USAD: Uncertainty-aware Statistical Adversarial Detection

Zhijian Zhou* **Xunye Tian*** **Jiacheng Zhang*** **Zesheng Ye†**
Yiyi Guo **Donghao Zhang** **Liuhua Peng** **Feng Liu†**
The University of Melbourne
fengliu.ml@gmail.com zesheng.ye@unimelb.edu.au

Abstract

Statistical adversarial detection (SAD) treats adversarial detection as a two-sample test. Given a reference set of clean examples (CEs) and a batch of queries, potentially containing an unknown mixture of CEs and adversarial examples (AEs), SAD decides whether the query distribution drifts away from the CE distribution while controlling the false-alarm rate. Existing SAD-based methods mainly use *maximum mean discrepancy* (MMD) to measure the distributional discrepancy. However, MMD's distributional properties limit its ability to capture characteristic uncertainty patterns of AEs that are crucial for detection: AEs typically exhibit abnormal feature spread (i.e., global uncertainty) and instability under perturbations (i.e., local uncertainty). To close the gap, we propose *Uncertainty-aware Statistical Adversarial Detection* (USAD), which explicitly captures these uncertainty patterns with two new statistics: (1) **V**ariance **D**iscrepancy (VD), which measures the difference in feature spread between AEs and CEs to capture global uncertainty differences, and (2) **P**erturbation-based **C**ovariance **D**iscrepancy (PCD), which compares feature covariance under Gaussian perturbations to capture local uncertainty differences. By aggregating VD and PCD, USAD achieves superior detection performances over baseline methods against various adversarial attacks, highlighting the importance of considering characteristic behaviors of AEs for effective SAD. Our code is available at: https://github.com/yeager20001118/USAD.

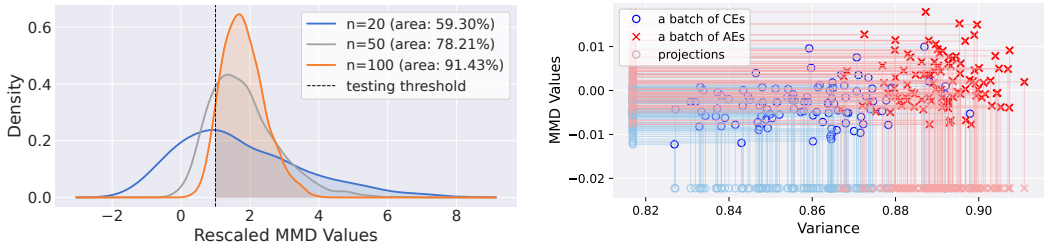
1 Introduction

Defending against *adversarial examples* (AEs) remains a long-standing challenge in deep learning [1–12]. The most lightweight defense against adversarial attacks is adversarial detection, which identifies AEs in input data before they reach target systems [7, 13–15]. Existing methods often consider adversarial detection as a binary classification task [13–15]. They typically train a binary detector tailored to specific classifiers or attack types, which limits the effectiveness against unknown attacks [16, 17]. Although a study leverages diffusion models [18–20] to remove such dependency [15], it is vulnerable to diffusion-based adaptive attacks [21, 22]. These limitations motivate us to pursue a model- and attack-agnostic detection framework.

Statistical adversarial detection (SAD) offers a complementary direction by formulating adversarial detection as a two-sample hypothesis test: given a reference pool of CEs and a set of queries, SAD asks whether the query *distribution* has drifted away from the clean one (see Section 2.1 for formal setup). Concretely, the defender needs to maintain a reference set X of CEs and, for a window Y of incoming queries, compute a test statistic $\mathcal{T}(X, Y)$ that measures their distributional discrepancy [7]. A permutation test then determines a threshold t_α such that $\Pr(\mathcal{T}(X, Y) > t_\alpha \mid Y \text{ is clean}) \leq \alpha$, providing guarantees on statistically-controlled Type-I error, which is often desired in high-stakes

*Equal contribution. †Corresponding authors.

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



(a) Distribution of MMD values for AE batch sizes.

(b) MMD–variance relationship for CE and AE batches.

Figure 1: Statistical characteristics of MMD-based SAD using semantic features [7]. **(a):** Kernel density estimates of MMD values for three AE batch sizes ($n = \{20, 50, 100\}$). For visualization, the MMD values of each batch size are rescaled by its own permutation threshold, so that the three batch sizes share a common threshold of 1, denoted by the vertical dashed line. The shaded regions to the right of the threshold indicate detection power, quantified by the area percentages shown in the legend (91.43% for $n = 100$, 78.21% for $n = 50$, and 59.30% for $n = 20$). It clearly illustrates that MMD-based SAD exhibits diminishing detection power as the batch size decreases. **(b):** Sensitivity of MMD to uncertainty differences between AEs (red crosses) and CEs (blue circles), with projections onto the x- and y-axes for comparison, where the variance on the x-axis refers to the empirical RKHS variance of semantic features computed with the kernel (i.e., $\widehat{V}(\cdot;\ell)$ in Section 3.1). Despite the markedly higher variance of AEs and their clear separation from CEs, the MMD values exhibit substantial overlap along the y-axis, indicating the insensitivity of MMD-based SAD to variance differences.

contexts. Operationally, this corresponds to maintaining a queue of recent queries and running tests whenever a window has accumulated, or in a sliding-window over the stream; each such window may contain an *unknown mixture* of CEs and AEs [23]. This way SAD acts as a statistical monitoring layer: once a query set is flagged as suspicious, the defending system can trigger other defenses or route the traffic for human inspection, all while keeping the rate of spurious alarms under control [7]. See Appendix A for further discussions on the practicality of SAD and Appendix B for related works.

Current SAD methods mainly use *maximum mean discrepancy* (MMD) [24] as the test statistic $\mathcal{T}(X, Y)$ [7, 23]. They have shown strong performance against *unknown* and even *adaptive* attacks, while inheriting rigorous false-alarm control. However, existing MMD-based SAD methods are *sample-inefficient*: they often require query windows with a sufficiently large effective sample size (and a non-trivial fraction of AEs therein) for the *test statistic* to reliably identify the distributional discrepancy with high test power. As Figure 1a shows, we confirm that the test power of MMD-based SAD degrades rapidly as the window size of queries shrinks. This limits practical deployment, as waiting for large query sets is often unrealistic and attack volume is inherently uncontrollable.

To understand this limitation, we examine which distributional properties the *test statistic* captures. Since MMD captures mean-embedding shifts (see Eq. (3)), we hypothesize that it is insensitive to geometric properties, such as feature spread [13] and local instability [15, 25, 26], that distinguish AEs from CEs. These geometric cues are closely related to *uncertainty*: AEs often reside near decision boundaries and exhibit higher feature variance and predictive uncertainty than CEs [27]. To validate this hypothesis, we test whether MMD captures AE–CE uncertainty differences, using *variance* as the uncertainty measure. As shown in Figure 1b, CE and AE MMD values largely overlap despite substantial variance differences, indicating that MMD is insensitive to this uncertainty measurement. This finding reveals a fundamental mismatch: *MMD is not tailored for SAD, and therefore, MMD-based SAD methods are insensitive to distinguish AEs from CEs when AEs display their characteristic uncertainty patterns rather than mean embedding shifts.* This motivates new *test statistics* directly sensitive to AE uncertainty patterns.

To close the gap, this paper proposes ***U**ncertainty-aware **S**tatistical **A**dversarial **D**etection* (USAD), which explicitly accounts for two well-established uncertainty-induced patterns, namely (1) AEs are displaced off the clean data manifold, causing abnormal spread in their semantic feature distributions [13, 27] (i.e., *global uncertainty*), and (2) norm-bounded AEs are generated by small perturbations that cross local decision boundaries [28, 29] and often exploit brittle non-robust features [30], leading to atypical local response patterns under added perturbations [25, 26] (i.e., *local uncertainty*), with two new *test statistics*: (1) ***V**ariance **D**iscrepancy* (VD), which measures the difference in feature spread between AEs and CEs (see Section 3.1), and (2) ***P**erturbation-based **C**ovariance **D**iscrepancy* (PCD), which compares feature covariance under Gaussian perturbations to capture local stability

differences (see Section 3.2). VD is designed to capture *global uncertainty differences* arising from manifold displacement, whereas PCD seeks to capture *local uncertainty differences* induced by perturbation sensitivity. To capture both global and local uncertainty differences, we follow Zhou et al. [31] and aggregate VD and PCD while accounting for their mutual dependencies (see Section 4). The false alarm rate of USAD is theoretically controlled (see Section 5).

Section 6 evaluates the effectiveness of USAD on benchmark image datasets such as CIFAR-10 [32] and ImageNet-1K [33]. Specifically, USAD *consistently* outperforms baseline SAD methods by a notable margin against various unseen adversarial attacks under different threat models, including different norms (e.g., ℓ_∞ and ℓ_2), perturbation budgets (e.g., ϵ from $8/255$ to $1/255$), batch sizes (e.g., from 80 to 10), and ratio of AEs in the query batch (e.g., from 100% to 20%). Notably, even with a query batch size as small as $|Y| = 10$, our method achieves test power essentially equal to 1 at $\epsilon = 4/255$, outperforming baselines by at least 24.8%. This marks a substantial step toward overcoming the sample-size limitations that constrain detection power in hypothesis-testing-based methods. More importantly, our method remains robust against a well-designed adaptive attack.

Our contributions are: (1) We identify that MMD-based SADs fail to capture the distributional uncertainty differences crucial for distinguishing AEs from CEs; (2) To address this, we propose ***U**ncertainty-aware **S**tatistical **A**dversarial **D**etection* (USAD) with new test statistics, i.e., ***V**ariance **D**iscrepancy* (VD) and ***P**erturbation-based **C**ovariance **D**iscrepancy* (PCD), to explicitly capture *global* and *local* uncertainty differences between AEs and CEs; (3) Theoretically, we prove that USAD guarantees valid false-alarm rate control. Empirically, we show that USAD consistently outperforms strong baseline methods across diverse unseen, transfer, and adaptive attacks, under small query window sizes and when AEs are sparsely *mixed* with CEs within each query batch.

2 Statistical Adversarial Detection

We begin by formalizing *statistical adversarial detection* (SAD) as a hypothesis testing task for distinguishing *adversarial examples* (AEs) from *clean examples* (CEs) by measuring their distributional discrepancy (Section 2.1). We then identify limitations of existing SAD practices and motivate the need for new measures (Section 2.2).

2.1 Detecting AEs from a Statistical Perspective

Adversarial Attacks. The algorithms for generating AEs are commonly referred to as adversarial attacks [1, 2]. Given a well-trained classifier $f : \mathcal{X} \to \mathcal{C}$ on a dataset $D = \{(\boldsymbol{x}_i, c_i)\}_{i=1}^N$ with $\boldsymbol{x}_i$ being a sample from the input space $\mathcal{X} \subseteq \mathbb{R}^d$ and c_i being its ground-truth label defined in a label space $\mathcal{C}$, an AE $\boldsymbol{y}_i$ regarding $\boldsymbol{x}_i$ with perturbation $\boldsymbol{\zeta}_i$ can be generated by solving the optimization:

$$\boldsymbol{y}_i = \boldsymbol{x}_i + \underset{\boldsymbol{\zeta}_i}{\operatorname{argmax}}\, \mathcal{L}(f(\boldsymbol{x}_i + \boldsymbol{\zeta}_i), c_i), \text{ s.t. } ||\boldsymbol{\zeta}_i||_p \leq \epsilon, \tag{1}$$

where ϵ is the perturbation budget, $||\cdot||_p$ is the ℓ_p-norm (e.g., ℓ_∞ or ℓ_2) and $\mathcal{L}$ is an objective function.

Eq. (1) can be solved by many methods to generate AEs. For example, the *fast gradient sign method* (FGSM) is a one-step attack that perturbs clean data in the direction of the loss gradient [2]. Building on FGSM, the *basic iterative method* (BIM) applies iterative perturbations along the gradient direction, updating the input at each step to generate stronger AEs [34]. *Projected gradient descent* (PGD) further enhances BIM by introducing random initialization prior to iterative gradient-based updates [3]. Beyond non-targeted attacks, the *Carlini & Wagner* (C&W) attack generates targeted AEs by optimizing a carefully designed objective function that explicitly encourages misclassification toward a chosen label [4]. *AutoAttack* (AA) [6] aggregates multiple attacks, which include three non-targeted white-box attacks [6, 35] and one targeted black-box attack [36], which makes AA widely adopted for evaluating adversarial robustness.

SAD Problem Setup. Given a reference set of CEs $X = \{\boldsymbol{x}_i\}_{i=1}^n \sim \mathbb{P}$ and a query set $Y = \{\boldsymbol{y}_j\}_{j=1}^m$ drawn from some distribution $\mathbb{Q}$, we aim to decide whether the queries Y follow the same distribution as CEs. Under the null hypothesis $\boldsymbol{H}_0$, *all* queries are clean and $\mathbb{P} = \mathbb{Q}$. Under the alternative $\boldsymbol{H}_1$, at least a *non-zero fraction* of queries in Y are AEs generated by *unseen* adversarial attacks against f. SAD therefore tests

$$\boldsymbol{H}_0 : \mathbb{P} = \mathbb{Q} \quad \text{and} \quad \boldsymbol{H}_1 : \mathbb{P} \neq \mathbb{Q}, \tag{2}$$

and reject $\boldsymbol{H}_0$ at significance level α when the chosen *test statistic* exceeds a threshold, providing Type-I control. Here, Type-I error means falsely declaring a clean query set as adversarial. Note that Eq. (2) tests distributional equality rather than adversariality: $\boldsymbol{H}_0$ is rejected whenever $\mathbb{Q}$ differs from $\mathbb{P}$ in the distributional properties captured by the chosen test statistic, so benign shifts that alter the same properties may also be detected (see Appendix A and Appendix G.11).

SAD as Hypothesis Test. The SAD starts from designing a *test statistic* $\mathcal{T}(X, Y)$ to measure the discrepancy between $\mathbb{P}$ and $\mathbb{Q}$ (by observing X and Y). At a chosen level α, $\mathcal{T}(X, Y)$ is then compared against a *testing threshold* t_α defined as the $(1-\alpha)$-quantile of the *null distribution*, which characterizes the sampling behavior of $\mathcal{T}(X, Y)$ under $\boldsymbol{H}_0$, satisfying $\Pr(\mathcal{T}(X, Y) > t_\alpha | \boldsymbol{H}_0) = \alpha$. It means that $\boldsymbol{H}_0$ will be rejected (i.e., declare Y adversarial) if $\mathcal{T}(X, Y) > t_\alpha$.

As the null distribution is rarely known analytically, we typically estimate t_α using a *permutation test*, which builds an empirical null distribution by randomly re-partitioning the pooled examples $X \cup Y$ into new sets $(X_{(r)}, Y_{(r)})$, and computing $\mathcal{T}(X_{(r)}, Y_{(r)})$ for each permutation $r = 1, \ldots, R$. The empirical threshold t_α is then set to the $(1-\alpha)$-quantile of these permuted statistic values, ensuring the probability of falsely rejecting $\boldsymbol{H}_0$ is controlled at the chosen level α (see Appendix C for details).

MMD as Test Statistic. Critically, the effectiveness of SAD relies on whether the chosen test statistic $\mathcal{T}(X, Y)$ can capture the distributional discrepancy if X indeed comes from a distribution different from Y (i.e., $\mathbb{P} \neq \mathbb{Q}$). Existing SAD practices [7, 23, 37, 38] have predominantly employed the MMD [24], which measures the distance between $\mathbb{P}$ and $\mathbb{Q}$ by mapping their elements to a *reproducing kernel Hilbert space* (RKHS) and comparing the *kernel mean embeddings* [39] therein. Formally, for a characteristic kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with RKHS $\mathcal{H}_\kappa$ and its corresponding map $\kappa(\cdot, \boldsymbol{x}) \in \mathcal{H}_\kappa$, we have the kernel mean embeddings of $\mathbb{P}$ and $\mathbb{Q}$ given by $\boldsymbol{\mu}_\mathbb{P} = E_{\boldsymbol{x}\sim\mathbb{P}}[\kappa(\cdot, \boldsymbol{x})]$ and $\boldsymbol{\mu}_\mathbb{Q} = E_{\boldsymbol{y}\sim\mathbb{Q}}[\kappa(\cdot, \boldsymbol{y})]$, respectively. The MMD-based SAD [7, 23] then instantiates Eqn.(2) on testing the equality of such embeddings, namely $\boldsymbol{H}_0^{\mathrm{M}} : \boldsymbol{\mu}_\mathbb{P} = \boldsymbol{\mu}_\mathbb{Q}$ versus $\boldsymbol{H}_1^{\mathrm{M}} : \boldsymbol{\mu}_\mathbb{P} \neq \boldsymbol{\mu}_\mathbb{Q}$ via squared distance between $\boldsymbol{\mu}_\mathbb{P}$ and $\boldsymbol{\mu}_\mathbb{Q}$:

$$\begin{aligned} \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}, \kappa) &\triangleq \|\boldsymbol{\mu}_\mathbb{P} - \boldsymbol{\mu}_\mathbb{Q}\|^2_{\mathcal{H}_\kappa} \\ &= E[\kappa(\boldsymbol{x}, \boldsymbol{x}') + \kappa(\boldsymbol{y}, \boldsymbol{y}') - 2\kappa(\boldsymbol{x}, \boldsymbol{y})], \end{aligned} \tag{3}$$

where $\boldsymbol{x}, \boldsymbol{x}' \overset{\text{i.i.d.}}{\sim} \mathbb{P}$, $\boldsymbol{y}, \boldsymbol{y}' \overset{\text{i.i.d.}}{\sim} \mathbb{Q}$, and $\|\cdot\|^2_{\mathcal{H}_\kappa} = \langle \cdot, \cdot \rangle_{\mathcal{H}_\kappa}$ defines the inner product in RKHS $\mathcal{H}_\kappa$. In practice, we empirically estimate $\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}, \kappa)$ with samples X and Y, using unbiased U-statistics [24] (Appendix D). Recently, SAMMD [7] showed that applying MMD to semantic features from the target classifier f, combined with proper *testing threshold*, can elevate the detection power of MMD-based SAD.

2.2 SAD Needs Uncertainty-Aware Statistics

When MMD-based SAD Fails. Despite maintaining Type-I error control, MMD-based SAD methods exhibit diminishing power as query size $|Y|$ decreases (Fig. 1a). Such small-sample weakness is problematic for SAD: practical defense requires early detection when only a few query examples have been observed. It is often unrealistic to wait for large query sets to accumulate [23]. Moreover, this degradation persists even with carefully tuned kernel embeddings and thresholds [7], implying that the bottleneck is not merely kernel choice or threshold but in which distributional properties the *test statistic* itself targets.

MMD (and its variant SAMMD), by construction, is most sensitive to mean shifts $\|\boldsymbol{\mu}_\mathbb{P} - \boldsymbol{\mu}_\mathbb{Q}\|^2_{\mathcal{H}_\kappa}$, capturing where distributions are centered (i.e., first-order moments) in the RKHS $\mathcal{H}_\kappa$, yet providing less information about their internal structures, such as spread and local geometry [40, 41]. These second-order properties, however, are precisely where adversarial attacks leave strong artifacts that can be valuable in distinguishing AEs.

Adversarial Artifacts as Distributional Uncertainty. AEs result from solving (1), where adversarial perturbations are crafted to enforce AEs to cross target classifier f's decision boundaries. This exploits two geometric properties, leading to two well-established characteristic behaviors of AEs:

Factor 1 (Global Uncertainty from Manifold Displacement)**.** *AEs are systematically pushed off clean data manifold [27]. This displacement alters the spread of AEs' semantic feature distribution [13].*

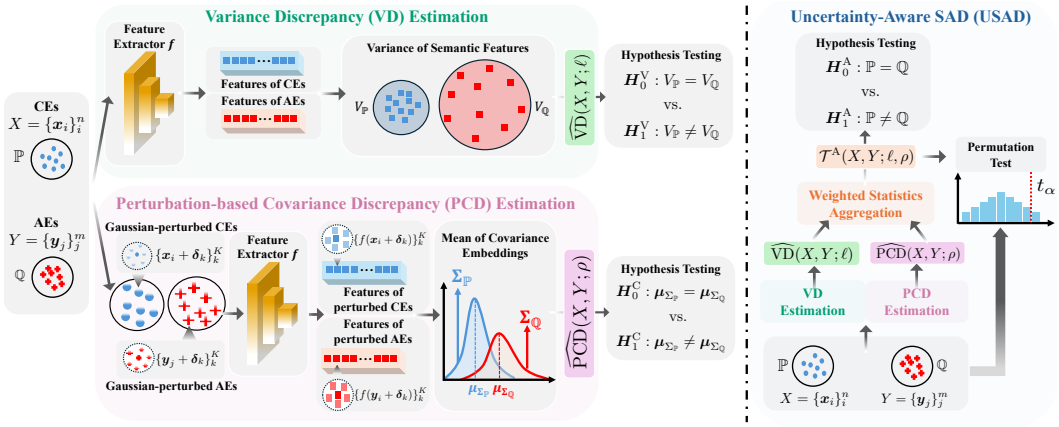


Figure 2: ***U***ncertainty-aware ***S****tatistical* ***A****dversarial* ***D****etection* (USAD). The semantic features of CEs $X \sim \mathbb{P}$ and queries $Y \sim \mathbb{Q}$ (suspected AEs) are extracted by using the penultimate layer of classifier f. From these features, USAD estimates (i) *variance discrepancy* (VD) measuring shifts in feature-spread between X and Y and (ii) *perturbation-based covariance discrepancy* (PCD) comparing their covariance mean-embeddings under Gaussian perturbations. These VD and PCD statistics are adaptively aggregated via *correlation-aware weighting* into a single *test statistic*, and a *permutation test* determines, at level α, whether Y is adversarial.

Consequently, AEs may become more or less dispersed than CEs, without necessarily shifting the mean embedding, reflecting a shift in the global uncertainty over the distribution.

Factor 2 (Local Uncertainty from Perturbation Sensitivity)**.** *AEs tend to reside near classifier f's decision boundaries and exploit non-robust features [28, 30], making them sensitive to local isotropic perturbations. Motivated by prior random-perturbation-based detection studies [15, 25, 26] and the connection between adversarial robustness and stability under additive Gaussian noise [42], we examine differences in the feature-response patterns of AEs and CEs under Gaussian perturbations through perturbation-induced covariance, where AEs are expected to exhibit less stable feature responses than CEs. This instability marks elevated local uncertainty at individual examples.*

Appendix G.9 summarizes the arguments from prior work for these two factors and analyzes them directly on synthetic data and real AEs.

Bridging the Gap. These findings reveal a fundamental mismatch: existing MMD-based SAD practices compare first-order distributional differences (i.e., mean shifts in the RKHS). Thus, they may struggle to distinguish $\mathbb{P}$ (i.e., clean distribution) from $\mathbb{Q}$ (i.e., adversarial distribution) when adversarial artifacts appear as second-order structural changes (i.e., uncertainty pattern shifts) without substantially moving mean embeddings of AEs. This calls for new discrepancy measures that are directly sensitive to these changes in distributional uncertainty caused by adversarial attacks. We introduce two such *test statistics* to close the gap.

3 Uncertainty-Aware Test Statistics

We now develop *test statistics* that target how adversarial perturbations affect uncertainty in the classifier f's semantic feature space identified in Section 2. For clarity, we slightly abuse notation and denote $f(\cdot) \in \mathcal{F} \subseteq \mathbb{R}^q$ as semantic features of an input $\cdot \in \mathcal{X}$ extracted by f, where $\mathcal{F}$ is a q-dimensional semantic feature space (often from the penultimate layer of f). Section 3.1 introduces *variance discrepancy* (VD) that quantifies *global uncertainty* by comparing within-distribution variances, detecting altered concentration in query distribution $\mathbb{Q}$ relative to clean distribution $\mathbb{P}$. Section 3.2 introduces *perturbation-based covariance discrepancy* (PCD) that measures whether query examples (suspected to be AEs) exhibit higher *local uncertainty* to small, random isotropic noise compared to CEs, encoded by Gaussian-noise-induced covariance at each example.

3.1 Variance Discrepancy

Global Uncertainty as Variance. We quantify the variance of semantic features within-distribution to detect whether Y's features spread differently from those of X in the RKHS. Let $\ell : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ be a kernel with RKHS $\mathcal{H}_\ell$ (which also defines map $\ell(\cdot, f(\boldsymbol{x})) \in \mathcal{H}_\ell$); in our experiments, ℓ is a

Gaussian kernel on semantic features (see Appendix F). For distributions $\mathbb{P}$ and $\mathbb{Q}$, the semantic feature variances are

$$V_{\mathbb{P}} \triangleq E_{\boldsymbol{x}\sim\mathbb{P}}\left[\ell(f(\boldsymbol{x}), f(\boldsymbol{x}))\right] - E_{\boldsymbol{x},\boldsymbol{x}'\sim\mathbb{P}^2}[\ell(f(\boldsymbol{x}), f(\boldsymbol{x}'))],$$
$$V_{\mathbb{Q}} \triangleq E_{\boldsymbol{y}\sim\mathbb{Q}}\left[\ell(f(\boldsymbol{y}), f(\boldsymbol{y}))\right] - E_{\boldsymbol{y},\boldsymbol{y}'\sim\mathbb{Q}^2}[\ell(f(\boldsymbol{y}), f(\boldsymbol{y}'))].$$

The first term of $V_{\mathbb{P}}$ establishes the self-similarity of examples within $\mathbb{P}$, while the second captures their pairwise similarity. Thus, small $V_{\mathbb{P}}$ indicates tight clustering, while large $V_{\mathbb{P}}$ reflects greater dispersion (likewise for $V_{\mathbb{Q}}$). VD is related to the *maximum variance discrepancy* (MVD) [43]: VD compares the traces of the RKHS covariance operators of $\mathbb{P}$ and $\mathbb{Q}$, whereas MVD compares the full operators. VD thus directly targets changes in the total feature spread, which correspond to the global uncertainty in Factor 1, and requires estimating only a scalar quantity from limited samples; richer local covariance changes are instead captured by PCD (Section 3.2), making VD and PCD complementary (see Appendix G.12 for an empirical comparison with MVD).

Test. We formalize the comparison of variances $V_{\mathbb{P}}$ and $V_{\mathbb{Q}}$ by instantiating Eq.(2) as the following hypothesis test:

$$\boldsymbol{H}_0^{\mathrm{V}} : V_{\mathbb{P}} = V_{\mathbb{Q}} \quad \text{vs.} \quad \boldsymbol{H}_1^{\mathrm{V}} : V_{\mathbb{P}} \neq V_{\mathbb{Q}}\ , \tag{4}$$

with *test statistic* defined as the *variance discrepancy* $\mathrm{VD}(\mathbb{P}, \mathbb{Q}; \ell) = \left(V_{\mathbb{P}} - V_{\mathbb{Q}}\right)^2$ between $\mathbb{P}$ and $\mathbb{Q}$.

Estimator. Since $\mathbb{P}$ and $\mathbb{Q}$ are analytically unknown, we empirically estimate $\mathrm{VD}(\mathbb{P}, \mathbb{Q}; \ell)$ using X and Y, such that

$$\widehat{\mathrm{VD}}(X, Y; \ell) = \left(\widehat{V}(X; \ell) - \widehat{V}(Y; \ell)\right)^2\ , \tag{5}$$

where for $Z \in \{X, Y\}$, we define

$$\widehat{V}(Z; \ell) = \sum_{\boldsymbol{z}\in Z} \frac{\ell(f(\boldsymbol{z}), f(\boldsymbol{z}))}{|Z|} - \sum_{\boldsymbol{z}_i\neq\boldsymbol{z}_j}^{Z} \frac{\ell(f(\boldsymbol{z}_i), f(\boldsymbol{z}_j))}{|Z|(|Z|-1)}\ .$$

If $\widehat{\mathrm{VD}}(X, Y; \ell)$ exceeds the *threshold* estimated via *permutation test*, we consider $\mathbb{P}$ and $\mathbb{Q}$ to be significantly different.

This way VD completes our response to Factor 1: it detects uncertainty shifts at the *global* level, comparing how tightly $\mathbb{P}$ and $\mathbb{Q}$ concentrate across their support in the RKHS. Next, we will examine *local* uncertainty shifts—how features fluctuate around individual examples under small, isotropic perturbations, which differ between CEs and AEs as noted in Factor 2. This motivates a perturbation-based covariance comparison that complements VD.

3.2 Perturbation-based Covariance Discrepancy

Recall that CEs often show consistent responses to isotropic perturbations [44], while AEs are constructed to be near the decision boundaries of f (to cause mis-classification) [28, 30, 45]. Accordingly, AEs are expected to exhibit less stable feature responses to even small perturbations, especially in directions that matter for f's decisions, indicating higher local sensitivity than CEs (Factor 2).

Local Uncertainty as Perturbation-Induced-Covariance. To probe local sensitivity, we specifically compare the distributions of per-example feature covariances under isotropic Gaussian noises. For each example $\boldsymbol{x} \in \mathcal{X}$, we draw K random perturbations $\{\boldsymbol{\delta}_k\}_{k=1}^K \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \sigma^2 I)$, extract features $\{f(\boldsymbol{x}+\boldsymbol{\delta}_k)\}_{k=1}^K$, and compute the covariance matrix of $\boldsymbol{x}$:

$$\Sigma_{\boldsymbol{x}} = \frac{1}{K-1}\sum_{k=1}^{K}(f(\boldsymbol{x}+\boldsymbol{\delta}_k) - \overline{f_{\boldsymbol{x}}})(f(\boldsymbol{x}+\boldsymbol{\delta}_k) - \overline{f_{\boldsymbol{x}}})^\top, \tag{6}$$

where $\overline{f_{\boldsymbol{x}}} \triangleq K^{-1}\sum_{k=1}^K f(\boldsymbol{x}+\boldsymbol{\delta}_k)$. The covariance matrix $\Sigma_{\boldsymbol{x}}$ encodes *local uncertainty* via its eigenstructure: large eigenvalues indicate directions where features vary substantially under Gaussian perturbation, small eigenvalues indicate stability [15, 46].

Test. To compare differences in *local uncertainty* (as distributional discrepancy) between $\mathbb{P}$ and $\mathbb{Q}$, we instantiate Eq.(2) regarding the perturbation–induced covariance embeddings by testing the equality of the corresponding kernel mean embeddings with null and alternative hypotheses as:

$$\boldsymbol{H}_0^{\mathrm{C}} : \boldsymbol{\mu}_{\Sigma_{\mathbb{P}}} = \boldsymbol{\mu}_{\Sigma_{\mathbb{Q}}} \quad \text{vs.} \quad \boldsymbol{H}_1^{\mathrm{C}} : \boldsymbol{\mu}_{\Sigma_{\mathbb{P}}} \neq \boldsymbol{\mu}_{\Sigma_{\mathbb{Q}}}\ , \tag{7}$$

where $\boldsymbol{\mu}_{\Sigma_{\mathbb{P}}} \triangleq E_{\boldsymbol{x}\sim\mathbb{P}}\left[\rho(\Sigma_{\boldsymbol{x}}, \cdot)\right]$ and $\boldsymbol{\mu}_{\Sigma_{\mathbb{Q}}} \triangleq E_{\boldsymbol{y}\sim\mathbb{Q}}\left[\rho(\Sigma_{\boldsymbol{y}}, \cdot)\right]$ are mean embeddings of the covariance distributions for $\mathbb{P}$ and $\mathbb{Q}$ in the RKHS $\mathcal{H}_\rho$[2]. We define *perturbation-based covariance discrepancy* (PCD) as $\text{PCD}(\mathbb{P}, \mathbb{Q}, \rho) \triangleq \|\boldsymbol{\mu}_{\Sigma_{\mathbb{P}}} - \boldsymbol{\mu}_{\Sigma_{\mathbb{Q}}}\|^2_{\mathcal{H}_\rho}$, which serves as *test statistic*. Equivalently, $\text{PCD}(\mathbb{P}, \mathbb{Q}, \rho)$ mirrors Eq.(3) but measures the difference between distributions of covariance matrices under kernel ρ. In our experiments, ρ is either the log-RBF kernel or the Gaussian kernel defined directly on covariance matrices (see Appendix F).

Estimator. In line with Eq.(3), we empirically estimate $\text{PCD}(\mathbb{P}, \mathbb{Q}, \rho)$ with finite examples X and Y, such that

$$\widehat{\text{PCD}}(X, Y; \rho) = \sum_{i\neq j}^{n} \frac{\rho(\Sigma_{\boldsymbol{x}_i}, \Sigma_{\boldsymbol{x}_j})}{n(n-1)} - \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{2\rho(\Sigma_{\boldsymbol{x}_i}, \Sigma_{\boldsymbol{y}_j})}{nm} + \sum_{i\neq j}^{m} \frac{\rho(\Sigma_{\boldsymbol{y}_i}, \Sigma_{\boldsymbol{y}_j})}{m(m-1)}. \tag{8}$$

This completes our response to Factor 2: PCD detects uncertainty shifts between $\mathbb{P}$ and $\mathbb{Q}$ at the *local* level by comparing how semantic features fluctuate under isotropic noise around individual examples. Together with VD's *global* spread comparison, we now possess two complementary lenses through which adversarial artifacts manifest as measurable distributional uncertainty signals.

4 Making SAD Aware of Local and Global Uncertainty, Adaptively

While VD and PCD quantify complementary uncertainty patterns in the semantic feature space, attacks vary in how strongly they activate each effect: some primarily displace AEs from the clean manifold (i.e., strong VD signal) [27]; others tend to exploit local gradients near boundaries (i.e., strong PCD signal) [30]. For SAD detection to be effective across diverse attacks, it should adapt to the uncertainty present in the data based on each statistic's informativeness for the query data rather than treat each statistic uniformly.

In light of this, we aggregate VD and PCD into *a unified test* that accounts for their joint correlation structure [31], yielding an adaptive detector sensitive to both global and local adversarial artifacts, termed ***U**ncertainty-aware **S**tatistical **A**dversarial **D**etection* (USAD).

USAD as Hypothesis Test. Still, USAD instantiates Eq.(2) by testing distributional equality:

$$\boldsymbol{H}_0^{\text{A}} : \mathbb{P} = \mathbb{Q} \quad \text{vs.} \quad \boldsymbol{H}_1^{\text{A}} : \mathbb{P} \neq \mathbb{Q},$$

and defining the corresponding aggregated *test statistic* as

$$\mathcal{T}^{\text{A}}(X, Y; \ell, \rho) = \mathcal{T}(X, Y; \ell, \rho)^\top \widehat{\Sigma}_{\text{A}}^{-1} \mathcal{T}(X, Y; \ell, \rho), \tag{9}$$

where $\mathcal{T}(X, Y; \ell, \rho) \triangleq \left(\widehat{\text{VD}}(X, Y; \ell), \widehat{\text{PCD}}(X, Y; \rho)\right)^\top$ and $\widehat{\Sigma}_{\text{A}}$ denotes the correlation between the statistics VD and PCD.

Weighted Statistics Aggregation. $\mathcal{T}^{\text{A}}(X, Y; \ell, \rho)$ combines both global $\widehat{\text{VD}}(X, Y; \ell)$ and local uncertainty shifts measure $\widehat{\text{PCD}}(X, Y; \rho)$, following *correlation-aware* statistics ensembling [31], with the key idea to weight $\widehat{\text{VD}}(X, Y; \ell)$ and $\widehat{\text{PCD}}(X, Y; \rho)$ according to their empirical dependence, so that redundant components are down-weighted while complementary ones are emphasized.

Statistics Correlation Estimation. In practice, we estimate their correlation structure under the null hypothesis $\boldsymbol{H}_0^{\text{A}}$ from CEs $X \sim \mathbb{P}$, which characterizes how $\mathcal{T}^{\text{A}}(X, Y; \ell, \rho)$ behaves under $\boldsymbol{H}_0^{\text{A}}$ [31, 47]. Specifically, we perform B bootstrap resamplings from X. In each iteration b, we draw two disjoint subsets X_b and Y_b from the clean pool (with $|X_b| = n$ and $|Y_b| = m$), and compute the statistics vector $\mathcal{T}_b(X, Y; \ell, \rho)$. From $\{\mathcal{T}_b\}_{b=1}^B$, we estimate the correlation

$$\widehat{\Sigma}_{\text{A}} = \frac{1}{B-1}\sum_{b=1}^{B}\left(\mathcal{T}_b - \widehat{\mathcal{T}_{\boldsymbol{\mu}}}\right)\left(\mathcal{T}_b - \widehat{\mathcal{T}_{\boldsymbol{\mu}}}\right)^\top, \tag{10}$$

where $\widehat{\mathcal{T}_{\boldsymbol{\mu}}} = B^{-1}\sum_{b=1}^{B}\mathcal{T}_b$ is the mean estimated with all B iterations. The diagonal entries of $\widehat{\Sigma}_{\text{A}}$ capture individual variabilities of VD and PCD, while off-diagonal entries reflect their correlation under the null.

[2]Mathematically, the covariance matrix $\Sigma_{\boldsymbol{x}} \in \mathcal{S}_+(\mathcal{F})$ defines a positive semi-definite (PSD) covariance operator on the semantic feature space $\mathcal{F}$, with $\mathcal{S}_+(\mathcal{F})$ being the cone of PSD operators on $\mathcal{F}$. Thus, the RKHS $\mathcal{H}_\rho$ where we embed $\Sigma_{\boldsymbol{x}}$ into and define $\text{PCD}(\mathbb{P}, \mathbb{Q}, \rho)$ is characterized by the kernel $\rho : \mathcal{S}_+(\mathcal{F}) \times \mathcal{S}_+(\mathcal{F}) \to \mathbb{R}$ and corresponding map $\rho(\cdot, \Sigma_{\boldsymbol{x}}) \in \mathcal{H}_\rho$.

Permutation Test. Following the SAD pipeline (Section 2), we apply a *permutation test* with R random permutations to $\mathcal{T}^{\mathrm{A}}(X, Y; \ell, \rho)$ (Eq.(9)). For each $r = 1, \ldots, R$, we compute $\mathcal{T}^{\mathrm{A}}_{(r)}$ while keeping $\widehat{\Sigma}_{\mathrm{A}}$ fixed at Eq.(10), so the aggregation weights reflect the null covariance structure across permutations. At significance level α, the testing threshold is $t_\alpha = \mathrm{Quantile}_{1-\alpha}(\{\mathcal{T}^{\mathrm{A}}_{(r)}\}_{r=1}^{R})$.
Detection. We reject $\boldsymbol{H}_0^{\mathrm{A}}$ (i.e., classifying Y as adversarial) if and only if $\mathcal{T}^{\mathrm{A}}(X, Y; \ell, \rho) > t_\alpha$.

5 Theoretical Analysis

In this section, we present theoretical guarantees for our test statistics, with full proofs in Appendix E. These results are obtained with standard tools, namely concentration results for U-statistics [48], permutation tests [49], and the correlation-aware statistics aggregation of Zhou et al. [31]; their purpose is to establish the validity and consistency of the new statistics VD and PCD and of the USAD framework, rather than to develop new proof techniques. We first establish Type-I error control as follows.

Theorem 1. *Given the test batch Y consists of clean data, the VD, PCD, or USAD, when combined with a permutation test at significance level $\alpha \in (0, 1)$, controls the Type I error at level α; that is, the probability of falsely identifying the test batch Y as adversarial is at most α.*

We next study the test power of VD as follows.

Theorem 2. *Let ℓ be a bounded kernel. Under the alternative hypothesis $\boldsymbol{H}_1^{\mathrm{V}}$, the power of the VD test, that is, the probability of correctly rejecting the null hypothesis $\boldsymbol{H}_0^{\mathrm{V}}$, converges to 1 as $n, m \to \infty$ with $n/m \to c \in (0, \infty)$.*

This guarantees that VD is a consistent statistical test; a similar result holds for PCD, as below.

Theorem 3. *Let ρ be a bounded kernel. Under the alternative hypothesis $\boldsymbol{H}_1^{\mathrm{C}}$, the power of the PCD test converges to 1 as $n, m \to \infty$ with $n/m \to c \in (0, \infty)$.*

We now extend these results to the USAD.

Corollary 4. *If **either or both** the alternative hypotheses $\boldsymbol{H}_1^{\mathrm{V}}$ and $\boldsymbol{H}_1^{\mathrm{C}}$ hold, the power of the USAD test converges to 1 as $n, m \to \infty$ with $n/m \to c \in (0, \infty)$, provided that the condition number of $\widehat{\Sigma}_{\mathrm{A}}$ remains bounded.*

In Theorem 1 and Corollary 4, although our analysis focuses on USAD, the aggregation framework is general and can incorporate additional statistics such as MMD. In particular, Corollary 4 covers alternatives under which $\boldsymbol{H}_1^{\mathrm{V}}$ or $\boldsymbol{H}_1^{\mathrm{C}}$ holds, so alternatives that change the mean embedding while leaving $V_{\mathbb{Q}}$ and $\boldsymbol{\mu}_{\Sigma_{\mathbb{Q}}}$ unchanged are not targeted by the current USAD; such alternatives can be addressed by incorporating MMD, and Appendix G.8 reports a VD+PCD+MMD variant, showing that adding more statistics does not necessarily improve finite-sample power and that the statistics should be selected according to the target alternatives.

Remark 1. *Our analysis concerns **distribution-level detection** and does* NOT *require the query batch to be purely adversarial or purely clean: we test whether the query distribution $\mathbb{Q}$* significantly differs *from a trusted clean reference $\mathbb{P}$. If Y contains a **non-negligible fraction** of adversarial examples, the induced mixture yields a detectable distribution shift, and the test power approaches 1 for sufficiently large batches (Theorems 2–3 and Corollary 4). Conversely, if adversarial examples are diluted to a **vanishing fraction** (CE mixing level $\to 100\%$), the mixture can become effectively indistinguishable from clean data; in this regime, the rejection probability is controlled by the significance level α with finite-sample Type-I error guarantees (Theorem 1). Quantitatively, the population discrepancies underlying VD and PCD decrease with the fraction of adversarial examples in Y, so that a smaller fraction requires a larger query batch for detection (see Remark 2 in Appendix G.10).*

6 Experiments

We evaluate our approaches on ResNet-18/50 [50] and ViT-B-16 [51] trained on *CIFAR-10* [32] and *ImageNet-1K* [33]. We report CIFAR-10 results here and defer ImageNet-1K results, Type-I error, transfer attack, ℓ_0, ℓ_1, ℓ_2 attacks and adversarial video detection on UCF101 [52] to Appendix G. See the experimental setup in Appendix F.

Table 1: Test power for different input example sizes $m = |Y|$ decreasing from 50 to 10 under PGD attack with $\epsilon = 4/255$. The test power results are averaged over $1,000$ repetitions. The target model is ResNet-18 trained on *CIFAR-10* dataset.

Method	$m = 10$	$m = 20$	$m = 30$	$m = 40$	$m = 50$
SAMMD [7]	0.292 ± 0.029	0.486 ± 0.022	0.650 ± 0.008	0.756 ± 0.019	0.832 ± 0.009
PCD (ours)	0.486 ± 0.024	0.828 ± 0.013	0.958 ± 0.007	0.998 ± 0.002	$\mathbf{1.000 \pm 0.000}$
VD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
MMDAgg [53]	0.342 ± 0.016	0.476 ± 0.022	0.632 ± 0.020	0.816 ± 0.017	0.912 ± 0.009
MMD-FUSE [54]	0.524 ± 0.017	0.892 ± 0.014	0.994 ± 0.004	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
MMD-DUAL [31]	0.752 ± 0.022	0.978 ± 0.003	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
USAD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$

Table 2: Test power under adaptive attack with $\epsilon = 4/255$. We increase the input example sizes $m = |Y|$ from 10 to 50 to show that all the SAD methods can gain power from example size, even though the adaptive attack can effectively avoid the detection with insufficient examples. The test power results are averaged over $1,000$ repetitions. The target model is ResNet-18 trained on *CIFAR-10* dataset.

Method	Attack	$m = 10$	$m = 20$	$m = 30$	$m = 40$	$m = 50$
PCD	PGD	0.486 ± 0.024	0.828 ± 0.013	0.958 ± 0.007	0.998 ± 0.002	1.000 ± 0.000
	Adaptive	0.388 ± 0.027	0.714 ± 0.020	0.768 ± 0.033	0.882 ± 0.010	0.914 ± 0.014
VD	PGD	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
	Adaptive	0.186 ± 0.023	0.326 ± 0.007	0.416 ± 0.022	0.546 ± 0.021	0.614 ± 0.011
USAD	PGD	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
	Adaptive	0.818 ± 0.013	0.998 ± 0.002	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
Avg. Decrease		-0.365 ± 0.207	-0.263 ± 0.193	-0.258 ± 0.168	-0.190 ± 0.131	-0.157 ± 0.111

Baselines. We compare our methods against the state-of-the-art SAD method: **SAMMD** [7], an MMD-based detector that uses semantic features, which significantly outperforms all the single MMD-based two-sample testing methods (e.g., MMD-D [55], C2ST [56]), we refer readers interested in these experimental results to SAMMD. We do not directly compare with per-example detectors (e.g., LID [13], EPS-AD [15], and PFD [57]), which classify each input separately and do not provide explicit Type-I error control; we view them as complementary to SAD (see Appendix A).

Evaluation Metric. We evaluate detection performance using test power, which measures the ability to correctly identify adversarial examples while maintaining a significance level of $\alpha = 0.05$. Figure 3(a–d) and Figure 3(e–h) report the test power under five attacks, where the query batches consist entirely of adversarial examples: *AutoAttack* (AA) [6], *basic iterative method* (BIM) [34], *Carlini & Wagner* (CW) attack [4], *fast gradient sign method* (FGSM) [2] and *projected gradient descent* (PGD) [3] under different target models that are trained on either *CIFAR-10* or *ImageNet-1K*.

Overall Result. Our results demonstrate that USAD consistently achieves superior performance across all attacks. Notably, for all attacks, USAD reaches perfect detection (test power ≈ 1.0) whenever the perturbation budget $\epsilon \geq 4/255$ under a relatively small example size $|Y| = 50$, and even our proposed single uncertainty-aware methods PCD and VD can reach perfect detection whenever $\epsilon \geq 6/255$, all three significantly outperforming the baseline SAMMD.

Impact of Sample Size. To study robustness to varying example sizes, we compare PCD and VD to the single-kernel MMD baseline (SAMMD), and compare USAD to representative multi-kernel MMD tests (e.g., **MMDAgg** [53], **MMD-FUSE** [54], and **MMD-DUAL** [31]) that boost power in small-sample regimes via kernel aggregation. Table 1 reports results as the query size ranges from 10 to 50. PCD and VD consistently outperform SAMMD, and USAD consistently outperforms MMD-based kernel aggregation methods, maintaining power 1.0 even with *only 10 query examples*. An ablation comparing the correlation-aware aggregation of USAD with a simple equal-weight sum of standardized VD and PCD is reported in Appendix G.8.

Performance against Adaptive Attack. Inspired by Gao et al. [7], Kaya et al. [58], we evaluate adaptive attacks designed to evade SAD methods. Specifically, we use adaptive PGD variants that maximize the adversarial cross-entropy loss while minimizing the discrepancy measured by the target statistic (VD, PCD, or USAD; see Eqs. (5), (8), and (9)). Table 2 shows that although power drops compared to non-adaptive attacks, USAD still achieves 0.818 power with only 10 examples, indicating strong robustness. The component-wise results further suggest that PCD better retains power under adaptive attacks: such attacks can suppress the global spread signal used by VD, whereas

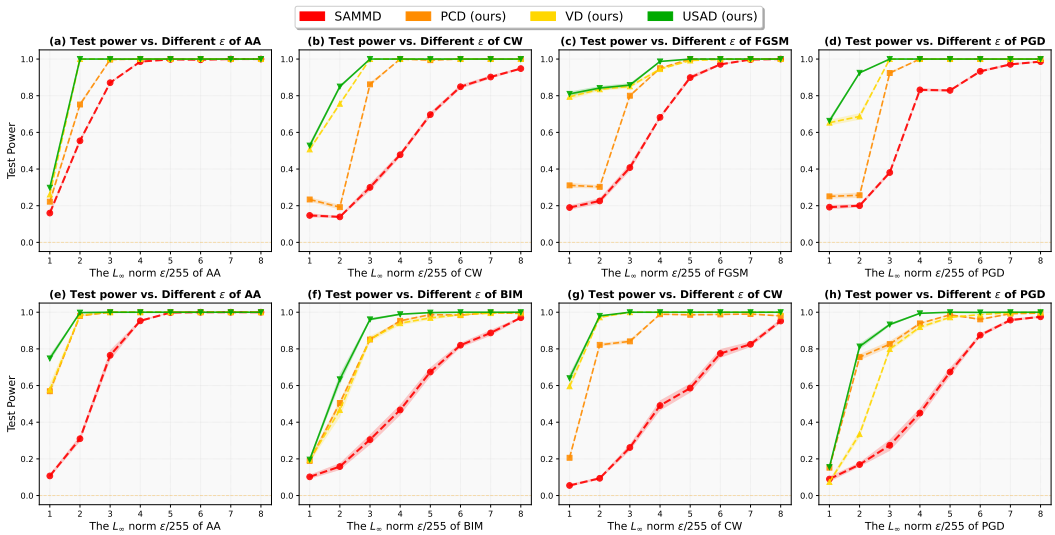


Figure 3: Results $(a-h)$ are test power (detection rate) under different adversarial attacks with different ϵ, the given adversarial samples all share the same sample size $|Y| = 50$. The results are averaged over $1,000$ repetitions and the ideal test power is 1. The target model is ResNet-18 trained on *CIFAR-10* dataset in $(a-d)$, and ResNet-50 trained on *ImageNet* dataset in $(e-h)$.

Table 3: Rejection rates for different *clean examples* (CE) mixing levels from 0% to 100% with $|Y| = 50$, under PGD attack with $\epsilon = 4/255$. The results are averaged over $1,000$ repetitions. The target model is ResNet-18 trained on *CIFAR-10* dataset.

Method	0	20	40	60	80	100
SAMMD	0.828 ± 0.016	0.620 ± 0.013	0.344 ± 0.010	0.196 ± 0.012	0.052 ± 0.010	0.040 ± 0.009
PCD (ours)	0.996 ± 0.002	0.952 ± 0.005	0.356 ± 0.020	0.142 ± 0.010	0.066 ± 0.011	0.056 ± 0.004
VD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.998 \pm 0.002}$	$\mathbf{0.778 \pm 0.016}$	0.060 ± 0.006
MMDAgg	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.992 ± 0.003	0.548 ± 0.025	0.096 ± 0.005	0.050 ± 0.009
MMD-FUSE	0.908 ± 0.010	0.682 ± 0.013	0.330 ± 0.022	0.168 ± 0.015	0.050 ± 0.006	0.030 ± 0.009
MMD-DUAL	$\mathbf{1.000 \pm 0.000}$	0.990 ± 0.003	0.806 ± 0.012	0.274 ± 0.012	0.068 ± 0.009	0.040 ± 0.006
USAD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.900 ± 0.014	0.350 ± 0.018	0.038 ± 0.007

reducing the local instability captured by PCD is harder while preserving adversarial effectiveness. More importantly, these results highlight a key advantage of SAD over single-example detectors (e.g., LID [13] and EPS-AD [15]): adaptive attacks can reduce power in small batches, but as batch size grows, SAD consistently separates AEs from CEs. Appendix G.12 further compares USAD with the per-example detector PFD [57] under an adaptive C&W attack [38].

Impact of CE mixing levels. We evaluate all methods as the fraction of *clean examples (CEs)* in the query batch increases from 0% to 100%. As reported in Table 3, detection becomes harder as the CE proportion grows (the induced shift weakens), yet our methods attain the highest power across mixing levels. When the query batch is purely clean (100% CE), the problem reduces to the null setting and the rejection rate (i.e., Type-I error) stays around the significance level $\alpha = 0.05$. Notably, even at a high mixing level (e.g., 80% CE), VD still achieves strong power (around 0.78), while competing methods degrade to near-chance performance. We provide a more detailed discussion of this distribution-level perspective and the mixture regimes in Remark 1. Lower AE fractions in the query batch, together with larger batch sizes, are studied in Appendix G.10.

7 Conclusion

This work reveals that the general-purpose metric, MMD, is insensitive to how adversarial perturbations actually manifest as distributional uncertainty changes. We addressed this by developing two statistics to measure global and local uncertainties and combining them into an uncertainty-aware SAD detector, achieving high power in the small-sample regime with theoretically false alarm control. Our formulation offers a practical template: task-aligned features, statistics that encode the relevant geometry, and principled aggregation to discount redundancy. We expect this recipe to extend beyond SAD to general distribution shift testing.

Acknowledgements

This research was supported by the University of Melbourne's Research Computing Services and the Petascale Campus Initiative. ZJZ and XYT are supported by the Melbourne Research Scholarship and the ARC with grant number DE240101089. JCZ and YYG are supported by the Melbourne Research Scholarship. LHP is supported by the ARC with grant number LP240100101. FL is supported by the ARC with grant numbers DP230101540, DE240101089 and DP260103996.

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[4] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017.

[5] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[7] Ruize Gao, Feng Liu, Jingfeng Zhang, Bo Han, Tongliang Liu, Gang Niu, and Masashi Sugiyama. Maximum mean discrepancy test is aware of adversarial attacks. In *ICML*, 2021.

[8] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *IEEE Symposium on Security and Privacy*, pages 176–194, 2021.

[9] Pengfei Jing, Qiyi Tang, Yuefeng Du, Lei Xue, Xiapu Luo, Ting Wang, Sen Nie, and Shi Wu. Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations. In *USENIX Security Symposium*, pages 3237–3254, 2021.

[10] Jiacheng Zhang, Feng Liu, Dawei Zhou, Jingfeng Zhang, and Tongliang Liu. Improving accuracy-robustness trade-off via pixel reweighted adversarial training. In *ICML*, 2024.

[11] Yuhao Sun, Jiacheng Zhang, Zesheng Ye, Chaowei Xiao, and Feng Liu. Sample-specific noise injection for diffusion-based adversarial purification. In *ICML*, 2025.

[12] Bo Han, Jiangchao Yao, Tongliang Liu, Bo Li, Sanmi Koyejo, and Feng Liu. *Trustworthy Machine Learning: From Data to Models*. Now Foundations and Trends, 2025.

[13] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

[14] Zhijie Deng, Xiao Yang, Shizhen Xu, Hang Su, and Jun Zhu. LiBRe: A practical Bayesian approach to adversarial detection. In *CVPR*, 2021.

[15] Shuhai Zhang, Feng Liu, Jiahao Yang, Yifan Yang, Changsheng Li, Bo Han, and Mingkui Tan. Detecting adversarial data by probing multiple perturbations using expected perturbation score. In *ICML*, 2023.

[16] Florian Tramèr. Detecting adversarial examples is (nearly) as hard as classifying them. In *ICML*, 2022.

[17] Oliver Bryniarski, Nabeel Hingun, Pedro Pachuca, Vincent Wang, and Nicholas Carlini. Evading adversarial example detection defenses with orthogonal projected gradient descent. In *ICLR*, 2022.

[18] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

[19] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[20] Chin-Wei Huang, Jae Hyun Lim, and Aaron C. Courville. A variational perspective on diffusion-based generative models and score matching. In *NeurIPS*, 2021.

[21] Haotian Xue, Alexandre Araujo, Bin Hu, and Yongxin Chen. Diffusion-based adversarial sample generation for improved stealthiness and controllability. In *NeurIPS*, 2023.

[22] Mintong Kang, Dawn Song, and Bo Li. Diffattack: Evasion attacks against diffusion-based adversarial purification. In *NeurIPS*, 2023.

[23] Jiacheng Zhang, Benjamin I. P. Rubinstein, Jingfeng Zhang, and Feng Liu. One stone, two birds: Enhancing adversarial defense through the lens of distributional discrepancy. In *ICML*, 2025.

[24] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

[25] Bo Huang, Yi Wang, and Wei Wang. Model-agnostic adversarial detection by random perturbations. In *IJCAI*, 2019.

[26] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In *ICML*, 2019.

[27] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv*, 2017.

[28] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[29] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *NeurIPS*, 2016.

[30] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

[31] Zhijian Zhou, Xunye Tian, Liuhua Peng, Chao Lei, Antonin Schrab, Danica J Sutherland, and Feng Liu. DUAL: Learning diverse kernels for aggregated two-sample and independence testing. In *NeurIPS*, 2025.

[32] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 (canadian institute for advanced research), 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[34] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR, Workshop Track Proceedings*, 2017.

[35] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020.

[36] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *ECCV*, 2020.

[37] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv*, 2017.

[38] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *AISec*, 2017.

[39] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[40] Zaid Harchaoui, Eric Moulines, and Francis Bach. Kernel change-point analysis. In *NeurIPS*, 2008.

[41] Zhijian Zhou, Jie Ni, Jia-He Yao, and Wei Gao. On the exploration of local significant differences for two-sample test. In *NeurIPS*, 2023.

[42] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *NeurIPS*, 2019.

[43] Natsumi Makigusa. Two-sample test based on maximum variance discrepancy. *Communications in Statistics – Theory and Methods*, 53(15):5421–5438, 2024.

[44] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

[45] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. Robustness of classifiers to universal perturbations: A geometric perspective. In *ICLR*, 2018.

[46] Zhuolin Yang, Linyi Li, Xiaojun Xu, Bhavya Kailkhura, Tao Xie, and Bo Li. On the certified robustness for ensemble models and beyond. In *ICLR*, 2022.

[47] Anirban Chatterjee and Bhaswar B Bhattacharya. Boosting the power of kernel two-sample tests. *Biometrika*, 112(1):asae048, 2025.

[48] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426, New York, NY, 1994. Springer.

[49] Ilmun Kim, Sivaraman Balakrishnan, and Larry Wasserman. Minimax optimality of permutation tests. *Annals of Statistics*, 50(1):225–251, 2022.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[51] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[52] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *AAAI*, 2019.

[53] Antonin Schrab, Ilmun Kim, Mélisande Albert, Béatrice Laurent, Benjamin Guedj, and Arthur Gretton. MMD aggregated two-sample test. *Journal of Machine Learning Research*, 24: 194:1–194:81, 2023.

[54] Felix Biggs, Antonin Schrab, and Arthur Gretton. MMD-Fuse: Learning and combining kernels for two-sample testing without data splitting. In *NeurIPS*, 2023.

[55] Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J Sutherland. Learning deep kernels for non-parametric two-sample tests. In *ICML*, 2020.

[56] X.-Y. Cheng and A. Cloninger. Classification logit two-sample testing by neural networks. *arXiv*, 2019.

[57] Qian Wang, Chen Li, Yuchen Luo, Hefei Ling, Shijuan Huang, Ruoxi Jia, and Ning Yu. Detecting adversarial data using perturbation forgery. In *CVPR*, 2025.

[58] Yigitcan Kaya, Muhammad Bilal Zafar, Sergul Aydore, Nathalie Rauschmayr, and Krishnaram Kenthapadi. Generating distributional adversarial examples to evade statistical detectors. In *ICML*, 2022.

[59] Z. Harchaoui, F. R. Bach, and E. Moulines. Testing for homogeneity with kernel fisher discriminant analysis. In *NeurIPS*, 2007.

[60] M. Kirchler, S. Khorasani, M. Kloft, and C. Lippert. Two-sample testing using deep learning. In *AISTATS*, 2020.

[61] J. Kübler, W. Jitkrittum, B. Schölkopf, and K. Muandet. Learning kernel tests without data splitting. In *Advances in Neural Information Processing Systems 33*, pages 6245–6255, Dutchess, NY, 2020. Curran Associates.

[62] O. Hagrass, B. Sriperumbudur, and B. Li. Spectral regularized kernel two-sample tests. *Annals of Statistics*, 52(3):1076–1101, 2024.

[63] Zhijian Zhou, Liuhua Peng, Xunye Tian, Mingming Gong, and Feng Liu. Are two datasets close enough with statistical significance? A kernel distributional closeness testing approach. In *ICML*, 2026.

[64] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

[65] Jesse Hemerik and Jelle Goeman. Exact testing with random permutations. *Test*, 27(4):811–825, 2018.

[66] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, 2001.

[67] Antonin Schrab, Ilmun Kim, Benjamin Guedj, and Arthur Gretton. Efficient aggregated kernel tests using incomplete U-statistics. In *NeurIPS*, 2022.

[68] Xuyang Zhong and Chen Liu. Sparse-pgd: A unified framework for sparse adversarial perturbations generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[69] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv*, 2018.

Appendix Contents

A More Discussions on the Scope, Position, and Limitations of SAD **16**

B Related Works **17**

C Permutation Test **17**

D Maximum Mean Discrepancy **19**

E Proofs of Theoretical Analysis **20**

E.1 Detailed Proofs of Theorem 1 21

E.2 Detailed Proofs of Theorem 2 21

E.3 Detailed Proofs of Theorem 3 23

E.4 Detailed Proofs of Corollary 4 24

F Experimental Details **24**

F.1 Aggregation Strategy 24

F.2 Inference Time Complexity 25

F.3 Implementation Details. 26

G Additional Experiments **27**

G.1 Detection under L_2 Norm Attacks 27

G.2 Detailed Numerical Results 28

G.3 Robustness to Transfer Attacks and Additional Architectures 29

G.4 Type-I Error Control Check 31

G.5 Computational Efficiency. 32

G.6 ℓ_0 (Sparse-PGD) and ℓ_1 Attacks. 32

G.7 Adversarial Video Detection on UCF101. 32

G.8 Ablation on the Correlation-aware Aggregation and Incorporating MMD 33

G.9 Analysis of the Two Uncertainty Factors 34

G.10 Detection under Low AE Fractions in the Query Batch 35

G.11 Behavior under a Benign Distribution Shift 37

G.12 Comparison with MVD and with a Per-example Detector under an Adaptive C&W Attack 38

G.13 Sensitivity to the Kernel Bandwidth and the Number of Perturbations 39

A More Discussions on the Scope, Position, and Limitations of SAD

This paper focuses on *statistical adversarial detection* (SAD) as an active research area and a promising paradigm complementary to existing adversarial defense strategies. We have identified the fundamental limitation of existing SAD practices and proposed a concrete variant termed *Uncertainty-aware Statistical Adversarial Detection* (USAD) accordingly.

Scope of SAD. By framing adversarial detection as a two-sample hypothesis test, SAD offers two practical advantages: it requires neither modification of the classifier nor attack-specific training, and it provides a *lightweight monitoring primitive*. Once a query window is flagged as distributionally suspicious, the system can throttle the source, trigger more expensive defenses, or route the traffic for human inspection, while maintaining controlled false-alarm rates. Beyond test-time monitoring, the same batch-level tests can also support *data cleaning* in large-scale training pipelines, where coordinated manipulation may contaminate entire mini-batches from distributed or third-party data sources. In this scope, USAD strengthens SAD by substantially improving detection reliability in small and early-stage windows, where MMD-based methods often lack sufficient power. This makes USAD particularly well-suited for timely monitoring and data vetting in realistic deployment settings [7]. Our target setting is provider-side deployment, where the model owner has access to the classifier f and its intermediate representations. Therefore, USAD assumes a white-box setting from the defender's perspective, rather than a strict label-only black-box setting; this assumption is also common in feature-based adversarial detection and prior SAD methods, such as LID [13], SAMMD [7], and Zhang et al. [23], which use internal model features for detection. Extending USAD to settings where only predicted labels or confidence scores are available is an interesting direction that we leave for future work.

Position of USAD. We emphasize that USAD (and other SAD variants) is not intended to replace per-example detectors. Rather, it serves as a statistical monitor operating on short query windows. Detecting individual AEs in isolation is known to be nearly as hard as robust classification and typically relies on strong attack-specific assumptions [16]. In contrast, SAD deliberately trades per-example granularity for rigorous false-alarm control and reliable detection power at the window level, making it a natural building block for monitoring and throttling in deployed systems. In Section 6, we further show that USAD maintains high power even when AEs are sparsely mixed with CEs within each window.

Towards Per-example Localization. We view USAD and per-example detectors as complementary: USAD first flags a suspicious batch with explicit false-alarm control, after which a per-example detector can be used to localize or filter individual AEs. Moreover, the VD and PCD statistics can also be decomposed into per-example contributions. For VD, define, for each query $\boldsymbol{y}_j \in Y$,

$$v_j = \frac{\ell(f(\boldsymbol{y}_j), f(\boldsymbol{y}_j))}{m} - \sum_{i \neq j}^{m} \frac{\ell(f(\boldsymbol{y}_i), f(\boldsymbol{y}_j))}{m(m-1)},$$

so that $\widehat{V}(Y;\ell) = \sum_{j=1}^{m} v_j$. Then, by Eq. (5),

$$\widehat{\mathrm{VD}}(X, Y; \ell) = \sum_{j=1}^{m} \left(\widehat{V}(X;\ell) - \widehat{V}(Y;\ell)\right) \left(\frac{\widehat{V}(X;\ell)}{m} - v_j\right).$$

Similarly, by Eq. (8), $\widehat{\mathrm{PCD}}(X, Y; \rho) = \sum_{j=1}^{m} c_j^{\mathrm{PCD}}$, where

$$c_j^{\mathrm{PCD}} = \frac{1}{m} \sum_{i \neq i'}^{n} \frac{\rho(\Sigma_{\boldsymbol{x}_i}, \Sigma_{\boldsymbol{x}_{i'}})}{n(n-1)} - \sum_{i=1}^{n} \frac{2\rho(\Sigma_{\boldsymbol{x}_i}, \Sigma_{\boldsymbol{y}_j})}{nm} + \sum_{i \neq j}^{m} \frac{\rho(\Sigma_{\boldsymbol{y}_i}, \Sigma_{\boldsymbol{y}_j})}{m(m-1)},$$

in which the first term distributes the clean–clean term of Eq. (8) evenly over the queries. These contributions can be combined using the USAD aggregation weights (i.e., $\widehat{\Sigma}_{\mathrm{A}}^{-1}$ in Eq. (9)) to rank the query examples: larger positive contributions indicate examples that contribute more strongly to the rejection and can be prioritized for filtering or further examination. We note that the current guarantees remain at the batch level, and we leave a formal per-example analysis for future work.

Limitations of SAD. That said, we acknowledge the limitations of SAD in its current forms. Prior to USAD, SAD is instantiated almost exclusively with *maximum mean discrepancy* (MMD) [24] as the

test statistic $\mathcal{T}(X, Y)$ [7, 23]. These MMD-based SAD methods are inherently *sample-inefficient* (see Section 1 and Figure 1a): reliable detection typically requires query windows with sufficiently large effective sample sizes and non-trivial adversarial fractions to achieve high test power. This limitation stems from the fact that MMD primarily captures discrepancies in kernel mean embeddings, whose estimation variance can be substantial in small-sample regimes. As a result, MMD-based SAD methods can fail precisely in early-stage or low-volume attack regimes where timely detection is most important.

Limitations of USAD. We also acknowledge the limitations of USAD itself. First, USAD targets changes in the global feature spread or the perturbation-induced local covariance, which are specifically motivated by the uncertainty patterns of AEs rather than arbitrary distributional differences. However, as Eq. (2) tests distributional equality, these statistics are not formally specific to adversarial attacks, and benign shifts that alter the same quantities may also be detected; the response of USAD to benign shifts thus depends on whether they change the uncertainty patterns captured by VD or PCD, and USAD relies on a clean reference set X that is representative of the deployment distribution. Appendix G.11 examines a benign shift empirically and shows that benign-shift and adversarial batches remain highly separable according to VD and PCD. Second, detection power depends jointly on the fraction of AEs in the query batch, the batch size, and the attack signal: when AEs constitute only a very small fraction of the query batch, reliable detection requires substantially larger batches (Appendix G.10), and detection may also become more challenging when AEs induce only weak changes in the quantities captured by VD or PCD. Third, USAD is not sensitive to every possible alternative: distributional differences that leave $V_{\mathbb{Q}}$ and $\boldsymbol{\mu}_{\Sigma_{\mathbb{Q}}}$ unchanged (e.g., pure mean-embedding shifts) are not targeted by the current statistics, although the USAD framework can incorporate additional statistics such as MMD for this purpose (Appendix G.8). Finally, USAD requires access to the semantic features of the classifier f (see the scope above), and PCD incurs a higher computational cost than VD (Appendix G.5).

B Related Works

Variance Discrepancy. Prior studies have utilized second-order information in RKHS in several ways. Some analyze covariance operators to derive asymptotic null distributions for two-sample tests [40, 59–61], while others spectrally regularize MMD to incorporate covariance structure into a mean-embedding statistic [62]. NAMMD [63] further employs the trace of the covariance operator to improve comparability in distribution closeness testing using a simple finite-sample estimator, and the maximum variance discrepancy (MVD) [43] compares the full covariance operators (see Section 3.1 for its relationship to VD). In contrast, our *variance discrepancy* (VD) captures a signal particularly relevant for adversarial detection by directly comparing the within-distribution variances of semantic features. This focuses on the concentration of features within each distribution, which reflects the global uncertainty in Factor 1 and provides additional sensitivity to manifold deviations characteristic of adversarial examples [13, 27].

Perturbation-based Covariance Discrepancy. Previous score-based detectors typically perturb a test point and read out label uncertainty. Some methods craft gradient-guided perturbations at one or multiple layers of a deep classifier to probe local sensitivity [64]; others use random perturbations to measure prediction instability [15, 25, 26, 57], in line with the connection between adversarial robustness and stability under additive noise established by randomized smoothing and related certification methods [42, 44, 46]. A complementary line estimates uncertainty from model randomness, such as Monte-Carlo dropout or Bayesian parameter sampling, and then perturbs predictions accordingly [14, 38]. Inspired by these ideas, our approach moves from label space to feature space: instead of only tracking how the predicted label flips, we estimate per-example perturbation covariances of deep features and compare their distributions via a kernel on PSD matrices (Section 3.2). This captures richer local geometry, yields a simple, scalable statistic that aligns with manifold-based accounts of clean vs. adversarial behavior.

C Permutation Test

We detail the permutation test used throughout *statistical adversarial detection* (SAD) to calibrate the decision threshold t_α for our proposed statistics (VD, PCD, and their aggregate USAD).

From SAD to Calibration. For completeness, we briefly review the problem setup of SAD and the reason for conducting the permutation test. Section 2.1 formulates SAD as a two-sample hypothesis test problem: given a reference set of *clean examples* (CEs) $X = \{\boldsymbol{x}_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} \mathbb{P}$ and a query set $Y = \{\boldsymbol{y}_j\}_{j=1}^m \overset{\text{i.i.d.}}{\sim} \mathbb{Q}$, we test

$$\boldsymbol{H}_0 : \mathbb{P} = \mathbb{Q} \quad \text{vs.} \quad \boldsymbol{H}_1 : \mathbb{P} \neq \mathbb{Q}.$$

At significance level α, we want a *test statistic* $\mathcal{T}(X, Y)$ and a *testing threshold* t_α such that $\Pr(\mathcal{T}(X, Y) > t_\alpha \mid \boldsymbol{H}_0) \leq \alpha$, so that rejecting $\boldsymbol{H}_0$ (declaring Y adversarial) has a controlled false alarm rate. That is, the probability of falsely identifying the query data Y as adversarial is at most α. In this paper, $\mathcal{T}(X, Y)$ can be instantiated by VD (Section 3.1 and Eq.(5)), PCD (Section 3.2 and Eq.(8)), or their aggregate USAD (Section 4 and Eq.(9)).

Here, the missing piece is t_α, which is defined as the $(1-\alpha)$-quantile of the null distribution of $\mathcal{T}(X, Y)$ under $\boldsymbol{H}_0$. For most SAD methods and modern statistics (including MMD, VD, PCD, and USAD), this distribution is not available in closed form. We therefore exploit the *exchangeability* property of the SAD setup: *under $\boldsymbol{H}_0 : \mathbb{P} = \mathbb{Q}$, samples from X and Y are i.i.d., so the labels "reference" and "query" are arbitrary* and construct an empirical null distribution using *permutation test* with X and Y. We will finally determine the *testing threshold* t_α calibrated on this empirical null distribution.

Statistics Under Test. For notational convenience, we write

$$\text{STAT} \in \left\{ \widehat{\text{VD}}(\cdot, \cdot; \ell),\ \widehat{\text{PCD}}(\cdot, \cdot; \rho),\ \mathcal{T}^{\text{A}}(\cdot, \cdot; \ell, \rho) \right\}$$

and use ϕ to denote the corresponding kernel choice ($\phi = \ell$ for VD, $\phi = \rho$ for PCD, and $\phi = \{\ell, \rho\}$ for USAD).

Permutation Null Construction. We have previously mentioned that X and Y are exchangeable under $\boldsymbol{H}_0$ and will construct the empirical null distribution with pooled samples

$$Z \triangleq X \cup Y = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{n+m}\}.$$

Let $\boldsymbol{\Pi}_{n+m}$ be the set of permutations of $\{1, \ldots, n+m\}$. For each permutation $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{n+m}) \in \boldsymbol{\Pi}_{n+m}$, we form

$$X_{\boldsymbol{\pi}} = \{\boldsymbol{z}_{\pi_i}\}_{i=1}^n \quad \text{and} \quad Y_{\boldsymbol{\pi}} = \{\boldsymbol{z}_{\pi_i}\}_{i=n+1}^{n+m},$$

and define

$$\text{STAT}(\boldsymbol{\pi} Z; \phi) \triangleq \text{STAT}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \phi),$$

where $\boldsymbol{\pi} Z = \{\boldsymbol{z}_{\pi_1}, \ldots, \boldsymbol{z}_{\pi_n}, \boldsymbol{z}_{\pi_{n+1}}, \ldots, \boldsymbol{z}_{\pi_{n+m}}\}$ denotes the shuffled pooled samples.

Since under $\boldsymbol{H}_0$ the entries of Z are i.i.d., Z is exchangeable: permuting the indices does not change its distribution. Hence, under $\boldsymbol{H}_0$, $\text{STAT}(X, Y; \phi)$ has the same distribution as $\text{STAT}(\boldsymbol{\pi} Z; \phi)$ for a uniformly random $\boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}$ independent of Z.

Monte-Carlo Approximation. Enumerating all $(n+m)!$ permutations in $\boldsymbol{\Pi}_{n+m}$ (or, equivalently, all $\binom{n+m}{n}$ distinct splits $(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}})$, on which $\text{STAT}(\boldsymbol{\pi} Z; \phi)$ depends) is usually impossible, so we approximate the permutation distribution by sampling R permutations. For each $r = 1, \ldots, R$, we

1. draw $\boldsymbol{\pi}_{(r)}$ uniformly from $\boldsymbol{\Pi}_{n+m}$;
2. construct $X_{(r)} = X_{\boldsymbol{\pi}_{(r)}}$ and $Y_{(r)} = Y_{\boldsymbol{\pi}_{(r)}}$;
3. compute
$$\text{STAT}^{(r)} = \text{STAT}(\boldsymbol{\pi}_{(r)} Z; \phi) = \text{STAT}(X_{(r)}, Y_{(r)}; \phi),$$

leading to R such statistics, i.e., $\text{STAT}^{(r)}$ for $r = 1, \ldots, R$, in total. Then, the empirical upper-α *testing threshold* is defined as

$$t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R) = \arg\min_t \left\{ \sum_{r=1}^R \frac{\mathbb{I}[\text{STAT}(\boldsymbol{\pi}_{(r)} Z; \phi) \leq t]}{R+1} \geq 1 - \alpha \right\}.$$

This is the empirical $(1-\alpha)$-quantile t_α as discussed in Section 2.1, where the denominator $R+1$ accounts for the observed statistic $\mathrm{STAT}(X, Y; \phi)$, which is included among the permutation statistics as the one obtained with the identity permutation [65]. Equivalently, $\boldsymbol{H}_0$ is rejected if and only if $\big(1 + \sum_{r=1}^{R} \mathbb{I}[\mathrm{STAT}(\boldsymbol{\pi}_{(r)} Z; \phi) \geq \mathrm{STAT}(X, Y; \phi)]\big)/(R+1) \leq \alpha$. This guarantees Type-I error control at level α for any number of permutations R (see Appendix E.1); we assume $\alpha(R+1) \geq 1$, since otherwise $\boldsymbol{H}_0$ is never rejected.

Decision Rule. Finally, we have the following test with the *testing threshold* t_α with observed statistic $\mathrm{STAT}(X, Y; \phi)$ on samples X and Y as follows

$$\delta_\alpha = \mathbb{I}\left[\mathrm{STAT}(X, Y; \phi) > t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^{R})\right] \ . \tag{11}$$

D Maximum Mean Discrepancy

Section 2.1 frames SAD as a two-sample testing (TST) between the reference clean distribution $\mathbb{P}$ and a query distribution $\mathbb{Q}$ of potentially *adversarial examples* (AEs). The *maximum mean discrepancy* (MMD) is a popular kernel two-sample statistic used by existing SAD methods and by the SAMMD baseline evaluated in this paper. This section provides a self-contained account of kernel-based TST and of the population and empirical forms of MMD that underlie Eq.(3).

Kernel-based Two-sample Testing (TST) asks whether two independent samples are generated from the same distribution [24]. Formally, let $\mathcal{X}$ be the input space and $X = \{\boldsymbol{x}_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} \mathbb{P}$ and a query set $Y = \{\boldsymbol{y}_j\}_{j=1}^{m} \overset{\text{i.i.d.}}{\sim} \mathbb{Q}$. The goal of TST is to decide whether $\mathbb{P} = \mathbb{Q}$ (*cf.* Eq.(2)). When inputs are in high-dimensional spaces (e.g., images), directly estimating the densities of $\mathbb{P}$ and $\mathbb{Q}$ is fragile. Kernel methods emerge as a widely used class of non-parametric approaches to sidestep this fragility. These methods map each distribution into a *reproducing kernel Hilbert space* (RKHS) where distributions are represented by their mean embeddings, and differences between distributions become distances between points in that RKHS [24, 39]. By definition, let $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel with RKHS $\mathcal{H}_\kappa$ and associated map $\kappa(\cdot, \boldsymbol{x}) \in \mathcal{H}_\kappa$. For any probability measure $\mathbb{W}$ on $\mathcal{X}$ with $E_{\boldsymbol{z}\sim\mathbb{W}}\left[\sqrt{\kappa(\boldsymbol{z}, \boldsymbol{z})}\right] < \infty$, its kernel mean embedding is

$$\boldsymbol{\mu}_{\mathbb{W}} = E_{\boldsymbol{z}\sim\mathbb{W}}\left[\kappa(\cdot, \boldsymbol{z})\right] \in \mathcal{H}_\kappa.$$

Specializing to $\mathbb{P}$ and $\mathbb{Q}$ gives

$$\boldsymbol{\mu}_{\mathbb{P}} = E_{\boldsymbol{x}\sim\mathbb{P}}\left[\kappa(\cdot, \boldsymbol{x})\right] \quad \text{and} \quad \boldsymbol{\mu}_{\mathbb{Q}} = E_{\boldsymbol{y}\sim\mathbb{Q}}\left[\kappa(\cdot, \boldsymbol{y})\right].$$

If κ is *characteristic*, this embedding is injective [54]:

$$\boldsymbol{\mu}_{\mathbb{P}} = \boldsymbol{\mu}_{\mathbb{Q}} \quad \textit{if and only if} \quad \mathbb{P} = \mathbb{Q}.$$

Kernel-based TST, therefore, rephrases the hypotheses as

$$\boldsymbol{H}_0^{\mathrm{M}} : \boldsymbol{\mu}_{\mathbb{P}} = \boldsymbol{\mu}_{\mathbb{Q}} \quad \text{and} \quad \boldsymbol{H}_1^{\mathrm{M}} : \boldsymbol{\mu}_{\mathbb{P}} \neq \boldsymbol{\mu}_{\mathbb{Q}} \ , \tag{12}$$

and bases the test on a norm of $\boldsymbol{\mu}_{\mathbb{P}} - \boldsymbol{\mu}_{\mathbb{Q}}$. Specifically, the *maximum mean discrepancy* (MMD) statistic [24] uses the squared RKHS distance (i.e., ℓ_2 norm) between these mean embeddings, which admits a simple expression in terms of kernel evaluations.

Maximum Mean Discrepancy (MMD) [24] is a popular kernel-based metric that assesses the kernel mean embedding equality with kernel κ:

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}, \kappa) = \|\boldsymbol{\mu}_{\mathbb{P}} - \boldsymbol{\mu}_{\mathbb{Q}}\|_{\mathcal{H}_\kappa}^2.$$

Using the reproducing property $\langle \kappa(\cdot, \boldsymbol{x}), \kappa(\cdot, \boldsymbol{x}')\rangle_{\mathcal{H}_\kappa} = \kappa(\boldsymbol{x}, \boldsymbol{x}')$ and expanding the square gives us the following:

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}, \kappa) = E_{\boldsymbol{x}, \boldsymbol{x}' \overset{\text{i.i.d.}}{\sim} \mathbb{P}}\left[\kappa(\boldsymbol{x}, \boldsymbol{x}')\right] + E_{\boldsymbol{y}, \boldsymbol{y}' \overset{\text{i.i.d.}}{\sim} \mathbb{Q}}\left[\kappa(\boldsymbol{y}, \boldsymbol{y}')\right] - 2E_{\boldsymbol{x}\sim\mathbb{P}, \boldsymbol{y}\sim\mathbb{Q}}\left[\kappa(\boldsymbol{x}, \boldsymbol{y})\right].$$

For a characteristic kernel, $\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}, \kappa) = 0$ if and only if $\mathbb{P} = \mathbb{Q}$ [24, 54].

Empirical Estimator. In practice, the distributions $\mathbb{P}$ and $\mathbb{Q}$ are rarely available in closed form, as well as the population embeddings $\boldsymbol{\mu}_{\mathbb{P}}$ and $\boldsymbol{\mu}_{\mathbb{Q}}$. For the samples X and Y, we can construct an unbiased estimator of $\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}, \kappa)$ by the U-statistic:

$$\widehat{\mathrm{MMD}}^2(X, Y; \kappa) = \sum_{i \neq j}^{n} \frac{\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)}{n(n-1)} - \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{2\kappa(\boldsymbol{x}_i, \boldsymbol{y}_j)}{nm} + \sum_{i \neq j}^{m} \frac{\kappa(\boldsymbol{y}_i, \boldsymbol{y}_j)}{m(m-1)} .$$

Under the sampling $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathbb{P}$ and $\boldsymbol{y}_i \overset{\text{i.i.d.}}{\sim} \mathbb{Q}$, this estimator satisfies $E\left[\widehat{\mathrm{MMD}}^2(X, Y; \kappa)\right] = \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}, \kappa)$ [24].

In the main text, we denote the population quantity by $\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}, \kappa)$ (Eq.(3)). When MMD is used as a SAD *test statistic*, its empirical estimate $\widehat{\mathrm{MMD}}^2(X, Y; \kappa)$ plays the role of $\mathcal{T}(X, Y)$ in Section 2.1. Then, a permutation test (detailed in Section C) is applied to $\widehat{\mathrm{MMD}}^2(X, Y; \kappa)$ to obtain a threshold t_α such that $\Pr(\mathcal{T}(X, Y) > t_\alpha \mid \boldsymbol{H}_0) \approx \alpha$.

MMD on Semantic Features. In existing SAD practices [7, 23], MMD-based SAD methods do not work directly on raw pixels, but on semantic features $f(\boldsymbol{x}) \in \mathcal{F}$ typically extracted from the penultimate layer of the threat classifier (i.e., the classifier under attack). This is equivalent to composing the kernel with the feature map $\kappa(f(\boldsymbol{x}), f(\boldsymbol{x}'))$ and computing $\widehat{\mathrm{MMD}}^2(X, Y; \kappa)$. The SAMMD baseline used in our experiments applies this feature-space MMD, together with calibrated thresholds, to detect adversarial deviations; the mathematical definitions above remain unchanged under this composition.

E Proofs of Theoretical Analysis

We provide here the proofs of the theoretical results presented in the main text. We begin by presenting several foundational concepts and concentration results for U-statistics, which serve as key tools in our analysis.

Definition 5. *[49] Let $g(\boldsymbol{x}, \boldsymbol{y})$ be a bivariate function, which is symmetric in its arguments, i.e., $g(\boldsymbol{x}, \boldsymbol{y}) = g(\boldsymbol{y}, \boldsymbol{x})$. We define a function for a two-sample U-statistic as $h(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y}_1, \boldsymbol{y}_2)$, which is composed of evaluations of g over the variables $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y}_1$ and $\boldsymbol{y}_2$, and is symmetric, i.e., $h(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y}_1, \boldsymbol{y}_2) = h(\boldsymbol{x}_2, \boldsymbol{x}_1; \boldsymbol{y}_2, \boldsymbol{y}_1)$. Suppose we have two samples $X = \{\boldsymbol{x}_i\}_{i=1}^n \sim \mathbb{P}^n$ and $Y = \{\boldsymbol{y}_j\}_{j=1}^m \sim \mathbb{Q}^m$. The two-sample U-statistic is given by:*

$$U_{n,m}(X, Y; \phi) = \binom{n}{2}^{-1} \binom{m}{2}^{-1} \sum_{i<j}^{n} \sum_{k<l}^{m} h(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{y}_k, \boldsymbol{y}_l) .$$

Next, we present a standard large-deviation bound for bounded U-statistics.

Theorem 6. *[48] If the function h is bounded, that is, $\|h\|_\infty \triangleq \sup |h(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y}_1, \boldsymbol{y}_2)| < \infty$, and assume $n \leq m$, then for any $t > 0$,*

$$\Pr(|U_{n,m}(X, Y; \phi) - \theta| \geq t) \leq 2 \exp\left(-\lfloor n/2 \rfloor t^2 / (2\|h\|_\infty^2)\right) ,$$

where $\theta = E[h(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y}_1, \boldsymbol{y}_2)]$.

We now introduce the notation for permutation-based statistics. Let $\boldsymbol{\Pi}_{n+m}$ denote the set of all possible permutations of $\{1, \ldots, n, n+1, \ldots, n+m\}$ over the pooled sample $Z = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n, \boldsymbol{z}_{n+1}, \ldots, \boldsymbol{z}_{n+m}\}$. Given a permutation $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n, \pi_{n+1}, \ldots, \pi_{n+m}) \in \boldsymbol{\Pi}_{n+m}$, define $X_{\boldsymbol{\pi}} = \{\boldsymbol{z}_{\pi_i}\}_{i=1}^n$ and $Y_{\boldsymbol{\pi}} = \{\boldsymbol{z}_{\pi_i}\}_{i=n+1}^{n+m}$.

We next state a useful concentration inequality for permuted U-statistics, which will be repeatedly used in the proofs below.

Theorem 7. *[49, Theorem 6.1] Consider the permuted two-sample U-statistic $U_{n,m}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \phi)$ with $h(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y}_1, \boldsymbol{y}_2) = g(\boldsymbol{x}_1, \boldsymbol{x}_2) + g(\boldsymbol{y}_1, \boldsymbol{y}_2) - g(\boldsymbol{x}_1, \boldsymbol{y}_2) - g(\boldsymbol{x}_2, \boldsymbol{y}_1)$ in Definition 5, where $\boldsymbol{\pi}$ is uniformly distributed on $\boldsymbol{\Pi}_{n+m}$, and assume $n \leq m$. Define*

$$\Gamma_{n,m}^2 := \frac{1}{n^2 (n-1)^2} \sup_{\boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}} \left\{ \sum_{i<j}^{n} g^2 \left(\boldsymbol{z}_{\pi_i}, \boldsymbol{z}_{\pi_j}\right) \right\} .$$

Then, for every $t > 0$ and some constant $C > 0$, we have

$$\Pr\left(U_{n,m}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \phi) \geq t \mid Z\right) \leq \exp\left(-C \min\left(\frac{t^2}{\Gamma_{n,m}^2}, \frac{t}{\Gamma_{n,m}}\right)\right) .$$

E.1 Detailed Proofs of Theorem 1

We begin with a useful definition as follows.

Definition 8. *[65] Let Z be the pooled sample, and let $\boldsymbol{\Pi}_{n+m}$ be the finite set of permutations $\boldsymbol{\pi}$ acting on Z, such that $\boldsymbol{\Pi}_{n+m}$ is a group with respect to the operation of composition of transformations. Let $\boldsymbol{H}_0$ be any null hypothesis which implies that the joint distribution of the test statistics $\mathrm{STAT}(\boldsymbol{\pi} Z; \phi)$, $\boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}$, is invariant under all transformations in $\boldsymbol{\Pi}_{n+m}$ of Z. We have, under $\boldsymbol{H}_0$,*

$$\left(\mathrm{STAT}(\boldsymbol{\pi}' Z; \phi)\right)_{\boldsymbol{\pi}' \in \boldsymbol{\Pi}_{n+m}} \overset{d}{=} \left(\mathrm{STAT}(\boldsymbol{\pi} \cdot \boldsymbol{\pi}' Z; \phi)\right)_{\boldsymbol{\pi}' \in \boldsymbol{\Pi}_{n+m}} \quad \textit{for all } \boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m} ,$$

where $\overset{d}{=}$ denotes equality in distribution.

We present the proofs of Theorem 1 as follows

Proof. It is easy to see that the permutation set $\boldsymbol{\Pi}_{n+m}$ forms a group under the operation of composition of transformations. Moreover, when the test batch Y consists of clean data, X and Y are drawn from the same distribution, hence Z is exchangeable w.r.t. $\boldsymbol{\Pi}_{n+m}$ and the joint law of $\{\mathrm{STAT}(\boldsymbol{\pi} Z; \phi) : \boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}\}$ is invariant under permutation by any $\boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}$. Therefore Definition 8 applies. For USAD, $\widehat{\Sigma}_{\mathrm{A}}$ is estimated from CEs before the test and kept fixed across permutations (Section 4); as long as it is estimated independently of the tested samples X and Y, the arguments below hold conditionally on $\widehat{\Sigma}_{\mathrm{A}}$, under which $\mathcal{T}^{\mathrm{A}}(\cdot, \cdot; \ell, \rho)$ is a fixed function of the two samples. Given the pooled sample Z, we write the multiset

$$\{\mathrm{STAT}(\boldsymbol{\pi} Z; \phi) : \boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}\}$$

of $(n + m)!$ statistics, and note that the identity permutation is included, so $\mathrm{STAT}(Z; \phi) = \mathrm{STAT}(X, Y; \phi)$ is one element of this multiset. By Definition 8, for a uniformly distributed $\boldsymbol{\pi} \sim \mathrm{Unif}(\boldsymbol{\Pi}_{n+m})$ independent of Z,

$$\left(\mathrm{STAT}(\boldsymbol{\pi}' Z; \phi)\right)_{\boldsymbol{\pi}' \in \boldsymbol{\Pi}_{n+m}} \overset{d}{=} \left(\mathrm{STAT}(\boldsymbol{\pi} \cdot \boldsymbol{\pi}' Z; \phi)\right)_{\boldsymbol{\pi}' \in \boldsymbol{\Pi}_{n+m}}.$$

Hence, the rank of $\mathrm{STAT}(Z; \phi)$ in this multiset has the same distribution as the rank of $\mathrm{STAT}(\boldsymbol{\pi} Z; \phi)$ in it, which is uniform on $\{1, \ldots, (n + m)!\}$ if there are no ties; with ties, using the conservative convention (counting ties as exceedances) increases the right tail exceedance count, which makes rejection less likely, keeping the type I error conservative. In practical implementation (see Appendix C), we approximate the permutation distribution by $R < (n + m)!$ random permutations $\boldsymbol{\pi}_{(1)}, \ldots, \boldsymbol{\pi}_{(R)} \overset{i.i.d.}{\sim} \mathrm{Unif}(\boldsymbol{\Pi}_{n+m})$, drawn independently of Z, and use the testing threshold $t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)$ in Appendix C. By the same exchangeability argument, the $R + 1$ statistics $\mathrm{STAT}(X, Y; \phi), \mathrm{STAT}(\boldsymbol{\pi}_{(1)} Z; \phi), \ldots, \mathrm{STAT}(\boldsymbol{\pi}_{(R)} Z; \phi)$ are exchangeable under $\boldsymbol{H}_0$ [65], so the rank of $\mathrm{STAT}(X, Y; \phi)$ among them is uniform on $\{1, \ldots, R+1\}$ (conservative under ties). By the definition of $t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)$, $\mathrm{STAT}(X, Y; \phi) > t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)$ holds only if at least $\lceil (1-\alpha)(R+1) \rceil$ of the R permuted statistics are smaller than $\mathrm{STAT}(X, Y; \phi)$, i.e., only if its rank among the $R + 1$ statistics exceeds $\lceil (1 - \alpha)(R + 1) \rceil$. Hence

$$\Pr(\delta_\alpha = 1) \equiv \Pr(\mathrm{STAT}(X, Y; \phi) > t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)) \leq \frac{R + 1 - \lceil (1 - \alpha)(R + 1) \rceil}{R + 1} \leq \alpha.$$

□

E.2 Detailed Proofs of Theorem 2

We present the proofs of Theorem 2 as follows

Proof. Recall the definition of VD as

$$\mathrm{VD}(\mathbb{P},\mathbb{Q};\ell) = \left(V_{\mathbb{P}} - V_{\mathbb{Q}}\right)^2 .$$

The corresponding estimator is

$$\widehat{\mathrm{VD}}(X,Y;\ell) \quad = \quad \left(\widehat{V}(X;\ell) - \widehat{V}(Y;\ell)\right)^2 ,$$

where

$$\widehat{V}(X;\ell) = \sum_{i}^{n} \frac{\ell(f(\boldsymbol{x}_i), f(\boldsymbol{x}_i))}{n} - \sum_{i\neq j}^{n} \frac{\ell(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j))}{n(n-1)} ,$$
$$\widehat{V}(Y;\ell) = \sum_{i}^{m} \frac{\ell(f(\boldsymbol{y}_i), f(\boldsymbol{y}_i))}{m} - \sum_{i\neq j}^{m} \frac{\ell(f(\boldsymbol{y}_i), f(\boldsymbol{y}_j))}{m(m-1)} .$$

These two estimators can be equivalently written as second-order U-statistics:

$$\widehat{V}(X;\ell) = \binom{n}{2}^{-1} \sum_{i<j}^{n} \frac{\ell(f(\boldsymbol{x}_i), f(\boldsymbol{x}_i)) + \ell(f(\boldsymbol{x}_j), f(\boldsymbol{x}_j))}{2} - \ell(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) ,$$
$$\widehat{V}(Y;\ell) = \binom{m}{2}^{-1} \sum_{i<j}^{m} \frac{\ell(f(\boldsymbol{y}_i), f(\boldsymbol{y}_i)) + \ell(f(\boldsymbol{y}_j), f(\boldsymbol{y}_j))}{2} - \ell(f(\boldsymbol{y}_i), f(\boldsymbol{y}_j)) .$$

Assume ℓ is bounded on the range of f, i.e., $\|\ell\|_\infty \triangleq \sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{X}} |\ell(f(\boldsymbol{x}), f(\boldsymbol{y}))| < \infty$. Then $|V_{\mathbb{P}}|, |V_{\mathbb{Q}}|, |\widehat{V}(X;\ell)|, |\widehat{V}(Y;\ell)| \leq 2\|\ell\|_\infty$, and hence

$$\begin{aligned}
&\left|\mathrm{VD}(\mathbb{P},\mathbb{Q};\ell) - \widehat{\mathrm{VD}}(X,Y;\ell)\right| \\
&= \left|(V_{\mathbb{P}} - V_{\mathbb{Q}})^2 - \left(\widehat{V}(X;\ell) - \widehat{V}(Y;\ell)\right)^2\right| \\
&= \left(\widehat{V}(X;\ell) - \widehat{V}(Y;\ell) - (V_{\mathbb{P}} - V_{\mathbb{Q}})\right)\left(\widehat{V}(X;\ell) - \widehat{V}(Y;\ell) + (V_{\mathbb{P}} - V_{\mathbb{Q}})\right) \\
&\leq \left|\widehat{V}(X;\ell) - \widehat{V}(Y;\ell) - (V_{\mathbb{P}} - V_{\mathbb{Q}})\right| \cdot \left|\widehat{V}(X;\ell) - \widehat{V}(Y;\ell) + (V_{\mathbb{P}} - V_{\mathbb{Q}})\right| \\
&\leq 8\|\ell\|_\infty \cdot \left|\widehat{V}(X;\ell) - \widehat{V}(Y;\ell) - (V_{\mathbb{P}} - V_{\mathbb{Q}})\right| . \qquad (13)
\end{aligned}$$

Here, the estimator of $V_{\mathbb{P}} - V_{\mathbb{Q}}$ can be further presented as a two-sample U-statistic as in Definition 5

$$\widehat{V}(X;\ell) - \widehat{V}(Y;\ell) = \binom{n}{2}^{-1}\binom{m}{2}^{-1} \sum_{i<j}^{n}\sum_{k<l}^{m} h_{\mathrm{VD}}(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{y}_k, \boldsymbol{y}_l) ,$$

where

$$\begin{aligned}
h_{\mathrm{VD}}(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{y}_k, \boldsymbol{y}_l) \quad = \quad & \frac{\ell(f(\boldsymbol{x}_i), f(\boldsymbol{x}_i)) + \ell(f(\boldsymbol{x}_j), f(\boldsymbol{x}_j))}{2} - \ell(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) \\
& - \frac{\ell(f(\boldsymbol{y}_k), f(\boldsymbol{y}_k)) + \ell(f(\boldsymbol{y}_l), f(\boldsymbol{y}_l))}{2} + \ell(f(\boldsymbol{y}_k), f(\boldsymbol{y}_l)) .
\end{aligned}$$

Under the alternative hypothesis $\boldsymbol{H}_1^{\mathrm{V}} : V_{\mathbb{P}} \neq V_{\mathbb{Q}}$, we have that

$$\widehat{V}(X;\ell) - \widehat{V}(Y;\ell) \xrightarrow{P} V_{\mathbb{P}} - V_{\mathbb{Q}} \neq 0 ,$$

by Theorem 6 (note that $\|h_{\mathrm{VD}}\|_\infty \leq 4\|\ell\|_\infty$), where $\xrightarrow{P}$ denotes convergence in probability, and hence

$$\widehat{\mathrm{VD}}(X,Y;\ell) \equiv (\widehat{V}(X;\ell) - \widehat{V}(Y;\ell))^2 \xrightarrow{P} (V_{\mathbb{P}} - V_{\mathbb{Q}})^2 > 0 ,$$

by Eq. (13).

Under a uniformly random permutation $\boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}$, conditionally on Z, $X_{\boldsymbol{\pi}}$ is a sample of size n drawn without replacement from Z, and $Y_{\boldsymbol{\pi}}$ consists of the remaining m examples.

Writing $g(\boldsymbol{z}_i, \boldsymbol{z}_j) = \frac{\ell(f(\boldsymbol{z}_i), f(\boldsymbol{z}_i)) + \ell(f(\boldsymbol{z}_j), f(\boldsymbol{z}_j))}{2} - \ell(f(\boldsymbol{z}_i), f(\boldsymbol{z}_j))$, so that $h_{\mathrm{VD}}(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{y}_k, \boldsymbol{y}_l) = g(\boldsymbol{x}_i, \boldsymbol{x}_j) - g(\boldsymbol{y}_k, \boldsymbol{y}_l)$ and $|g| \leq 2\|\ell\|_\infty$, both $\widehat{V}(X_{\boldsymbol{\pi}}; \ell)$ and $\widehat{V}(Y_{\boldsymbol{\pi}}; \ell)$ are averages of g over the pairs of examples within $X_{\boldsymbol{\pi}}$ and within $Y_{\boldsymbol{\pi}}$, respectively, and thus have the same conditional mean $\binom{n+m}{2}^{-1} \sum_{i<j}^{n+m} g(\boldsymbol{z}_i, \boldsymbol{z}_j)$. Moreover, among the pairs of summands of $\widehat{V}(X_{\boldsymbol{\pi}}; \ell)$, the $O(n^3)$ pairs that share an example have covariances bounded by $4\|\ell\|_\infty^2$, whereas the remaining pairs involve four distinct examples drawn without replacement from Z and have covariances of order $O(\|\ell\|_\infty^2/(n+m))$; hence the conditional variance of $\widehat{V}(X_{\boldsymbol{\pi}}; \ell)$ is $O(\|\ell\|_\infty^2/n)$, and likewise that of $\widehat{V}(Y_{\boldsymbol{\pi}}; \ell)$ is $O(\|\ell\|_\infty^2/m)$. By Chebyshev's inequality, the two-sample U-statistic with permuted samples $X_{\boldsymbol{\pi}}$ and $Y_{\boldsymbol{\pi}}$ exhibits the convergence that

$$\widehat{V}(X_{\boldsymbol{\pi}}; \ell) - \widehat{V}(Y_{\boldsymbol{\pi}}; \ell) \xrightarrow{P} 0 \,,$$

as $n, m \to \infty$ with $n/m \to c \in (0, \infty)$. Consequently, the permutation distribution of $\widehat{V}(X_{\boldsymbol{\pi}}; \ell) - \widehat{V}(Y_{\boldsymbol{\pi}}; \ell)$ concentrates at zero and

$$\widehat{\mathrm{VD}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell) = (\widehat{V}(X_{\boldsymbol{\pi}}; \ell) - \widehat{V}(Y_{\boldsymbol{\pi}}; \ell))^2 \xrightarrow{P} 0 \,.$$

Since R is fixed and $\alpha(R+1) \geq 1$, the testing threshold $t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)$ (see Appendix C with $\mathrm{STAT}(X, Y; \phi) = \widehat{\mathrm{VD}}(X, Y; \ell)$) lies between the smallest and the largest of $\widehat{\mathrm{VD}}(X_{(r)}, Y_{(r)}; \ell)$, $r = 1, \ldots, R$, and hence also converges to 0 in probability, which implies, by Slutsky's theorem [66],

$$\Pr(\delta_\alpha = 1) \equiv \Pr(\widehat{\mathrm{VD}}(X, Y; \ell) > t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)) \longrightarrow 1 \,,$$

where the test δ_α is the permutation procedure (see Eq. (11) in Appendix C with $\mathrm{STAT}(X, Y; \phi) = \widehat{\mathrm{VD}}(X, Y; \ell)$). □

E.3 Detailed Proofs of Theorem 3

We present the proofs of Theorem 3 as follows

Proof. Recall the definition of PCD as

$$\mathrm{PCD}(\mathbb{P}, \mathbb{Q}, \rho) = \|\boldsymbol{\mu}_{\Sigma_\mathbb{P}} - \boldsymbol{\mu}_{\Sigma_\mathbb{Q}}\|_{\mathcal{H}_\rho}^2 = E[\rho(\Sigma_{\boldsymbol{x}}, \Sigma_{\boldsymbol{x}'}) + \rho(\Sigma_{\boldsymbol{y}}, \Sigma_{\boldsymbol{y}'}) - 2\rho(\Sigma_{\boldsymbol{x}}, \Sigma_{\boldsymbol{y}})] \,.$$

The corresponding estimator is

$$\widehat{\mathrm{PCD}}(X, Y; \rho) = \sum_{i \neq j}^{n} \frac{\rho(\Sigma_{\boldsymbol{x}_i}, \Sigma_{\boldsymbol{x}_j})}{n(n-1)} - \sum_{i}^{n} \sum_{j}^{m} \frac{2\rho(\Sigma_{\boldsymbol{x}_i}, \Sigma_{\boldsymbol{y}_j})}{nm} + \sum_{i \neq j}^{m} \frac{\rho(\Sigma_{\boldsymbol{y}_i}, \Sigma_{\boldsymbol{y}_j})}{m(m-1)} \,.$$

which can be further presented as a two-sample U-statistic as in Definition 5

$$\widehat{\mathrm{PCD}}(X, Y; \rho) = \binom{n}{2}^{-1} \binom{m}{2}^{-1} \sum_{i<j}^{n} \sum_{k<l}^{m} h_{\mathrm{PCD}}(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{y}_k, \boldsymbol{y}_l) \,,$$

where

$$h_{\mathrm{PCD}}(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{y}_k, \boldsymbol{y}_l) = \rho(\Sigma_{\boldsymbol{x}_i}, \Sigma_{\boldsymbol{x}_j}) + \rho(\Sigma_{\boldsymbol{y}_k}, \Sigma_{\boldsymbol{y}_l}) - \rho(\Sigma_{\boldsymbol{x}_i}, \Sigma_{\boldsymbol{y}_l}) - \rho(\Sigma_{\boldsymbol{x}_j}, \Sigma_{\boldsymbol{y}_k}) \,.$$

Under the alternative hypothesis $\boldsymbol{H}_1^{\mathrm{C}} : \boldsymbol{\mu}_{\Sigma_\mathbb{P}} \neq \boldsymbol{\mu}_{\Sigma_\mathbb{Q}}$, we have that

$$\widehat{\mathrm{PCD}}(X, Y; \rho) \xrightarrow{P} \mathrm{PCD}(\mathbb{P}, \mathbb{Q}, \rho) > 0 \,,$$

by Theorem 6, since $\|h_{\mathrm{PCD}}\|_\infty \leq 4\|\rho\|_\infty$ with $\|\rho\|_\infty \triangleq \sup|\rho(\cdot, \cdot)| < \infty$.

Under a uniformly random permutation $\boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}$, the two-sample U-statistic with permuted samples $X_{\boldsymbol{\pi}}$ and $Y_{\boldsymbol{\pi}}$ exhibits the convergence that

$$\widehat{\mathrm{PCD}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \rho) \xrightarrow{P} 0 \,,$$

based on Theorem 7 as $n, m \to \infty$ with $n/m \to c \in (0, \infty)$: h_{PCD} has the form required there with $g(\boldsymbol{z}_i, \boldsymbol{z}_j) = \rho(\Sigma_{\boldsymbol{z}_i}, \Sigma_{\boldsymbol{z}_j})$, $\Gamma_{n,m}^2 \leq \|\rho\|_\infty^2/(2n(n-1)) \to 0$ (assuming $n \leq m$ without loss

of generality, since $\widehat{\text{PCD}}$ is symmetric in its two samples), and applying Theorem 7 to both ρ and $-\rho$ controls both tails. Consequently, since R is fixed and $\alpha(R+1) \geq 1$, the testing threshold $t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)$ (see Appendix C with $\text{STAT}(X, Y; \phi) = \widehat{\text{PCD}}(X, Y; \rho)$), which lies between the smallest and the largest of $\widehat{\text{PCD}}(X_{(r)}, Y_{(r)}; \rho)$, $r = 1, \ldots, R$, also converges to 0 in probability. This implies, by Slutsky's theorem [66],

$$\Pr(\delta_\alpha = 1) \equiv \Pr(\widehat{\text{PCD}}(X, Y; \rho) > t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)) \longrightarrow 1 ,$$

where the test δ_α is the permutation procedure (see Eq. (11) in Appendix C with $\text{STAT}(X, Y; \phi) = \widehat{\text{PCD}}(X, Y; \rho)$). □

E.4 Detailed Proofs of Corollary 4

We present the proofs of Corollary 4 as follows

Proof. As shown in proofs of Theorems 2 and 3, under a uniformly random permutation $\boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}$, we have

$$\widehat{\text{VD}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell) = (\widehat{V}(X_{\boldsymbol{\pi}}; \ell) - \widehat{V}(Y_{\boldsymbol{\pi}}; \ell))^2 \xrightarrow{P} 0 ,$$

and

$$\widehat{\text{PCD}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \rho) \xrightarrow{P} 0 ,$$

which imply that

$$\mathcal{T}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell, \rho) \equiv (\widehat{\text{VD}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell), \widehat{\text{PCD}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \rho))^\top \xrightarrow{P} (0, 0)^\top$$

by Slutsky's theorem [66].

Then, we have

$$\mathcal{T}^{\text{A}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell, \rho) = \mathcal{T}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell, \rho)^\top \widehat{\Sigma}_{\text{A}}^{-1} \mathcal{T}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell, \rho) \xrightarrow{P} 0 ,$$

where $\widehat{\Sigma}_{\text{A}}$ is estimated once and kept fixed across permutations (Section 4). Since rescaling $\widehat{\Sigma}_{\text{A}}$ by a positive constant rescales the observed and all permuted USAD statistics by the same factor, the test δ_α does not depend on the scale of $\widehat{\Sigma}_{\text{A}}$, and we normalize $\widehat{\Sigma}_{\text{A}}$ to have unit trace without loss of generality. As the condition number of $\widehat{\Sigma}_{\text{A}}$ remains bounded (Corollary 4), the eigenvalues of $\widehat{\Sigma}_{\text{A}}^{-1}$ are then at least 1 and bounded from above, which gives the convergence above.

Consequently, the permutation distribution of $\mathcal{T}^{\text{A}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell, \rho)$ concentrates at zero, and the testing threshold $t_\alpha\big(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R\big)$ (see Appendix C, with $\text{STAT}(X, Y; \phi) = \mathcal{T}^{\text{A}}(X, Y; \ell, \rho)$), which is at most $\max_{1 \leq r \leq R} \mathcal{T}^{\text{A}}(X_{(r)}, Y_{(r)}; \ell, \rho)$, converges to 0 in probability. Similarly, if **either or both** of the alternatives $\boldsymbol{H}_1^{\text{V}}$ and $\boldsymbol{H}_1^{\text{C}}$ hold, then at least one of VD or PCD converges in probability to a positive constant, and thus $\mathcal{T}(X, Y; \ell, \rho)$ converges in probability to a nonzero vector. It follows that the observed USAD value $\mathcal{T}^{\text{A}}(X, Y; \ell, \rho) \geq \|\mathcal{T}(X, Y; \ell, \rho)\|_2^2$ is bounded away from zero with probability tending to one.

Therefore, by Slutsky's theorem [66],

$$\Pr(\delta_\alpha = 1) \equiv \Pr(\mathcal{T}^{\text{A}}(X, Y; \ell, \rho) > t_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)) \longrightarrow 1 ,$$

where the test δ_α is the permutation procedure (see Eq. (11) in Appendix C with $\text{STAT}(X, Y; \phi) = \mathcal{T}^{\text{A}}(X, Y; \ell, \rho)$). □

F Experimental Details

F.1 Aggregation Strategy

DUAL [31] can be extended to aggregate heterogeneous statistics because it operates on their joint correlation structure, without requiring the components to share the same U-statistic form, scaling, or null distribution. In contrast, MMDAgg [53] relies on all components having identical U-statistic structures and comparable bootstrap quantiles, making it unsuitable for mixing statistics with different

functional forms or null behaviors. MMD-FUSE [54] restricts aggregation by requiring all inputs to follow the same kernel-based MMD formulation, so that its log-sum-exp fusion remains theoretically calibrated; as a result, it cannot accommodate statistics that differ in scale or structural definition. By standardizing each statistic through its empirical covariance under the null and combining them via a correlation-aware quadratic form, DUAL enables principled fusion of heterogeneous statistics that capture complementary aspects of distributional discrepancy.

F.2 Inference Time Complexity

For clarity, we first introduce the notations used in the time–complexity analysis. Let n denote the batch size of clean examples, i.e., $X = \{\boldsymbol{x}_i\}_{i=1}^n$, and let m denote the batch size of test examples, i.e., $Y = \{\boldsymbol{y}_j\}_{j=1}^m$. We use K to denote the number of random perturbations used in PCD, and q to denote the dimension of the semantic feature, as in the main text. We denote by C_ℓ and C_ρ the computational cost of evaluating the kernel functions ℓ and ρ once, respectively, and by C_f the cost of extracting semantic features from the target classifier f. We denote by B the number of bootstrap resampling repeats used to estimate the correlation $\widehat{\Sigma}_{\mathrm{A}}$, and by R the number of repeats used in the permutation test.

Time Complexity of VD Estimator. For the clean batch X, evaluating the kernel over all diagonal and off-diagonal pairs of semantic features requires $\mathcal{O}(n^2)$ operations. The test batch Y contributes an additional $\mathcal{O}(m^2)$ operations. With a per-evaluation cost of C_ℓ, the total kernel computation cost is $\mathcal{O}((n^2+m^2)C_\ell)$. Extracting semantic features from the classifier f adds $\mathcal{O}((n+m)C_f)$, which is typically negligible compared with the quadratic kernel terms. Overall, the time complexity of VD is

$$\mathcal{O}\big((n^2+m^2)C_\ell + (n+m)C_f\big).$$

Time Complexity of PCD Estimator. For two batches X and Y, each with K random perturbations per example, extracting semantic features through the classifier f requires $\mathcal{O}((n+m)KC_f)$. Computing the covariance matrix for each example involves K outer-product operations in a q-dimensional feature space, resulting in a total cost of $\mathcal{O}((n+m)Kq^2)$. After obtaining the covariance matrix for every example, the empirical PCD estimator in Eqn. (8) requires evaluating the kernel ρ over all pairs of covariances within and across the two batches. This involves $n(n-1)$, $m(m-1)$, and nm kernel computations, yielding a total complexity of $\mathcal{O}((n+m)^2C_\rho)$. Overall, the time complexity of PCD is

$$\mathcal{O}\big((n+m)(KC_f+Kq^2) + (n+m)^2C_\rho\big).$$

Time Complexity of USAD Estimator. For estimating the correlation $\widehat{\Sigma}_{\mathrm{A}}$ (i.e., Eqn. (10)) from B bootstrap replicates incurs an offline cost of

$$\mathcal{O}\left(B\left((n^2+m^2)C_\ell + (n+m)^2C_\rho + (n+m)(KC_f+Kq^2)\right)\right),$$

which is dominated by computing VD and PCD B times. Notably, this is a one-off preprocessing cost: once $\widehat{\Sigma}_{\mathrm{A}}$ is estimated, it is fixed and reused in all subsequent permutation tests.

Given the precomputed $\widehat{\Sigma}_{\mathrm{A}}$, evaluating the aggregated test statistic requires computing VD and PCD only once for the test batch Y. Therefore, the inference-time complexity of USAD is

$$\mathcal{O}\left((n^2+m^2)C_\ell + (n+m)^2C_\rho + (n+m)(KC_f+Kq^2)\right),$$

which corresponds exactly to the cost of evaluating VD and PCD on the observed data.

Time Complexity of Permutation Test. During the permutation test, the kernel matrices of ℓ and ρ over the pooled examples $X \cup Y$ are computed only once: for PCD, they are already computed when evaluating the statistic on the original batches, while for VD, the permuted statistics additionally require the nm cross-batch kernel values $\ell(f(\boldsymbol{x}_i), f(\boldsymbol{y}_j))$, an extra one-off cost of $\mathcal{O}(nmC_\ell)$. For each permutation, we simply rearrange the precomputed kernel matrices according to the permuted indices, and then aggregate the corresponding entries to obtain the permuted VD and PCD values. No kernel evaluations or covariance computations are repeated. Since each permutation only involves indexing and summation over existing matrices, the per-permutation cost is linear in the number of matrix entries.

Therefore, for R permutations, the time complexity of the permutation test is $\mathcal{O}\big(R(n^2+m^2)\big)$ for VD, $\mathcal{O}\big(R(n+m)^2\big)$ for PCD, and $\mathcal{O}\big(R(n+m)^2\big)$ for USAD, which is dominated by the cost of repeatedly aggregating entries from the precomputed kernel and covariance matrices.

In addition to the complexity analysis, we also report the actual elapsed time in Section G.5.

F.3 Implementation Details.

Computational Resources. We implement all methods using Python 3.9.18 with PyTorch 2.7 and CUDA 12.4 on two platforms. One platform is an NVIDIA RTX 4090 GPU PC with PyTorch framework. Another platform is a High-performance Computer cluster with several NVIDIA A100 GPUs with Pytorch framework. The memory of two platforms are both 64 GB. The storage of disk of two platforms are both over 4 TB. The *CIFAR-10* dataset and *ImageNet-1K* dataset can be downloaded via torchvision or from their official websites. Since our proposed methods do not require training, we directly utilize the full CIFAR-10 test set (10,000 images) and the ImageNet-1K validation set (50,000 images) for evaluation. We employ pre-trained ResNet-18, WideResNet-28, WideResNet-70, ResNet-50, and ViT-B-16 models with weights obtained from torchvision or downloaded checkpoints (provided in our GitHub repository; see the link in the abstract) for feature extraction and classification.

Experimental Setup for Datasets. We employ two primary datasets: *CIFAR-10* [32] and *ImageNet-1K* [33]. For clean examples, we independently and randomly sample reference examples from each dataset. For adversarial examples in CIFAR-10, we apply each adversarial attack to the entire CIFAR-10 test set (10,000 images) to generate 10,000 adversarial examples. For ImageNet-1K, following Gao et al. [7], we independently and randomly select 500 clean examples from the full ImageNet-1K validation set (50,000 images), then apply each adversarial attack to these selected images to generate 500 adversarial examples. For statistical power experiments, we independently and randomly select $|Y| = 50$ adversarial examples for testing (except when explicitly varying the number of examples). To assess type-I error rates, we partition both the CIFAR-10 test set and ImageNet-1K validation set into two equal subsets. We designate one subset as the reference set and the other as the input set, then independently and randomly select equal numbers of examples from both subsets to evaluate whether detection methods incorrectly classify clean examples as adversarial.

Experimental Setup for Baselines. For the single-statistic SAD method SAMMD, we adopt all default experimental settings from Gao et al. [7]. For the aggregated-statistic two-sample testing methods MMDAgg, MMD-FUSE, and MMD-DUAL, we follow the default experimental settings from Zhou et al. [31], with one modification: we set the number of aggregations to 2 to ensure a fair comparison with our method. While our method can be extended to kernel aggregation, such extensions are beyond the scope of this work on adversarial detection.

Experimental Setup for Our Methods. For PCD, we first apply principal component analysis (PCA) for dimensionality reduction, where the target dimension is automatically determined based on the original feature size. We set the Gaussian perturbation mean to 0 and standard deviation to $1/255$, with the number of perturbations fixed at 200 across all experiments. For both PCD and VD, we set the bandwidth parameters as $b_{\text{PCD}} = b_{\text{VD}} = 4\sqrt{2}$. For USAD, we use bandwidth $b_{\text{USAD}} = 13\sqrt{2}$. In the permutation testing procedure, we set the significance level at $\alpha = 0.05$ and the number of permutations to 100. All reported results are computed over 10 independent experimental trials with different random seeds, where each trial conducts 100 sampling iterations to estimate the mean test power. Thus, every result is averaged over $1{,}000$ repetitions in total. A sensitivity analysis with respect to the number of perturbations K and the kernel bandwidth is reported in Appendix G.13.

Illustrations of Kernel Types. For VD, we use the Gaussian kernel

$$\ell(x, y) = \exp\left(-\frac{\|x - y\|^2}{b^2}\right),$$

where the bandwidth b is selected using the median heuristic [54, 67] computed from the clean feature set.

For PCD, we use the log-RBF kernel applied to covariance matrices,

$$\rho_{\log}(\Sigma_x, \Sigma_y) = \exp\left(-\frac{\|\log \Sigma_x - \log \Sigma_y\|_F^2}{b^2}\right),$$

or the Gaussian kernel defined directly on covariance matrices,

$$\rho_{\text{gau}}(\Sigma_x, \Sigma_y) = \exp\left(-\frac{\|\Sigma_x - \Sigma_y\|_F^2}{b^2}\right),$$

where the bandwidth b is selected using the median heuristic computed from the clean feature set.

Implementation Details of Adversarial Attacks (Including Adaptive Attack). In our experiments, we detect AEs that *successfully* fool the threat model, and all implementation details are designed around this principle. Specifically, we use a diverse set of attack methods under both ℓ_∞-norm and ℓ_2-norm to comprehensively evaluate the detection ability of USAD, including *AutoAttack* (AA) [6], *basic iterative method* (BIM) [34], *Carlini & Wagner* (CW) attack [4], *fast gradient sign method* (FGSM) [2] and *projected gradient descent* (PGD) [3]. For ℓ_∞-norm attacks, following Gao et al. [7], we set the maximum allowed perturbation budget ϵ to range from $1/255$ to $8/255$. For ℓ_2-norm attacks, to ensure successful attacks against the threat model, we increase ϵ over a set of discrete values from $48/255$ to $104/255$, with successive values differing by $8/255$ (i.e., $48/255$, $56/255$, $64/255$, $72/255$, $80/255$, $88/255$, $96/255$ and $104/255$). For both ℓ_∞-norm and ℓ_2-norm attacks, we set the step size to be $\epsilon/5$. For all iterative attacks (i.e., except for FGSM), we set the iteration number to 5.

Adaptive Attack. To evaluate robustness under adaptive attacks, we follow the strategy of Gao et al. [7] and generate adversarial examples on CIFAR-10 and ImageNet using a PGD-based white-box attack, where both the classifier and our detector are fully accessible to the attacker. Given the trained classifier $f : \mathcal{X} \to \mathcal{C}$ and a dataset $D = \{(\boldsymbol{x}_i, c_i)\}_{i=1}^N$ with $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and ground-truth label $c_i \in \mathcal{C}$, the adaptive adversarial examples $\tilde{D} = \{\boldsymbol{x}_i + \boldsymbol{\zeta}_i^\star\}_{i=1}^N$ are produced by solving

$$\{\boldsymbol{\zeta}_i^\star\}_{i=1}^N = \underset{\{\boldsymbol{\zeta}_i\}_{i=1}^N}{\arg\max} \left[\sum_{i=1}^N \mathcal{L}\big(f(\boldsymbol{x}_i + \boldsymbol{\zeta}_i), c_i\big) - \mathrm{STAT}\big(D, \{\boldsymbol{x}_i + \boldsymbol{\zeta}_i\}_{i=1}^N; \phi\big) \right], \quad \text{s.t.} \quad \|\boldsymbol{\zeta}_i\|_\infty \le \epsilon, \quad \forall i \in [N].$$

where $\mathcal{L}$ is the attack loss function (e.g., cross-entropy), and $\mathrm{STAT}(\cdot)$ denotes the detection statistic used by our method, which can be VD, PCD, or USAD with the corresponding kernel function ϕ. The associated kernels $\phi \in \{\ell, \rho, \{\ell, \rho\}\}$ correspond respectively to VD, PCD, and USAD.

G Additional Experiments

This section complements the results reported in Section 6 with thirteen additional groups of experiments. In summary, we (i) evaluate the proposed detector under ℓ_2 norm attacks and compare it against baselines (Section G.1); (ii) report detailed numerical results across perturbation budgets and additional architectures (Section G.2); (iii) evaluate robustness to transfer attacks from surrogate target models (Section G.3); (iv) empirically verify Type-I error control (Section G.4); (v) measure inference-time computational efficiency against competing detectors (Section G.5); (vi) evaluate the proposed detector against sparse ℓ_0 (Sparse-PGD) and ℓ_1 adversarial attacks (Section G.6); (vii) extend our evaluation to adversarial video detection on UCF101 (Section G.7); (viii) ablate the correlation-aware aggregation of USAD and incorporate MMD into the framework (Section G.8); (ix) summarize the arguments from prior work for the two uncertainty factors (Factor 1 and Factor 2) and directly analyze them on synthetic data and real AEs (Section G.9); (x) analyze how the AE fraction affects test power and study low AE fractions in the query batch together with larger batch sizes (Section G.10); (xi) evaluate USAD under a benign distribution shift (Section G.11); (xii) compare VD with MVD and compare with a per-example detector under an adaptive attack (Section G.12); and (xiii) analyze the sensitivity to the kernel bandwidth and the number of perturbations K (Section G.13). Unless stated otherwise, we follow the hypothesis-testing protocol with the main text: significance level $\alpha = 0.05$ and adversarial batch size $|Y| = 50$.

G.1 Detection under L_2 Norm Attacks

The main text (Section 6) primarily reports the detection performance on ℓ_∞-bounded adversarial perturbations, following the previous studies [7]. To check that our detectors are not tied to this particular norm, we also evaluate our method under ℓ_2-bounded attacks.

Figure 4 reports test power for PCD, VD, and USAD, together with the SAMMD baseline, against AA, CW, FGSM, and PGD attacks targeting a ResNet-18 trained on *CIFAR-10*, constrained in ℓ_2 norm. We fix the adversarial batch size to $|Y| = 50$ and vary the perturbation budget ϵ, averaging over 1,000 repetitions.

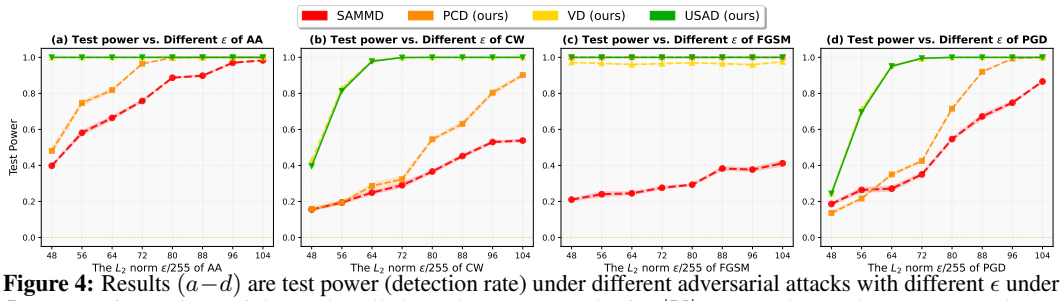


Figure 4: Results $(a-d)$ are test power (detection rate) under different adversarial attacks with different ϵ under L_2 norm, given adversarial samples all share the same sample size $|Y| = 50$. The results are averaged over $1,000$ repetitions and the ideal test power is 1 (the same as 100% detection rate). The target model is ResNet-18 trained on *CIFAR-10* dataset.

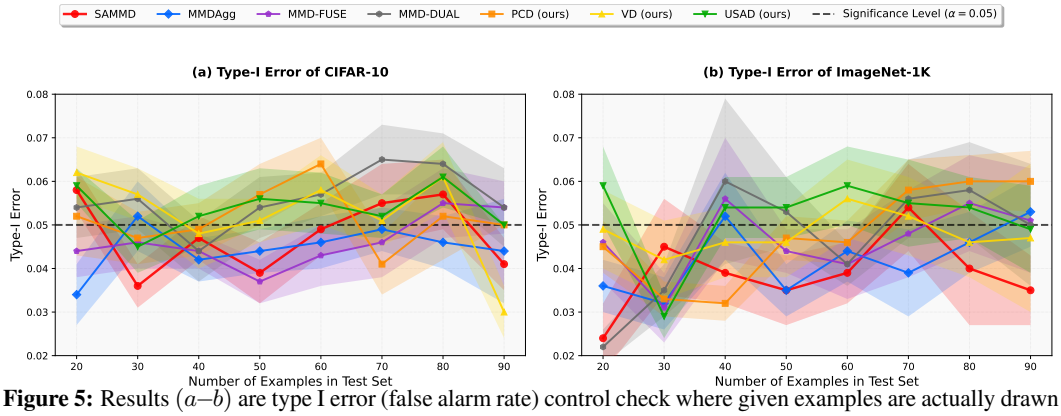


Figure 5: Results $(a–b)$ are type I error (false alarm rate) control check where given examples are actually drawn from clean examples under difference clean example sizes. The results are averaged over $1,000$ repetitions and the ideal type I error is around the significance level $\alpha = 0.05$ (only $\alpha\%$ of the chance to reject clean examples). The target model is ResNet-18 trained on *CIFAR-10* dataset in (a), and the target model is ResNet-50 trained on *ImageNet-1K* dataset in (b).

Across all four attacks, the uncertainty-aware statistics maintain high detection power. USAD and PCD achieve near-perfect detection once $\epsilon \geq 64$, while SAMMD remains noticeably weaker at the same budgets. For AA and FGSM (panels (a) and (c)), USAD reaches test power essentially equal to one already around $\epsilon \approx 48$, substantially outperforming SAMMD at small perturbation levels. For PGD (panel (d)), both USAD and VD exceed 0.95 test power at $\epsilon \approx 64$, whereas SAMMD requires considerably larger perturbations to approach this regime. These results indicate that the statistics underlying USAD capture distributional shifts that manifest under both ℓ_∞- and ℓ_2-bounded attacks.

G.2 Detailed Numerical Results

Tables 4 and 5 present the detailed numerical results corresponding to Figure 3 in the main paper, providing exact test power values with standard deviations for all evaluated methods across different perturbation budgets $\epsilon \in \{1, 2, \ldots, 8\}$. Table 4 corresponds to Figure 3 $(a-d)$ and reports the detection performance of our proposed methods (PCD, VD, and USAD) against the baseline SAMMD on adversarial examples generated against a ResNet-18 model trained on the *CIFAR-10* dataset. Table 5 corresponds to Figure 3 $(e-h)$ and presents results on adversarial examples generated against a ResNet-50 model trained on the *ImageNet-1K* dataset. The superior performance of our methods is even more pronounced on this larger-scale, more complex dataset. While the main paper visualizes a subset of attacks for clarity—specifically AA, CW, FGSM, and PGD for CIFAR-10 (Figure 3 $a-d$), and AA, BIM, CW, and PGD for ImageNet-1K (Figure 3 $e-h$)—these tables include all five attacks (AA, BIM, CW, FGSM, and PGD) for comprehensive evaluation with exact values. These results indicate that our methods maintain their superiority over the baseline, consistent with the observed trends across all other attacks and datasets. The standard deviations (computed over 1,000

repetitions with $|Y| = 50$ adversarial examples per trial) are consistently small, particularly for larger perturbation budgets where test power approaches 1.0, indicating the stability and reliability of our detection methods. The boldface values in the tables highlight the best-performing method(s) for each attack and perturbation budget, clearly demonstrating that our proposed methods achieve state-of-the-art detection performance across the vast majority of experimental configurations.

Table 4: Test power of dataset *CIFAR-10* under five different attacks with different perturbation budget ϵ. The given adversarial examples all share the same number of examples $|Y| = 50$. The results are averaged over $1,000$ repetitions. The target model is ResNet-18 trained on *CIFAR-10* dataset.

Attack	Method	ϵ							
		1	2	3	4	5	6	7	8
AA	SAMMD	0.160±0.012	0.554±0.011	0.871±0.005	0.986±0.003	0.998±0.001	0.996±0.002	**1.000±0.000**	**1.000±0.000**
	PCD (ours)	0.222±0.016	0.752±0.012	0.997±0.001	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.261±0.010	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**0.298**±0.016	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
BIM	SAMMD	0.148±0.010	0.250±0.013	0.577±0.011	0.792±0.011	0.894±0.008	0.926±0.008	0.973±0.005	0.980±0.004
	PCD (ours)	0.125±0.008	0.274±0.011	0.523±0.019	0.971±0.005	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.211±0.015	0.984±0.003	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**0.219**±0.010	**0.985**±0.004	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
CW	SAMMD	0.147±0.011	0.139±0.006	0.300±0.019	0.478±0.013	0.697±0.013	0.849±0.011	0.902±0.009	0.948±0.005
	PCD (ours)	0.234±0.012	0.192±0.013	0.862±0.011	**1.000±0.000**	0.994±0.003	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.507±0.010	0.756±0.010	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**0.528**±0.015	**0.850**±0.010	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
FGSM	SAMMD	0.190±0.012	0.226±0.015	0.408±0.019	0.682±0.009	0.899±0.012	0.971±0.005	0.997±0.002	**1.000±0.000**
	PCD (ours)	0.311±0.014	0.303±0.009	0.800±0.008	0.950±0.003	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.794±0.016	0.835±0.008	0.851±0.011	0.945±0.006	0.992±0.004	0.998±0.001	0.999±0.001	**1.000±0.000**
	USAD (ours)	**0.810**±0.013	**0.842**±0.011	**0.858**±0.011	**0.987**±0.002	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
PGD	SAMMD	0.192±0.012	0.200±0.009	0.381±0.011	0.832±0.009	0.829±0.006	0.933±0.007	0.971±0.006	0.986±0.003
	PCD (ours)	0.251±0.012	0.257±0.015	0.924±0.008	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.653±0.015	0.687±0.018	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**0.663**±0.014	**0.925**±0.004	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**

Table 5: Test power of dataset *ImageNet-1K* under five different attacks with different perturbation budget ϵ. The given adversarial examples all share the same number of examples $|Y| = 50$. The results are averaged over $1,000$ repetitions. The target model is ResNet-50 trained on *ImageNet-1K* dataset.

Attack	Method	ϵ							
		1	2	3	4	5	6	7	8
AA	SAMMD	0.107±0.010	0.310±0.017	0.765±0.028	0.953±0.007	0.998±0.001	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	PCD (ours)	0.569±0.010	0.980±0.004	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.577±0.035	0.984±0.005	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**0.748**±0.019	**0.998**±0.002	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
BIM	SAMMD	0.102±0.012	0.158±0.019	0.305±0.028	0.467±0.029	0.674±0.025	0.820±0.013	0.887±0.016	0.970±0.005
	PCD (ours)	0.189±0.016	0.505±0.014	0.851±0.008	0.953±0.006	0.987±0.003	0.984±0.003	0.998±0.001	0.993±0.002
	VD (ours)	0.194±0.014	0.467±0.040	0.854±0.016	0.940±0.011	0.969±0.011	0.986±0.005	0.995±0.003	0.999±0.001
	USAD (ours)	**0.196**±0.010	**0.634**±0.030	**0.961**±0.008	**0.989**±0.004	**0.998**±0.001	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
CW	SAMMD	0.055±0.007	0.094±0.007	0.262±0.018	0.491±0.028	0.586±0.025	0.775±0.024	0.825±0.015	0.952±0.011
	PCD (ours)	0.206±0.012	0.822±0.009	0.841±0.011	0.989±0.003	0.986±0.003	0.988±0.003	0.990±0.002	0.980±0.003
	VD (ours)	0.596±0.022	0.972±0.007	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**0.640**±0.024	**0.980**±0.003	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
FGSM	SAMMD	0.051±0.012	0.075±0.014	0.187±0.025	0.402±0.033	0.727±0.025	0.940±0.011	0.997±0.002	**1.000**±0.000
	PCD (ours)	0.092±0.008	0.210±0.011	0.525±0.035	0.931±0.011	0.999±0.001	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000
	VD (ours)	0.043±0.005	0.127±0.012	0.254±0.015	0.516±0.021	0.726±0.017	0.924±0.011	0.967±0.006	0.997±0.001
	USAD (ours)	**0.135**±0.014	**0.312**±0.014	**0.620**±0.025	**0.966**±0.008	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000
PGD	SAMMD	0.090±0.015	0.169±0.012	0.274±0.031	0.450±0.027	0.674±0.024	0.875±0.010	0.957±0.008	0.975±0.005
	PCD (ours)	0.152±0.010	0.755±0.017	0.826±0.017	0.939±0.006	0.986±0.002	0.961±0.004	0.992±0.003	0.995±0.002
	VD (ours)	0.074±0.012	0.335±0.020	0.799±0.013	0.919±0.013	0.972±0.007	0.987±0.003	0.997±0.001	**1.000±0.000**
	USAD (ours)	**0.217±0.013**	**0.813±0.016**	**0.933±0.008**	**0.994±0.003**	**1.000±0.000**	**0.999±0.001**	**1.000±0.000**	**1.000±0.000**

G.3 Robustness to Transfer Attacks and Additional Architectures

We next investigate robustness to transfer attacks, where adversarial examples are generated on a surrogate target model but detected using features from a distinct threat classifier, e.g., the AEs are generated with respect to a Swin Transformer, while the threat model is a ResNet-18. This setting reflects a realistic threat model in which the defender is unaware of the attacker's architecture.

Transfer Attacks. Table 8 evaluates the detection performance under transfer attack scenarios, where adversarial examples are generated by attacking different target models (ResNet-18 (Res18), WideResNet-28 (WRN28), WideResNet-70 (WRN70), and Shifted window (Swin) Transformer) but detected using features extracted from Res18. Our proposed methods demonstrate exceptional robustness to transfer attacks across all target models and attack types. Notably, both PCD and

Table 6: Test power of dataset *ImageNet-1K* under five different attacks with different perturbation budget ϵ. The given adversarial examples all share the same number of examples $|Y| = 50$. The results are averaged over $1,000$ repetitions. The target model is ViT-B-16 trained on *ImageNet-1K* dataset.

Attack	Method	ϵ							
		1	2	3	4	5	6	7	8
AA	SAMMD	0.691±0.052	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	PCD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
BIM	SAMMD	0.758±0.110	0.846±0.025	0.973±0.012	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	PCD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.593±0.030	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
CW	SAMMD	0.472±0.066	0.997±0.003	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	PCD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.999±0.001	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
FGSM	SAMMD	0.732±0.065	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	PCD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.533±0.039	0.765±0.040	0.811±0.019	0.766±0.026	0.721±0.033	0.655±0.037	0.704±0.023	0.643±0.019
	USAD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
PGD	SAMMD	0.545±0.074	0.606±0.039	0.819±0.023	0.949±0.014	0.991±0.003	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	PCD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	VD (ours)	0.495±0.042	0.988±0.004	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**
	USAD (ours)	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**	**1.000±0.000**

Table 7: Test power of dataset *CIFAR-10* under different attacks with perturbation budget $\epsilon = 4/255$ across different target models WideResNet-28 (WRN28) or WideResNet-70 (WRN70), when the feature extractor are the same as target model. Both of WRN28 and WRN70 are trained on *CIFAR-10* dataset. The given adversarial examples all share the same number of examples $|Y| = 50$. The results are averaged over $1,000$ repetitions.

Target Model	Detection Method	AA	BIM	CW	FGSM	PGD
WRN28	SAMMD	0.748 ± 0.029	0.287 ± 0.023	0.356 ± 0.013	0.497 ± 0.026	0.367 ± 0.024
	PCD (ours)	0.997 ± 0.002	0.613 ± 0.016	**0.910 ± 0.017**	**0.999 ± 0.001**	**0.937 ± 0.014**
	VD (ours)	**1.000 ± 0.000**	**0.737 ± 0.055**	0.484 ± 0.105	**0.999 ± 0.001**	0.321 ± 0.094
	MMDAgg	0.992 ± 0.002	0.628 ± 0.015	0.610 ± 0.010	0.735 ± 0.015	0.682 ± 0.012
	MMD-FUSE	**1.000 ± 0.000**	0.789 ± 0.013	0.742 ± 0.013	0.967 ± 0.006	0.748 ± 0.012
	MMD-DUAL	**1.000 ± 0.000**	0.523 ± 0.018	0.938 ± 0.007	0.978 ± 0.005	0.882 ± 0.010
	USAD (ours)	**1.000 ± 0.000**	**0.860 ± 0.028**	**0.953 ± 0.011**	**1.000 ± 0.000**	**0.959 ± 0.005**
WRN70	SAMMD	0.943 ± 0.011	0.416 ± 0.012	0.486 ± 0.010	0.500 ± 0.011	0.485 ± 0.020
	PCD (ours)	**1.000 ± 0.000**	0.773 ± 0.016	0.930 ± 0.006	**1.000 ± 0.000**	**0.953 ± 0.007**
	VD (ours)	**1.000 ± 0.000**	**0.938 ± 0.010**	**0.983 ± 0.002**	0.994 ± 0.002	0.844 ± 0.011
	MMDAgg	0.976 ± 0.003	0.577 ± 0.010	0.623 ± 0.012	0.686 ± 0.010	0.634 ± 0.012
	MMD-FUSE	**1.000 ± 0.000**	0.700 ± 0.019	0.778 ± 0.013	0.915 ± 0.011	0.698 ± 0.014
	MMD-DUAL	**1.000 ± 0.000**	0.749 ± 0.016	0.837 ± 0.011	0.857 ± 0.009	0.735 ± 0.016
	USAD (ours)	**1.000 ± 0.000**	**0.956 ± 0.006**	**0.992 ± 0.002**	**1.000 ± 0.000**	**0.982 ± 0.005**

USAD achieve perfect detection (1.000 ± 0.000) across all five attacks (AA, BIM, CW, FGSM, PGD) and all four target models (Res18, WRN28, WRN70, Swin), indicating near-universal detection capability regardless of the architectural characteristics of the model used to generate adversarial examples. VD similarly achieves perfect or near-perfect detection in the vast majority of scenarios, with only minor degradation on CW attacks against WRN28 and WRN70 (0.999 ± 0.001) and BIM attacks against WRN28 (0.993 ± 0.001), which remains substantially superior to baseline methods. In contrast, the baseline SAMMD exhibits significant vulnerability to transfer attacks, with detection rates often dropping below 0.4 (AA on WRN28, WRN70 and SWIN) when it performs 0.995 on Res18 originally. This performance gap suggests that single-statistic method based on maximum mean discrepancy alone may be overly sensitive to the specific characteristics of the attack generation process and fail to capture the fundamental distributional shifts induced by adversarial perturbations across different model architectures. The aggregated-statistic baselines (MMDAgg, MMD-FUSE, MMD-DUAL) show improved robustness compared to SAMMD, which aligns with the theoretical motivation for aggregation—combining multiple test statistics can provide more comprehensive distributional characterization and reduce sensitivity to specific attack characteristics. However, these methods still fall short of our proposed approaches. For example, on WRN28, MMDAgg achieves only 0.459 for AA attacks and 0.245 for BIM attacks, while MMD-FUSE and MMD-DUAL perform better but still exhibit substantial degradation on certain attack-model combinations (e.g., MMD-FUSE: 0.794 for CW on WRN28, MMD-DUAL: 0.764 for BIM on WRN28). Our USAD

Table 8: Test power of dataset *CIFAR-10* under different transfer attacks with perturbation budget $\epsilon = 4/255$ across different target models, while the feature extractor ResNet-18 keeps the same, which is trained on *CIFAR-10* dataset. The given adversarial examples all share the same number of examples $|Y| = 50$. The results are averaged over $1,000$ repetitions.

Target Model	Detection Method	AA	BIM	CW	FGSM	PGD
Res18	SAMMD	0.995 ± 0.002	0.689 ± 0.017	0.434 ± 0.012	0.610 ± 0.011	0.562 ± 0.019
	PCD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
	VD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.975 ± 0.003	$\mathbf{1.000 \pm 0.000}$
	MMDAgg	$\mathbf{1.000 \pm 0.000}$	0.978 ± 0.003	0.788 ± 0.012	0.582 ± 0.021	0.871 ± 0.010
	MMD-FUSE	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.894 ± 0.011	$\mathbf{1.000 \pm 0.000}$
	MMD-DUAL	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.952 ± 0.005	$\mathbf{1.000 \pm 0.000}$
	USAD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
WRN28	SAMMD	0.387 ± 0.018	0.193 ± 0.010	0.227 ± 0.015	0.769 ± 0.010	0.319 ± 0.015
	PCD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
	VD (ours)	$\mathbf{1.000 \pm 0.000}$	0.993 ± 0.001	0.999 ± 0.001	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
	MMDAgg	0.459 ± 0.022	0.245 ± 0.011	0.295 ± 0.017	0.812 ± 0.020	0.522 ± 0.014
	MMD-FUSE	0.844 ± 0.011	0.625 ± 0.007	0.794 ± 0.012	0.992 ± 0.004	0.993 ± 0.002
	MMD-DUAL	0.937 ± 0.006	0.764 ± 0.007	0.900 ± 0.009	0.998 ± 0.002	0.998 ± 0.002
	USAD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
WRN70	SAMMD	0.398 ± 0.009	0.207 ± 0.015	0.227 ± 0.010	0.792 ± 0.015	0.199 ± 0.013
	PCD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
	VD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.999 ± 0.001	$\mathbf{1.000 \pm 0.000}$	0.999 ± 0.001
	MMDAgg	0.524 ± 0.016	0.289 ± 0.010	0.352 ± 0.014	0.764 ± 0.015	0.324 ± 0.015
	MMD-FUSE	0.970 ± 0.004	0.810 ± 0.011	0.847 ± 0.008	0.982 ± 0.005	0.795 ± 0.014
	MMD-DUAL	0.988 ± 0.003	0.912 ± 0.007	0.911 ± 0.009	0.996 ± 0.002	0.882 ± 0.008
	USAD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
SWIN	SAMMD	0.365 ± 0.015	0.306 ± 0.017	0.279 ± 0.016	0.600 ± 0.015	0.403 ± 0.015
	PCD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
	VD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
	MMDAgg	0.630 ± 0.010	0.535 ± 0.021	0.515 ± 0.007	0.730 ± 0.015	0.420 ± 0.010
	MMD-FUSE	0.999 ± 0.001	0.992 ± 0.003	0.987 ± 0.002	$\mathbf{1.000 \pm 0.000}$	0.894 ± 0.010
	MMD-DUAL	$\mathbf{1.000 \pm 0.000}$	0.995 ± 0.002	0.994 ± 0.003	$\mathbf{1.000 \pm 0.000}$	0.946 ± 0.007
	USAD (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$

method, which also employs aggregation but with our novel uncertainty-aware test statistics (i.e., VD and PCD), achieves perfect detection across all scenarios, demonstrating that the choice of base statistics is as critical as the aggregation strategy itself.

Additional Architectures. To demonstrate the broader applicability of our detection framework beyond ResNet architectures, we evaluate our methods on vision transformers and wide residual networks. Table 6 presents results on the ViT-B-16 model trained on ImageNet-1K, where our methods achieve perfect or near-perfect detection across all attacks and perturbation budgets. Notably, all three of our proposed methods (PCD, VD, USAD) attain 1.000 ± 0.000 test power starting from $\epsilon = 1$ for AA, BIM, CW, and PGD attacks, while SAMMD exhibits substantially lower detection rates at small perturbation budgets (e.g., 0.691 for AA at $\epsilon = 1$, 0.758 for BIM at $\epsilon = 1$). The only exception is VD on FGSM attacks, where detection rates remain moderate across different ϵ values, suggesting potential architectural-specific sensitivities of certain base statistics to particular attack types on transformer models. Table 7 further validates our framework on WRN28 and WRN70 architectures trained on CIFAR-10, where adversarial examples are generated by attacking the same model used for feature extraction. Our methods maintain strong performance with USAD achieving detection rates exceeding 0.860 across all attacks on both architectures, while SAMMD struggles with rates often below 0.500. Interestingly, the performance patterns differ across architectures: VD achieves perfect detection on WRN28 for AA attacks but exhibits more variable performance on other attacks, whereas on WRN70, both PCD and VD achieve perfect detection for AA while showing complementary strengths on other attacks. This architectural diversity in the results underscores the importance of aggregation in USAD, which consistently delivers robust detection by combining multiple complementary statistics that capture different aspects of adversarial perturbations across varied model architectures, from convolutional networks to transformers.

G.4 Type-I Error Control Check

Controlling type-I error (false alarm rate) is crucial for practical deployment. Figure 5 shows that all SAD and MMD testing methods maintain Type-I error rates close to the theoretical significance

Table 9: Running Time (seconds) per test trial. We record the total inference time for four methods across 100 trials, then we divide each total time by 100. For PCD and USAD, it is also possible to pre-compute the distance matrix between each covariance matrix generated by CEs offline, which can make the computation inference time at least twice faster than current implementation.

Dataset	MMDAgg	MMD-FUSE	MMD-DUAL	SAMMD	PCD	VD	USAD
CIFAR-10	0.072	0.047	0.047	0.024	0.117	0.016	0.132
ImageNet-1K	0.051	0.044	0.037	0.097	0.759	0.015	0.772

level of $\alpha = 0.05$ across different number of examples in the test set. Our methods (PCD, VD, and USAD) also exhibit stable type-I error rates, confirming the significance of all the statistical adversarial detection methods.

G.5 Computational Efficiency.

Table 9 reports the inference time per test trial averaged over 100 runs, with all experiments conducted using reference example size $|X|$ equal to input example size $|Y| = 50$ on the platform described in Appendix F.3. Our VD method demonstrates superior computational efficiency across both datasets, achieving 0.016s on CIFAR-10 and 0.015s on ImageNet-1K—representing 33% and 85% speedups over the baseline SAMMD, respectively. Notably, VD's inference time remains nearly constant across datasets, indicating excellent scalability to higher-dimensional data. While SAMMD shows moderate computational cost on CIFAR-10 (0.024s), it increases substantially to 0.097s on ImageNet-1K, suggesting quadratic or higher complexity with respect to feature dimensionality. In contrast, PCD and USAD exhibit significantly longer inference times (0.117s and 0.132s on CIFAR-10; 0.759s and 0.772s on ImageNet-1K), due to their reliance on computationally intensive distance matrix operations or optimization procedures. The minimal overhead of VD, combined with its strong detection performance, makes it particularly attractive for real-time or large-scale deployment scenarios where computational budget is constrained. Although PCD and USAD are slower than SAMMD, their inference time remains below one second per query batch, and, even in large-scale deployment, USAD only needs to be applied to small query windows rather than the entire data stream. The extra cost is useful because PCD can detect attacks where VD is weak: for example, under the adaptive attack with $m = 10$ in Table 2, VD achieves 0.186 power, while USAD reaches 0.818. Thus, VD can be used when latency is critical, while USAD can be used when reliable detection is more important; the trade-off between the number of perturbations K in PCD and the running time is further studied in Appendix G.13.

G.6 ℓ_0 (Sparse-PGD) and ℓ_1 Attacks.

While the main paper focuses on dense ℓ_∞ and ℓ_2 adversarial perturbations, sparse attacks induce qualitatively different geometric distortions and thus warrant separate empirical verification. To assess the robustness of our detection framework against this regime, we additionally evaluate against ℓ_0-bounded perturbations generated by Sparse-PGD (SPGD) [68] and ℓ_1-bounded perturbations generated by ℓ_1-APGD. For SPGD, we vary the sparsity budget $k \in \{10, 20, 50, 100\}$ (i.e., the number of perturbed pixels), and for ℓ_1-APGD we sweep the perturbation norm $N \in \{10, 20, 30\}$. All experiments follow the same evaluation protocol as the main paper and are verified to maintain a valid type-I error control.

Table 10 reports the detection power of PCD, VD, and USAD across these settings. Our methods, particularly VD and USAD, remain highly effective against sparse adversarial perturbations: both attain detection power of 1.000 across nearly all configurations, with only marginal degradation under the most restrictive ℓ_1 budget (ℓ_1-N10). PCD shows a sharper sensitivity to the perturbation budget under ℓ_1 attacks, but recovers to near-perfect detection as the budget grows. These results confirm that the proposed framework generalizes beyond dense ℓ_∞/ℓ_2 threat models and is robust to the geometrically distinct distortions induced by sparse adversarial attacks.

G.7 Adversarial Video Detection on UCF101.

A natural extension of our framework is to the video domain, where adversaries typically inject perturbations across multiple frames and video models routinely process frames in batches. This

Table 10: Detection power against ℓ_1-APGD (varying norm N) and Sparse-PGD (varying sparsity k) on CIFAR-10. All results are averaged over multiple trials and reported as mean $\pm$ standard deviation. Type-I error is controlled in all settings.

Method	ℓ_1-N10	ℓ_1-N20	ℓ_1-N30	SPGD-k10	SPGD-k20	SPGD-k50	SPGD-k100
PCD	0.231 ± 0.009	0.476 ± 0.009	0.719 ± 0.011	0.991 ± 0.003	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
VD	0.977 ± 0.005	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
USAD	0.981 ± 0.004	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000

setting is particularly well-suited to multi-sample detectors such as ours, since the temporal structure of a video provides a natural batch of correlated samples for two-sample testing. To verify that our method transfers to this regime, we conduct an additional evaluation on the UCF101 action recognition benchmark using the sparse video attack proposed by [52].

Setup. We attack 65 sample videos drawn from the first two classes in alphabetical order (*ApplyEyeMakeup* and *Archery*). For each successfully attacked video, we extract a 40-frame clip from both its clean and adversarial versions, treating each clip as a sample set for two-sample testing. Detection is performed by running a permutation test between the clean and adversarial frame sets drawn from the same class, yielding a binary detection decision per video. The reported detection rate (test power) is averaged over all successfully attacked videos, and type-I error is computed under the null using clean-vs-clean clip pairs.

Results. Table 11 summarizes the outcome. Our method achieves a detection rate of 81.5% while maintaining 0.0% type-I error, demonstrating that it transfers effectively to the video domain without modification. This result is consistent with our intuition: because adversarial video attacks distribute perturbations across many frames, the per-frame distributional shift accumulates into a strong multi-sample signal that our framework is well-equipped to detect.

Table 11: Adversarial video detection on UCF101 against the sparse video attack of [52]. We evaluate on the first two classes (alphabetical), comparing 40-frame clean and adversarial clips from the same class via a permutation test. Type-I error is controlled.

Class	Total N	Avg. Detection	Type-I Error
ApplyEyeMakeup + *Archery*	65	81.5%	0.0%

G.8 Ablation on the Correlation-aware Aggregation and Incorporating MMD

Equal-weight Aggregation. To isolate the effect of the correlation-aware aggregation in Section 4, we compare USAD with a simple equal-weight sum of standardized VD and PCD, where VD and PCD are first standardized by their null standard deviations (i.e., the square roots of the diagonal entries of $\widehat{\Sigma}_{\mathrm{A}}$ in Eq. (10)) and then added with equal weights, with the testing threshold calibrated by the same permutation test. The experiments are conducted on CIFAR-10 using ResNet-18 as the target model under a five-step PGD ℓ_∞ attack, with $|X| = 100$ and $|Y| = 50$. All methods control the Type-I error at $\alpha = 0.05$, and the results are averaged over 10 experiments with 100 tests each. Table 12 shows that USAD matches or outperforms the equal-weight sum across all settings, improving power by 0.045 and 0.024 at $\epsilon = 1/255$ and $2/255$, respectively. This supports the benefit of accounting for the correlation between VD and PCD.

Table 12: Test power of PCD, VD, an equal-weight sum of standardized VD and PCD, and USAD under different perturbation budgets ϵ. All methods control the Type-I error at $\alpha = 0.05$, and the results are averaged over 10 experiments with 100 tests each. The experiments are conducted on CIFAR-10 using ResNet-18 under a five-step PGD ℓ_∞ attack, with $|X| = 100$ and $|Y| = 50$.

ϵ	PCD	VD	Equal-weight sum	USAD
$1/255$	0.344	0.769	0.752	**0.797**
$2/255$	0.175	0.777	0.755	**0.779**
$3/255$	0.313	1.000	1.000	**1.000**

Incorporating MMD. As discussed in Section 5, the aggregation framework is general and can incorporate additional statistics. To demonstrate this flexibility, we add MMD to the USAD framework, i.e., we aggregate VD, PCD, and $\widehat{\mathrm{MMD}}^2$ with the same correlation-aware weighting as in Eq. (9). The experiments are conducted on CIFAR-10 using ResNet-18 as the target model under a five-step PGD ℓ_∞ attack, with $|X| = 100$ and $|Y| = 50$. The results are averaged over 10 experiments with 150 tests each, with standard error at most 0.016. Table 13 confirms that USAD can incorporate MMD. However, MMD provides limited additional signal in the weaker-attack settings and reduces the finite-sample power. This shows that adding more statistics does not necessarily improve performance; the statistics should be selected according to the target alternatives.

Table 13: Test power of USAD (VD+PCD) and its variant incorporating MMD (VD+PCD+MMD) under different perturbation budgets ϵ. The results are averaged over 10 experiments with 150 tests each, with standard error at most 0.016. The experiments are conducted on CIFAR-10 using ResNet-18 under a five-step PGD ℓ_∞ attack, with $|X| = 100$ and $|Y| = 50$.

Method	$\epsilon = 1/255$	$\epsilon = 2/255$	$\epsilon = 3/255$	$\epsilon = 4/255$	$\epsilon = 5/255$–$8/255$
USAD (VD+PCD)	**0.769**	**0.775**	**1.000**	**1.000**	**1.000**
USAD (VD+PCD+MMD)	0.611	0.697	**1.000**	**1.000**	**1.000**

G.9 Analysis of the Two Uncertainty Factors

Arguments from Prior Work. Factor 1 builds on the observation that clean and adversarial data reside on different manifolds in the semantic feature space, with AEs typically lying slightly off the clean data submanifold and exhibiting altered neighborhood structures. In particular, Feinman et al. [27] detect AEs through kernel density estimates in the feature space of the last hidden layer together with Bayesian uncertainty estimates, and Ma et al. [13] show that adversarial regions exhibit higher local intrinsic dimensionality (LID) than normal data regions; both indicate that AEs alter how semantic features are spread, which VD measures through the within-distribution variance. Factor 2 builds on two complementary observations. On the one hand, under isotropic Gaussian perturbations, randomized smoothing shows that predictions can be certified to be locally stable, providing evidence that clean inputs tend to exhibit greater stability to small noise [42, 44]. On the other hand, norm-bounded AEs are generated by small perturbations that cross local decision boundaries [28, 29] and often exploit brittle non-robust features [30], indicating increased sensitivity to certain perturbations. Accordingly, random-perturbation-based detectors [15, 25, 26] distinguish AEs from CEs through their atypical local response patterns under added perturbations, which motivates PCD to compare the covariances of semantic features under Gaussian perturbations.

VD and PCD are designed to capture these two factors, respectively. While the test power of VD and PCD across real attacks (e.g., Table 4) supports this design, it does not directly show how strongly the two factors are present in each setting. We therefore provide a more direct analysis, first in a controlled synthetic setting and then on real AEs.

Controlled Synthetic Experiment. We use a two-dimensional classifier whose decision boundary is the line on which the first input coordinate equals 0, and repeat each setting 200 times with $|X| = |Y| = 50$. For Factor 1, we keep the clean and query samples at the same distance from the decision boundary, so that their local sensitivity is unchanged, and only increase the global spread of the query distribution. For Factor 2, we keep the global spread fixed and move the query samples closer to the decision boundary by decreasing their boundary distance. Table 14 reports the values of $\widehat{\mathrm{VD}}$ and $\widehat{\mathrm{PCD}}$ in both cases. VD increases consistently as the global spread increases, while PCD remains unchanged; conversely, as the samples approach the decision boundary, PCD increases substantially, while VD remains close to zero. These controlled results support that VD captures changes in the global feature spread, whereas PCD captures the increased local sensitivity near the decision boundary.

Analysis on Real AEs. Following the setup of the synthetic study, we further measure the two factors on real AEs generated by different attacks. We consider ResNet-18 trained on *CIFAR-10* with $\epsilon = 1/255$ and $|Y| = 50$, i.e., the setting of Table 4, from which the VD and PCD power below are taken. Both quantities are computed using the classifier's 64-dimensional penultimate features. For Factor 1, we measure the spread change as the absolute difference in empirical RKHS variance

Table 14: Controlled synthetic analysis of Factor 1 (left: increasing the global spread of the query distribution with the boundary distance fixed) and Factor 2 (right: decreasing the boundary distance of the query samples with the global spread fixed). We report $\widehat{\mathrm{VD}}$ (mean $\pm$ standard deviation) and $\widehat{\mathrm{PCD}}$ over 200 repetitions with $|X| = |Y| = 50$, using a two-dimensional classifier whose decision boundary is the line on which the first input coordinate equals 0.

Query spread	VD	PCD
0.05	0.0035 ± 0.0050	0.0000
0.10	0.0617 ± 0.0248	0.0000
0.20	0.1502 ± 0.0335	0.0000
0.30	0.1982 ± 0.0386	0.0000
0.40	0.2178 ± 0.0335	0.0000

Boundary distance	VD	PCD
0.50	0.0030 ± 0.0044	0.0000
0.20	0.0035 ± 0.0044	0.0001
0.10	0.0030 ± 0.0043	0.1206
0.05	0.0030 ± 0.0047	1.0694
0.02	0.0027 ± 0.0036	1.6369

between the query batch and the clean reference, $|\widehat{V}(Y;\ell) - \widehat{V}(X;\ell)|$, using the Gaussian kernel $\ell(f(\boldsymbol{x}), f(\boldsymbol{x}')) = \exp(-\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\|_2^2/32)$. For Factor 2, we measure the Euclidean distance from the semantic feature $f(\boldsymbol{y})$ of each AE $\boldsymbol{y}$ to the nearest boundary of its predicted-class region under the final linear classification layer:

$$\mathrm{dist}(\boldsymbol{y}) = \min_{c \neq \hat{c}} \frac{s_{\hat{c}}(\boldsymbol{y}) - s_c(\boldsymbol{y})}{\|w_{\hat{c}} - w_c\|_2},$$

where $s_c(\boldsymbol{y})$ is the logit for class $c \in \mathcal{C}$, which is affine in $f(\boldsymbol{y})$ with weight vector w_c in the final linear classification layer, and $\hat{c} = \arg\max_{c \in \mathcal{C}} s_c(\boldsymbol{y})$ is the predicted class. Boundary distances are first averaged over the AEs in each query batch, and both quantities are then averaged over 1,000 batches with $|X| = |Y| = 50$. Table 15 shows that a larger spread change is associated with higher VD power across these attacks; in particular, FGSM has both the largest spread change and the highest VD power. Similarly, a smaller boundary distance tends to be associated with higher PCD power, and FGSM has both the smallest boundary distance and the highest PCD power. Overall, these trends are consistent with our interpretation that VD captures global feature-spread changes, while PCD captures local perturbation sensitivity.

Table 15: Analysis of Factor 1 (left: change in the feature spread of the query batch and VD power) and Factor 2 (right: average boundary distance of the AEs and PCD power) on real AEs generated by different attacks against ResNet-18 trained on *CIFAR-10*, with $\epsilon = 1/255$ and $|Y| = 50$.

Attack	Spread change	VD power
AA	0.0464	0.261 ± 0.010
CW	0.0699	0.507 ± 0.010
PGD	0.0800	0.653 ± 0.015
FGSM	**0.1402**	$\mathbf{0.794 \pm 0.016}$

Attack	Boundary distance	PCD power
AA	2.112	0.222 ± 0.016
CW	1.326	0.234 ± 0.012
PGD	1.246	0.251 ± 0.012
FGSM	**0.954**	$\mathbf{0.311 \pm 0.014}$

G.10 Detection under Low AE Fractions in the Query Batch

In practical monitoring scenarios, AEs may constitute only a small fraction of the query batch. Our consistency results (Theorems 2–3 and Corollary 4) show that, for any fixed non-zero AE fraction,

the detection power approaches 1 as the batch size increases, provided the AEs change the uncertainty captured by VD or PCD. Therefore, there is no universal minimum AE fraction: reliable finite-sample detection depends jointly on the AE fraction, the batch size, and the attack signal, and a smaller AE fraction must be compensated by a larger batch.

Effect of the AE Fraction on Test Power. We make this dependence explicit as follows. Let $\eta \in (0, 1]$ denote the fraction of AEs in the query batch, and write the query distribution as the mixture $\mathbb{Q}_\eta = (1-\eta)\mathbb{P} + \eta\,\mathbb{A}$, where $\mathbb{A}$ denotes the distribution of AEs. We define

$$\Delta_V = \left|V_{\mathbb{P}} - V_{\mathbb{A}} - (1-\eta)\,\mathrm{MMD}^2(\mathbb{P}, \mathbb{A}, \ell)\right| \quad \text{and} \quad \Delta_C = \|\boldsymbol{\mu}_{\Sigma_{\mathbb{P}}} - \boldsymbol{\mu}_{\Sigma_{\mathbb{A}}}\|_{\mathcal{H}_\rho},$$

where $V_{\mathbb{A}}$ and $\boldsymbol{\mu}_{\Sigma_{\mathbb{A}}}$ are defined as in Section 3.1 and Section 3.2, and $\mathrm{MMD}^2(\mathbb{P}, \mathbb{A}, \ell)$ denotes the MMD in Eq. (3) computed on semantic features with kernel ℓ (Appendix D). Here, Δ_V and Δ_C measure how different the clean and adversarial distributions are in the global variance captured by VD and in the perturbation-induced covariance captured by PCD, respectively. We also write $N_{\text{eff}} = nm/(n+m)$ for the effective sample size.

Remark 2 (Informal)**.** *Let ℓ and ρ be bounded kernels, and let Δ be either Δ_V (for VD) or Δ_C (for PCD). Up to kernel-dependent constants and lower-order terms, whenever $\eta\Delta \gtrsim \sqrt{\log(1/\alpha)/N_{\text{eff}}}$, the probability that the corresponding test fails to reject $\boldsymbol{H}_0$ when Y is drawn from $\mathbb{Q}_\eta$ (i.e., its Type-II error) satisfies*

$$\Pr\left(\mathcal{T}(X, Y) \le t_\alpha \mid Y \sim \mathbb{Q}_\eta\right) \lesssim \exp\left\{-C N_{\text{eff}}\,\eta^2\Delta^2\right\}$$

for some constant $C > 0$. Consequently, $N_{\text{eff}}\,\eta^2\Delta^2 \gtrsim \log(1/\alpha) + \log(1/\beta)$ suffices for a Type-II error of at most β.

Proof sketch. (1) *Population discrepancies.* For VD, expanding the RKHS variance under the mixture gives $V_{\mathbb{Q}_\eta} = (1-\eta)V_{\mathbb{P}} + \eta V_{\mathbb{A}} + \eta(1-\eta)\,\mathrm{MMD}^2(\mathbb{P}, \mathbb{A}, \ell)$, and hence $|V_{\mathbb{P}} - V_{\mathbb{Q}_\eta}| = \eta\Delta_V$. For PCD, drawing from $\mathbb{Q}_\eta$ produces the covariance matrix of a CE with probability $1-\eta$ and that of an AE with probability η; by linearity of kernel mean embeddings, $\boldsymbol{\mu}_{\Sigma_{\mathbb{Q}_\eta}} = (1-\eta)\boldsymbol{\mu}_{\Sigma_{\mathbb{P}}} + \eta\,\boldsymbol{\mu}_{\Sigma_{\mathbb{A}}}$, and hence $\|\boldsymbol{\mu}_{\Sigma_{\mathbb{P}}} - \boldsymbol{\mu}_{\Sigma_{\mathbb{Q}_\eta}}\|_{\mathcal{H}_\rho} = \eta\Delta_C$. Thus, the population discrepancies underlying both VD and PCD decrease linearly with η. (2) *Concentration and permutation thresholds.* The empirical VD gap $\widehat{V}(X;\ell) - \widehat{V}(Y;\ell)$ is a difference of bounded second-order U-statistics (Appendix E.2), and Hoeffding's inequality [48] gives $\Pr\left(|(\widehat{V}(X;\ell) - \widehat{V}(Y;\ell)) - (V_{\mathbb{P}} - V_{\mathbb{Q}_\eta})| \ge t\right) \lesssim \exp(-C N_{\text{eff}}\,t^2)$ for any $t > 0$. For PCD, the RKHS distance between the empirical mean embeddings of $\{\Sigma_{\boldsymbol{x}_i}\}_{i=1}^n$ and $\{\Sigma_{\boldsymbol{y}_j}\}_{j=1}^m$ deviates from $\eta\Delta_C$ by more than $O(n^{-1/2} + m^{-1/2}) + t$ with probability $\lesssim \exp(-C N_{\text{eff}}\,t^2)$ [24], and, for a bounded kernel ρ, the unbiased estimator in Eq. (8) differs from the squared distance between these empirical embeddings by at most $O(1/n + 1/m)$; these are lower-order terms once $\eta\Delta_C$ is above the sampling noise. Under permutation, the two groups are formed from the same pooled sample, and concentration bounds for permuted U-statistics [49] (Theorem 7 for PCD; for VD, the permuted gap is a difference of U-statistics of samples drawn without replacement, see Appendix E.2) imply that the critical radius of the VD gap or the PCD embedding distance is of order $\sqrt{\log(1/\alpha)/N_{\text{eff}}}$, or, equivalently, that the critical value of the corresponding squared statistic is of order $\log(1/\alpha)/N_{\text{eff}}$. (3) *Type-II error.* If $\eta\Delta$ is larger than this critical radius and the lower-order terms above, then a failure to reject $\boldsymbol{H}_0$ requires the empirical discrepancy to differ from its population value by at least a constant fraction of $\eta\Delta$; substituting $t \asymp \eta\Delta$ into the concentration bounds above yields the bound in Remark 2. □

Remark 2 gives a quadratic sample-size dependence on the AE fraction: when Δ remains of the same order, halving the AE fraction requires roughly four times as many effective samples to maintain the same Type-II error. Moreover, if n is fixed, increasing only m has diminishing benefit because $N_{\text{eff}} \to n$, which explains why very low AE fractions remain difficult even with large query batches. The remark also requires the relevant discrepancy Δ to remain non-negligible: if VD or PCD captures little difference between the clean and adversarial distributions, increasing the sample size cannot compensate for an absent signal.

Setup. We evaluate USAD and the MMD-based methods under different AE fractions and query batch sizes on CIFAR-10 with ResNet-18 under PGD-ℓ_∞ with $\epsilon = 8/255$, using a fixed reference size $|X| = 100$ and $|Y| \in \{50, 100, 200, 500, 1000\}$, with 5 independent experiments, 100 tests per

experiment, and 100 permutations. Since the number of AEs in a query batch must be an integer, the implementation rounds the product of the AE fraction and $|Y|$ down to an integer; thus, for $|Y| = 50$ and a 1% AE fraction, the query batch contains no AE, and this entry measures the Type-I error rather than the power (for $|Y| = 50$, the smallest non-zero AE fraction is 2%, corresponding to one AE).

Results. Table 16 shows that larger batches can accumulate weak evidence from diluted AEs: at a 10% AE fraction, for example, the power of USAD increases from 0.650 to 0.961 as the batch size grows from 50 to 1000. This is an advantage of batch-level detection, which can combine information across multiple queries. USAD achieves a clear advantage over the MMD-based methods at 10% contamination and remains consistently stronger at 5%. At 1%–2%, however, all methods have limited power. Extremely low contamination, such as 0.1% or 1%, is challenging for adversarial detection in general, including both batch-level and per-example methods. For example, among $1{,}000$ queries, a 5% per-example false-alarm rate produces about 50 false alarms, while 1% contamination contains only 10 AEs and 0.1% contamination contains only one AE. Thus, false alarms can easily outnumber true AEs, and reliable detection in this regime requires a false-alarm rate substantially below the AE prevalence and may require much larger batches.

Table 16: Rejection rates of USAD and MMD-based methods for different AE fractions in the query batch and different query batch sizes $|Y|$, on CIFAR-10 with ResNet-18 under PGD-ℓ_∞ with $\epsilon = 8/255$ and a fixed reference size $|X| = 100$. The results are averaged over 5 independent experiments with 100 tests each, using 100 permutations. The number of AEs is the product of the AE fraction and $|Y|$ rounded down, so the entries with a 1% AE fraction and $|Y| = 50$ contain no AE and measure the Type-I error.

AE fraction	Method	$\|Y\| = 50$	$\|Y\| = 100$	$\|Y\| = 200$	$\|Y\| = 500$	$\|Y\| = 1000$
10%	USAD	**0.650**	**0.839**	**0.889**	**0.928**	**0.961**
	MMD-DUAL	0.189	0.311	0.378	0.511	0.578
	MMDAgg	0.054	0.056	0.070	0.072	0.038
	MMD-FUSE	0.052	0.044	0.054	0.054	0.044
	SAMMD	0.044	0.056	0.067	0.033	0.056
5%	USAD	**0.150**	**0.273**	**0.319**	**0.366**	**0.396**
	MMD-DUAL	0.056	0.067	0.144	0.122	0.056
	MMDAgg	0.038	0.044	0.040	0.048	0.026
	MMD-FUSE	0.040	0.044	0.028	0.044	0.030
	SAMMD	0.033	0.067	0.056	0.022	0.056
2%	USAD	0.050	0.111	0.094	0.094	0.111
	MMD-DUAL	0.067	0.011	0.033	0.100	0.033
	MMDAgg	0.030	0.038	0.036	0.034	0.024
	MMD-FUSE	0.038	0.044	0.022	0.036	0.024
	SAMMD	0.022	0.022	0.011	0.078	0.067
1%	USAD	0.044	0.089	0.061	0.061	0.067
	MMD-DUAL	0.044	0.033	0.000	0.011	0.022
	MMDAgg	0.042	0.040	0.030	0.036	0.022
	MMD-FUSE	0.042	0.042	0.024	0.032	0.024
	SAMMD	0.011	0.044	0.011	0.000	0.033

G.11 Behavior under a Benign Distribution Shift

As discussed in Section 2.1 and Appendix A, VD and PCD are not formally specific to adversarial attacks, and benign shifts that alter the global feature spread or the perturbation-induced local covariance may also be detected. We therefore evaluate how USAD responds to a benign distribution shift.

Setup. We use a ResNet-18 classifier trained on *CIFAR-10* as the target model. We use clean CIFAR-10.1 [69] as a benign shift and CIFAR-10 PGD with $\epsilon = 4/255$ as an adversarial shift. We use 30 reference sets with $|X| = 100$ and $|Y| = 50$, and 100 query batches per reference set, and report the results for each component averaged over different random seeds. We report the benign false-alarm rate (BFAR), i.e., the proportion of CIFAR-10.1 batches incorrectly flagged as adversarial, the true-positive rate (TPR), i.e., the proportion of PGD batches correctly detected, and

the AUROC, which measures how well the statistics separate benign-shift and adversarial batches across all thresholds; lower BFAR and higher TPR and AUROC are better.

Results. Table 17 shows that both PCD and VD remain reasonably robust to this benign shift, with BFAR below 10% while maintaining 100% PGD detection. More importantly, the near-perfect AUROC shows that benign-shift and adversarial batches remain highly separable according to the VD and PCD statistics. Thus, the main issue is not a lack of adversarial information, but that a threshold calibrated under the strict equality null may be sensitive to small but acceptable natural shifts. To illustrate this more clearly, we vary the decision threshold and measure the PGD detection rate when the false-alarm rate on CIFAR-10.1 is limited to 1%, 3%, or 5% (Table 18). Even at the stringent 1% BFAR operating point, PCD and VD retain 99.57% and 99.93% PGD detection, respectively, showing that a threshold that tolerates small benign shifts can still reliably distinguish the larger adversarial shift.

Table 17: Behavior of PCD and VD under a benign distribution shift (clean CIFAR-10.1) versus an adversarial shift (CIFAR-10 PGD with $\epsilon = 4/255$), using 30 reference sets with $|X| = 100$, $|Y| = 50$, and 100 query batches per reference set. The target model is a ResNet-18 trained on *CIFAR-10*. BFAR is the proportion of CIFAR-10.1 batches incorrectly flagged as adversarial, TPR is the proportion of PGD batches correctly detected, and AUROC measures the separability of benign-shift and adversarial batches across all thresholds.

Component	CIFAR-10.1 BFAR ↓	PGD TPR ↑	Benign-vs-adversarial AUROC ↑
PCD	7.77%	$\mathbf{100.00\%}$	$\mathbf{0.9996}$
VD	9.75%	$\mathbf{100.00\%}$	$\mathbf{1.0000}$

Table 18: PGD detection rate (TPR) of PCD and VD when the decision threshold is chosen such that the false-alarm rate on CIFAR-10.1 batches (BFAR) is limited to 1%, 3%, or 5%, under the same setting as Table 17.

Component	TPR at 1% BFAR ↑	TPR at 3% BFAR ↑	TPR at 5% BFAR ↑
PCD	$\mathbf{99.57\%}$	$\mathbf{99.90\%}$	$\mathbf{99.93\%}$
VD	$\mathbf{99.93\%}$	$\mathbf{100.00\%}$	$\mathbf{100.00\%}$

Discussion. Following Zhou et al. [63], these observations motivate viewing the problem as distribution closeness testing rather than strict equality testing, where the null hypothesis allows the discrepancy between $\mathbb{P}$ and $\mathbb{Q}$ to lie within an acceptable range of benign shift and is rejected only when the discrepancy exceeds this range. For example, the threshold can be chosen so that only 1%, 3%, or 5% of CIFAR-10.1 batches are flagged; a new batch is then classified as adversarial only when its VD or PCD discrepancy exceeds this acceptable benign-shift range. As shown above, even at 1% BFAR, almost all PGD batches are still detected, which means that small natural shifts can be tolerated while larger adversarial shifts remain detectable. Building on this interpretation, our VD and PCD statistics could be extended to distribution closeness testing, and developing a formal closeness-testing procedure with theoretical guarantees is an interesting direction for future work.

G.12 Comparison with MVD and with a Per-example Detector under an Adaptive C&W Attack

VD versus MVD. Following the discussion in Section 3.1, we examine whether the simpler trace-level VD offers a practical advantage over MVD [43], which compares the full RKHS covariance operators. We directly compare the two methods under identical experimental conditions, using the same AEs, semantic features, Gaussian kernel, sampled batches, and permutation assignments, with $|X| = |Y| = 50$. The experiments are conducted on CIFAR-10 using ResNet-18 as the target model, under the corresponding perturbation budgets used for each adversarial attack. Table 19 shows that, across the four attacks, VD achieves higher test power than MVD, with the largest improvements on FGSM and PGD, and that MVD requires approximately $1.8\times$ the runtime of VD (0.714 ms versus 0.397 ms). These results support using the simpler trace-level VD for small-query SAD, where both test power and computational cost matter.

PCD versus Existing Perturbation-based Detectors. Random perturbations have been used by prior per-example detectors to reveal differences between clean and adversarial inputs [15, 25, 26, 57]. To the best of our knowledge, PCD is the first batch-level detector to represent each input through the

Table 19: Test power (left) and runtime (right) of VD and MVD under identical experimental conditions (the same AEs, semantic features, Gaussian kernel, sampled batches, and permutation assignments). The experiments are conducted on CIFAR-10 using ResNet-18 as the target model, with $|X| = |Y| = 50$.

Attack	VD	MVD
CW	**0.453**	0.432
FGSM	**0.957**	0.522
PGD	**0.597**	0.455
AA	**1.000**	0.947

Runtime	VD	MVD
Total runtime (ms)	**0.397**	0.714
Relative total runtime	**1.00**×	1.80×

covariance of its semantic-feature responses under Gaussian perturbations and to test whether these covariance distributions differ between clean and query batches. PCD uses a permutation-calibrated two-sample test with Type-I error control, whereas prior per-example detectors classify each input separately and do not provide explicit Type-I error control.

Comparison with PFD under an Adaptive C&W Attack. Since per-example detectors and SAD methods make decisions at different granularities, a direct comparison requires a common setting. Following Carlini and Wagner [38], we evaluate both our methods and PFD [57], a recent perturbation-based per-example detector, under an adaptive C&W attack, in which the attacker has access to the detector and optimizes each adversarial input to simultaneously mislead the classifier, evade the detector, and maintain a small ℓ_2 distortion. Specifically, the attack minimizes the ℓ_2 distortion plus the C&W classifier loss and a detector-evasion loss constructed for either PFD or our detectors, over an unconstrained reparameterization of the input through the hyperbolic tangent, which is solved with Adam. For a fair comparison, we keep all other attack settings fixed and change only the detector-specific loss: we optimize each adversarial example for 200 steps, with the standard C&W trade-off coefficient set to 0.01, the confidence parameter set to 0, and the coefficient of the detector-evasion loss set to 1.0. The experiments are conducted on CIFAR-10 using ResNet-18 as the target model. For PCD and USAD, we use query batches of size $|Y| = 50$. Their detection rate is the proportion of adversarial query batches for which the permutation test rejects the null hypothesis at $\alpha = 0.05$, whereas PFD reports the proportion of individual adversarial examples detected as adversarial. Table 20 shows that, under the same adaptive attack, PFD does not detect the generated AEs, whereas PCD and USAD retain detection rates of 41.5% and 44.8%, respectively. This provides a direct comparison between our detectors and a strong perturbation-based per-example detector under an adaptive attack, and demonstrates that USAD is substantially more robust than PFD against the adaptive C&W attack.

Table 20: Detection rates of PFD, PCD, and USAD under an adaptive C&W attack [38] that optimizes each adversarial example for 200 steps, changing only the detector-specific loss. The experiments are conducted on CIFAR-10 using ResNet-18, with $|Y| = 50$ for the batch-level detectors PCD and USAD.

Setting	Mean ℓ_2	PFD detection rate	PCD detection rate	USAD detection rate
Adaptive C&W	0.072	0.0%	41.5%	$\mathbf{44.8}\%$

G.13 Sensitivity to the Kernel Bandwidth and the Number of Perturbations

Kernel Bandwidth. The bandwidth parameters of our kernels are set using the median heuristic and fixed values (see Appendix F). To examine the sensitivity to this choice, we vary the bandwidth from $0.25\times$ to $4\times$ the median-heuristic value. The experiments are conducted on CIFAR-10 using ResNet-18 as the target model under a five-step PGD ℓ_∞ attack with $\epsilon = 2/255$, $|X| = 100$, and $|Y| = 50$. Table 21 shows that USAD, VD, and PCD remain stable across this $16\times$ bandwidth range, with maximum power variations of only 0.008, 0.002, and 0.002, respectively, compared with 0.038 for SAMMD. This shows that our methods are not sensitive to the bandwidth choice.

Number of Perturbations. Since the inference cost of PCD (and hence USAD) grows with the number of perturbations K (see the time complexity analysis in Appendix F), we study whether K can be reduced while maintaining good detection. We vary K on CIFAR-10 with ResNet-18 under PGD with $\epsilon = 4/255$, using $|X| = 100$ and $|Y| = 10$. We report the test power and Type-I error of PCD, together with the end-to-end runtime. As shown in Table 22, reducing K from 200 to 100 decreases

Table 21: Test power when the kernel bandwidth is varied from $0.25\times$ to $4\times$ the median-heuristic value. The last column (spread) is the difference between the maximum and minimum power across the five bandwidths. The experiments are conducted on CIFAR-10 using ResNet-18 under a five-step PGD ℓ_∞ attack with $\epsilon = 2/255$, $|X| = 100$, and $|Y| = 50$.

Method	$0.25\times$	$0.5\times$	$1\times$	$2\times$	$4\times$	Spread
USAD (VD+PCD)	0.766	0.770	0.764	0.766	0.762	0.008
VD	0.776	0.778	0.776	0.776	0.776	0.002
PCD	0.172	0.174	0.174	0.174	0.174	0.002
SAMMD	0.318	0.290	0.280	0.280	0.280	0.038

the end-to-end runtime from 0.643s to 0.374s while maintaining comparable power and valid Type-I error, whereas further reducing K to 50 causes a clear power drop. Thus, $K = 100$ provides a good performance–runtime trade-off; all other results in this paper use $K = 200$ (Appendix F).

Table 22: Test power, Type-I error, and end-to-end runtime for different numbers of perturbations K in PCD, on CIFAR-10 with ResNet-18 under PGD with $\epsilon = 4/255$, $|X| = 100$, and $|Y| = 10$.

K	Power	Type-I error	End-to-end runtime
50	0.348	0.061	0.239s
100	**0.500**	**0.050**	**0.374**s
200	0.498	0.054	0.643s

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions of USAD, including the proposed uncertainty-aware statistics, their aggregation, theoretical Type-I error control, and empirical evaluation against standard and adaptive attacks.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations regarding the scope of SAD, the reliance on batch-level statistical testing, and the dependence of detection power on the query size, attack type, and feature representation; Appendix A further discusses the limitations of USAD, including its response to benign distribution shifts, low fractions of AEs in the query batch, and the alternatives targeted by VD and PCD.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The theoretical results explicitly state the required assumptions, and complete proofs are provided in the appendix. The main text also explains how these results justify the validity of the proposed statistical tests. The informal analysis of the effect of the AE fraction (Remark 2 in Appendix G.10) is accompanied by a proof sketch.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper reports the datasets, attacks, baselines, evaluation metrics, significance level, query sizes, and implementation details needed to reproduce the main experimental findings.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The paper uses publicly available datasets and releases the code with instructions for reproducing the experiments (see the link in the abstract).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental section specifies the datasets, target classifiers, attack configurations, baselines, query sizes, testing thresholds, and hyperparameters used for VD, PCD, and USAD.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The paper reports detection power and Type-I error across repeated trials, and the statistical testing procedure controls the false alarm rate at the chosen significance level.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The experiments of the work are conducted on two platforms. One platform is an NVIDIA RTX 4090 GPU PC with PyTorch framework. Another platform is a High-performance Computer cluster with several NVIDIA A100 GPUs with Pytorch framework. The memory of two platforms are both 64 GB. The storage of disk of two platforms are both over 4 TB.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research uses standard public datasets and adversarial robustness benchmarks, does not involve human subjects, and is intended to improve the reliability and security of machine learning systems.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: The purpose of this paper is to improve adversarial detection and model reliability, without any negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, or generative systems with substantial misuse risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

Justification: The paper uses standard public datasets, models, and attack methods, and cites the corresponding original works. The use of these assets follows their intended research usage.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [No]

Justification: The paper cites the original datasets, models, and attack methods, but does not explicitly list all licenses and terms of use. We will add these details in the appendix.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve human subjects, so IRB approval is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core methodology does not use LLMs as an important, original, or non-standard component. Any use of LLMs for writing or editing assistance does not affect the scientific content of the paper.

Guidelines:

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