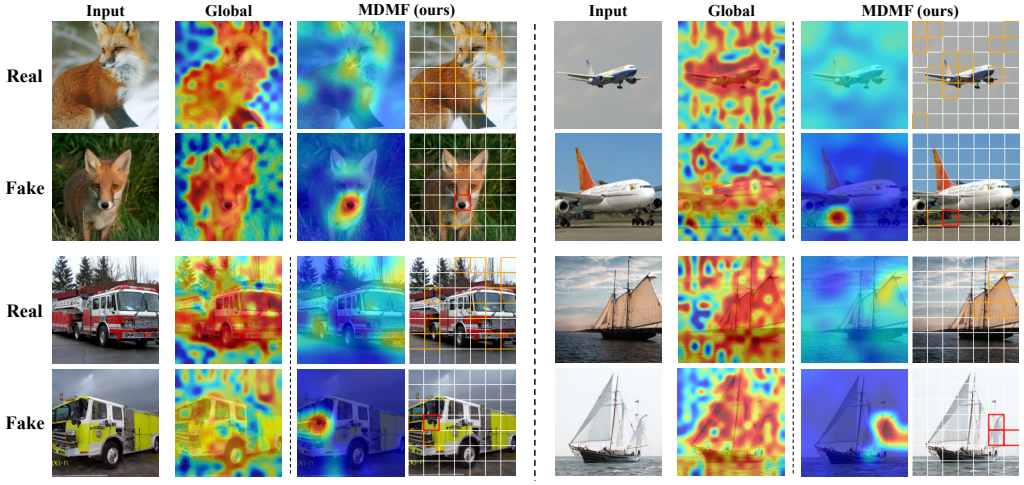


Figure 10: **Additional qualitative visualization on DiT-XL/2.** We compare real images and category-matched generated images, visualizing the responses of a global pooling baseline versus MDMF. Warmer colors indicate higher predicted likelihood of being fake.

G Limitations and Discussion

We acknowledge two practical considerations. First, inference uses a stored bank of real-image references and the runtime of the reference comparison grows with bank size. Appendix E.7 shows little change when bank size varies from 1k to 10k within a fixed sampling seed, while the difference across sampled banks is larger. Reference composition therefore remains relevant. Second, strong blur and noise reduce detection performance for both MDMF and F-ConV (Figure 4(c)). The high-scoring real examples in Appendix E.11 illustrate that atypical real images can also receive fake-side scores, but the visualizations do not establish the cause of each error. These limitations motivate testing calibrated operating points and reference selection under the intended deployment distribution.

H Broader Impact

The rapid proliferation of high-quality AI-generated images has substantially amplified the risk of visual disinformation, identity impersonation, and erosion of trust in digital media. By introducing a principled distributional perspective for AI-generated image detection, MDMF directly contributes to mitigating these harms: a more reliable detector enables platforms, fact-checkers, and end users to flag synthetic content with reduced false-negative rates, supporting the integrity of online discourse, journalism, and forensic investigation. The patch-level forensic signatures we learn also yield interpretable evidence (Figures 5 and 7–10), which can be inspected and audited, in line with calls for transparent decision-making in AI-driven content moderation. We are aware of two potential negative effects worth noting. First, any forensic detector can become a target of adversarial attack: malicious actors with knowledge of MDMF's PFS-MMD pipeline could attempt to craft perturbations that evade detection; we therefore recommend that production deployments combine MDMF with complementary signals (e.g., provenance metadata or watermarking) and continual updates to keep pace with emerging generators. Second, false positives, i.e., real images mistaken for AI-generated, can adversely affect photographers and artists; deployers should expose calibrated confidence levels and provide a redress mechanism rather than treating MDMF outputs as final verdicts.