# USAD: Uncertainty-aware Statistical Adversarial Detection

**Zhijian Zhou** [*1] **Xunye Tian** [*1] **Jiacheng Zhang** [*1] **Zesheng Ye** [1]
**Yiyi Guo** [1] **Donghao Zhang** [1] **Liuhua Peng** [1] **Feng Liu** [1]

## Abstract

*Statistical adversarial detection* (SAD) treats 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/tmlr-group/USAD`.

## 1. Introduction

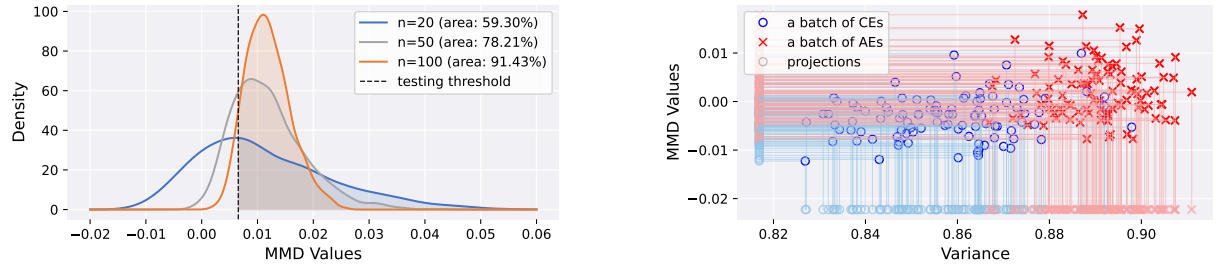
Defending against *adversarial examples* (AEs) remains a long-standing challenge in deep learning (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018; Carlini & Wagner, 2017b; Zhang et al., 2019; Croce & Hein, 2020b; Gao et al., 2021; Cao et al., 2021; Jing et al., 2021; Zhang et al., 2024; Sun et al., 2025; Han et al., 2025). The most lightweight defense against adversarial attacks is adversarial detection, which identifies AEs in input data before they reach target systems (Ma et al., 2018; Deng et al., 2021; Zhang et al., 2023; Gao et al., 2021). Existing methods often consider adversarial detection as a binary classification task (Ma et al., 2018; Deng et al., 2021; Zhang et al., 2023). They typically train a binary detector tailored to specific classifiers or attack types, which limits the effectiveness against unknown attacks (Tramèr, 2022; Bryniarski et al., 2022). Although a study leverages diffusion models (Song & Ermon, 2019; Song et al., 2021; Huang et al., 2021) to remove such dependency (Zhang et al., 2023), it is vulnerable to diffusion-based adaptive attacks (Xue et al., 2023; Kang et al., 2023). 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 (Gao et al., 2021). A permutation test then determines a threshold $t_\alpha$ such that $\Pr(T(X, Y) > t_\alpha | \text{Y is clean}) \leq \alpha$, providing guarantees on statistically-controlled Type-I error, which is often desired in high-stakes 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 (Zhang et al., 2025). 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 (Gao et al., 2021). See Appendix A for further discussions on the practicality of SAD.

Current SAD mainly use *maximum mean discrepancy*

[*]Equal contribution [1]The University of Melbourne. Correspondence to: Feng Liu <fengliu.ml@gmail.com>, Zesheng Ye <zesheng.ye@unimelb.edu.au>.

*ICML 2026 Workshop on Hypothesis Testing*, Seoul, South Korea, 2026. Copyright 2026 by the author(s).




**(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 (Gao et al., 2021). **(a):** Kernel density estimates of MMD values for three AE batch sizes ($n = \{20, 50, 100\}$). The vertical dashed line denotes the test threshold. 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. 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.

(MMD) (Gretton et al., 2012) as the test statistic $\mathcal{T}(X, Y)$ (Gao et al., 2021; Zhang et al., 2025). 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 sample size (and non-trivial fraction AE) 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 (Ma et al., 2018) and local instability (Li et al., 2019), 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 (Feinman et al., 2017). 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 (Feinman et al., 2017; Ma et al., 2018) (i.e., *global uncertainty*), and (2) norm-bounded AEs are generated by small perturbations that cross local decision boundaries (Moosavi-Dezfooli et al., 2016; Fawzi et al., 2016) and often exploit brittle non-robust features (Ilyas et al., 2019), leading to atypical local response patterns under added perturbations (Huang et al., 2019; Roth et al., 2019) (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. (2025) 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 (Krizhevsky et al., 2009) and ImageNet-1K (Deng et al., 2009). 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., $\ell_\infty$ and $\ell_2$), perturbation budgets (e.g., $\epsilon$ 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 (Szegedy et al., 2014; Goodfellow et al., 2015). 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:

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

where $\epsilon$ is the perturbation budget, $||\cdot||_p$ is the $\ell_p$-norm (e.g., $\ell_\infty$ or $\ell_2$) 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 (Goodfellow et al., 2015). 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 (Kurakin et al., 2017). *Projected gradient descent* (PGD) further enhances BIM by introducing random initialization prior to iterative gradient-based updates (Madry et al., 2018). 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 (Carlini & Wagner, 2017b). *AutoAttack* (AA) (Croce & Hein, 2020b) aggregates multiple attacks, which include three non-targeted white-box attacks (Croce & Hein, 2020b,a) and one targeted black-box attack (Andriushchenko et al., 2020), which makes AA widely adopted for evaluating adversarial robustness.

**SAD Problem Setup.** Given a reference set of CEs $X = \{\boldsymbol{x}_j\}_{i=1}^n \sim \mathbb{P}$ 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

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

and reject $\boldsymbol{H}_0$ at significance level $\alpha$ when the chosen <u>*test statistic*</u> exceeds a threshold, providing Type-I control. Here, Type-I error means falsely declaring a clean query set as adversarial.

**SAD as Hypothesis Test.** The SAD starts from designing a <u>*test statistic*</u> $\mathcal{T}(X, Y)$ to measure the discrepancy between $\mathbb{P}$ and $\mathbb{Q}$ (by observing $X$ and $Y$). At a chosen level $\alpha$, $\mathcal{T}(X, Y)$ is then compared against a <u>*testing threshold*</u> $t_\alpha$ 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 $\boldsymbol{H}_0$ is rejected (i.e., declare $Y$ adversarial) if $\mathcal{T}(X, Y) > t_\alpha$.

As the null distribution is rarely known analytically, we typically estimate $t_\alpha$ using a <u>*permutation test*</u>, 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 $t_\alpha$ is then set to the adjusted $(1-\alpha)$-quantile of these permuted statistic values, ensuring the probability of falsely rejecting $\boldsymbol{H}_0$ is controlled at the chosen level $\alpha$ (see Appendix B 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 (Grosse et al., 2017; Carlini & Wagner, 2017a; Gao et al., 2021; Zhang et al., 2025) have predominantly employed the MMD (Gretton et al., 2012), 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* (Muandet et al., 2017) 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 (Gao et al., 2021; Zhang et al., 2025) 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 (Gretton et al., 2012) (Appendix C). Recently, SAMMD (Gao et al., 2021) 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 (Zhang et al., 2025). Moreover, this degradation persists even with carefully tuned kernel embeddings and thresholds (Gao et al., 2021), 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 (Harchaoui et al., 2008; Zhou et al., 2023). 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 (Feinman et al., 2017). This displacement alters the spread of AEs' semantic feature distribution (Ma et al., 2018). 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 (Moosavi-Dezfooli et al., 2016; Ilyas et al., 2019), making them sensitive to local isotropic perturbations. Under random Gaussian noises, AEs show larger and more directionally varied feature changes than CEs (Li et al., 2019). This instability marks elevated local uncertainty at individual examples.*

**Bridging the Gap.** These findings suggest a practical sensitivity gap in existing MMD-based SAD methods. MMD compares distributions through differences between their kernel mean embeddings, which can be viewed as first-order moment differences in the chosen RKHS. Although a characteristic kernel enables MMD, in principle, to distinguish any two distributions, its finite-sample power still depends on the kernel choice and on how strongly a particular distributional change is reflected in the empirical mean-embedding difference. Consequently, existing MMD-based detectors may be less effective 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 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 $\ell(\cdot, f(\boldsymbol{x})) \in \mathcal{H}_\ell$). For distributions $\mathbb{P}$ and $\mathbb{Q}$, the semantic feature variances are

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

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

The first term of $V_{\mathbb{P}}$ establishes the self-similarity of examples within $\mathbb{P}$, while the second captures their pairwise

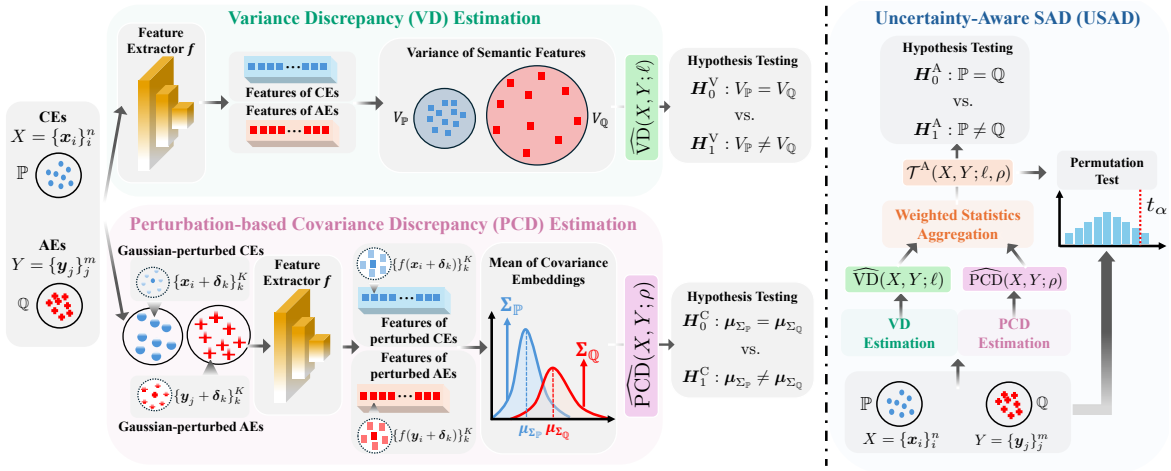



**Figure 2.** ***U***ncertainty-aware ***S***tatistical ***A***dversarial ***D***etection (USAD). The semantic features of CEs $X \sim \mathbb{P}$ and queries $Y \sim \mathbb{Q}$ (suspected AEs) are extracted by using the penultimate layer of classifier $f$. From these features, USAD estimates (i) *variance 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 $\alpha$, whether $Y$ is adversarial.

similarity. Thus, small $V_{\mathbb{P}}$ indicates tight clustering, while large $V_{\mathbb{P}}$ reflects greater dispersion (likewise for $V_{\mathbb{Q}}$).

**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* $\mathrm{VD}(\mathbb{P}, \mathbb{Q}; \ell) = (V_{\mathbb{P}} - V_{\mathbb{Q}})^2$ 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 (Cohen et al., 2019), while AEs are constructed to be near the decision boundaries of $f$ (to cause mis-classification) (Moosavi-Dezfooli et al., 2016, 2018; Ilyas et al., 2019). 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 (Li et al., 2019).

**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 $\{\boldsymbol{\delta}_i\}_{k=1}^K \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \sigma^2 I)$, extract features $\{f(\boldsymbol{x} + \boldsymbol{\delta}_i)\}_{k=1}^K$, 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 $\overline{f_{\boldsymbol{x}}} \triangleq k^{-1} \sum_{i=1}^{k} f(\boldsymbol{x} + \boldsymbol{\delta}_i)$. 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 (Zhang et al., 2023; Yang et al., 2022).

**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 $\mathcal{H}_\rho$[1]. We define *perturbation-based covariance discrepancy* (PCD) as $\mathrm{PCD}(\mathbb{P}, \mathbb{Q}, \rho) \triangleq$

[1]Mathematically, the covariance matrix $\Sigma_{\boldsymbol{x}} \in \mathcal{S}_+(\mathcal{F})$ de-

$\|\boldsymbol{\mu}_{\Sigma_{\mathbb{P}}} - \boldsymbol{\mu}_{\Sigma_{\mathbb{Q}}}\|^2_{\mathcal{H}_\rho}$, which serves as *test statistic*. Equivalently, $\mathrm{PCD}(\mathbb{P}, \mathbb{Q}, \rho)$ mirrors Eq.(3) but measures the difference between distributions of covariance matrices under kernel $\rho$.

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

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

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) (Feinman et al., 2017); others tend to exploit local gradients near boundaries (i.e., strong PCD signal) (Ilyas et al., 2019). 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 (Zhou et al., 2025), 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:

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

and defining the corresponding aggregated *test statistic* as

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

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

**Weighted Statistics Aggregation.** $\mathcal{T}^{\mathrm{A}}(X, Y; \ell, \rho)$ combines both global $\widehat{\mathrm{VD}}(X, Y; \ell)$ and local uncertainty shifts measure $\widehat{\mathrm{PCD}}(X, Y; \rho)$, following *correlation-aware* statistics ensembling (Zhou et al., 2025), with the key idea to weight $\widehat{\mathrm{VD}}(X, Y; \ell)$ and $\widehat{\mathrm{PCD}}(X, Y; \rho)$ 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 $\boldsymbol{H}_0^{\mathrm{A}}$ using an auxiliary clean calibration set $X_{\mathrm{cal}} \sim \mathbb{P}$ that is independent of the reference and query sets $(X, Y)$, which characterizes how the component statistics in $\mathcal{T}^{\mathrm{A}}(X, Y; \ell, \rho)$ behave jointly under $\boldsymbol{H}_0^{\mathrm{A}}$ (Chatterjee & Bhattacharya, 2025; Zhou et al., 2025). Specifically, we perform $B$ repeated resampling iterations using $X_{\mathrm{cal}}$. In each iteration $b$, we draw two disjoint subsets $X_b$ and $Y_b$ from the clean calibration pool (with $|X_b| = n$ and $|Y_b| = m$), and compute the statistics vector $\mathcal{T}_b \equiv \mathcal{T}(X_b, Y_b; \ell, \rho)$. From $\{\mathcal{T}_b\}_{b=1}^B$, we estimate their covariance structure as

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

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 $\widehat{\Sigma}_{\mathrm{A}}$ capture individual variabilities of VD and PCD, while off-diagonal entries reflect their correlation under the null. Additionally, by estimating this from CEs only, we ensure the aggregation weights reflect behavior when $\boldsymbol{H}_0^{\mathrm{A}}$ holds (Zhou et al., 2025).

**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 $\alpha$, we reject $\boldsymbol{H}_0^{\mathrm{A}}$ when $\mathcal{T}^{\mathrm{A}} > t_\alpha$, where $t_\alpha$ is the adjusted $(1-\alpha)$-quantile of the permuted statistic values, as defined in Appendix B.

**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 D. 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 $\alpha$; that is, the probability of falsely identifying the test batch $Y$ as adversarial is at most $\alpha$.*

We next study the test power of VD as follows.

**Theorem 2.** *Let $\ell$ 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 $n/m \to c \in (0, \infty)$.*

fines 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 $\mathrm{PCD}(\mathbb{P}, \mathbb{Q}, \rho)$ is characterized by the kernel $\rho : \mathcal{S}_+(\mathcal{F}) \times \mathcal{S}_+(\mathcal{F}) \to \mathbb{R}$ and corresponding map $\rho(\cdot, \Sigma_{\boldsymbol{x}}) \in \mathcal{H}_\rho$.

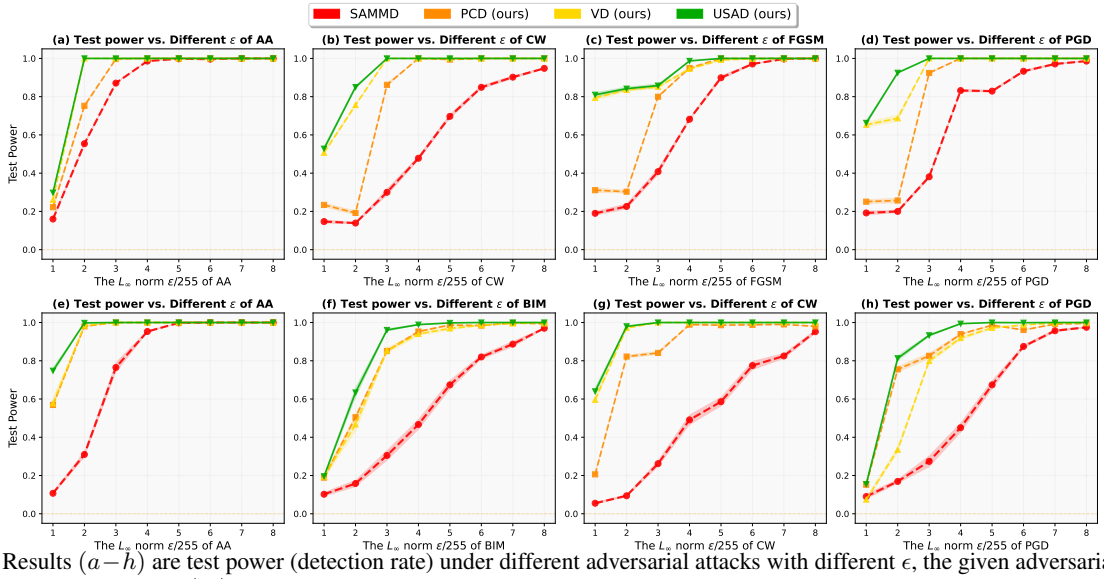



**Figure 3.** Results $(a-h)$ are test power (detection rate) under different adversarial attacks with different $\epsilon$, the given adversarial samples all share the same sample size $|Y| = 50$. The results are averaged over $1,000$ repetitions and the ideal test power is 1. The target model is ResNet-50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

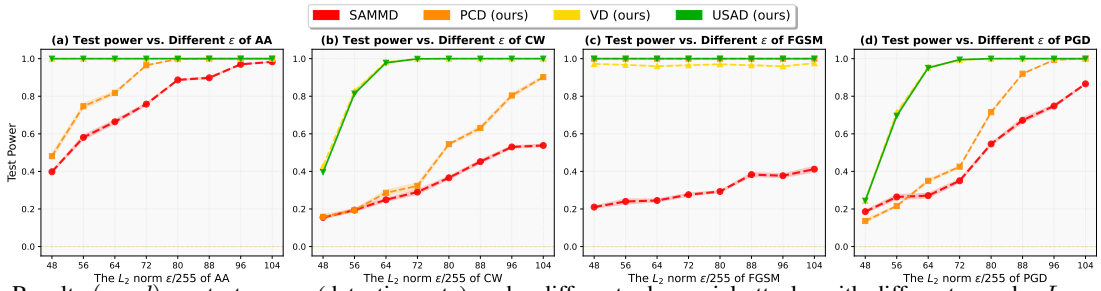



**Figure 4.** Results $(a-d)$ are test power (detection rate) under different adversarial attacks with different $\epsilon$ under $L_2$ norm, given adversarial samples all share the same sample size $|Y| = 50$. The results are averaged over $1,000$ repetitions and the ideal test power is 1 (the same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

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

**Theorem 3.** *Let $\rho$ 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 $n/m \to c \in (0, \infty)$.*

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.

**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 $\alpha$ with finite-sample Type-I error guarantees (Theorem 1).*

## 6. Experiments

We evaluate our approaches on ResNet-18/50 (He et al., 2016) and ViT-B-16 (Dosovitskiy et al., 2021) trained on *CIFAR-10* (Krizhevsky et al., 2009) and *ImageNet-1K* (Deng et al., 2009). We report the main detection results under $\ell_\infty$ and $\ell_2$ attacks, adaptive attacks, CE-mixing settings, and empirical Type-I error control in the main text. Full experimental details are deferred to Appendix F and E.

**Baselines.** We compare our methods against the state-of-the-art SAD method: **SAMMD** (Gao et al., 2021), a MMD-based detector that uses semantic features, which significantly outperforms all the single MMD-based two-

**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 (Gao et al., 2021) | $0.292 \pm 0.029$ | $0.486 \pm 0.022$ | $0.650 \pm 0.008$ | $0.756 \pm 0.019$ | $0.832 \pm 0.009$ |
| PCD (ours) | $0.486 \pm 0.024$ | $0.828 \pm 0.013$ | $0.958 \pm 0.007$ | $0.998 \pm 0.002$ | $\mathbf{1.000 \pm 0.000}$ |
| 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 (Schrab et al., 2023) | $0.342 \pm 0.016$ | $0.476 \pm 0.022$ | $0.632 \pm 0.020$ | $0.816 \pm 0.017$ | $0.912 \pm 0.009$ |
| MMD-FUSE (Biggs et al., 2023) | $0.524 \pm 0.017$ | $0.892 \pm 0.014$ | $0.994 \pm 0.004$ | $\mathbf{1.000 \pm 0.000}$ | $\mathbf{1.000 \pm 0.000}$ |
| MMD-DUAL (Zhou et al., 2025) | $0.752 \pm 0.022$ | $0.978 \pm 0.003$ | $\mathbf{1.000 \pm 0.000}$ | $\mathbf{1.000 \pm 0.000}$ | $\mathbf{1.000 \pm 0.000}$ |
| 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 | $0.486 \pm 0.024$ | $0.828 \pm 0.013$ | $0.958 \pm 0.007$ | $0.998 \pm 0.002$ | $1.000 \pm 0.000$ |
| | Adaptive | $0.388 \pm 0.027$ | $0.714 \pm 0.020$ | $0.768 \pm 0.033$ | $0.882 \pm 0.010$ | $0.914 \pm 0.014$ |
| VD | PGD | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | Adaptive | $0.186 \pm 0.023$ | $0.326 \pm 0.007$ | $0.416 \pm 0.022$ | $0.546 \pm 0.021$ | $0.614 \pm 0.011$ |
| USAD | PGD | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | Adaptive | $0.818 \pm 0.013$ | $0.998 \pm 0.002$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| Avg. Decrease | | $-0.365 \pm 0.207$ | $-0.263 \pm 0.193$ | $-0.258 \pm 0.168$ | $-0.190 \pm 0.131$ | $-0.157 \pm 0.111$ |

sample testing methods (e.g., MMD-D (Liu et al., 2020), C2ST (Cheng & Cloninger, 2019)), we refer readers interested in these experimental results to SAMMD.

**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) (Croce & Hein, 2020b), *basic iterative method* (BIM) (Kurakin et al., 2017), *Carlini & Wagner* (CW) attack (Carlini & Wagner, 2017b), *fast gradient sign method* (FGSM) (Goodfellow et al., 2015) and *projected gradient descent* (PGD) (Madry et al., 2018) under different target models that trained on either *CIFAR-10* or *ImageNet-1K*.

**Adaptive Attack Formulation.** To evaluate robustness under adaptive attacks, we follow the strategy of Gao et al. (2021) 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. The attack is designed to optimize two competing objectives: it increases the classification loss to ensure the generated examples remain adversarial, while simultaneously suppressing the detector's test statistic so that the adversarial batch appears statistically closer to the clean reference batch. Implementation details of the proposed adaptive attack can be found in Appendix E.3.

**Overall Result.** Our results demonstrate that USAD consistently achieves superior performance across all attacks. Notably, for all attacks, USAD reaches perfect detection (test power $\approx 1.0$) 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. Additional results in Figures 4 further show that USAD remains effective under $\ell_2$ attacks.

**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** (Schrab et al., 2023), **MMD-FUSE** (Biggs et al., 2023), and **MMD-DUAL** (Zhou et al., 2025)) 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*.

**Performance against Adaptive Attack.** Inspired by Gao et al. (2021); Kaya et al. (2022), 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 results further suggest that PCD better retains power under adaptive attacks: such attacks can suppress the global spread

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

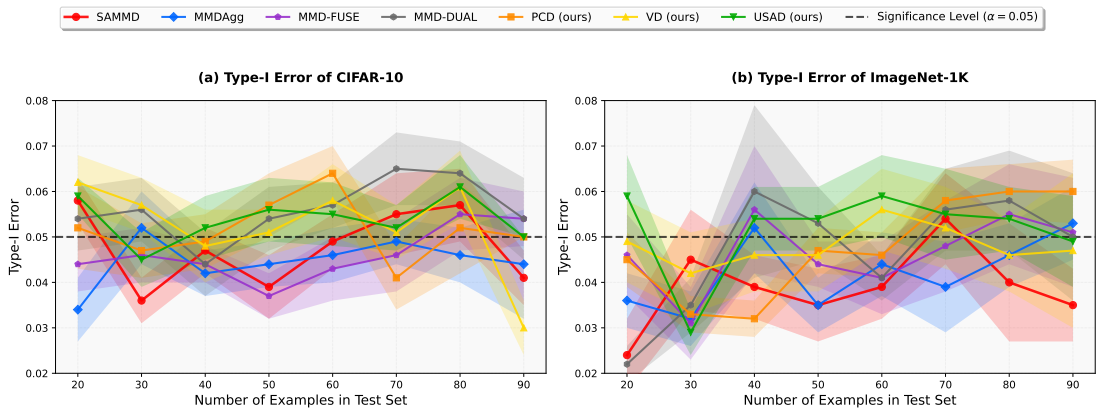



**Figure 5.** Results $(a-b)$ are type I error (false alarm rate) control check where given examples are actually drawn from clean examples under different clean example sizes. The results are averaged over $1,000$ repetitions and the ideal type I error is around the significance level $\alpha = 0.05$ (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 $(b)$.

signal used by VD, whereas 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 (Ma et al., 2018) and EPS-AD (Zhang et al., 2023)): adaptive attacks can reduce power in small batches, but as batch size grows, SAD consistently separates AEs from CEs.

**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 method attains the highest power across mixing levels. In particular, a non-negligible adversarial fraction induces a detectable shift, whereas a vanishing fraction (i.e., 100% CE) can become effectively indistinguishable from clean data, in which case the rejection rate (Type-I error) stays around the significance level $\alpha = 0.05$. Notably, even at a high CE mixing level of 80%, our VD statistic retains strong detection power (0.778), while USAD achieves 0.350 and remains the strongest aggregated method; in contrast, the MMD-based baselines achieve at most 0.096 power. We provide a more detailed discussion of this distribution-level perspective and the mixture regimes in Remark 1. Figure 5 further verifies that USAD maintains Type-I error close to the prescribed significance level $\alpha = 0.05$ on clean query batches.

**Additional Results.** Beyond the main experiments, Appendix F provides further evaluations under transfer attacks, additional architectures, sparse $\ell_0/\ell_1$ attacks, adversarial video detection, and computational efficiency. These experiments test whether the observed gains persist beyond the standard $\ell_\infty$ image-attack setting, including cases where AEs are generated from surrogate models or induce different geometric perturbation patterns. Overall, the results show that USAD remains effective across diverse threat models, architectures, and data modalities, while maintaining practical inference-time efficiency.

## 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 number DE240101089, LP240100101, DP230101540 and the NSF&CSIRO Responsible AI program with grant number 2303037.

## Impact Statement

This work on uncertainty-aware statistical adversarial detection raises important considerations regarding the reliability and robustness of machine learning systems deployed in security-critical settings. We carefully evaluate the proposed method on widely adopted benchmark datasets, multiple model architectures, and diverse adversarial attacks, following established hypothesis-testing protocols that ensure controlled false-alarm rates and fair comparison with prior statistical detection approaches. Our experimental results demonstrate that explicitly modeling distributional uncertainty can substantially improve detection power in small-sample regimes, addressing a key limitation of existing MMD-based SAD methods.

From a broader perspective, our method operates entirely at inference time and does not rely on adversarial training, attack-specific assumptions, or external supervision, reducing the risk of overfitting to known threats or amplifying unintended biases. We hope this work contributes to the development of more reliable and trustworthy defenses against adversarial attacks, and encourages further research into uncertainty-aware and statistically principled approaches for safe and robust machine learning deployment.

## References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: A query-efficient black-box adversarial attack via random search. In *ECCV*, 2020.

Biggs, F., Schrab, A., and Gretton, A. MMD-Fuse: Learning and combining kernels for two-sample testing without data splitting. In *NeurIPS*, 2023.

Bryniarski, O., Hingun, N., Pachuca, P., Wang, V., and Carlini, N. Evading adversarial example detection defenses with orthogonal projected gradient descent. In *ICLR*, 2022.

Cao, Y., Wang, N., Xiao, C., Yang, D., Fang, J., Yang, R., Chen, Q. A., Liu, M., and Li, B. 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*, pp. 176–194, 2021.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *AISec*, 2017a.

Carlini, N. and Wagner, D. A. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pp. 39–57. IEEE, 2017b.

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

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

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

Croce, F. and Hein, M. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020a.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020b.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Deng, Z., Yang, X., Xu, S., Su, H., and Zhu, J. LiBRe: A practical Bayesian approach to adversarial detection. In *CVPR*, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Fawzi, A., Moosavi-Dezfooli, S.-M., and Frossard, P. Robustness of classifiers: from adversarial to random noise. In *NeurIPS*, 2016.

Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B. Detecting adversarial samples from artifacts. *arXiv*, 2017.

Gao, R., Liu, F., Zhang, J., Han, B., Liu, T., Niu, G., and Sugiyama, M. Maximum mean discrepancy test is aware of adversarial attacks. In *ICML*, 2021.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P. On the (statistical) detection of adversarial examples. *arXiv*, 2017.

Han, B., Yao, J., Liu, T., Li, B., Koyejo, S., and Liu, F. *Trustworthy Machine Learning: From Data to Models*. Now Foundations and Trends, 2025.

Harchaoui, Z., Moulines, E., and Bach, F. Kernel change-point analysis. In *NeurIPS*, 2008.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hemerik, J. and Goeman, J. Exact testing with random permutations. *Test*, 27(4):811–825, 2018.

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

Huang, B., Wang, Y., and Wang, W. Model-agnostic adversarial detection by random perturbations. In *IJCAI*, 2019.

Huang, C., Lim, J. H., and Courville, A. C. A variational perspective on diffusion-based generative models and score matching. In *NeurIPS*, 2021.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

Jing, P., Tang, Q., Du, Y., Xue, L., Luo, X., Wang, T., Nie, S., and Wu, S. Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations. In *USENIX Security Symposium*, pp. 3237–3254, 2021.

Kang, M., Song, D., and Li, B. Diffattack: Evasion attacks against diffusion-based adversarial purification. In *NeurIPS*, 2023.

Kaya, Y., Zafar, M. B., Aydore, S., Rauschmayr, N., and Kenthapadi, K. Generating distributional adversarial examples to evade statistical detectors. In *ICML*, 2022.

Kim, I., Balakrishnan, S., and Wasserman, L. Minimax optimality of permutation tests. *Annals of Statistics*, 50 (1):225–251, 2022.

Krizhevsky, A., Nair, V., and Hinton, G. CIFAR-10 (canadian institute for advanced research). 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. In *ICLR, Workshop Track Proceedings*, 2017.

Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. In *NeurIPS*, 2019.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., and Sutherland, D. J. Learning deep kernels for non-parametric two-sample tests. In *ICML*, 2020.

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S. N. R., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. DeepFool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., Frossard, P., and Soatto, S. Robustness of classifiers to universal perturbations: A geometric perspective. In *ICLR*, 2018.

Muandet, K., Fukumizu, K., Sriperumbudur, B., and Schölkopf, B. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

Papoulis, A. and Pillai, S. U. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, 2001.

Roth, K., Kilcher, Y., and Hofmann, T. The odds are odd: A statistical test for detecting adversarial examples. In *ICML*, 2019.

Schrab, A., Kim, I., Guedj, B., and Gretton, A. Efficient aggregated kernel tests using incomplete U-statistics. In *NeurIPS*, 2022.

Schrab, A., Kim, I., Albert, M., Laurent, B., Guedj, B., and Gretton, A. MMD aggregated two-sample test. *Journal of Machine Learning Research*, 24:194:1–194:81, 2023.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Sun, Y., Zhang, J., Ye, Z., Xiao, C., and Liu, F. Sample-specific noise injection for diffusion-based adversarial purification. In *ICML*, 2025.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Tramèr, F. Detecting adversarial examples is (nearly) as hard as classifying them. In *ICML*, 2022.

Wei, X., Zhu, J., Yuan, S., and Su, H. Sparse adversarial perturbations for videos. In *AAAI*, 2019.

Xue, H., Araujo, A., Hu, B., and Chen, Y. Diffusion-based adversarial sample generation for improved stealthiness and controllability. In *NeurIPS*, 2023.

Yang, Z., Li, L., Xu, X., Kailkhura, B., Xie, T., and Li, B. On the certified robustness for ensemble models and beyond. In *ICLR*, 2022.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

Zhang, J., Liu, F., Zhou, D., Zhang, J., and Liu, T. Improving accuracy-robustness trade-off via pixel reweighted adversarial training. In *ICML*, 2024.

Zhang, J., Rubinstein, B. I. P., Zhang, J., and Liu, F. One stone, two birds: Enhancing adversarial defense through the lens of distributional discrepancy. In *ICML*, 2025.

Zhang, S., Liu, F., Yang, J., Yang, Y., Li, C., Han, B., and Tan, M. Detecting adversarial data by probing multiple perturbations using expected perturbation score. In *ICML*, 2023.

Zhong, X. and Liu, C. Sparse-pgd: A unified framework for sparse adversarial perturbations generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Zhou, Z., Ni, J., Yao, J.-H., and Gao, W. On the exploration of local significant differences for two-sample test. In *NeurIPS*, 2023.

Zhou, Z., Tian, X., Peng, L., Lei, C., Schrab, A., Sutherland, D. J., and Liu, F. DUAL: Learning diverse kernels for aggregated two-sample and independence testing. In *NeurIPS*, 2025.

## 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 treats the classifier as a black box without requiring 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 (Gao et al., 2021).

**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 (Tramèr, 2022). 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.

**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) (Gretton et al., 2012) as the test statistic $\mathcal{T}(X, Y)$ (Gao et al., 2021; Zhang et al., 2025). These MMD-based SAD methods have shown strong performance against *unknown* and even *adaptive* attacks, while enjoying rigorous false-alarm control through hypothesis testing. However, they are inherently *sample-inefficient*: 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 shown in Figure 1a, the test power of MMD-based SAD degrades rapidly as the query window size decreases, even under strong adversarial perturbations, where the induced distributional shift is often overwhelmed by statistical noise. This sample inefficiency critically limits practical deployment, since defenders cannot realistically wait for large query windows to accumulate in streaming or online settings, and attackers inherently control the number and timing of adversarial queries. As a result, MMD-based SAD methods can fail precisely in early-stage or low-volume attack regimes where timely detection is most important.

## B. Permutation Test

We detail the permutation test used throughout *statistical adversarial detection* (SAD) to calibrate the decision threshold $t_\alpha$ 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 testing 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 $\alpha$, we want a *test statistic* $\mathcal{T}(X, Y)$ and a *testing threshold* $t_\alpha$ 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 $\alpha$. 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 $t_\alpha$, which may be defined as an upper $(1 - \alpha)$-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 exchangeable*, and construct an empirical null distribution using a *permutation test* with $X$ and $Y$. We finally determine the *testing threshold* $t_\alpha$ calibrated using this empirical null distribution.

**Statistics Under Test.** For notational convenience, we write

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

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

For USAD, $\widehat{\Sigma}_{\mathrm{A}}$ is estimated as in Eq. (10) using an auxiliary clean calibration set $X_{\mathrm{cal}} \sim \mathbb{P}$ that is independent of $(X, Y)$. Let $\mathcal{A}$ denote $X_{\mathrm{cal}}$ and the randomness used in estimating $\widehat{\Sigma}_{\mathrm{A}}$. Conditional on $\mathcal{A}$, $\widehat{\Sigma}_{\mathrm{A}}$ is fixed, and the same matrix is used to compute both the observed and all permuted USAD statistics. For VD and PCD, $\mathcal{A}$ may be taken to be empty. For PCD and USAD, each sample-specific covariance matrix $\Sigma_{\boldsymbol{z}}$ is constructed using perturbation randomness independent of the reference/query labels, and the pair $(\boldsymbol{z}, \Sigma_{\boldsymbol{z}})$ is permuted together.

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

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

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

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

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

Under $\boldsymbol{H}_0$, the entries of $Z$ are i.i.d., so $Z$ is exchangeable: permuting its indices does not change its distribution. Let

$$\mathcal{O}(Z) \triangleq \{\boldsymbol{\pi} Z : \boldsymbol{\pi} \in \boldsymbol{\Pi}_{n+m}\}$$

denote the permutation orbit of $Z$. Conditional on $\mathcal{A}$ and $\mathcal{O}(Z)$, the observed reference/query assignment is uniformly distributed over the assignments induced by $\boldsymbol{\Pi}_{n+m}$. Because the same statistic, including the same fixed $\widehat{\Sigma}_{\mathrm{A}}$ for USAD, is applied to every assignment, the resulting permutation distribution is a valid conditional null distribution.

**Monte-Carlo Approximation.** Enumerating all $\binom{n+m}{n}$ distinct reference/query partitions is usually impossible, so we approximate the permutation distribution by sampling $R$ permutations. Let

$$\mathrm{STAT}^{(0)} \triangleq \mathrm{STAT}(X, Y; \phi)$$

denote the observed statistic. For each $r = 1, \ldots, R$, we

1. draw $\boldsymbol{\pi}_{(r)} \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(\boldsymbol{\Pi}_{n+m})$, independently of $Z$ and $\mathcal{A}$;
2. construct $X_{(r)} = X_{\boldsymbol{\pi}_{(r)}}$ and $Y_{(r)} = Y_{\boldsymbol{\pi}_{(r)}}$;
3. compute
$$\mathrm{STAT}^{(r)} = \mathrm{STAT}(\boldsymbol{\pi}_{(r)} Z; \phi) = \mathrm{STAT}(X_{(r)}, Y_{(r)}; \phi).$$

This gives $R$ permutation statistics $\{\mathrm{STAT}^{(r)}\}_{r=1}^{R}$. Let

$$\mathrm{STAT}_{[1]} \leq \cdots \leq \mathrm{STAT}_{[R]}$$

denote their ordered values, and define

$$k_{\alpha,R} = \lceil (1-\alpha)(R+1) \rceil .$$

The finite-$R$ rank-corrected *testing threshold* is

$$t_\alpha\big(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R\big) = \begin{cases} \mathrm{STAT}_{[k_{\alpha,R}]}, & k_{\alpha,R} \le R, \\ +\infty, & k_{\alpha,R} = R+1. \end{cases} \tag{11}$$

Equivalently, the corresponding Monte-Carlo permutation $p$-value is

$$\widehat{p}_R = \frac{1 + \sum_{r=1}^R \mathbb{I}\Big[\mathrm{STAT}^{(r)} \ge \mathrm{STAT}^{(0)}\Big]}{R+1}. \tag{12}$$

The additive one accounts for the observed assignment, while the use of "$\ge$" makes the test conservative in the presence of ties. By construction,

$$\widehat{p}_R \le \alpha \quad \Longleftrightarrow \quad \mathrm{STAT}^{(0)} > t_\alpha\big(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R\big).$$

**Decision Rule.** Finally, the test with observed statistic $\mathrm{STAT}(X, Y; \phi)$ is defined as

$$\begin{aligned} \delta_\alpha &= \mathbb{I}\left[\mathrm{STAT}(X, Y; \phi) > t_\alpha\big(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R\big)\right] \\ &= \mathbb{I}\left[\widehat{p}_R \le \alpha\right]. \end{aligned} \tag{13}$$

Conditional on $\mathcal{A}$ and $\mathcal{O}(Z)$, the observed labeling and the $R$ randomly permuted labelings are exchangeable. Therefore, $\widehat{p}_R$ is super-uniform under $\boldsymbol{H}_0$, yielding

$$\Pr\left(\delta_\alpha = 1 \mid \mathcal{A}\right) \le \alpha.$$

Averaging over the independent calibration set and its calibration randomness gives

$$\Pr_{\boldsymbol{H}_0}(\delta_\alpha = 1) \le \alpha.$$

Thus, the false-alarm probability is controlled at the prescribed significance level.

## C. 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 (Gretton et al., 2012). 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 (Gretton et al., 2012; Muandet et al., 2017). 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 $\mathbb{R}$ on $\mathcal{X}$ with $E_{r\sim\mathbb{R}}\left[\sqrt{\kappa(r,r)}\right] < \infty$, its kernel mean embedding is

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

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 $\kappa$ is *characteristic*, this embedding is injective (Biggs et al., 2023):

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

Kernel-based TST, therefore, rephrases the hypotheses as

$$\boldsymbol{H}_0^{\mathrm{M}} : \boldsymbol{\mu}_{\mathbb{P}} = \boldsymbol{\mu}_{\mathbb{Q}} \quad \text{and} \quad \boldsymbol{H}_1^{\mathrm{M}} : \boldsymbol{\mu}_{\mathbb{P}} \neq \boldsymbol{\mu}_{\mathbb{Q}} \,, \tag{14}$$

and bases the test on a norm of $\boldsymbol{\mu}_{\mathbb{P}} - \boldsymbol{\mu}_{\mathbb{Q}}$. Specifically, the *mean maximum discrepancy* (MMD) statistic (Gretton et al., 2012) uses the squared RKHS distance (i.e., $\ell_2$ norm) between these mean embeddings, which admits a simple expression in terms of kernel evaluations.

**Maximum Mean Discrepancy (MMD)** (Gretton et al., 2012) is a popular kernel-based metric that assesses the kernel mean embedding equality with kernel $\kappa$:

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

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 $\mathbb{P} = \mathbb{Q}$ (Gretton et al., 2012; Biggs et al., 2023).

**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 $E\left[\widehat{\mathrm{MMD}}^2(X, Y; \kappa)\right] = \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; \kappa)$ (Gretton et al., 2012).

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 B) is applied to $\widehat{\mathrm{MMD}}^2(X, Y; \kappa)$ to obtain a threshold $t_\alpha$ such that $\Pr(\mathcal{T}(X, Y) > t_\alpha \mid \boldsymbol{H}_0) \approx \alpha$.

**MMD on Semantic Features.** In existing SAD practices (Gao et al., 2021; Zhang et al., 2025), 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.

## D. 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.** *(Kim et al., 2022) 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 $Y = \{\boldsymbol{y}_i\}_{i=1}^m \sim \mathbb{Q}^m$. 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.** *(Hoeffding, 1994) If the function $h$ is bounded, that is, $a \leq h(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y}_1, \boldsymbol{y}_2) \leq b$, and assume $n \leq m$, then for any $t > 0$,*

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

*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 $\mathbf{\Pi}_{n+m}$ 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 $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n, \pi_{n+1}, \ldots, \pi_{n+m}) \in \mathbf{\Pi}_{n+m}$, 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.** *(Kim et al., 2022, Theorem 6.1) Consider the permuted two-sample $U$-statistic $U(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \phi)$ and assume $n \leq m$. Define*

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

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

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

### D.1. Detailed Proofs of Theorem 1

We begin with a useful definition as follows.

**Definition 8.** *(Hemerik & Goeman, 2018) Let $Z$ be the sample taking values in the instance space $\mathcal{X}$. Let $\mathcal{G}$ be a finite set of transformations $g : \mathcal{X} \to \mathcal{X}$, such that $\mathcal{G}$ is a group with respect to the operation of composition of transformations. Let $\mathcal{H}_0$ be any null hypothesis which implies that the joint distribution of the test statistics $T(gZ)$, $g \in \mathcal{G}$, is invariant under all transformations in $\mathcal{G}$ of $Z$. Denote by $R$ the cardinality of the set $\mathcal{G}$ and write $\mathcal{G} = \{g_1, ..., g_R\}$. We have, under $\mathcal{H}_0$,*

$$(T(g_1 Z), ..., T(g_R Z)) \stackrel{d}{=} (T(g \cdot g_1 Z), ..., T(g \cdot g_R Z)) \quad \textit{for all } g \in \mathcal{G} ,$$

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

We present the proofs of Theorem 1 as follows

*Proof.* For USAD, let $\mathcal{A}$ denote the auxiliary clean calibration set $X_{\text{cal}}$ and the randomness used to estimate $\widehat{\Sigma}_{\text{A}}$. By construction, $\mathcal{A}$ is independent of $(X, Y)$. We condition throughout on $\mathcal{A}$, under which $\widehat{\Sigma}_{\text{A}}$ is fixed and the same matrix is used for the observed statistic and all permuted statistics. For VD and PCD, $\mathcal{A}$ may be taken to be trivial. For PCD, each sample and its perturbation-induced covariance matrix are treated as one augmented observation and are permuted together.

It is easy to see that the permutation set $\mathbf{\Pi}_{n+m}$ 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, conditional on $\mathcal{A}$, $Z$ is exchangeable with respect to $\mathbf{\Pi}_{n+m}$, and the joint law of

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

is invariant under left composition by any $\boldsymbol{\pi} \in \mathbf{\Pi}_{n+m}$. Therefore, Definition 8 applies.

Let

$$M = |\mathbf{\Pi}_{n+m}| = (n+m)!$$

and write $\mathbf{\Pi}_{n+m} = \{\boldsymbol{\pi}_{(1)}, \ldots, \boldsymbol{\pi}_{(M)}\}$. The identity permutation is included, so $\text{STAT}(Z; \phi) = \text{STAT}(X, Y; \phi)$ is one element of the multiset

$$\left\{ \text{STAT}(\boldsymbol{\pi}_{(1)} Z; \phi), \ldots, \text{STAT}(\boldsymbol{\pi}_{(M)} Z; \phi) \right\} .$$

By Definition 8, for a uniformly distributed $\boldsymbol{\pi} \sim \text{Unif}(\mathbf{\Pi}_{n+m})$ independent of $Z$,

$$\begin{aligned} &\left( \text{STAT}(\boldsymbol{\pi}_{(1)} Z; \phi), \ldots, \text{STAT}(\boldsymbol{\pi}_{(M)} Z; \phi) \right) \\ &\quad \stackrel{d}{=} \left( \text{STAT}(\boldsymbol{\pi} \cdot \boldsymbol{\pi}_{(1)} Z; \phi), \ldots, \text{STAT}(\boldsymbol{\pi} \cdot \boldsymbol{\pi}_{(M)} Z; \phi) \right), \end{aligned}$$

conditional on $\mathcal{A}$. Since left composition by $\boldsymbol{\pi}$ is a bijection of $\mathbf{\Pi}_{n+m}$, conditional on the permutation orbit

$$\mathcal{O}(Z) \triangleq \{\boldsymbol{\pi} Z : \boldsymbol{\pi} \in \mathbf{\Pi}_{n+m}\},$$

the observed labeling is uniformly distributed over the orbit. Consequently, the full permutation $p$-value

$$p_{\text{full}} = \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}\left[\text{STAT}(\boldsymbol{\pi}_{(i)}Z;\phi) \geq \text{STAT}(Z;\phi)\right]$$

is super-uniform under $\boldsymbol{H}_0$. Equivalently, the permutation rank is uniform after random tie-breaking; counting ties as exceedances makes the resulting test conservative.

In practical implementation (see Appendix B), we approximate the full permutation distribution using $R < M$ independently sampled permutations

$$\boldsymbol{\pi}_{(1)}, \ldots, \boldsymbol{\pi}_{(R)} \overset{\text{i.i.d.}}{\sim} \text{Unif}(\boldsymbol{\Pi}_{n+m}),$$

independently of $Z$ and $\mathcal{A}$. Define

$$\text{STAT}^{(0)} = \text{STAT}(Z;\phi), \qquad \text{STAT}^{(r)} = \text{STAT}(\boldsymbol{\pi}_{(r)}Z;\phi), \quad r = 1, \ldots, R.$$

Conditional on $\mathcal{A}$ and $\mathcal{O}(Z)$, the observed labeling and the $R$ randomly permuted labelings are exchangeable. Hence,

$$\left(\text{STAT}^{(0)}, \text{STAT}^{(1)}, \ldots, \text{STAT}^{(R)}\right)$$

is exchangeable, and the Monte-Carlo permutation $p$-value

$$\widehat{p}_R = \frac{1 + \sum_{r=1}^{R} \mathbb{I}\left[\text{STAT}^{(r)} \geq \text{STAT}^{(0)}\right]}{R+1}$$

is super-uniform under $\boldsymbol{H}_0$.

To express the same test using a threshold, let

$$\text{STAT}_{[1]} \leq \cdots \leq \text{STAT}_{[R]}$$

be the ordered values of $\{\text{STAT}^{(r)}\}_{r=1}^{R}$ and define

$$k_{\alpha,R} = \lceil (1-\alpha)(R+1) \rceil,$$

and

$$\tau_\alpha\big(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^{R}\big) = \begin{cases} \text{STAT}_{[k_{\alpha,R}]}, & k_{\alpha,R} \leq R, \\ +\infty, & k_{\alpha,R} = R+1. \end{cases}$$

By construction,

$$\widehat{p}_R \leq \alpha \quad \Longleftrightarrow \quad \text{STAT}(Z;\phi) > \tau_\alpha\big(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^{R}\big).$$

Therefore,

$$\begin{aligned} \Pr(\delta_\alpha = 1 \mid \mathcal{A}) &\equiv \Pr\left(\text{STAT}(Z;\phi) > \tau_\alpha\big(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^{R}\big) \;\middle|\; \mathcal{A}\right) \\ &= \Pr(\widehat{p}_R \leq \alpha \mid \mathcal{A}) \leq \alpha. \end{aligned}$$

Finally, averaging over the independent calibration data and its calibration randomness yields

$$\Pr_{\boldsymbol{H}_0}(\delta_\alpha = 1) \leq \alpha.$$

□

### D.2. Detailed Proofs of Theorem 2

We present the proofs of Theorem 2 as follows

*Proof.* Recall the definition of VD as

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

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 $\ell$ is bounded on the range of $f$: $|\ell(f(\boldsymbol{x}), f(\boldsymbol{y}))| \leq K$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ with the constant $K > 0$. Then $|V_{\mathbb{P}}|, |V_{\mathbb{Q}}|, |\widehat{V}(X; \ell)|, |\widehat{V}(Y; \ell)| \leq K$, and hence

$$\begin{aligned}
\left|\mathrm{VD}(\mathbb{P}, \mathbb{Q}; \ell) - \widehat{\mathrm{VD}}(X, Y; \ell)\right| \quad &= \quad \left|(V_{\mathbb{P}} - V_{\mathbb{Q}})^2 - \left(\widehat{V}(X; \ell) - \widehat{V}(Y; \ell)\right)^2\right| \\
&= \quad \left(\widehat{V}(X; \ell) - \widehat{V}(Y; \ell) - (V_{\mathbb{P}} - V_{\mathbb{Q}})\right)\left(\widehat{V}(X; \ell) - \widehat{V}(Y; \ell) + (V_{\mathbb{P}} - V_{\mathbb{Q}})\right) \\
&\leq \quad \left|\widehat{V}(X; \ell) - \widehat{V}(Y; \ell) - (V_{\mathbb{P}} - V_{\mathbb{Q}})\right| \cdot \left|\widehat{V}(X; \ell) - \widehat{V}(Y; \ell) + (V_{\mathbb{P}} - V_{\mathbb{Q}})\right| \\
&< \quad 4K \cdot \left|\widehat{V}(X; \ell) - \widehat{V}(Y; \ell) - (V_{\mathbb{P}} - V_{\mathbb{Q}})\right| \ . \qquad (15)
\end{aligned}$$

Here, the estimator of $V_{\mathbb{P}} - V_{\mathbb{Q}}$ can be further present 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, where $\xrightarrow{P}$ denotes convergence in probability, and hence

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

by Eq. (15).

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

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

based on Theorem 7 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

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

The empirical adjusted $(1-\alpha)$-quantile $\tau_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)$ (See Appendix B with $\mathrm{STAT}(X, Y; \phi) = \widehat{\mathrm{VD}}(X, Y; \ell)$) of VD statistic also converges to 0.

which implies, by Slutsky's theorem (Papoulis & Pillai, 2001),

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

where the test $\delta_\alpha$ is the permutation procedure (see Eq. (13) in Appendix B with $\mathrm{STAT}(X, Y; \phi) = \widehat{\mathrm{VD}}(X, Y; \ell)$). □

### D.3. Detailed Proofs of Theorem 3

We present the proofs of Theorem 3 as follows

*Proof.* Recall the definition of PCD as

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

The corresponding estimator is

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

which can be further present 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

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

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

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

by Theorem 6.

Under a uniformly random permutation $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, the two-sample $U$-statistic with permuted samples $X_{\boldsymbol{\pi}}$ and $Y_{\boldsymbol{\pi}}$ exhibit 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 $n/m \to c \in (0, \infty)$. Consequently, the empirical adjusted $(1-\alpha)$-quantile $\tau_\alpha(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R)$ (See Appendix B with $\mathrm{STAT}(X, Y; \phi) = \widehat{\mathrm{PCD}}(X, Y; \rho)$) of PCD statistic also converges to 0. This implies, by Slutsky's theorem (Papoulis & Pillai, 2001),

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

where the test $\delta_\alpha$ is the permutation procedure (see Eq. (13) in Appendix B with $\mathrm{STAT}(X, Y; \phi) = \widehat{\mathrm{PCD}}(X, Y; \rho)$). □

### D.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 $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, we have

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

and

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

which imply that

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

by Slutsky's theorem (Papoulis & Pillai, 2001).

Then, we have

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

where $\Sigma_{\mathrm{A}}$ is fixed in the testing procedure.

Consequently, the permutation distribution of $\mathcal{T}^{\mathrm{A}}(X_{\boldsymbol{\pi}}, Y_{\boldsymbol{\pi}}; \ell, \rho)$ concentrates at zero, and the empirical adjusted $(1-\alpha)$-quantile $\tau_\alpha\big(Z, \{\boldsymbol{\pi}_{(r)}\}_{r=1}^R\big)$ (see Appendix B, with $\mathrm{STAT}(X, Y; \phi) = \mathcal{T}^{\mathrm{A}}(X, Y; \ell, \rho)$) converges to 0. Similarly, if **either or both** of the alternatives $\boldsymbol{H}_1^{\mathrm{V}}$ and $\boldsymbol{H}_1^{\mathrm{C}}$ hold, then at least one of VD or PCD converges to a positive constant, and thus $\mathcal{T}(X, Y; \ell, \rho)$ converges to a nonzero vector. It follows that the observed USAD value $\mathcal{T}^{\mathrm{A}}(X, Y; \ell, \rho)$ converges to a positive constant.

Therefore, by Slutsky's theorem (Papoulis & Pillai, 2001),

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

where the test $\delta_\alpha$ is the permutation procedure (see Eq. (13) in Appendix B with $\mathrm{STAT}(X, Y; \phi) = \mathcal{T}^{\mathrm{A}}(X, Y; \ell, \rho)$). □

## E. Experimental Details

### E.1. Aggregation Strategy

DUAL (Zhou et al., 2025) 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 (Schrab et al., 2023) 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 behaviours. MMDFuse (Biggs et al., 2023) 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.

### E.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., $X = \{\boldsymbol{x}_j\}_{j=1}^n$, 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 permutations used in PCD, and $p$ to denote the dimension of the semantic feature. We denote by $C_\ell$ and $C_\rho$ the computational cost of evaluating the kernel functions $\ell$ and $\rho$ 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 $C_\ell$, 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 $p$-dimensional feature space, resulting in a total cost of $\mathcal{O}((n+m)Kp^2)$. After obtaining the covariance matrix for every example, the empirical PCD estimator in Eqn. (8) requires evaluating the kernel $\rho$ 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

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

**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

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

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

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

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 inner products required by VD and PCD are already computed once when evaluating the statistics on the original batches. 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 F.5.

### E.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 (will be provided on our GitHub repository after reviewing period) for feature extraction and classification.

**Experimental Setup for Datasets.** We employ two primary datasets: *CIFAR-10* (Krizhevsky et al., 2009) and *ImageNet-1K* (Deng et al., 2009). 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. (2021), 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

**Table 4.** Test power of dataset *CIFAR-10* under five different attacks with different perturbation budget $\epsilon$. 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 | $\epsilon$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 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** |

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. (2021). For the aggregated-statistic two-sample testing methods MMDAgg, MMD-FUSE, and MMD-DUAL, we follow the default experimental settings from Zhou et al. (2025), 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.

**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 (Schrab et al., 2022; Biggs et al., 2023) computed from the clean feature set.

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

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

**Table 5.** Test power of dataset *ImageNet-1K* under five different attacks with different perturbation budget $\epsilon$. 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 | $\epsilon$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 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** |

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 $\ell_\infty$-norm and $\ell_2$-norm to comprehensively evaluate the detection ability of USAD, including *AutoAttack* (AA) (Croce & Hein, 2020b), *basic iterative method* (BIM) (Kurakin et al., 2017), *Carlini & Wagner* (CW) attack (Carlini & Wagner, 2017b), *fast gradient sign method* (FGSM) (Goodfellow et al., 2015) and *projected gradient descent* (PGD) (Madry et al., 2018). For $\ell_\infty$-norm attacks, following Gao et al. (2021), we set the maximum allowed perturbation budget $\epsilon$ to range from $1/255$ to $8/255$. For $\ell_2$-norm attacks, to ensure successful attacks against the threat model, we increase $\epsilon$ 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 $\ell_\infty$-norm and $\ell_2$-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. (2021) 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

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

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

**Table 6.** Test power of dataset *ImageNet-1K* under five different attacks with different perturbation budget $\epsilon$. 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 | $\epsilon$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 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** |

## F. Additional Experiments

This section complements the results reported in Section 6 with six additional groups of experiments. In summary, we (i) evaluate the proposed detector under $\ell_2$ norm attacks and compare it against baselines (Section F.1); (ii) report detailed numerical results across perturbation budgets and additional architectures (Section F.2); (iii) evaluate robustness to transfer attacks from surrogate target models (Section F.3); and (iv) empirically verify Type-I error control (Section F.4); (v) measure inference-time computational efficiency against competing detectors (Section F.5); (vi) evaluate the proposed detector against sparse $\ell_0$ (Sparse-PGD) and $\ell_1$ adversarial attacks (Section F.6); and (vii) extend our evaluation to adversarial video detection on UCF101 (Section F.7). Unless stated otherwise, we follow the hypothesis-testing protocol with the main text: significance level $\alpha = 0.05$ and adversarial batch size $|Y| = 50$.

### F.1. Detection under $L_2$ Norm Attacks

The main text (Section 6) primarily reports the detection performance on $\ell_\infty$-bounded adversarial perturbations, following the previous studies (Gao et al., 2021). To check that our detectors are not tied to this particular norm, we also evaluate our method under $\ell_2$-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 $\ell_2$ norm. We fix the adversarial batch size to $|Y| = 50$ and vary the perturbation budget $\epsilon$, averaging over $1{,}000$ repetitions.

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 $\ell_\infty$- and $\ell_2$-bounded attacks.

### F.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

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

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.

### F.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 ViT-B/16, 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 (WRN-70), and Shifted window (Swin) Transformer) but detected using features extracted from Res18. This setting tests the transferability and generalizability of our detection methods across different model architectures, which is crucial for practical deployment where the exact attack generation process is unknown. Our proposed methods demonstrate exceptional robustness to transfer attacks across all target models and attack types. Notably, both PCD and USAD achieve perfect detection ($1.000 \pm 0.000$) 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 ($0.999 \pm 0.001$) and BIM attacks against WRN28 ($0.993 \pm 0.001$), 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

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

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 method, which also employs aggregation but with our novel permutation-based and variational distance-based test statistics, 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 $1.000 \pm 0.000$ 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 $\epsilon$ 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

**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 |

**Table 10.** Detection power against $\ell_1$-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 | $\ell_1$-N10 | $\ell_1$-N20 | $\ell_1$-N30 | SPGD-$k$10 | SPGD-$k$20 | SPGD-$k$50 | SPGD-$k$100 |
|---|---|---|---|---|---|---|---|
| PCD | $0.231 \pm 0.009$ | $0.476 \pm 0.009$ | $0.719 \pm 0.011$ | $0.991 \pm 0.003$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| VD | $0.977 \pm 0.005$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| USAD | $0.981 \pm 0.004$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |

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.

### F.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 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.

### F.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 we mentioned in E.3. Our VD method demonstrates superior computational efficiency across both datasets, achieving $0.016$s on CIFAR-10 and $0.015$s 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 $(0.024\text{s})$, it increases substantially to $0.097$s on ImageNet-1K, suggesting quadratic or higher complexity with respect to feature dimensionality. In contrast, PCD and USAD exhibit significantly longer inference times ($0.117$s and $0.132$s on CIFAR-10; $0.759$s and $0.772$s 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.

### F.6. $\ell_0$ (Sparse-PGD) and $\ell_1$ Attacks.

While the main paper focuses on dense $\ell_\infty$ and $\ell_2$ 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 $\ell_0$-bounded perturbations generated by Sparse-PGD (SPGD) (Zhong & Liu, 2025) and $\ell_1$-bounded perturbations generated by $\ell_1$-APGD. For SPGD, we vary the sparsity budget $k \in \{10, 20, 50, 100\}$ (i.e., the number of perturbed pixels), and for $\ell_1$-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

**Table 11.** Adversarial video detection on UCF101 against the sparse video attack of (Wei et al., 2019). 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 | $81.5\%$ | $0.0\%$ |

all configurations, with only marginal degradation under the most restrictive $\ell_1$ budget ($\ell_1$-N10). PCD shows a sharper sensitivity to the perturbation budget under $\ell_1$ attacks, but recovers to near-perfect detection as the budget grows. These results confirm that the proposed framework generalizes beyond dense $\ell_\infty/\ell_2$ threat models and is robust to the geometrically distinct distortions induced by sparse adversarial attacks.

### F.7. Extension to Adversarial Video Detection on UCF101.

We also include a preliminary study that explores whether our sample-wise distributional detection framework can be applied beyond static images. Video data provides a natural setting for such an extension, since a short clip contains multiple temporally related frames that can be treated as a sample set. However, we emphasize that our current method does not explicitly model temporal correlations across frames. Therefore, this experiment should be viewed as a proof-of-concept rather than a dedicated adversarial video detection method. Specifically, adversarial video attacks often introduce perturbations across multiple frames, which may induce a distributional discrepancy between clean and adversarial frame sets. This makes video detection a potentially interesting future direction for multi-sample detectors such as ours, although fully exploiting temporal structure would require additional modeling beyond the present framework.

**Setup.** We attack 65 sample videos drawn from the first two classes in alphabetical order (*ApplyEyeMakeup* and *Archery*) of the UCF101 action recognition benchmark, using the sparse video attack proposed by Wei et al. (2019). For each successfully attacked video, we extract a 40-frame clip from both its clean and adversarial versions, and treat the frames within each clip as a sample set. 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 reports the result of this preliminary evaluation. Our method achieves a detection rate of $81.5\%$ while maintaining $0.0\%$ type-I error in this small-scale setting. These results suggest that sample-wise distributional testing may transfer to video-domain adversarial detection when frame-level perturbations accumulate into a detectable distributional shift. At the same time, this experiment does not establish a complete adversarial video detection framework, as it does not compare against specialized video detection baselines or explicitly account for temporal dependencies. Developing detectors that combine distributional testing with temporal modeling is an interesting direction for future work.