RealStats: A Rigorous Real-Only Statistical Framework for Fake Image Detection

Haim Zisman
Bar-Ilan University

Uri Shaham
Bar-Ilan University

Abstract

As generative models continue to evolve, detecting AI-generated images remains a critical challenge. While effective detection methods exist, they often lack formal interpretability and may rely on implicit assumptions about fake content, potentially limiting their robustness to distributional shifts. In this work, we introduce a rigorous, statistically grounded framework for fake image detection that focuses on producing a probability score interpretable with respect to the real-image population. Our method leverages the strengths of multiple existing detectors by combining strong training-free statistics. We compute p-values over a range of test statistics and aggregate them using classical statistical ensembling to assess alignment with the unified real-image distribution. This framework is generic, flexible, and training-free, making it well-suited for robust fake image detection across diverse and evolving settings.

Code available at: `https://github.com/shaham-lab/RealStats`.

1 INTRODUCTION

The ability to distinguish real images from AI-generated content has become increasingly critical as generative models grow in both realism and accessibility. Recent advances in generative AI have significantly improved the visual fidelity of synthetic images, making them nearly indistinguishable from real ones to the human eye.

Addressing this growing challenge requires detection methods that are explicitly designed to balance two key properties central to our approach: interpretability and adaptability. Interpretability refers to the ability to assess the reliability of detection outputs by producing scores with well-defined statistical meaning. This is essential to ensure that the results can be understood and trusted in practice.

Adaptability denotes the ability to remain effective under distribution shifts caused by emerging generators. It requires operating without relying on assumptions about fake image distributions, while allowing future refinement and extension as new detection signals or insights become available.

Detection methods for AI-generated images are rapidly advancing. A major direction has focused on supervision, where models or ensembles are trained on labeled fake images to learn patterns that distinguish real from synthetic content Wang et al. (2020); Martin-Rodriguez et al. (2023); Epstein and et al. (2023). Although effective at capturing training distributions, their reliance on fake data limits adaptability, as performance often degrades under distribution shifts introduced by evolving generators Gragnaniello et al. (2021).

Recent work explores strong pre-trained representations Ojha et al. (2023); Cozzolino et al. (2024) and hand-crafted heuristics Ricker et al. (2024); He et al. (2024); Brokman et al. (2025) as detection statistics, using real images as a reference. These approaches enhance adaptability by avoiding training on fake data and offer a degree of interpretability by calibrating thresholds based on real image distributions. However, their adaptability remains limited, as the distribution shifts introduced by evolving generators can still affect performance. This may result from assumptions about how fake images differ from real ones, such as fixed magnitude relationships that do not generalize, or from the inability of individual statistics to consistently separate real and synthetic content across different generative models.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

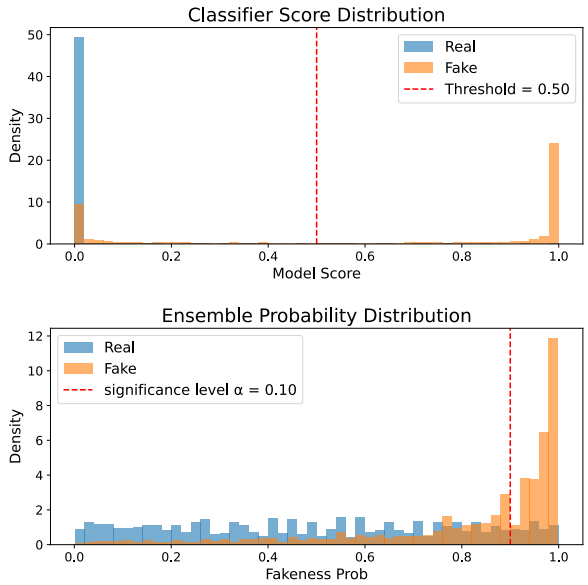


Figure 1: Illustration of the score interpretability gap between a supervised classifier Wang et al. (2020) and our statistical method. **Top:** A supervised model outputs scores that can separate real from fake images, but these scores are not inherently interpretable, as they lack clear statistical meaning and are often overconfident. **Bottom:** Our method produces calibrated p-values based on real image distributions. These values have a precise interpretation: the probability of observing such a result if the image were real. This enables principled decisions using a standard significance level.

We extend recent training-free detection methods by proposing a statistically rigorous framework based solely on real images. To improve adaptability, the method integrates multiple detection statistics to capture a broader range of deviations introduced by evolving generators, without making any assumptions about fake image distributions.

Our method returns a p-value under the null hypothesis that an image is drawn from the real distribution, providing a statistically grounded and interpretable reliability measure. This value is derived by aggregating several intermediate p-values, each capturing deviation under a different statistic, into a coherent score reflecting the overall evidence against the image being real. In summary, **our contribution** is a real-only detection framework that formulates AI-image detection as a hypothesis test on the real distribution and integrates multiple training-free statistics into a calibrated p-value. This yields an interpretable and adaptable structure with formal statistical meaning, allowing the method to incorporate new statistics without relying on fake data.

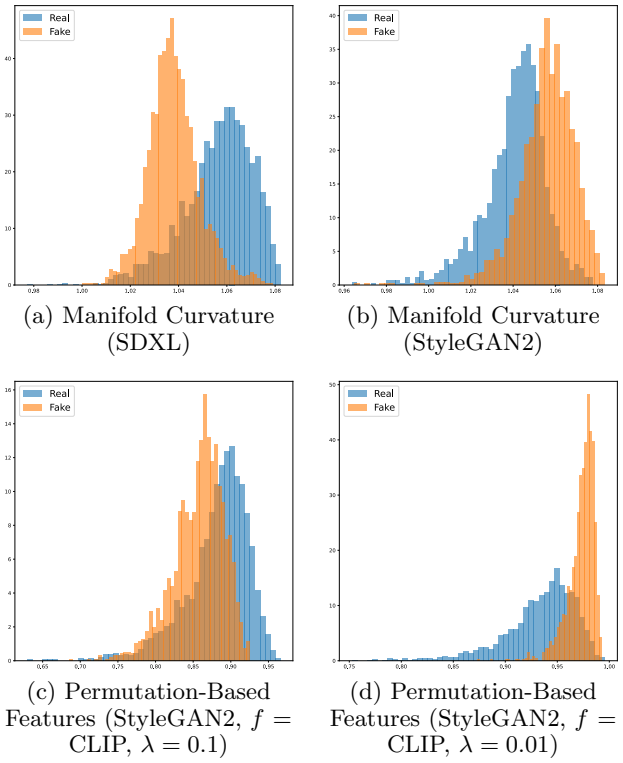
2 RELATED WORK

Efforts to detect AI-generated images have evolved across three main paradigms: fully supervised methods, few-shot and zero-shot detectors leveraging pretrained models, and unsupervised approaches rooted in statistical inference. Each class of methods has contributed different insights into the detection problem.

Supervised approaches are a widely explored direction, training discriminative models on labeled real and synthetic datasets. CNN-based methods Wang et al. (2020); Baraheem and Nguyen (2023) have demonstrated strong performance, and follow-up studies explored frequency cues Frank et al. (2020); Bammey (2024) and handcrafted features Martin-Rodriguez et al. (2023) to enhance robustness. However, these models tend to degrade when applied to images from unseen generative sources Gragnaniello et al. (2021). Online adaptations Epstein and et al. (2023) attempt to mitigate this by continuously updating with new fakes, yet remain tightly coupled to evolving synthetic distributions, limiting long-term adaptability. Moreover, they produce outputs that, while separating classes, often lack statistical meaning and interpretability, as illustrated in Figure 1.

To address these limitations, recent work has shifted toward few-shot detectors that aim to reduce reliance on large labeled datasets. These approaches typically leverage powerful pre-trained encoders, such as Dino Oquab et al. (2023), CLIP Radford et al. (2021); Cozzolino et al. (2024); Sha et al. (2023), GAN Goodfellow et al. (2014); Ojha et al. (2023), Diffusion Models Chu et al. (2024); Wang et al. (2023) and adapt them using a small number of synthetic examples, fine-tuning or calibrating on limited fakes. These methods retain a degree of supervision. As a result, they require retraining to handle emerging models and remain tied to internal model-specific scores, limiting prediction interpretability and extensibility to future generators.

A distinct class of methods avoids supervision in a training-free, zero-shot regime. These approaches draw on intrinsic statistical properties of real images and evaluate test statistics without access to fake samples. Perturbation-based detectors, such as RIGID He et al. (2024), assume that real images exhibit greater embedding stability under noise, using a similarity threshold to distinguish them from fakes. A related approach Brokman et al. (2025) leverages the geometry of a generative model's log-probability manifold, using curvature as a statistic. It assumes that generated images concentrate near local maxima, while real images lie in flatter regions. Similarly, reconstruction-based methods such as AEROBLADE Ricker et al. (2024) use latent diffusion model Rombach et al. (2022) au-



(a) Manifold Curvature (SDXL)

(b) Manifold Curvature (StyleGAN2)

(c) Permutation-Based Features (StyleGAN2, $f =$ CLIP, $\lambda = 0.1$)

(d) Permutation-Based Features (StyleGAN2, $f =$ CLIP, $\lambda = 0.01$)

Figure 2: Illustration of the adaptability gap in existing training-free methods. Each row shows statistics from a different method (top: manifold curvature; bottom: permutation-based features), and each column corresponds to a different test condition. **Top:** The manifold curvature statistic shows opposite behavior across generators. In (a) SDXL, real images have higher curvature than real ones, while in (b) StyleGAN2, the pattern reverses. **Bottom:** The permutation-based feature statistic is sensitive to the perturbation strength λ. In (c), real images yield higher scores than fakes, but in (d), the ordering flips. These shifts highlight that handcrafted statistics often rely on assumptions, such as fixed magnitude differences, that do not generalize across models or test settings.

toencoders to compute the reconstruction error, assuming that the generated images are better reconstructed than the real ones. These techniques rely on meaningful statistical measures that effectively distinguish real from generated images, offering a level of interpretability by grounding detection in well-defined statistics. However, they often assume a one-sided hypothesis test, which does not always hold in practice, as illustrated in Figure 2, which can reduce their ability to adapt to the behaviors of emerging generative models. In many cases, these methods are based on a single hand-crafted statistic and do not incorporate multiple signals. As a result, their generalization tends to be more effective within specific families of generative models.

Inspired by the effectiveness of recent zero-shot methods that leverage strong, model-independent statistics, we build on these insights to develop a unified, statistically rigorous framework. While prior approaches often rely on single handcrafted tests and assume fixed differences between real and fake statistics, typically calibrated through thresholds, our method extends these ideas without such assumptions. It integrates multiple detection signals using only real image distributions. By transforming scalar statistics into p-values and aggregating them with statistical methods, our approach provides interpretable outputs, principled error control, and improved adaptability to emerging generative models.

3 RATIONALE

3.1 Detection as Statistical Hypothesis Testing

We frame fake image detection as a statistical hypothesis multi-test Wasserman (2004). For each scalar statistic $s(x)$, we test whether an image x is plausible under the real image distribution.

Under the null hypothesis $H_0 \colon x \sim \mathbb{P}_{\text{real}}$, we evaluate the extremeness of $s(x)$ relative to a reference distribution estimated from real images $\mathcal{D}_{\text{real}} = \{x_1, \ldots, x_N\}$. The empirical cumulative distribution function (ECDF) is defined as:

$$\widehat{F}_N(t) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{s(x_i) \leq t\}, \quad x_i \sim \mathbb{P}_{\text{real}}. \tag{1}$$

Then, we compute a two-sided empirical p-value:

$$p(x) = 2 \cdot \min\left(\widehat{F}_N(s(x)),\, 1 - \widehat{F}_N(s(x))\right). \tag{2}$$

This testing procedure is applied across multiple scalar statistics $s_1(x), \ldots, s_K(x)$, each designed to capture a distinct aspect of deviation from the real image distribution. In next sections, we describe how these individual p-values (as defined in Eq. (2)) are aggregated into a single p-value.

3.2 Validity Conditions

The validity of the empirical p-value $p(x)$ relies on the following assumptions about the real image dataset $\mathcal{D}_{\text{real}} = \{x_1, \ldots, x_N\}$ and the test statistic $s(x)$:

1. **i.i.d. sampling**: The reference samples $x_1, \ldots, x_N$ are drawn independently and identically from the true real image distribution: $x_i \sim \mathbb{P}_{\text{real}}$.

2. **Independent statistics** The statistics $s_1(X), \ldots, s_K(X)$ used in aggregation are required to be independent under $\mathbb{P}_{\text{real}}$. Rather than assuming this, the method enforces independence by selecting a subset that satisfies formal independence criteria as explained in subsequent sections.

3. **distributional match**: A real test image x is also drawn from the same distribution: $x \sim \mathbb{P}_{\text{real}}$.

Under these conditions, the p-value defined in Eq. (2) is uniformly distributed on $[0, 1]$ under the null hypothesis H_0 (see Appendix B, Lemma 1).

4 PROPOSED METHOD

We propose a training-free detection method that tests whether an image is drawn from the distribution of real images, based on a rigorous statistical framework. The method consists of two phases. In the *null distribution modeling* phase, we compute a diverse collection of scalar statistics over real images and estimate their ECDFs. Then we find a subset of independent statistics that allow valid multi-test aggregation under the null hypothesis (see Fig. 3). In the *inference* phase, selected statistics are computed and mapped to p-values via stored ECDFs, then aggregated into a single interpretable p-value under the null distribution.

Each step of the null distribution modeling and detection pipeline is described in detail below.

4.1 Null Distribution Modeling Phase

This phase has two stages: *statistic extraction and empirical modeling* and *independent subset selection*. Given a reference dataset $\mathcal{D}_{\text{real}}$, it produces (1) set of stored ECDFs and (2) subset of independent statistics, enabling statistically valid p-value aggregation during detection.

Stage 1: Statistic Extraction and Empirical Modeling

In this stage, we compute a diverse set of scalar statistics $\mathcal{S}(x)$ over real images $x \in \mathcal{D}_{\text{real}}$, These statistics are computed using a broad set of feature extractors. For each statistic in $\mathcal{S}$, we compute and store its ECDF, forming the reference needed for hypothesis testing during detection.

Step 1.1: Multi-Detector Processing

Each image x is processed by a set of frozen feature extractors $\mathcal{F} = \{f_1, \ldots, f_m\}$, where each $f_j \colon \mathbb{R}^{h \times w \times C} \to \mathbb{R}^d$ maps an image to a high-dimensional embedding.

Inspired by RIGID He et al. (2024), we assess the stability of these embeddings by applying small Gaussian perturbations to the input and measuring the change in the resulting features. Specifically, we define additive noise $\delta \sim \mathcal{N}(0, I)$ and select a perturbation strength $\lambda_k \in \{\lambda_1, \ldots, \lambda_n\}$. The image score is then computed as the cosine similarity between clean and perturbed features:

$$s_{j,k}(x) = \text{sim}\left(f_j(x), f_j(x + \lambda_k \delta)\right). \tag{3}$$

This score reflects the robustness of the embedding under noise: real images typically produce more stable features, leading to higher similarity scores. However, this behavior is not universally consistent across encoders or noise level. To increase robustness and sensitivity, we evaluate a wide set of detector configurations, each defined by a pair (f_j, λ_k) combining a vision backbone and a perturbation level.

The complete set of scalar statistics extracted from image x is defined as:

$$\mathcal{S}(x) = \{s_{j,k}(x)\}_{j \le m,\, k \le n}.$$

The modular construction of $\mathcal{S}(x)$ allows new detectors to be easily incorporated, providing a clear path for future extension and adaptability.

Step 1.2: Empirical Two-Sided p-Value Estimation

For each scalar statistic in $\mathcal{S}(x)$, we evaluate its extremeness relative to the distribution of the same statistic computed over a reference set of real images.

For each configuration independently, we construct an ECDF as defined in Eq. (1).

Given image x, we compute a two-sided p-value for each statistic using Eq. (2), resulting in a p-value vector:

$$\mathbf{p}(x) = [p_{j,k}(x)]_{(j,k)}.$$

The two-sided formulation avoids assuming a fixed direction of deviation and accommodates diverse behaviors across detectors.

The ECDFs $\widehat{F}_s$ are estimated using real images only, which we treat as a representative sample drawn from the real population, allowing the method to remain adaptable as generative techniques evolve.

Stage 2: Independent Subset Selection

In this stage, we select a subset of scalar statistics that are independent under the null hypothesis. This enables statistically valid aggregation of their corresponding p-values. The selection is performed by test-

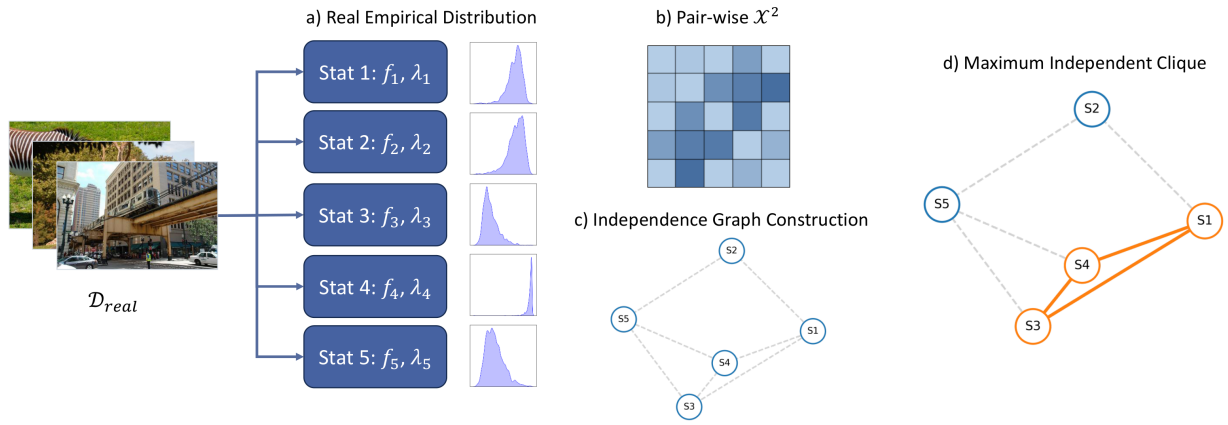


Figure 3: Overview of the null distribution modeling phase. (a) Real images from the reference dataset $\mathcal{D}_{\text{real}}$ are processed using multiple detector configurations (f_j, λ_k), producing scalar statistics whose empirical distributions are estimated and stored as ECDFs. (b) Pairwise statistical dependence among statistics is assessed via χ^2 tests over real samples. (c) The resulting relationships define an independence graph, where edges indicate accepted independence. (d) A maximal clique is extracted and regularized via a uniformity constraint to select a subset $\mathcal{I} \subseteq \mathcal{S}$ of independent statistics.

ing pairwise independence among statistics, constructing an independence graph, and finding a maximal clique under a uniformity constraint.

Step 2.1: Pairwise Dependence Testing

Stacking the vectors $\mathbf{p}(x_n)$ for all $x_n \in \mathcal{D}_{\text{real}}$, we form an $N \times T$ matrix of p-values, where $T = |\mathcal{S}|$.

To assess mutual dependence between statistics, we apply a pairwise χ^2 test to all $T(T-1)/2$ pairs of statistics, testing whether the joint distribution of p-values over real images deviates from independence.

Step 2.2: Independence Graph Construction

We construct an undirected graph $G = (\mathcal{V}, \mathcal{E})$, where each node $v_i \in \mathcal{V}$ represents a statistic $s_i \in \mathcal{S}$. For each pair (s_i, s_j), dependence is assessed via the χ^2 statistic and quantified using Cramér's V, which avoids the instability of χ^2 p-values in large samples Lin et al. (2013). An edge is added if the association is weak, i.e.,

$$(v_i, v_j) \in \mathcal{E} \quad \text{iff} \quad V(s_i, s_j) \leq \mathcal{V}_{\chi^2}.$$

where V_{χ^2} denotes the Cramér's V threshold. This graph captures the empirical structure of pairwise associations sufficiently weak to approximate independence.

Step 2.3: Maximum Clique Enumeration

We extract a maximal clique Bron and Kerbosch (1973) from the independence graph, corresponding to the largest subset $\mathcal{I} \subseteq \mathcal{S}$ of pairwise independent statistics. However, pairwise independence alone does not guarantee joint independence, which is required for valid multi-test aggregation. (details on aggregation methods in Section 4.2).

To address this, we select $\mathcal{I}$ by verifying that the aggregated p-values follow a uniform distribution under the null hypothesis. To mitigate large-sample artifacts analogous to those affecting χ^2 p-values, we apply the Kolmogorov–Smirnov test to a representative subsample of N_{KS} real images:

$$p\text{-value}_{\text{KS}} \geq \alpha_{\text{KS}}.$$

This ensures that the selected set supports valid downstream aggregation under the null hypothesis. Aggregation methods such as Stouffer's test or the min-p-value method (described in the next section) quantify the joint deviation from the null across statistics.

In practice, detection performance remains an important consideration. Although the set of statistics is chosen based on independence and null alignment, encoders such as DINOv2 and CLIP are known to produce particularly strong discriminative features. Thus, when multiple maximal cliques satisfy these conditions, we prioritize those containing these encoders to improve effectiveness.

4.2 Inference Phase

Given a candidate image x, we compute only the subset of scalar statistics $\mathcal{I} \subseteq \mathcal{S}$ that were selected during the null distribution modeling phase as independent (Step. 2.3), as illustrated in the detection pipeline

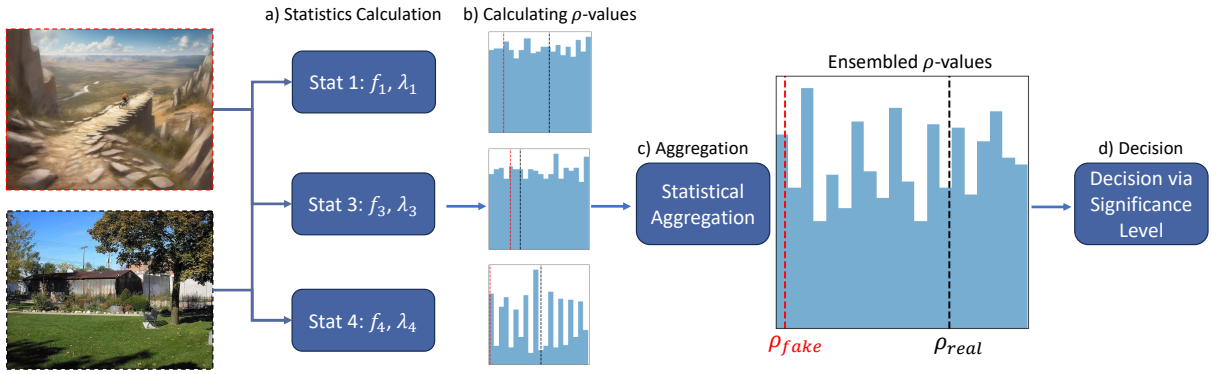


Figure 4: Overview of the inference phase. (a) A candidate image is processed using only the subset of detector configurations corresponding to the selected statistics $\mathcal{I}$. (b) Each resulting statistic is mapped to a two-sided p-value using the stored ECDFs. (c) The set of p-values is aggregated using a statistical aggregation method, such as Stouffer's test. (d) The final decision is taken by comparing the unified p-value to a predefined significance level.

(Fig. 4). To do so, we apply multi-detector processing (Step. 1.1), restricted to the relevant configurations that define $\mathcal{I}$.

Each statistic $s \in \mathcal{I}$ is then mapped to a two-sided p-value using the corresponding ECDF $\widehat{F}_s$ estimated from real images (Step. 1.2). These individual p-values quantify how extreme each selected statistic is under the null hypothesis.

To summarize this evidence into a single interpretable score, we apply one of two statistical aggregation methods, both of which assume independence among the selected statistics.

Stouffer's Test: This method transforms each p-value into a standard normal Z-score:

$$z_i = \Phi^{-1}(p_i) \tag{4}$$

$$Z = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} z_i, \quad P_{\text{Stouffer}} = \Phi(Z) \tag{5}$$

where Φ is the standard normal CDF and $K = |\mathcal{I}|$ is the number of aggregated statistics. Under the null, $P_{\text{Stouffer}} \sim \mathcal{U}[0,1]$ (see Appendix B, Lemma 2). This method is effective when several statistics show moderate deviations from the null that, while individually weak, become significant when combined.

Minimum p-Value: This method emphasizes the strongest individual evidence:

$$P_{\min} = \min_i p_i, \quad F_{P_{\min}}(t) = 1 - (1-t)^K \tag{6}$$

where $P_{\min}$ is the minimum of K independent p-values, and $F_{P_{\min}}$ is its CDF. Under the null hypothesis, and assuming p-values are computed as in Eq. (2), the resulting test remains valid with uniform distribution (see Appendix B, Lemma 3). This method is well suited when only a small subset of detectors is expected to distinguish fakes.

The resulting unified p-value provides a statistically valid and interpretable measure of authenticity, quantifying the deviation of the candidate image x from the distribution of real images.

The complete procedures for null modeling and inference are detailed in Appendix A.3, while a comprehensive study of runtime and memory usage is provided in Appendix D.1.

5 EXPERIMENTS

Datasets We evaluate our method on several large-scale benchmark datasets that collectively span a broad spectrum of generative models and content domains. The CNNSpot dataset Wang et al. (2020) consists of real and synthetic images across 20 LSUN Yu et al. (2015) categories, primarily generated using early convolutional and GAN-based models. The Universal Fake Detect dataset Ojha et al. (2023) expands this setup to include more recent latent diffusion architectures. To further assess robustness on high-fidelity diffusion-based content, we incorporate a Stable Diffusion Face dataset that provides high-quality generated images from SDXL and SDv2 tobecwb (2023). Finally, to evaluate performance on challenging real-world generative systems, we include the Synthbuster Bammey (2024) and GenImage Zhu et al. (2023) datasets.

Across these datasets, our aggregated dataset comprises a total of 187K images (93K real and 94K fake), approximately balanced between real and generated content. This large-scale and diverse benchmark enables evaluation of detection performance across multiple generation paradigms. A complete list of generative models is provided in Appendix C, and the exact dataset splits and reproducibility details are described in Appendix E.

Baselines We compare our method against three recent training-free, approaches that represent complementary statistical detection strategies. RIGID measures embedding stability under perturbation and serves both as a strong standalone baseline and as a foundational component within our framework. AEROBLADE evaluates reconstruction error using latent autoencoders, while ManifoldBias Brokman et al. (2025) quantifies statistical curvature in the latent space of pre-trained diffusion models.

Implementation Details Our method relies on frozen visual encoders and Gaussian perturbations. All detectors operate at image resolutions of 512x512. The specific encoders and perturbation strengths used in our experiments, including hyper-parameters are summarized in Appendix A.1 and Appendix A.2 respectively.

To ensure a statistically valid evaluation, we explicitly partition the real images into two disjoint subsets: 30% is allocated to the null distribution modeling phase (i.e., ECDF estimation), while the remaining 70% is held out for evaluation. This split ensures that no test-time inference image is used during calibration. Fake images are evaluated only against the held-out portion of real images. All detection methods, including ours and the baselines, are evaluated under the same protocol using shared real and fake subsets for each generator.

At the time of writing, the official implementation of RIGID is not publicly available. We implemented the method based on the description in the original paper, using its best-reported configuration: a DINOv2 backbone with a perturbation strength of 0.05. Due to variability in reported results across different works, we will release our full implementation, which includes both the original RIGID setup and extended variants with alternative feature extractors and perturbation levels.

Metrics We report two threshold-free metrics to evaluate detection performance: *Area Under the ROC Curve (AUC)* and *Average Precision (AP)*, each computed per generative model. Unlike methods that require selecting a fixed threshold, our approach produces p-values from hypothesis testing. Threshold-based metrics like accuracy are thus not directly comparable and may not reflect statistical confidence. AUC and AP provide a more appropriate view of performance across the decision range.

5.1 Interpretability Does Not Come at the Price of Performance

A central goal of our method is to provide interpretable and reliable detection. Interpretability is a direct result of returning a p-value for each inference image x, which is a value with clear statistical interpretation, rather than an ambiguous "realness score". Crucially, we demonstrate that despite this design choice, our method achieves performance on par with state-of-the-art training-free detectors.

Table 1: Average AUC and AP (with standard deviation) across all generators splits.

Model	AUC	AP
Manifold Bias	0.761 ± 0.179	0.753 ± 0.169
RIGID	0.769 ± 0.194	0.765 ± 0.189
AEROBLADE	0.697 ± 0.161	0.697 ± 0.163
Ours (Stouffer)	0.756 ± 0.135	0.743 ± 0.133
Ours (Min-p)	0.775 ± 0.126	0.756 ± 0.119

Note. Each split is balanced between real and fake samples using stratified sampling to ensure fair comparison across methods.

As shown in Table 1, our method achieves competitive AUC and AP compared to state-of-the-art training-free baselines. While Manifold Bias and RIGID obtain slightly higher peak scores, our approach offers comparable performance with substantially lower variance across generators, reflecting more consistent behavior. Importantly, this competitiveness comes without sacrificing interpretability

Empirically, different methods peak on some generators but drop on others, showing inconsistent stability (see Figure 5). Manifold Bias, for instance, scores highly on GauGAN but drops on StarGAN and SDv2. RIGID performs well on SDXL but struggles on SDv1.4, while AeroBlade is strong on ADM but weak on SDv1.5. Our method shows some weaknesses on datasets like CycleGAN, yet overall maintains more balanced results, supported by its multi-RIGID variants that prevent collapse when a single statistic fails. Complete AUC and AP values are provided in Appendix D.5.

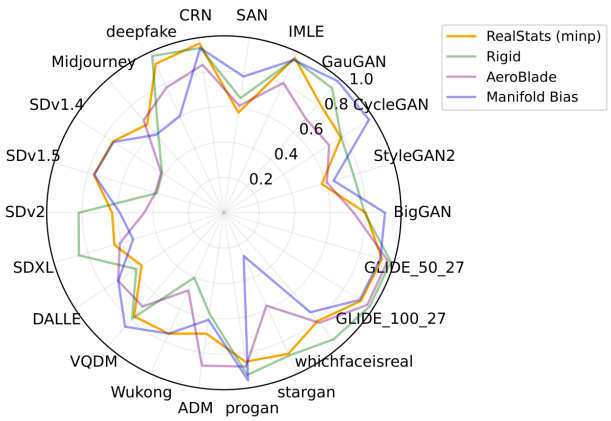


Figure 5: Per-generator AUC comparison across methods shown in radar format.

5.2 Adaptability in Action: Improving Performance on Challenging Generators

While our method performs on par with state-of-the-art training-free detectors overall, we observe weaker results on certain generators where *ManifoldBias* excels, including **GauGAN**, **CycleGAN**, and **SAN** (see Figure 5). To demonstrate adaptability, we revisit these cases and integrate the *ManifoldBias* statistic under the same experimental setup.

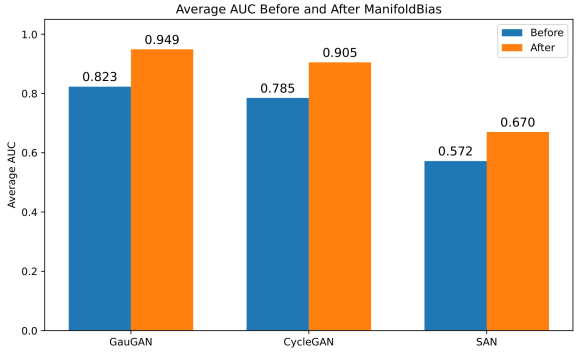
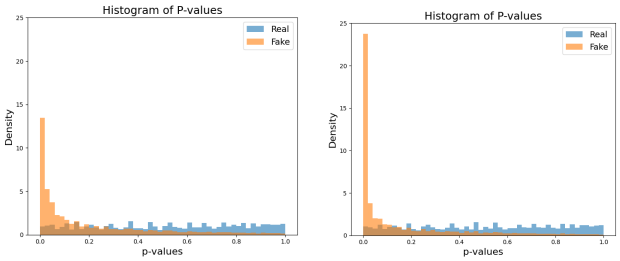


Figure 6: Average AUC of the Min-p ensemble before and after incorporating *ManifoldBias* on GauGAN, CycleGAN, and SAN.

As shown in Figure 6, the ensemble gains clear improvements on these generators, closing the gap with baselines by leveraging the strength of *ManifoldBias*.

To highlight the broader impact, the benefit extends to our full test benchmark of 160K images, where incorporating *ManifoldBias* improves p-value separation (Figure 7) and raises overall AUC scores. These results demonstrate the modularity and adaptability of our framework, showing it can respond to evolving generative content by selectively integrating new, independent statistics when needed.



(a) Entire Dataset, Min-p ensemble (before)

(b) Entire Dataset, Min-p ensemble (after)

Figure 7: Example of improved p-value separation with *ManifoldBias* in the Min-p ensemble. Before (left), real and fake distributions overlap considerably; after (right), fake samples shift toward zero while real remain uniform.

5.3 Interpretability Qualititive Comparison

Appendix D.4 provides a qualitative comparison illustrating the interpretability of our method, where p-value patterns reveal how image deviations from the reference distribution manifest in a transparent and meaningful way.

5.4 Fast, Scalable, and Memory-Efficient

Our method is designed for high-throughput inference and scales efficiently with GPU parallelism. Runtime improves with more workers, while memory remains moderate even with many statistics and large batches. In addition, we demonstrate that the computational overhead of the independence-testing stage (Phase 1) is negligible relative to the forward-pass cost.

Unlike training-free methods built on heavy autoencoders (e.g., Stable Diffusion), it achieves faster inference and lower memory use. The approach runs on single- or multi-GPU setups with minimal overhead, making it practical for large-scale deployment.

Full runtime, scalability, and memory analyses are provided in Appendix D.1.

5.5 Effect of Reference Distribution Misalignment

The empirical null modeling procedure relies on the assumption that the set of real images used to construct the ECDFs provides a representative sample of the real-image population encountered at inference time. Two distinct aspects can challenge this assumption.

The first is finite sampling, where the empirical ECDF may not perfectly approximate the true CDF of the

real-image population since the reference set is finite. In this case, the ECDF becomes a biased or coarse estimate: the outputs still provide useful relative ranking for distinguishing real from fake images, but the formal probabilistic meaning of a p-value under the null is no longer guaranteed. Because real images are generally easy to obtain, this limitation can often be mitigated by modestly enlarging or diversifying the reference set, and a practical way to assess representativeness is to compare the empirical distribution of p-values on a validation set to the expected uniform distribution.

The second is distributional shift, where the test-time real distribution differs structurally or semantically from the reference domain, even if the reference set is large. Such shifts occur, for example, when content categories are heavily unbalanced, resolution changes, or new semantic attributes appear. In such cases, the p-values deviate from the expected uniform behavior, meaning they no longer function as true p-values in the classical probabilistic sense. To examine this effect, we designed a controlled domain-shift experiment in which the reference ECDF was constructed from only 1k FFHQ Karras et al. (2020) faces ($< 4\%$ of the test size), while inference was performed on 30k high-resolution CelebA Liu et al. (2015) faces. In this setup, faces in the test domain are abundant and diverse, whereas in the reference set they represent only a small and less varied subset. Moreover, CelebA differs from FFHQ both in structural properties (in particular, higher resolution than the vast majority of reference samples) and in overall domain composition (e.g., pose and attribute distribution), making it an appropriate setting for examining how RealStats behaves when the reference distribution captures only a limited portion of the true test domain.

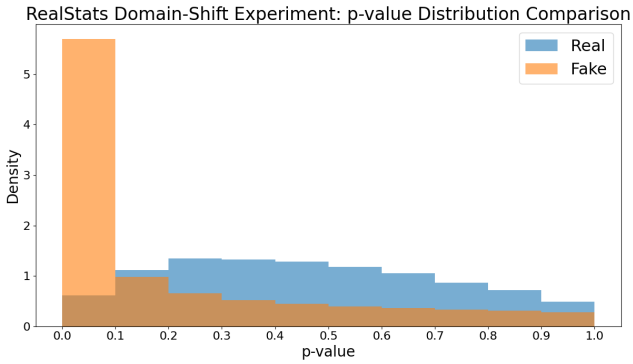


Figure 8: p-value density distributions under reference-test domain mismatch. Real samples (blue) deviate from uniform behaviour due to calibration mismatch, while fake samples (orange) remain concentrated near zero, demonstrating preserved discriminative signal.

Figure 8 visualizes the resulting p-value densities for real and fake samples under this mismatch. Real samples deviate from uniformity, indicating calibration misalignment, while fake samples remain concentrated near zero, showing preserved discriminative signal.

Even in this scenario, RealStats maintains strong real-fake separation (AUC = 0.79, AP = 0.845), outperforming a perturbation-based heuristic evaluated under identical conditions (AUC = 0.70, AP = 0.74). This indicates that while the misalignment violates the strict probabilistic meaning of the p-values, the aggregated scores remain highly discriminative.

5.6 Robustness Under Image Corruptions

We evaluate robustness to JPEG compression and Gaussian blur at test time without altering the reference distribution or evaluation setup. With the Min-p ensemble, blur keeps performance stable or slightly improved, while JPEG compression causes a moderate drop (5% AUC, 6.4% AP) but does not affect the validity of p-values. Overall, the framework remains resilient to these corruptions.

Additional results and visualizations are provided in Appendix D.2.

6 LIMITATIONS

Our framework is subject to two main limitations. First, the performance of our method depends on the selected statistic clique. In some cases, excluding correlated yet informative detectors may reduce separability. To mitigate this, we prioritize valid cliques that include strong and diverse detectors such as DINOv2 and CLIP.

Second, the validity of the resulting p-values depends on the quality of the reference distribution. As discussed earlier, both finite sampling and distributional shift can affect the representativeness of the empirical null model. Reliability can be assessed by examining the p-value distribution on a validation set and comparing it with the expected uniform pattern. In practice, misalignment is often manageable, as real data is typically easy to obtain.

7 CONCLUSION

We presented a statistical framework for fake image detection focused on two main properties: interpretability and adaptability. Through extensive experiments, we showed that our method achieves competitive performance with state-of-the-art training-free detectors, while remaining robust, scalable, and modular.

References

Bammey, Q. (2024). Synthbuster: Towards detection of diffusion model generated images.

Baraheem, S. S. and Nguyen, T. V. (2023). Ai vs. ai: Can ai detect ai-generated images?

Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Brokman, J., Giloni, A., Hofman, O., et al. (2025). Manifold induced biases for zero-shot and few-shot detection of generated images. *arXiv preprint arXiv:2504.15470*.

Bron, C. and Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph.

Chen, Q. and Koltun, V. (2017). Photographic image synthesis with cascaded refinement networks.

Choi, Y., Choi, M., Kim, M., et al. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). Stargan v2: Diverse image synthesis for multiple domains.

Chu, B., Xu, X., Wang, X., et al. (2024). Fire: Robust detection of diffusion-generated images via frequency-guided reconstruction error. *arXiv preprint arXiv:2412.07140*.

Cozzolino, D., Poggi, G., Corvi, R., et al. (2024). Raising the bar of ai-generated image detection with clip.

Dai, T., Cai, J., Zhang, Y., et al. (2019). Second-order attention network for single image super-resolution.

Deng, J., Dong, W., Socher, R., et al. (2009). Imagenet: A large-scale hierarchical image database. Ieee.

Egiazarian, V., Kuznedelev, D., Voronov, A., et al. (2024). Accurate compression of text-to-image diffusion models via vector quantization. *arXiv preprint arXiv:2409.00492*.

Epstein, D. and et al. (2023). Progressive online fake image detection with resnet adaptation.

Frank, J., Eisenhofer, T., and et al. (2020). Leveraging frequency analysis for deepfake image recognition.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. (2014). Generative adversarial nets.

Gragnaniello, D., Cozzolino, D., Marra, F., et al. (2021). Are gan generated images easy to detect? a critical analysis of the state-of-the-art. IEEE.

He, Z., Chen, P.-Y., and Ho, T.-Y. (2024). Rigid: A training-free and model-agnostic framework for robust ai-generated image detection. *arXiv preprint arXiv:2405.20112*.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T., Laine, S., Aittala, M., et al. (2020). Analyzing and improving the image quality of stylegan.

Li, J., Shen, T., Gu, Z., et al. (2024). Adm: Accelerated diffusion model via estimated priors for robust motion prediction under uncertainties. IEEE.

Li, K., Zhang, T., and Malik, J. (2019). Diverse image synthesis from semantic layouts via conditional imle. 2019 ieee.

Lin, M., Lucas, H. C., and Shmueli, G. (2013). Research commentary - too big to fail: Large samples and the p-value problem.

Lin, T.-Y., Maire, M., Belongie, S., et al. (2014). Microsoft coco: Common objects in context. Springer.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild.

Martin-Rodriguez, F., Garcia-Mojon, R., and Fernandez-Barciela, M. (2023). Detection of ai-created images using pixel-wise feature extraction and convolutional neural networks.

Midjourney (2023). Midjourney (v5.1) [text-to-image model]. `https://www.midjourney.com/`.

Nichol, A., Dhariwal, P., Ramesh, A., et al. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Ojha, U., Li, Y., and Lee, Y. J. (2023). Towards universal fake image detectors that generalize across generative models.

OpenAI (2023). Dall·e 3. `https://openai.com/dall-e-3`. Accessed: 2025-07-23.

Oquab, M., Darcet, T., Moutakanni, T., et al. (2023). Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization.

Podell, D., English, Z., Lacey, K., et al. (2023). Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. PmLR.

Ricker, B., McCloskey, S., and Kambhamettu, C. (2024). Aeroblade: Detection of ai-generated images via reconstruction errors from off-the-shelf autoencoders.

Rombach, R., Blattmann, A., Lorenz, D., et al. (2022). High-resolution image synthesis with latent diffusion models.

Rossler, A., Cozzolino, D., Verdoliva, L., et al. (2019). Learning to detect manipulated facial images. inproceedings of the ieee.

Schuhmann, C., Vencu, R., Beaumont, R., et al. (2021). Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Sha, Z., Li, Z., Yu, N., and Zhang, Y. (2023). De-fake: Detection and attribution of fake images generated by text-to-image generation models.

tobecwb (2023). Stable diffusion face dataset. `https://github.com/tobecwb/stable-diffusion-face-dataset`.

Wang, S.-Y., Wang, O., Zhang, R., et al. (2020). Cnn-generated images are surprisingly easy to spot... for now.

Wang, Z., Bao, J., Zhou, W., et al. (2023). Dire for diffusion-generated image detection.

Wasserman, L. (2004). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

Yu, F., Seff, A., Zhang, Y., et al. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks.

Zhu, M., Chen, H., Yan, Q., et al. (2023). Genimage: A million-scale benchmark for detecting ai-generated image.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

Appendix

This appendix contains supplementary material that supports the methods and findings in the main paper. It includes:

- A. Algorithmic Details: Pseudocode for the two phases of the framework, detector configurations, and hyperparameter settings.
- B. Statistical Validity: Lemmas justifying the validity of the proposed p-values construction and aggregation.
- C. List of Generative Models: Comprehensive list of generative models used in evaluation.
- D. Full Experimental Results: Detailed baseline comparison table, runtime and memory analysis, robustness evaluations, and qualitative visualization of interpretability.
- E. Dataset and Code: Description of released data splits, code and setup for reproducibility.

A Algorithmic Details

This section provides a detailed algorithmic overview of the phases and the components that constitute our framework.

A.1 Detectors Configurations

Table 2: Configuration of the statistics set used across experiments. Each row specifies the feature extractor and the perturbation strengths λ applied during statistic extraction.

Feature Extractor	**Perturbation Strengths λ**
CLIP ViT-L/14	0.05, 0.10
DINOv2 ViT-L/14	0.05, 0.10
DINOv3 ViT-S/16	0.05, 0.10
DINOv3 ViT-H/16	0.05, 0.10
ConvNeXT	0.05, 0.10
BEiT ViT-L/16	0.05, 0.10

A.2 Hyperparameter Configurations

The following hyperparameter values were used consistently across all experiments:

- Aggregation method: minimum p-value.
- Number of bins for pairwise χ^2 tests: $B_{\chi^2} = 15$.
- Number of bins for ECDF estimation: $B_{\text{ECDF}} = 400$.
- Significance level for KS test of uniformity: $\alpha_{\text{KS}} = 0.05$.
- Maximum Cramér's V threshold for accepting independence in χ^2 tests: $V_{\chi^2} = 0.07$.

These values were chosen after exploring a range of configurations during development. Specifically, we evaluated $B_{\text{ECDF}} \in [200, 1000]$, $B_{\chi^2} \in [10, 50]$. We also tested thresholds up to $V_{\chi^2} = 0.1$ for Cramér's V independence criterion and up to $\alpha_{\text{KS}} = 0.1$ for the KS uniformity test.

The final configuration was selected based on its empirical stability and the ability to maintain uniformity of aggregated p-values across different datasets and experimental setups.

A.3 Pseudo-code

Algorithm 1 Null Hypothesis Modeling

Input: Reference dataset $\mathcal{D}_{\text{real}}$
Output: ECDFs $\{\widehat{F}_{j,k}\}$, Subset $\mathcal{I} \subseteq \mathcal{S}$
// Statistic extraction and ECDF modeling (parallelizable)
for all detector (j, k) **do**
  for all $x_i \in \mathcal{D}_{\text{real}}$ **do**
    Compute $s_{j,k}(x_i)$
  end for
  Estimate ECDF $\widehat{F}_{j,k}$ from $\{s_{j,k}(x_i)\}$
end for
// Independence subset selection (pairwise tests are parallelizable)
Initialize independence graph $G = (\mathcal{V}, \mathcal{E})$
for all statistic pairs (s_i, s_j) **do**
  Compute Cramér's $V(s_i, s_j)$ from χ^2 statistic on p-values
  if $V(s_i, s_j) \leq V_{\chi^2}$ **then**
    Add edge (s_i, s_j) to graph G
  end if
end for
Find all cliques in G
Keep cliques passing KS-test with threshold α_{KS}
Select valid clique maximizing coverage of preferred statistics, breaking ties by size

Algorithm 2 Inference

Input: Image x, ECDFs $\{\widehat{F}_{j,k}\}$, subset $\mathcal{I}$, level α
Output: Decision: `REAL` or `FAKE`
// Statistic computation and mapping (parallelizable)
for all statistic $(j, k) \in \mathcal{I}$ **do**
  Compute statistic $s_{j,k}(x)$
  Compute p-value $p_{j,k}(x)$ using $\widehat{F}_{j,k}$
end for
Aggregate $\{p_{j,k}(x)\}$ into unified p-value (e.g., Stouffer or min-p)
if unified p-value $< \alpha$ **then return** `FAKE`
else return `REAL`

B Statistical Validity

This section presents lemmas that justify the statistical validity of our hypothesis testing and p-value aggregation procedures. The results establish that the constructed p-values are valid under the null hypothesis, assuming independence and i.i.d. real samples.

Lemma 1: Validity of Empirical p-Values

Let $x \sim \mathbb{P}_{\text{real}}$ and let $s(x)$ be a scalar statistic. Let $\widehat{F}_N$ denote the ECDF over N i.i.d. samples $x_1, \ldots, x_N \sim \mathbb{P}_{\text{real}}$. Define the two-sided empirical p-value as

$$p(x) = 2 \cdot \min\left(\widehat{F}_N(s(x)),\, 1 - \widehat{F}_N(s(x))\right).$$

Then under the null hypothesis,

$$p \sim \mathcal{U}[0,1] \quad \text{as } N \to \infty.$$

Note. This follows directly from the definition of empirical p-values. We treat this as a definitional result.

Lemma 2: Stouffer Aggregation Validity

Let $p_1, \ldots, p_K \sim \mathcal{U}[0,1]$ be independent. Define

$$Z = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} \Phi^{-1}(p_i) \tag{7}$$

$$P_{\text{Stouffer}} = \Phi(Z) \tag{8}$$

where Φ is the standard normal cumulative distribution function. Then under the null,

$$P_{\text{Stouffer}} \sim \mathcal{U}[0,1].$$

Proof. Since each $p_i \sim \mathcal{U}[0,1]$, the transformation $z_i = \Phi^{-1}(p_i)$ yields $z_i \sim \mathcal{N}(0,1)$. Independence of the p_i's implies independence of the z_i's. Therefore,

$$Z = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} z_i \sim \mathcal{N}(0,1),$$

and thus $P_{\text{Stouffer}} = \Phi(Z) \sim \mathcal{U}[0,1]$. □

Lemma 3: Minimum p-Value Aggregation Validity

Let $p_1, \ldots, p_K \sim \mathcal{U}[0,1]$ independently, and define

$$P_{\min} = \min_i p_i, \quad F_{P_{\min}}(t) = 1 - (1-t)^K.$$

Then under the null,

$$F_{P_{\min}}(P_{\min}) \sim \mathcal{U}[0,1].$$

Proof. For any $t \in [0,1]$,

$$\mathbb{P}(P_{\min} \le t) = 1 - \mathbb{P}(p_1 > t, \ldots, p_K > t) = 1 - (1-t)^K.$$

Define the transformation

$$U = F_{P_{\min}}(P_{\min}) = 1 - (1 - P_{\min})^K.$$

Then for any $u \in [0,1]$,

$$\begin{aligned}\mathbb{P}(U \le u) &= \mathbb{P}\left(P_{\min} \le 1 - (1-u)^{1/K}\right) \\ &= F_{P_{\min}}(1 - (1-u)^{1/K}) = u,\end{aligned}$$

so $U \sim \mathcal{U}[0,1]$. □

These lemmas ensure that our p-value construction and aggregation yield statistically valid outputs under the null hypothesis, assuming independence and i.i.d. sampling from the real distribution.

C List of Generative Models

Our evaluation includes synthetic images produced by a diverse range of generative models, drawn from several benchmark datasets: CNNSpot, Universal Fake Detect, Stable-Diffusion-Faces, Synthbuster and GenImage. These datasets collectively span multiple generations of image synthesis techniques and architectural families.

The following models were used in our experiments: ProGAN Karras et al. (2017) , StyleGAN (WhichFaceIsReal) , StyleGAN2 Choi et al. (2020) , BigGAN Brock et al. (2018) , GauGAN Park et al. (2019) , CycleGAN Zhu et al. (2017) , StarGAN Choi et al. (2018) , Cascaded Refinement Networks (CRN) Chen and Koltun (2017) , Implicit Maximum Likelihood Estimation (IMLE) Li et al. (2019) , Spatially-Adaptive Normalization (SAN) Dai et al. (2019) , DeepFake Rossler et al. (2019) , Stable Diffusion (v1.4, v1.5, v2, XL) Rombach et al. (2022); Podell et al. (2023) , ADM Li et al. (2024) , VQDM Egiazarian et al. (2024) , WuKong , Glide Nichol et al. (2021) , Midjourney v5 Midjourney (2023) , and DALL-E 3 OpenAI (2023).

As sources of real images, we relied on benchmark datasets including LSUN Yu et al. (2015), MSCOCO Lin et al. (2014), ImageNet Deng et al. (2009), and LAION Schuhmann et al. (2021). Both our reference and test sets were constructed from these datasets, while ensuring that they consist of disjoint samples. In addition, we incorporated a large number of additional images from MSCOCO that are not part of the predefined benchmark subsets, thereby broadening the diversity and coverage of our real image pool.

D Full Experimental Results

In this section, we present detailed experimental results.

D.1 Runtime and Memory Usage Analysis

Runtime Under Increasing Statistics and Parallelism We evaluated the run-time and memory efficiency of our method across the entire phases of the framework using a representative subset of the MSCOCO dataset and SDXL fake images, 2K images total. All experiments were conducted on a multi-core CPU system and a single NVIDIA A100 80GB GPU, using parallelization with multiple workers.

To assess the scalability of our statistic extraction, we measured its runtime under different GPU worker configurations as the number of selected statistics increases. Figure 9 confirm that while runtime naturally increases

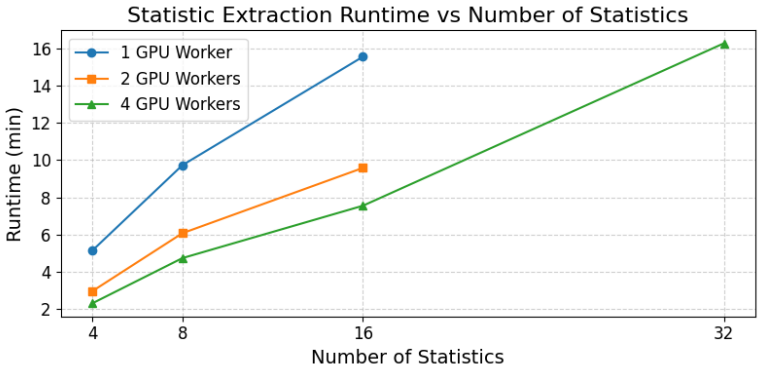


Figure 9: Statistic extraction runtime versus number of statistics under different levels of parallelism. Our method scales efficiently across multiple GPU workers.

with the number of statistics, our method benefits greatly from parallelism. For instance, with 16 statistics, runtime drops from over 15 minutes with 1 worker to under 8 minutes with 4 workers. At inference time, the per-sample runtime remains low: processing 2000 samples with 4 statistics on a single worker takes only 5.5 minutes, which corresponds to **0.165 seconds per sample**. This is substantially faster than ManifoldBias and AEROBLADE, which require 2.4 seconds and 5.1 seconds per sample, respectively.

To ensure fair comparison, we limited our evaluation to a single GPU, as baseline methods do not support multi-GPU or multi-process execution. Our implementation, however, is designed for scalability: it distributes

computation across multiple workers, reducing CPU-GPU context-switch overhead. Each worker can operate on a separate GPU, compute statistics independently, and return compact ECDFs to the CPU, enabling efficient deployment in both single and multi-GPU environments.

Empirical Complexity Analysis of Independence-Selection Phase Next, we assess the cost of independence testing and clique selection. Figure 10 demonstrates that both the pairwise χ^2 matrix computation and

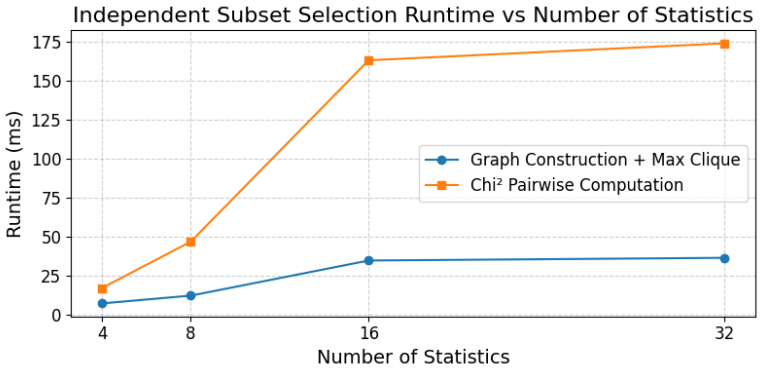


Figure 10: Runtime of statistical independence analysis and max-clique selection as the number of statistics increases. Both components scale efficiently.

the graph-based clique selection remain tractable, even with 32 statistics, requiring under 200ms end-to-end. This supports the practical deployment of our method.

Building on this analysis, we now examine the behaviour of the independence-selection stage under a wider range of conditions, including less favourable regimes that may produce denser independence graphs.

Empirical Complexity for Denser Independence Graphs To further assess computational efficiency, we conducted an experiment analysing the behaviour of maximal-clique enumeration within the independence-selection stage. Although maximal-clique search is NP-hard in theory, its practical cost is strongly influenced by the structure of the independence graph. In our settings, two characteristics ensure that this stage remains lightweight:

- independence graphs contain a relatively small number of statistics (up to 32 in all evaluated configurations).
- Cramer's V pruning produces sparse connectivity patterns before clique extraction.

As a result, clique enumeration consistently requires only tens of milliseconds in real executions, as shown in Figure 10.

To examine scalability under less favourable conditions, we simulated Phase 1, Stage 2 using synthetic statistics (200k values each, normalised to $[0, 1]$ with small perturbations) while varying the number of candidate statistics. For each configuration, pairwise Cramer's V values were computed, the corresponding independence graph was constructed, and Bron-Kerbosch enumeration was performed. The recorded runtimes are summarised in Table 3. Even under artificially dense configurations, runtime grows smoothly with graph size and remains within practical limits.

Table 3: Scaling behaviour of the independence-selection stage under synthetic dense conditions.

No. of Stats	Cramer's V (ms)	Graph + clique enumeration (ms)
8	160.3	0.52
16	692.3	1.55
32	2855.2	2.73
64	11514.9	8.26
128	46504.9	44.21

These findings show that, despite worst-case theoretical complexity, the independence graphs encountered in practice keep maximal-clique extraction computationally negligible relative to feature computation and statistic evaluation. Consequently, this stage does not present a scalability bottleneck within the RealStats framework.

Memory Consumption and Scalability We further evaluate peak GPU memory usage under different configurations and compare it to the baselines.

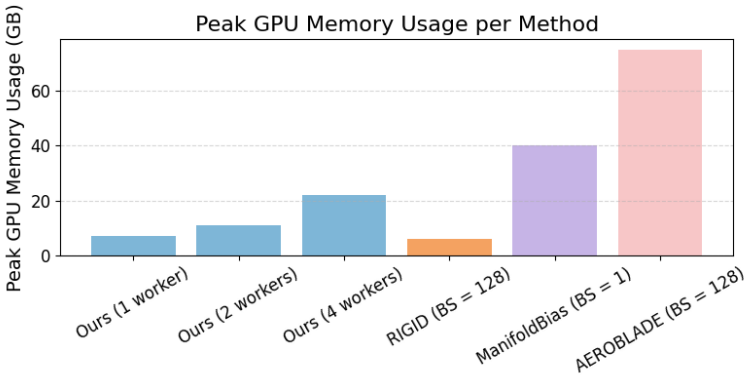


Figure 11: Peak GPU memory usage comparison with specified batch size. Our method, achieves high throughput with significantly reduced memory requirements.

Figure 11 shows that our method requires just 7-22GB of GPU memory with 1-4 workers at batch size 128. In contrast, AEROBLADE uses 76GB under the same setup, while ManifoldBias, with 8 internal perturbations, consumes 40GB at batch size 1 and cannot scale to larger batches. Despite using multiple statistics, our method remains memory-efficient and scalable. ECDF storage is also minimal, just 0.25MB for 32 statistics, making the approach practical even on constrained hardware.

Overall, these results demonstrate that our method is not only statistically principled, but also computationally efficient and offering fast inference, strong scalability with GPU workers, and memory usage that is well within practical bounds.

D.2 Robustness to Image Corruption

We evaluate robustness to two common real-world image degradations: JPEG compression and Gaussian blur. These corruptions are applied at test time without modifying the original reference distribution, computed ECDFs, or selected statistics $\mathcal{I}$. All experiments are conducted on the full dataset setup described in Section C, using the entire test set of both real and fake samples to ensure comprehensive evaluation.

Importantly, metrics are aggregated over all samples, rather than averaged across splits, so that results directly reflect the overall distributional behavior.

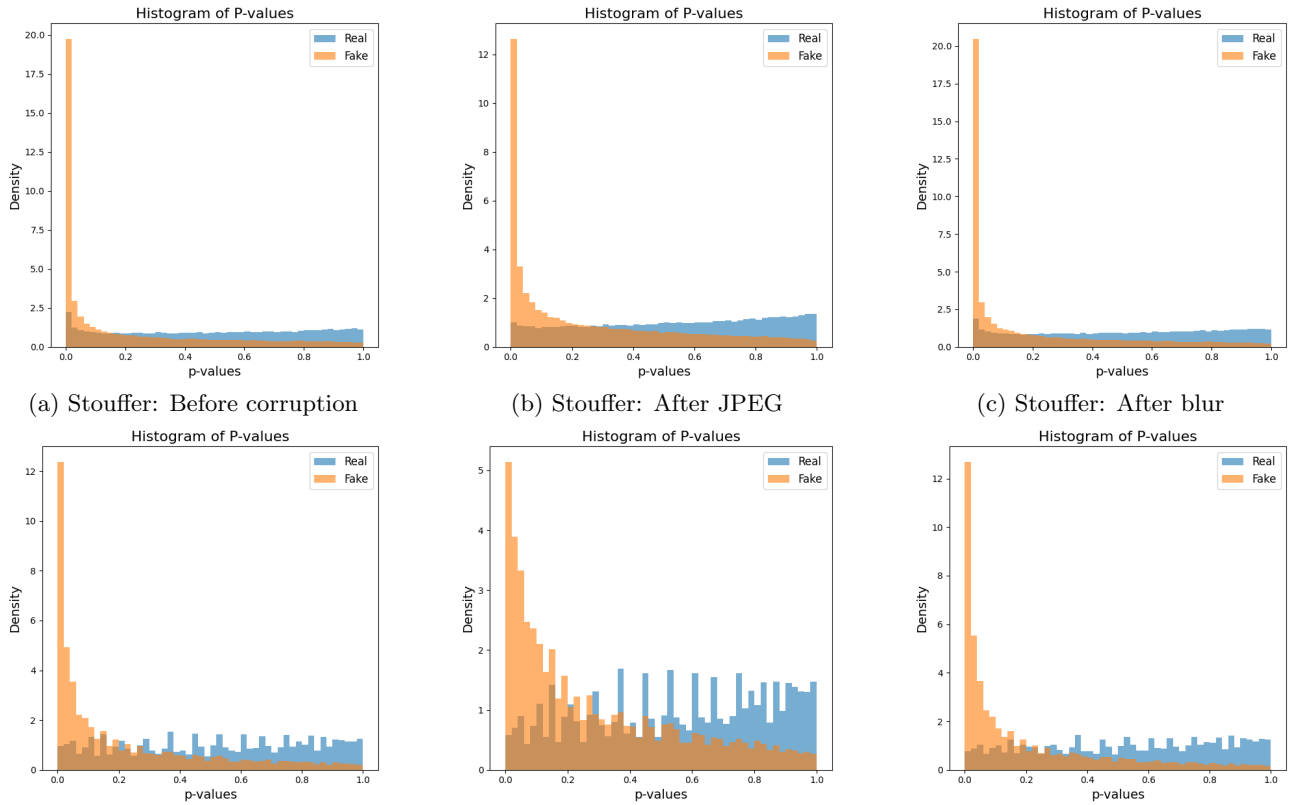
We evaluate robustness to two common degradations:

- **JPEG Compression** (Quality Factor = 0.75)
- **Standard Gaussian Blur** (3x3 Kernel)

Table 4: Performance under image corruptions for both ensemble strategies.

Metric	Stouffer			MinP		
	Without	**JPEG**	**Blur**	**Without**	**JPEG**	**Blur**
AUC	0.775	0.736	0.798	0.776	0.735	0.794
AP	0.798	0.750	0.818	0.786	0.734	0.807

Across both ensemble strategies, JPEG compression leads to a moderate reduction in performance (roughly 5-6% drop in AUC and AP), but does not induce a significant distributional shift in the p-values of real samples, as the null hypothesis remains valid. Gaussian blur introduces minor changes and even improves performance slightly, suggesting that the method is robust to mild smoothing effects regardless of the ensemble variant.



(a) Stouffer: Before corruption (b) Stouffer: After JPEG (c) Stouffer: After blur

(d) Min-p: Before corruption (e) Min-p: After JPEG (f) Min-p: After blur

Figure 12: p-value distributions for real (blue) and fake (orange) samples under the full dataset setup. **Top row (Stouffer)**: Before corruption (left), after JPEG compression (middle), and after Gaussian blur (right). **Bottom row (Min-p)**: Before corruption (left), after JPEG compression (middle), and after Gaussian blur (right). Across both ensemble variants, JPEG introduces only a mild shift reducing separability, while Gaussian blur leaves distributions well aligned with the reference, consistent with the observed negligible or positive performance changes.

Overall, these results show that our method remains statistically sound under realistic corruptions: JPEG compression introduces only moderate degradation without violating the null hypothesis, while mild Gaussian blur leaves performance unchanged or slightly improved. The robustness trends are consistent across both Stouffer and Min-p ensembles.

D.3 Preliminary Adversarial Perturbation Analysis

We conduct a focused preliminary experiment to examine the behaviour of RealStats under simple gradient-based adversarial perturbations. This analysis is not intended as a comprehensive adversarial robustness evaluation, but rather as an initial diagnostic aligned with the statistical structure of the framework.

We consider an FGSM-style iterative perturbation targeting a single perturbation-stability statistic (DINO-based RIGID configuration). For each selected fake image, we estimate the real-image reference distribution of the statistic and apply gradient steps designed to move the statistic toward the empirical mean of the real distribution. The perturbation is applied directly in pixel space, while all ECDFs, independence selection, and aggregation procedures remain fixed.

Across multiple samples and perturbation strengths, we did not observe cases in which the unified aggregated

p-value increased toward the real range. Instead, perturbed samples consistently shifted toward lower p-values, resulting in stronger confidence of being classified as fake. This behaviour was consistent across all tested configurations.

We hypothesise that this effect arises from two aspects of the framework: (i) the use of two-sided ECDF-based p-values calibrated on real-image distributions, and (ii) aggregation across multiple statistics that are empirically validated to satisfy independence under the null. A full adversarial robustness analysis is left for future work.

D.4 Interpretability Qualitative Comparison

Figure 13: **Qualitative comparison of interpretability across methods.** Each row shows scores assigned by a different method to the same set of fake images. RealStats produces p-values that vary consistently with deviation from the reference distribution, while the baselines often cluster visually diverse samples into similar scores.

Figure 13 shows a qualitative interpretability comparison across methods on a shared set of synthetic reference images. RealStats outputs calibrated p-values that quantify image-wise deviation from the reference distribution. The scores progress with visual realism: lower p-values indicate a stronger appearance of being fake.

By contrast, ManifoldBias and Rigid produce scores that are harder to interpret. Images with severe visual defects can receive values close to realistic samples. Images from the same semantic domain, such as faces, also often collapse to similar scores despite substantial corruptions. This weak calibration limits interpretability because the scores do not reflect meaningful distances between samples.

Overall, these results highlight the benefit of using reference-calibrated statistics: while RealStats may not always perfectly separate fake from real, it provides an interpretable scale for reasoning about deviations from the reference distribution.

We further illustrate this point with the p-value distributions in Figure 14. Each row corresponds to a different p-value interval and includes ground-truth labels. Fake images tend to cluster at lower p-values, though some appear at higher values, indicating statistical similarity to real images. This visualization shows how p-values provide a calibrated and interpretable measure of sample quality.

Figure 14: **Qualitative examples of p-value trends on test data.** Images are grouped by p-value intervals (low to high). Fake images concentrate at low p-values, while a few overlap with real images at higher values, showing statistical similarity to the reference distribution.

D.5 Full Table of AUC and AP Values

Table 5 reports the complete per-generator AUC and AP values across all evaluated methods. These detailed results complement the averaged scores reported in the paper (Table 1), providing a comprehensive view of method performance for each generator. As explained, individual baselines show strong variability across generators while our method maintains competitive and more consistent results overall.

Table 5: AUC and AP scores for each generator across methods.

Generator	AUC				AP			
	Ours (Min-p)	RIGID	AEROBLADE	ManifoldBias	Ours (Min-p)	RIGID	AEROBLADE	ManifoldBias
ADM	0.690	0.575	**0.873**	0.611	0.691	0.590	**0.845**	0.564
BigGAN	0.800	0.794	0.733	**0.909**	0.773	0.798	0.747	**0.918**
CRN	**0.967**	0.941	0.845	0.941	**0.930**	0.924	0.836	0.915
CycleGAN	0.785	0.787	0.706	**0.973**	0.790	0.779	0.721	**0.977**
DALLE	0.552	0.588	**0.710**	0.709	0.552	0.562	**0.689**	0.692
deepfake	0.926	**0.974**	0.779	0.600	0.886	**0.970**	0.764	0.569
GauGAN	0.823	0.932	0.705	**0.983**	0.790	0.930	0.683	**0.983**
GLIDE_100_27	0.920	**0.978**	0.962	0.911	0.889	**0.978**	0.939	0.923
GLIDE_50_27	0.929	**0.982**	0.964	0.920	0.905	**0.982**	**0.981**	0.920
IMLE	**0.963**	0.953	0.807	0.950	**0.942**	0.940	0.829	0.933
Midjourney	0.661	0.563	**0.692**	0.584	**0.661**	0.531	0.705	0.564
ProGAN	0.851	0.926	0.878	**0.957**	0.845	0.923	0.902	**0.964**
SAN	0.572	0.655	0.611	**0.778**	0.592	0.668	0.625	**0.780**
SDv1.4	0.745	0.415	0.421	**0.740**	**0.720**	0.423	0.444	0.646
SDv1.5	0.764	0.394	0.403	**0.763**	**0.750**	0.414	0.391	0.674
SDv2	0.631	**0.820**	0.450	0.587	0.623	**0.820**	0.463	0.635
SDXL	0.644	**0.853**	0.612	0.534	0.605	**0.853**	0.597	0.565
StarGAN	0.876	**0.887**	0.577	0.268	0.847	0.845	0.561	**0.367**
StyleGAN2	0.576	**0.762**	0.606	0.645	0.545	**0.747**	0.584	0.685
VQDM	0.774	0.791	0.701	**0.852**	0.774	**0.809**	0.718	0.848
whichfaceisreal	0.809	**0.946**	0.823	0.744	0.769	**0.925**	0.846	0.722
Wukong	0.751	0.403	0.482	**0.751**	**0.751**	0.420	0.468	0.720
Average	0.773	0.769	0.697	0.760	0.756	0.765	0.697	0.753
Std	0.126	0.194	0.161	0.178	0.119	0.189	0.163	0.169

D.6 Adaptability Scores with ManifoldBias

The results in Table 6 demonstrate the impact of incorporating adaptability into our ensemble methods. By allowing the aggregation strategy to adjust dynamically to different generators, our approach achieves consistently stronger performance, yielding notable improvements in both AUC and AP compared to our base model. This highlights the importance of adaptability as a key factor for generalization across diverse generative models.

Table 6: Per-generator AUC and AP scores of ensemble methods when integrated with ManifoldBias, with improvements (in %). Bold indicates positive improvement.

Generator	AUC		AP	
	Min-p	Stouffer	Min-p	Stouffer
ADM	0.644 (-6.67%)	0.610	0.641 (-7.24%)	0.594
BigGAN	**0.869 (+8.62%)**	0.865	**0.869 (+12.42%)**	0.862
CRN	**0.974 (+0.72%)**	0.990	**0.956 (+2.80%)**	0.986
CycleGAN	**0.905 (+15.29%)**	0.913	**0.914 (+15.70%)**	0.917
DALLE	**0.584 (+5.80%)**	0.558	**0.603 (+9.24%)**	0.570
DeepFake	**0.934 (+0.86%)**	0.917	**0.903 (+1.92%)**	0.886
GauGAN	**0.949 (+15.31%)**	0.936	**0.957 (+21.14%)**	0.939
Glide_100_27	**0.941 (+2.28%)**	0.951	**0.940 (+5.74%)**	0.957
Glide_50_27	**0.948 (+2.05%)**	0.962	**0.945 (+4.42%)**	0.966
IMLE	**0.973 (+1.04%)**	0.993	**0.953 (+1.17%)**	0.991
Midjourney	0.518 (-21.63%)	0.518	0.526 (-20.42%)	0.523
ProGAN	**0.942 (+10.69%)**	0.936	**0.951 (+12.54%)**	0.946
SAN	**0.670 (+17.13%)**	0.668	**0.688 (+16.22%)**	0.663
SDXL	**0.646 (+0.31%)**	0.645	**0.608 (+0.50%)**	0.622
SDv2	**0.722 (+14.42%)**	0.708	**0.704 (+13.00%)**	0.682
SDv1.4	0.689 (-7.52%)	0.679	0.679 (-5.69%)	0.660
SDv1.5	0.707 (-7.46%)	0.710	0.705 (-6.00%)	0.706
StarGAN	**0.887 (+1.26%)**	0.864	**0.871 (+2.83%)**	0.821
StyleGAN2	**0.609 (+5.73%)**	0.615	**0.604 (+10.83%)**	0.647
VQDM	**0.793 (+2.45%)**	0.781	**0.784 (+1.29%)**	0.766
WhichFaceIsReal	**0.820 (+1.36%)**	0.765	**0.802 (+4.29%)**	0.743
WuKong	0.707 (-5.86%)	0.709	0.702 (-6.52%)	0.706
Average	0.792	0.786	0.787	0.780
Std	0.143	0.151	0.141	0.151

E Datasets and Code

We release both the **data splits** used in our experiments and the full **codebase**. The package includes our implementations of **RIGID statistics** alongside the other statistical measures evaluated in this work, as well as scripts for inference and the complete pipeline. We provide the exact **random seeds** used in all benchmarks and evaluations: {0, 8, 12, 18, 22, 28, 30, 32, 36, 38}.

All resources (code and datasets) are available in our GitHub repository: `https://github.com/shaham-lab/RealStats`.