KURTOSIS-GUIDED DENOISING SCORE MATCHING FOR TABULAR ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Denoising models provide a way to learn data distributions from noise-corrupted samples. We show that at a fixed default noise scale, denoising score matching (DSM) provides strong tabular anomaly detection by evaluating the score norm on clean inputs in one forward pass. However, its performance depends on the noise scale: insufficient noise leaves sparse regions poorly represented, while excessive noise blurs local structure. A shared scale also overlooks differences in marginal shape, and tuning this trade-off is difficult without anomaly labels. We introduce kurtosis-guided denoising score matching (K-DSM), which allocates feature-specific noise scales from rearranged marginal histograms of the training data. Rearrangement separates marginal concentration from the locations of modes, while a local coverage analysis motivates assigning larger noise scales to profiles with higher kurtosis. The network trains on the original joint observations and scores clean inputs by the norm of its predicted correction in noise units. On 57 ADBench datasets with normal-only training, K-DSM raises mean AUC-PR from 0.585 to 0.627 and outperforms single-scale DSM at every tested base noise scale. It achieves the highest mean AUC-PR among evaluated baselines, with competitive AUC-ROC, while requiring one forward pass per input compared with fifteen for the strongest AUC-PR baseline. For contaminated training data, an exponential moving average teacher excludes likely anomalies from training updates, raising K-DSM's mean AUC-PR from 0.189 to 0.343 while preserving one-pass inference.

1 INTRODUCTION

Anomaly detection involves identifying patterns or observations that deviate from the expected or typical data distribution (Pang et al., 2020). Detecting anomalies is critical in fields including fraud detection (Lee et al., 2020), disease identification in healthcare (Li et al., 2018), and fault detection in manufacturing processes (Quiroz et al., 2017). In these applications, inference speed determines how much data can be examined and how quickly anomalies can be identified. Credit-card fraud screening requires processing millions of transactions (Carcillo et al., 2018). Network intrusion detectors must keep pace with incoming traffic (Mirsky et al., 2018), while industrial inspection can impose strict time limits set by production rates (Batzner et al., 2024). Low inference cost is therefore important for screening datasets within a fixed computational budget and for detecting anomalies in real time.

Denoising score matching (DSM) provides a simple approach to this requirement: a trained score network can assign an anomaly score with a single forward pass. Through the connection between score matching and denoising autoencoders (Vincent, 2011), DSM learns the gradient of the log-density of noise-corrupted data, whose norm can be evaluated directly on a clean test observation. However, existing denoising anomaly detection approaches use noise-conditioned networks (Song & Ermon, 2019) to combine score information across multiple noise scales (Mahmood et al., 2021), or aggregate responses to repeated test-time perturbations (Hirth et al., 2026), increasing inference cost. We show that single-scale DSM at a fixed noise scale already achieves strong tabular anomaly detection through direct score-norm evaluation. This makes the choice of training noise central: it must support accurate detection while retaining the simplicity and speed of a single-pass detector.

The effectiveness of this single-pass detector, however, depends on the training noise scale. Too little noise leaves sparse regions poorly represented in the denoising objective, while too much noise blurs local structure that may distinguish anomalies from normal observations. A shared scale makes

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

this trade-off particularly difficult for tabular data: standardization equalizes feature variances, but their marginal concentration and tail shape can still differ substantially. Increasing the noise to reach sparse regions of one feature may therefore introduce unnecessary perturbations in another. Selecting an appropriate scale is also difficult when anomaly labels are unavailable for validation. These limitations motivate calibrating the noise separately for each feature using the training data.

We introduce kurtosis-guided denoising score matching (K-DSM), which allocates feature-specific noise scales using the kurtosis of rearranged marginal histograms. Rearrangement produces a symmetric, unimodal profile that allows us to assess marginal concentration without retaining the original asymmetry or arrangement of modes. Its kurtosis then guides the noise allocation, with larger scales assigned to features whose rearranged profiles have more pronounced tails. We motivate this calibration through a coverage analysis that relates marginal concentration to the perturbation needed to reach low-density regions. Rearrangement is used only to determine the noise scales: the network trains on the original joint observations and retains the ability to learn dependencies between features. At inference, K-DSM scores clean inputs using the norm of its restoration field in noise units, preserving the single forward pass.

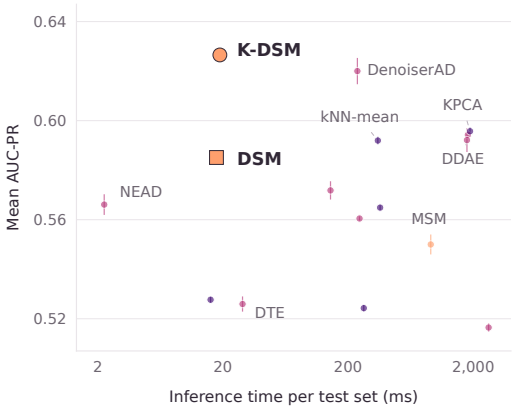


Figure 1: Mean AUC-PR versus mean inference time on ADBench (five seeds). K-DSM lies toward the upper left, combining strong detection performance with low inference cost. K-DSM gains 0.042 AUC-PR over DSM.

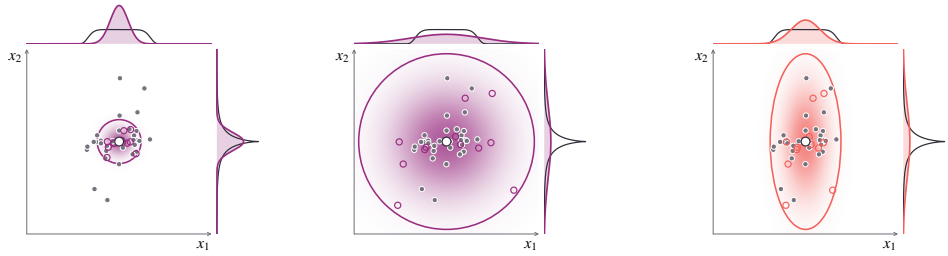
Our primary evaluation considers the semi-supervised setting, where training data contain only normal observations. Across 57 ADBench datasets (Han et al., 2022), K-DSM improves mean AUC-PR by 0.042 over single-scale DSM, winning on 38 of 57 datasets, and stays ahead at every tested base scale. Its mean AUC-PR exceeds every baseline, and it retains competitive AUC-ROC with one-pass inference (Figure 1). We also extend the method to the fully unsupervised setting, where training data may contain anomalies. An EMA-teacher filtering rule excludes likely anomalous observations before each gradient step, improving detection under contamination. Filtering adds computation during training but leaves inference unchanged. Our contributions are as follows:

- **Single-pass DSM for tabular anomaly detection.** Single-scale DSM at a fixed noise scale achieves *0.585 mean AUC-PR* by evaluating the score norm on clean inputs in one forward pass. This result complements recent findings on fixed-noise denoisers (Hirth et al., 2026).
- **Kurtosis-guided noise calibration.** K-DSM sets each feature's noise scale from the kurtosis of its rearranged histogram: higher kurtosis gives larger perturbations to reach sparse regions. A local coverage analysis motivates this rule, which preserves single-pass inference.
- **Strong detection performance with fast inference.** On 57 ADBench datasets, K-DSM reaches *0.627 mean AUC-PR*, above every baseline, and *0.823 mean AUC-ROC*, scoring each input in one forward pass (Figure 1). It improves on single-scale DSM by *0.042*, stays ahead at every tested base scale, and the gain persists when everything except the noise allocation is held fixed.
- **Extension to contaminated training.** An EMA-teacher filtering rule excludes likely anomalous training observations before each gradient step. It raises mean AUC-PR under contamination for K-DSM (*0.189 to 0.343*), DSM, and DDAE (Sattarov et al., 2025). K-DSM-EMA achieves the highest mean AUC-PR and F1 among the evaluated methods.

2 Kurtosis-Guided Denoising Score Matching

Denoising score matching. DSM (Vincent, 2011) fits a network s_θ to the score $\nabla_x \log p_{\text{data}}$, which is defined without the normalizing constant (Hyvärinen, 2005), by regressing on the conditional score of a known corruption kernel $q_\sigma(\tilde{x} \mid x)$:

$$\mathcal{L}_{\text{DSM}}(\theta) \;:=\; \frac{1}{2}\,\mathbb{E}_{x\sim p_{\text{data}}}\,\mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x}|x)}\left[\left\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x} \mid x)\right\|_2^2\right]. \tag{1}$$



(a) DSM: small global noise (b) DSM: large global noise (c) K-DSM: calibrated per-feature noise

Figure 2: **K-DSM improves coverage for better training supervision.** (a) Small shared noise limits coverage of the sparse tails of x_2, where anomalies may occur. (b) Large shared noise reduces the concentration of training inputs near the mode along x_1. (c) K-DSM uses feature-specific scales to extend tail coverage while retaining local supervision. Gray points show the data; colored circles show perturbed training points. Contours enclose 95% noise probability; marginal curves show data (black) and noise (colored).

Its minimizer is the score of the perturbed density $q_\sigma(\tilde{x}) = \int q_\sigma(\tilde{x} \mid x)\, p_{\text{data}}(x)\, dx$ (Vincent, 2011). For Gaussian corruption $\tilde{x} = x + \sigma\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$, the target is $-(\tilde{x} - x)/\sigma^2$, so the network learns the displacement back to the clean data, and the noise scale determines where it receives supervision. By *coverage* of a region, we mean the probability that a corrupted training input falls within it. Regions receiving few such inputs provide little direct supervision for score estimation, so coverage matters both near normal modes and in sparse regions where potential anomalies must be scored.

Standard DSM uses a shared noise scale across features. Figure 2 illustrates the trade-off for light-tailed x_1 and heavy-tailed x_2, both with unit variance. Small noise keeps corrupted inputs near their original observations, but perturbations from the dense core rarely reach the distant tails of x_2 (Figure 2a). Larger noise extends this reach but also spreads inputs away from the mode along x_1, reducing the fraction available for local supervision at a fixed training budget (Figure 2b). This can weaken estimation of local score structure, while stronger smoothing can further reduce sensitivity to nearby anomalies.

K-DSM instead sets each feature's noise scale from the shape of its marginal: larger perturbations along the heavy-tailed feature extend tail coverage, while smaller perturbations along the light-tailed feature preserve locality (Figure 2c). It computes a fixed noise vector $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)$ from normal training data through three steps:

1. **Histogram rearrangement.** Construct each feature's marginal histogram and rearrange its bins into a symmetric, unimodal profile.
2. **Kurtosis estimation.** Compute the kurtosis $\kappa_j^\star$ of each rearranged profile to summarize marginal concentration. Section 2.1 explains these two steps together.
3. **Noise calibration.** Map each $\kappa_j^\star$ to a feature-specific noise scale σ_j using the noise-scale formula motivated by the analysis in Section 2.3.

The resulting noise vector remains fixed during training. Unlike noise-conditioned networks (Song & Ermon, 2019) and multi-scale score matching (Mahmood et al., 2021), K-DSM trains a single network without noise conditioning. Calibration requires neither anomaly labels nor a dataset-specific validation set and preserves single-pass inference. Figure 3 summarizes calibration, training, and inference.

2.1 Noise Calibration with Rearranged Marginal Kurtosis

To calibrate noise, we need to characterize how each marginal's probability mass is distributed between dense and low-density regions. This helps us choose noise scales that reach sparse regions while preserving local structure. We obtain this information by combining histogram rearrangement and kurtosis.

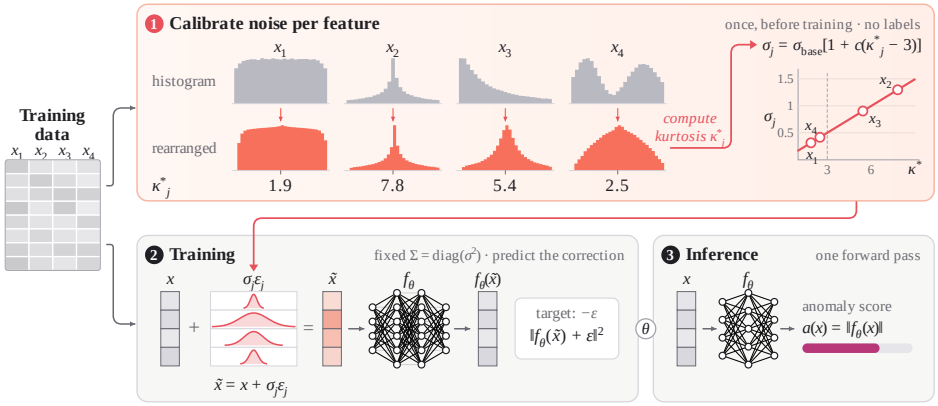


Figure 3: **K-DSM pipeline.** Rearranged marginal kurtosis determines fixed per-feature noise scales. A joint network learns to correct corrupted training observations and scores clean test inputs by its output norm in one forward pass. Rearrangement is used only for calibration.

For a histogram with normalized bin frequencies w_b, where $\sum_b w_b = 1$, and bin locations z_b, kurtosis is

$$\kappa(w, z) = \frac{\sum_b w_b (z_b - \mu)^4}{\left[\sum_b w_b (z_b - \mu)^2\right]^2}, \qquad \mu = \sum_b w_b z_b. \tag{2}$$

Ordinary kurtosis can change when modes move farther apart, even if their individual shapes remain unchanged. We therefore rearrange the histogram before computing it.

We sort equal-width bins by decreasing frequency and place them symmetrically around the origin, with the most frequent bins at the center and less frequent bins farther away (Figure 3, top). The bin frequencies remain unchanged; only the locations used to compute kurtosis are reassigned. For feature j, let w_j denote its normalized bin frequencies and $z_j^\star$ their rearranged locations. We compute $\kappa_j^\star = \kappa(w_j, z_j^\star)$, the kurtosis of the rearranged histogram. This construction approximates the symmetric decreasing rearrangement (Lieb & Loss, 2001), which preserves the distribution of density values. In practice, we place frequency-ranked bins alternately on either side of the origin; Appendix E gives the full construction.

In the rearranged histogram, a marginal with most of its mass in a few high-frequency bins and the remainder spread across many low-frequency bins has a concentrated center and extended tails. Kurtosis emphasizes the contribution of these outer bins through its fourth moment (Westfall, 2014), while normalization by the squared variance removes overall scale. Thus, $\kappa_j^\star$ summarizes marginal concentration with particular sensitivity to the extent of low-density regions, independently of their original positions. **Importantly, rearrangement is used only to compute the noise scales; it does not replace the distribution learned by the network.** Training uses noisy versions of the original joint observations, and scoring uses the original clean inputs. The network therefore learns a joint restoration field that can capture dependencies between features; using marginal statistics for calibration does not impose feature independence.

Noise-scale formula. Section 2.3 derives a first-order approximation to the noise scale meeting a fixed coverage requirement. Under the near-Gaussian shape assumption in Proposition 1, this scale is approximately the Gaussian reference scale plus a correction proportional to excess kurtosis. This motivates the initial noise-scale estimate:

$$\hat{\sigma}_j = \sigma_{\text{base}}\left[1 + c(\kappa_j^\star - 3)\right]. \tag{3}$$

Here, σ_{base} is the scale meeting the requirement for a Gaussian marginal and sets the overall perturbation level, while c sets how strongly the scale responds to rearranged kurtosis. The local analysis relates both constants to tail and coverage levels; we choose the defaults empirically. The reference $\kappa_j^\star = 3$ leaves the base scale unchanged; larger values increase the initial noise scale, while smaller values decrease it.

We use $\sigma_{\text{base}} = 0.5$ and $c = 0.33$ as defaults and obtain the final scale σ_j by clipping $\hat{\sigma}_j$ to the fixed, untuned range $[0.1, 3.0]$ (avoiding extreme noise). Appendix E.3 specifies the complete calibration, including discrete-feature and undefined-kurtosis rules; Appendix E details histogram rearrangement.

2.2 Training and Anomaly Scoring

We train K-DSM by adding Gaussian noise to the original observations using the per-feature scales from Section 2.1. These scales remain fixed throughout training. With $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$, a noisy input is $\tilde{x} = x + \Sigma^{1/2}\varepsilon$, where $\varepsilon \sim \mathcal{N}(0, I_d)$. We train a single network f_θ, with $f_\theta = \Sigma^{1/2} s_\theta$, to predict the correction $-\varepsilon$, measured in noise units:

$$\mathcal{L}_{\text{DSM}}^{\text{per-feature}}(\theta) = \frac{1}{2}\mathbb{E}_{x\sim p_{\text{data}},\, \varepsilon\sim\mathcal{N}(0,I_d)} \left[\left\| f_\theta(x + \Sigma^{1/2}\varepsilon) + \varepsilon \right\|_2^2 \right]. \tag{4}$$

For feature j, the target $-\varepsilon_j = (x_j - \tilde{x}_j)/\sigma_j$ is the correction back to the clean value, divided by the noise scale. This gives every target unit variance, even though features receive different amounts of noise. The normalization corresponds to weighting each feature's squared score error by σ_j^2 in the DSM objective (Vincent, 2011) (Appendix G.7). Like DDPMs, we train by predicting noise (Ho et al., 2020), but use a single fixed covariance Σ, without a noise schedule or a timestep input to the network.

The network estimates $\Sigma^{1/2}\nabla \log q_\Sigma$, where q_Σ is the joint density of the noisy observations: each score component is multiplied by its feature's noise scale (Appendix D). Although noise is added independently to each feature, the network processes all features together and can learn dependencies between them.

At inference, we pass a clean test observation x through the network and use $a(x) = \|f_\theta(x)\|_2$ as its anomaly score. Higher scores indicate larger predicted corrections in noise units. Each observation is scored in a single forward pass, without adding noise or repeating the denoising step (Appendix G.5).

2.3 Analysis of Noise Coverage and Score Estimation

This section motivates the calibration rule in three steps. First, we require corrupted training inputs to reach the low-density regions of each marginal with a prescribed probability. Second, we show that the rearranged profile is the most demanding case: a scale that meets this requirement for the rearranged profile also meets it for the original marginal, regardless of its skew or number of modes. Third, a first-order expansion near a Gaussian relates the smallest such scale to excess kurtosis, giving the affine form of Eq. (3) and its two constants.

Coverage requirement. Let p be the marginal density of one feature under normal data, and let A_τ be its highest-density region of probability $1 - \tau$ (Clémençon & Thomas, 2018). For a corrupted input $X + \sigma Z$, with $X \sim p$ and $Z \sim \mathcal{N}(0, 1)$ independent, we define *coverage* as the probability $\Pr(X + \sigma Z \notin A_\tau)$ of falling in the low-density region A_τ^c. We require coverage of at least α, with $0 < \tau < \alpha < 1$, and seek the smallest σ that achieves it.

Theorem 1 (Conservative coverage under rearrangement)**.** *Let $p^\star$ be the symmetric decreasing rearrangement of p, $X^\star \sim p^\star$, and $r_\tau = |A_\tau|/2$, so that $[-r_\tau, r_\tau]$ is a highest-density region of $p^\star$ with probability $1 - \tau$. For every $\sigma > 0$,*

$$\Pr(X + \sigma Z \notin A_\tau) \geq C_{p^\star}(\sigma) := \Pr(|X^\star + \sigma Z| > r_\tau), \tag{5}$$

with equality when $p = p^\star$. The function $C_{p^\star}$ increases continuously and strictly from τ to 1, so a unique smallest scale s_p satisfies $C_{p^\star}(s_p) = \alpha$.

Proof sketch. By the Riesz rearrangement inequality (Lieb & Loss, 2001), the symmetric decreasing profile maximizes the probability of staying inside the highest-density region among densities with the same distribution of density values. Taking complements gives the bound. Appendix B proves monotonicity and the full statement.

A scale that gives the rearranged profile coverage α therefore gives the original marginal at least the same coverage, regardless of mode locations. This guarantee applies to the exact population root s_p; K-DSM's first-order approximation from estimated histograms does not inherit it (standardizing the profile to unit variance preserves the bound; Appendix E). Since each feature's requirement is a separate lower bound $\sigma_j \geq s_j$, choosing $\sigma_j = s_j$ meets all requirements with the least total perturbation $\sum_j \sigma_j^2$, whereas the smallest shared scale is $\max_j s_j$ (Figure 2; Appendix B.3).

From coverage to kurtosis. Both the coverage bound and rearranged kurtosis depend on p only through its concentration curve $V_p(u)$, the smallest total length of a region containing probability u. The coverage radius is $r_\tau = V_p(1-\tau)/2$, and when the fourth moment is finite,

$$\kappa^\star = \frac{\int_0^1 V_p(u)^4\, du}{\left(\int_0^1 V_p(u)^2\, du\right)^2} \tag{6}$$

(Appendix B.1). Rearranged kurtosis thus summarizes, independently of scale, the growth of the concentration curve toward $u = 1$. This growth also determines the coverage radius, but kurtosis alone does not determine r_τ (Appendix C.1), so we linearize around the Gaussian.

Proposition 1 (Noise-scale correction at fixed coverage)**.** *Let p_δ be symmetric decreasing densities with mean zero, variance one, and kurtosis $3+\delta$ whose departure from the standard Gaussian density ϕ is, to first order, the fourth Hermite term: $p_\delta = \phi\,(1 + \delta H_4/24) + O_{L^1}(\delta^2)$ with $H_4(x) = x^4 - 6x^2 + 3$. Let $s(\delta)$ be the smallest scale giving coverage α outside the region of mass $1-\tau$. With $z = \Phi^{-1}(1-\tau/2)$, $w = \Phi^{-1}(1-\alpha/2)$, $v = (z/w)^2$, and $\sigma_G = \sqrt{v-1}$, where Φ is the standard Gaussian CDF,*

$$s(\delta) = \sigma_G[1 + c_{\mathrm{cov}}\delta] + O(\delta^2), \qquad c_{\mathrm{cov}} = \frac{z^2(1+v^{-1}+v^{-2}) - 3(1+v^{-1})}{24}. \tag{7}$$

A sufficient condition for $c_{\mathrm{cov}} > 0$ is $z^2 > 3$, equivalently $\tau < 2[1-\Phi(\sqrt{3})] \approx 0.0833$.

Proof sketch. For the Gaussian reference, coverage α is reached at scale σ_G. Differentiating the fixed-coverage equation $C(\delta, s(\delta)) = \alpha$ at this reference gives $c_{\mathrm{cov}} = -\partial_\delta C/(\sigma_G\,\partial_\sigma C)$; both derivatives follow from Gaussian convolution of the Hermite term and the shift of the region's boundary (Appendix C.3).

For feature j, set $\delta = \kappa_j^\star - 3$ and drop the $O(\delta^2)$ remainder. This gives the affine rule in Eq. (3) with $\sigma_{\mathrm{base}} = \sigma_G$ and $c = c_{\mathrm{cov}}$, both determined by (τ, α). Our fixed defaults $(0.5, 0.33)$ correspond to $(\tau, \alpha) \approx (0.0195, 0.0366)$ (Appendix C.4). These coverage levels have not been shown to be optimal for detection. The implemented rule also extrapolates the local approximation beyond its assumptions, and agreement with numerical coverage roots deteriorates at extreme kurtosis (Appendix C.6).

Coverage and anomaly detection. Coverage can help detection by improving score estimation. For an idealized kernel estimator with a noise matrix fixed in advance, more Gaussian-weighted training mass near a query tightens the pointwise error bound; errors below half the ideal anomaly–normal score gap preserve the ranking (Appendix D). However, noise also changes the score being estimated. For Gaussian normal data $X \sim \mathcal{N}(0, I_d)$, the ideal output is $f^*_{\Sigma,j}(x) = -\sigma_j x_j/(1+\sigma_j^2)$, so a feature's weight peaks at $\sigma_j = 1$ and decays as $1/\sigma_j$: large scales increase coverage but reduce that feature's contribution (Appendix D.3). Marginal coverage alone does not ensure local support in the joint distribution, and the bound does not cover K-DSM's data-dependent calibration or the network's approximation and optimization error.

3 Related Work

Denoising and score-based detection. Denoising autoencoders and score matching are closely related (Vincent, 2011), but anomaly detectors differ in how they use noise during training and inference. MSM (Mahmood et al., 2021) trains a score network at several noise levels (Song & Ermon, 2019), then uses a separate detector to combine the resulting score norms. NCSBAD (Hirth & Kasneci, 2025) also trains at multiple noise levels, but scores test points by repeatedly adding noise at one fixed level and combining the squared noise-prediction errors. Shin et al. (2023) detect image anomalies by measuring restoration error after adding noise. DDAE (Sattarov et al., 2025) trains an autoencoder with a diffusion noise schedule and sums reconstruction errors across noise levels. DenoiserAD (Hirth et al., 2026) trains at a single noise level and measures denoiser stability under repeated noise additions at inference. In comparison, K-DSM uses the kurtosis of rearranged training histograms to choose one fixed noise scale per feature. It scores each test point using the norm of the network output in a single forward pass, without adding noise at inference.

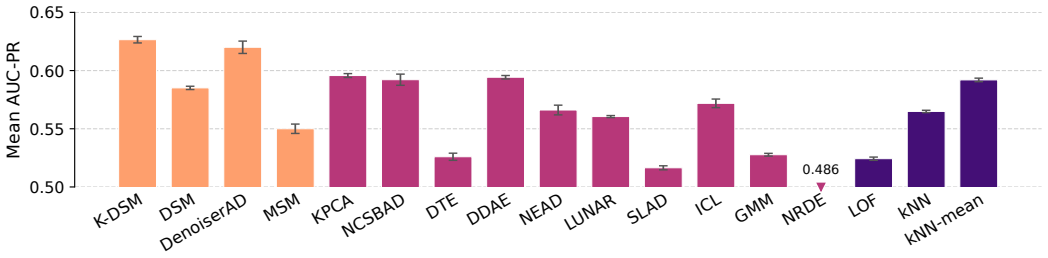


Figure 4: Mean AUC-PR on 57 ADBench datasets. Error bars show standard errors across five seed-level macro means; NRDE's standard error is reported in Table 26. Full AUC-PR, AUC-ROC, and F1 results appear in Appendix N.

Predicting noise levels. DTE (Livernoche et al., 2024) estimates a distribution over diffusion time, while NEAD (Dai et al., 2025) predicts element-wise noise magnitudes using diverse corruptions, including randomly varying scales across features. Their neural predictors also support direct inference. However, they learn to estimate how noisy an observation is, whereas K-DSM learns the score of the noise-smoothed data distribution through denoising.

Other tabular detectors. Classical baselines include kernel PCA (KPCA) (Hoffmann, 2007), robust covariance estimation (MCD) (Rousseeuw & Driessen, 1999), and Gaussian mixture models (GMMs). KPCA is often absent from modern benchmark comparisons, yet it attains the highest mean AUC-ROC in our evaluation (Appendix N). Neural methods include NRDE (Fu et al., 2026) for density estimation after separating noise, LUNAR (Goodge et al., 2022) for learned aggregation of neighbor distances, SLAD (Xu et al., 2023) for predicting representation scales, and ICL (Shenkar & Wolf, 2022) for learning feature dependencies through contrastive learning. Appendix F provides further details.

4 Experiments

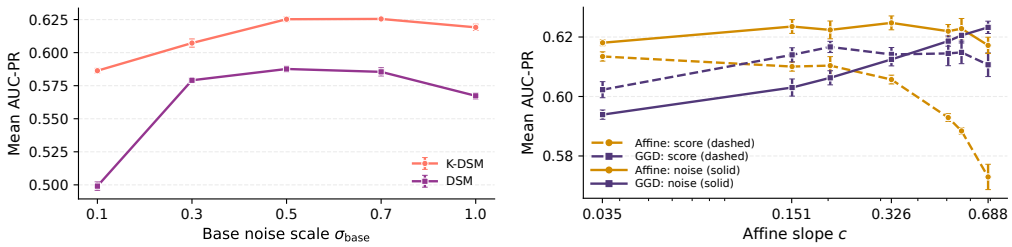
4.1 Protocol

Following Livernoche et al. (2024), we evaluate on 57 ADBench datasets (Han et al., 2022), capped at 50,000 observations per task before splitting. We train on a random 50% of normals and test on the remaining normals and all anomalies. All methods share splits, with scalers fitted on training normals only. We report dataset-macro means and standard errors across five seeds unless noted otherwise. Our primary metric, AUC-PR (average precision), emphasizes precision and recall when anomalies are rare (Davis & Goadrich, 2006; Saito & Rehmsmeier, 2015); class imbalance alone does not make it universally preferable to AUC-ROC (McDermott et al., 2024). Full AUC-PR, AUC-ROC, and F1 results appear in Appendix N. Settings are fixed across datasets, with DSM at the same base scale of 0.5 as K-DSM. K-DSM computes feature scales from training data without anomaly labels or validation. Benchmark results informed the global defaults; disjoint-dataset split-half selection yields mean held-out AUC-PR regret of 0.0033 for both base scale and noise-prediction slope (Appendix J.1). Architecture, baseline implementations, and aggregation details appear in Appendices G.6 and G.2.

4.2 Results

Detection performance. Figure 4 compares K-DSM with classical and deep anomaly detectors, including recent denoising methods. Two findings stand out. First, single-scale DSM already reaches 0.585 mean AUC-PR, competitive with DDAE and kNN-mean and above MSM, DTE, SLAD, and LOF. Second, K-DSM raises mean AUC-PR to 0.627, above every baseline (0.620 for DenoiserAD). K-DSM also achieves competitive AUC-ROC (0.823, versus 0.833 for KPCA) while retaining single-pass inference and low inference cost (Figure 1). Although K-DSM achieves a higher mean AUC-PR, the paired Wilcoxon signed-rank test does not establish a difference from DenoiserAD after Holm correction ($p_{\text{adj}} = 0.659$; Appendix N). Interestingly, combining K-DSM's calibration and single-pass scoring with DenoiserAD yields the highest mean AUC-PR overall (0.630) and nearly matches the best mean AUC-ROC (0.832, versus 0.833 for KPCA), suggesting that the two approaches are complementary. Implementation details appear in Appendix N.

Effect of feature-wise noise allocation. We test whether it matters which features receive more noise. Keeping K-DSM's training and anomaly-scoring procedures fixed, we compare two alternatives. The first uses one shared noise scale for all features, chosen to match K-DSM's total noise variance. The second keeps the same scales but randomly reassigns them to features. Mean AUC-PR is 0.595 with a shared scale, 0.610 with randomly reassigned scales, and 0.627 with kurtosis-guided scales. Thus, using different scales helps, and assigning them according to feature kurtosis improves performance further. A separate comparison using the same trained K-DSM model shows that the anomaly score also matters: scoring with the noise-unit output outperforms rescaling it to data units (Section 2.2). Experimental details, uncertainty estimates, and comparisons of scoring rules appear in Appendix K.



(a) Sensitivity to base noise scale σ_{base}.

(b) Affine vs. fitted GGD noise mapping.

Figure 5: Ablation studies on ADBench. **(a)** K-DSM outperforms DSM at every tested base scale, with the largest gain at the smallest scale. **(b)** Affine and generalized Gaussian (GGD) calibration with direct score prediction (dashed) or ε-prediction (solid). GGD tail probabilities are matched to the horizontal c-axis using the Cornish–Fisher radius coefficient $c_{\text{CF}}(\tau)$, distinct from the coverage coefficient c_{cov} in Proposition 1 (Appendix C.5).

Robustness to base noise scale. K-DSM outperforms single-scale DSM at every tested base scale (Figure 5a). The largest difference occurs at $\sigma_{\text{base}} = 0.1$, where DSM falls to 0.499 mean AUC-PR while K-DSM retains 0.586. Over $\sigma_{\text{base}} \in [0.3, 1.0]$, the two methods vary by similar amounts, with K-DSM consistently ahead. The practical benefit is therefore stronger performance throughout the sweep and less degradation at the smallest scale. Numerical results and cross-dataset parameter selection appear in Appendix J.

Affine vs. generalized Gaussian calibration. We use the generalized Gaussian distribution (GGD) as an alternative because it shares the symmetry and unimodality of the rearranged marginals and extends the Gaussian to lighter or heavier tails through a single shape parameter. With variance fixed to one, this parameter can be fitted from each feature's rearranged kurtosis. Central interval widths can then be computed directly for this family, without a first-order approximation. We compare intervals containing the same chosen probability mass under the fitted GGD and a standard Gaussian; their width ratio scales the base noise. This gives a nonlinear alternative using the same kurtosis information, allowing us to test whether an explicit model of tail shape improves detection over the affine rule. Appendix C.5 specifies the fitting procedure, the noise-scale formula, and the tail probabilities used in the comparison.

Figure 5b compares both calibrations with direct score prediction and noise (ε) prediction. The coefficient c controls how strongly kurtosis changes the affine noise scales; GGD tail probabilities are matched through the Cornish–Fisher radius map (Appendix C.5). With noise prediction, the affine rule achieves higher mean AUC-PR at most settings, although GGD is stronger at the largest tested c. The affine rule's mean AUC-PR remains between 0.617 and 0.625 across the sweep, supporting a fixed default without precise tuning of c. Noise prediction improves the affine rule at every shared setting, while its effect on GGD is mixed.

4.3 Unsupervised (Contaminated) Setting

In the fully unsupervised regime, the training set is unlabeled and contains anomalies. K-DSM is designed to increase training coverage in sparse regions. Under contamination, this can also provide more supervision around anomalous observations, making the network fit them as part of the training

distribution and weakening their anomaly signal. This is a potential cost of the coverage objective: it treats all training observations as valid, including contaminants.

EMA-teacher filtering. To reduce the influence of contaminants, we repurpose an exponential-moving-average (EMA) teacher (Tarvainen & Valpola, 2017) to filter training observations. A slowly updated copy of the network scores each minibatch, and the online model trains only on the lowest-scoring $\gamma\%$ of observations. Weight averaging slows changes in the selection signal. We use $\gamma = 80$, whose performance is close to the best tested value ($\gamma = 75$) in the ablation in Appendix I.1. The filter adds computation during training but leaves inference unchanged; implementation details and adaptations to other detectors appear in Appendix I.

Results. Without filtering, K-DSM scores slightly below DSM under contamination (Table 1). EMA filtering gives K-DSM the largest absolute gain in mean AUC-PR among the evaluated methods with an EMA variant, raising it from approximately 0.19 to 0.343. K-DSM-EMA leads the evaluated denoising and diffusion-based detectors in mean AUC-PR and F1, and achieves the highest means on these metrics across the full comparison. Its advantage does not extend to AUC-ROC. Classical baselines remain strong: MCD reaches 0.321 mean AUC-PR, matching DSM-EMA, while GMM and MCD both exceed K-DSM-EMA on mean AUC-ROC, with GMM leading overall. Full results and comparisons with alternative filtering strategies appear in Appendices L and I.

Table 1: Mean performance ($\times 100$) across 57 ADBench datasets under contaminated training. Results use five seeds. EMA variants appear above their base variants (grey); other baselines span both rows. kNN reports the distance to the fifth nearest training observation; kNN-mean averages the five distances. Best displayed mean per metric is bold; standard errors are in parentheses. The full comparison appears in Appendix L.

Metric	Variant	K-DSM	DSM	DDAE	DTE	MSM	GMM	KPCA	MCD	DenoiserAD	kNN	NCSBAD
AUC-PR	+EMA	**34.3** (0.4)	32.1 (0.5)	31.5 (0.4)	29.0 (0.2)	16.9 (0.1)	29.8 (0.2)	20.5 (0.1)	32.1 (0.5)	27.7 (0.8)	23.9 (0.1)	17.2 (0.2)
	base	18.9 (0.3)	19.4 (0.2)	22.8 (0.3)	28.1 (0.2)	17.7 (0.2)				14.6 (0.1)		
AUC-ROC	+EMA	70.1 (0.3)	71.0 (0.3)	72.5 (0.1)	72.5 (0.3)	64.1 (0.3)	**74.2** (0.1)	59.8 (0.3)	73.0 (0.4)	65.7 (0.3)	66.5 (0.1)	60.7 (0.3)
	base	60.2 (0.2)	60.4 (0.1)	68.5 (0.5)	72.6 (0.2)	62.8 (0.5)				53.4 (0.1)		
F1	+EMA	**34.4** (0.5)	30.8 (0.5)	31.2 (0.4)	28.4 (0.3)	16.6 (0.1)	28.8 (0.1)	22.0 (0.2)	30.8 (0.5)	28.2 (0.7)	25.0 (0.1)	18.3 (0.1)
	base	19.5 (0.3)	20.0 (0.3)	24.1 (0.3)	27.9 (0.4)	18.2 (0.2)				15.0 (0.0)		

5 Conclusion

We revisited denoising score matching for tabular anomaly detection and showed that a single noise scale already provides strong performance with one forward pass at inference. K-DSM builds on this simple detector by adapting the training noise to each feature using rearranged marginal kurtosis. On 57 ADBench datasets with normal-only training, K-DSM exceeds every baseline in mean AUC-PR, with competitive AUC-ROC, and low inference cost. Its improvement over DSM persists across all tested base noise scales, and combining its calibration and scoring with DenoiserAD suggests that the approaches are complementary. Under contaminated training, EMA-teacher filtering gives K-DSM the largest AUC-PR gain among the tested EMA variants, while classical methods remain strong competitors.

Limitations. The affine calibration relies on a specific near-Gaussian shape assumption and can substantially overestimate the noise needed for the coverage target at extreme kurtosis (Appendix C.6). Histogram resolution, clipping, and rules for discrete features or undefined kurtosis remain empirical choices. Greater marginal coverage does not guarantee better anomaly detection. K-DSM trails DSM on two small-sample image-embedding benchmarks outside our tabular scope, while improving over DSM on nine of the ten ADBench groups built from embeddings (Appendix M). Although feature scales are computed without anomaly labels on each dataset, benchmark results informed the global defaults. Under contamination, anomalies affect both calibration and the denoising objective, and their respective contributions to the performance loss have not been isolated. EMA filtering helps empirically, but depends on the teacher identifying contaminants and provides no guarantee across contamination levels.

REPRODUCIBILITY STATEMENT

We provide the K-DSM code in the supplementary material. The package includes method implementations, baseline wrappers, configuration files, and runners for normal-only and contaminated training, hyperparameter ablations, allocation and scoring controls, disjoint-dataset selection, and image-embedding experiments. Its README gives installation and data-acquisition instructions, commands for selecting datasets and random seeds, and guidance on saving model weights, anomaly scores, and metrics. Dependency lockfiles specify the benchmark and image-feature-extraction environments, and a results helper computes dataset-group aggregates and standard errors. Datasets and pretrained image encoders must be obtained separately as described in the README.

Section 4.1 and Appendix G.2 specify the evaluation splits, preprocessing, baseline adaptations, metrics, and aggregation across datasets and seeds. Appendices E and E.3 detail histogram construction, noise clipping, and fallback rules; Algorithm 2 and Table 5 specify training, inference, architecture, and optimization settings. Appendix I describes contamination filtering, and Appendix H.1 reports hardware and computational costs. The assumptions and proofs supporting the coverage and calibration results appear in Appendices B and C.

AI USE STATEMENT

Large language models were used in three supporting roles during this research. First, as a reasoning aid for mathematical proofs: LLM outputs were used to explore proof strategies and identify relevant lemmas, and the final theoretical claims were verified by the authors; the authors take responsibility for the submitted results. Second, as a writing assistant for drafting and editing manuscript text. Third, as a coding assistant for implementing evaluation pipelines and plotting utilities. In all cases, LLM-generated content was reviewed and either accepted, modified, or discarded by the authors.

REFERENCES

Kilian Batzner, Lars Heckler, and Rebecca König. EfficientAD: Accurate visual anomaly detection at millisecond-level latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 128–138, 2024. URL https://openaccess.thecvf.com/content/WACV2024/html/Batzner_EfficientAD_Accurate_Visual_Anomaly_Detection_at_Millisecond-Level_Latencies_WACV_2024_paper.html.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1lK_lBtvS.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD: A comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000.

Almut Burchard and Hichem Hajaiej. Rearrangement inequalities for functionals with monotone integrands. *Journal of Functional Analysis*, 233:561–582, 2006. URL https://arxiv.org/abs/math/0506336.

Fabrizio Carcillo, Andrea Dal Pozzolo, Yann-Aël Le Borgne, Olivier Caelen, Yannis Mazzer, and Gianluca Bontempi. SCARFF: A scalable framework for streaming credit card fraud detection with Spark. *Information Fusion*, 41:182–194, 2018. doi: 10.1016/j.inffus.2017.09.005.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Stéphan Clémençon and Albert Thomas. Mass volume curves and anomaly ranking. *Electronic Journal of Statistics*, 12(2):2806–2872, 2018. doi: 10.1214/18-EJS1474.

Wei Dai, Kai Hwang, and Jicong Fan. Unsupervised anomaly detection for tabular data using deep noise evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11553–11562, 2025. doi: 10.1609/aaai.v39i11.33257. URL https://ojs.aaai.org/index.php/AAAI/article/view/33257.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 233–240, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143874. URL https://doi.org/10.1145/1143844.1143874.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006. URL https://jmlr.org/papers/v7/demsar06a.html.

Luc Devroye. *A Course in Density Estimation*, volume 14 of *Progress in Probability and Statistics*. Birkhäuser, Boston, 1987. ISBN 978-0-8176-3365-3.

Dickson B. Dunlap. Thyroid function tests. In H. Kenneth Walker, William D. Hall, and J. Willis Hurst (eds.), *Clinical Methods: The History, Physical, and Laboratory Examinations*. Butterworths, Boston, 3rd edition, 1990. URL https://www.ncbi.nlm.nih.gov/books/NBK249/.

Dazhi Fu, Zhao Zhang, and Jicong Fan. Noise-robust density estimation for tabular data anomaly detection. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/63974.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In *KI-2012: Poster and Demo Track*, pp. 59–63, 2012.

Adam Goodge, Bryan Hooi, See-Kiong Ng, and Wee Siong Ng. LUNAR: Unifying local outlier detection methods via graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6737–6745, 2022. doi: 10.1609/aaai.v36i6.20629. URL https://ojs.aaai.org/index.php/AAAI/article/view/20629.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Songqiao Han, Xiyang Hu, Hailiang Huang, Mingqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. In *Neural Information Processing Systems (NeurIPS)*, 2022.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Manuel Hirth and Enkelejda Kasneci. Anomaly detection by estimating gradients of the tabular data distribution, 2025. URL https://openreview.net/forum?id=7QDIFrtAsB. Public ICLR 2025 submission.

Manuel Hirth, Lukas Koberg, Nasser Jazdi, and Enkelejda Kasneci. Denoising without diffusion: Fixed-noise denoiser anomaly detection in tabular data. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64348.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Heiko Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007. doi: 10.1016/j.patcog.2006.07.009.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Meng-Chieh Lee, Yue Zhao, Aluna Wang, Pierre Jinghong Liang, Leman Akoglu, Vincent S. Tseng, and Christos Faloutsos. Autoaudit: Mining accounting and time-evolving graphs. *CoRR*, abs/2011.00447, 2020. URL https://arxiv.org/abs/2011.00447.

Wenyuan Li, Yunlong Wang, Yong Cai, Corey W. Arnold, Emily Zhao, and Yilian Yuan. Semi-supervised rare disease detection using generative adversarial network. *CoRR*, abs/1812.00547, 2018. URL http://arxiv.org/abs/1812.00547.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: Copula-based outlier detection. *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1118–1123, 2020.

Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H. Chen. ECOD: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193, 2023. doi: 10.1109/TKDE.2022.3159580.

Elliott H. Lieb and Michael Loss. *Analysis*, volume 14 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2nd edition, 2001. ISBN 978-0-8218-2783-3.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Victor Livernoche, Vineet Jain, Yashar Hezaveh, and Siamak Ravanbakhsh. On diffusion modeling for anomaly detection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=lR3rk7ysXz.

Ahsan Mahmood, Junier Oliva, and Martin Andreas Styner. Multiscale score matching for out-of-distribution detection. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xoHdgbQJohv.

Matthew B.A. McDermott, Haoran Zhang, Lasse Hyldig Hansen, Giovanni Angelotti, and Jack Gallifant. A closer look at AUROC and AUPRC under class imbalance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=S3HvA808gk.

Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An ensemble of autoencoders for online network intrusion detection. In *Network and Distributed System Security Symposium (NDSS)*, 2018. doi: 10.14722/ndss.2018.23204.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *CoRR*, abs/2007.02500, 2020. URL https://arxiv.org/abs/2007.02500.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *International Conference on Machine Learning*, pp. 8703–8714. PMLR, 2021.

Chen Qiu, Aodong Li, Marius Kloft, Maja Rudolph, and Stephan Mandt. Latent outlier exposure for anomaly detection with contaminated data. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18153–18167, 2022. URL https://proceedings.mlr.press/v162/qiu22b.html.

Juan C. Quiroz, Norman Bin Mariun, Mohammad Rezazadeh Mehrjou, Mahdi Izadi, Norhisam Misron, and Mohd Amran Mohd Radzi. Fault detection of broken rotor bar in LS-PMSM using random forests. *CoRR*, abs/1711.02510, 2017. URL http://arxiv.org/abs/1711.02510.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, 29(2):427–438, May 2000. ISSN 0163-5808. doi: 10.1145/335191.335437. URL https://doi.org/10.1145/335191.335437.

Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ruff18a.html.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015. doi: 10.1371/journal.pone.0118432.

Timur Sattarov, Marco Schreyer, and Damian Borth. Diffusion-scheduled denoising autoencoders for anomaly detection in tabular data. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 2478–2489, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3736910. URL https://doi.org/10.1145/3711896.3736910.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_hszZbt46bT.

Woosang Shin, Jonghyeon Lee, Taehan Lee, Sangmoon Lee, and Jong Pil Yun. Anomaly detection using score-based perturbation resilience. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23372–23382, October 2023.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop*, pp. 172–179, 2003.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

Carole A. Spencer. Assay of thyroid hormone and related substances. In Kenneth R. Feingold, S. Faisal Ahmed, Bradley Anawalt, Alison Boyce, et al. (eds.), *Endotext [Internet]*. MDText.com, Inc., South Dartmouth (MA), 2025. URL https://www.ncbi.nlm.nih.gov/books/NBK279113/. Last updated July 3, 2025.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

V. V. Ulyanov, M. Aoshima, and Y. Fujikoshi. Non-asymptotic results for Cornish–Fisher expansions. *Journal of Mathematical Sciences*, 218(3):363–368, 2016. URL https://arxiv.org/abs/1604.00539.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Peter H. Westfall. Kurtosis as peakedness, 1905–2014. r.i.p. *The American Statistician*, 68(3):191–195, 2014. doi: 10.1080/00031305.2014.917055. URL https://doi.org/10.1080/00031305.2014.917055. PMID: 25678714.

Andre Wibisono, Yihong Wu, and Kaylee Yingxi Yang. Optimal score estimation via empirical Bayes smoothing. In *Proceedings of the 37th Conference on Learning Theory*, volume 247 of *PMLR*, pp. 4958–4991, 2024. URL https://proceedings.mlr.press/v247/wibisono24a.html.

Hongzuo Xu, Yijie Wang, Juhui Wei, Songlei Jian, Yizhou Li, and Ning Liu. Fascinating supervisory signals and where to find them: Deep anomaly detection with scale learning. In *International Conference on Machine Learning*, pp. 38655–38673. PMLR, 2023.

Jiaxin Yin, Yuanyuan Qiao, Zitang Zhou, Xiangchao Wang, and Jie Yang. MCM: Masked cell modeling for anomaly detection in tabular data. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=lNZJyEDxy4.

Jinsung Yoon, Kihyuk Sohn, Chun-Liang Li, Sercan Ö. Arik, Chen-Yu Lee, and Tomas Pfister. Self-supervise, refine, repeat: Improving unsupervised anomaly detection. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=b3v1UrtF6G.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJJLHbb0-.

Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-Difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision (ECCV)*, 2022.

APPENDIX

A Interpretability **17**

B Coverage: Complete Proofs **18**

B.1 Minimum-volume regions and rearranged kurtosis 18

B.2 Conservative coverage under rearrangement 19

B.3 Minimum-perturbation allocation 20

C First-Order Kurtosis Calibration **20**

C.1 Why kurtosis alone does not order tail radii 20

C.2 Assumptions and controlled quantile inversion 20

C.3 Averaged-coverage calibration 21

C.4 Illustrative calibration and numerical check 22

C.5 Generalized Gaussian calibration and ablation settings 22

C.6 Affine calibration versus numerical coverage roots 23

D Local Coverage and Score Learnability **24**

D.1 Population and empirical DSM solutions 24

D.2 Transfer to anomaly ranking 25

D.3 Gaussian example: local support and score separation 26

E From Rearranged Profiles to Histogram Calibration **26**

E.1 Standardization preserves a conservative certificate 26

E.2 Finite-bin approximation 27

E.3 Calibration used by K-DSM 27

F Additional Related Work **28**

G Implementation Details **29**

G.1 Benchmark Description 29

G.2 Baseline implementations and aggregation 31

G.3 Binary and categorical features: scope and implementation 31

G.4 Assigned noise scales across all ADBench features 32

G.5 Algorithm 33

G.6 Model architecture 33

G.7 Loss parameterisation 34

H Inference and Training Time **35**

H.1 Compute Cost 36

I EMA Teacher for Unsupervised Anomaly Detection **38**

I.1 Filtering Percentile Ablation 39
I.1.1 Filtering sensitivity by native contamination rate 39
I.2 Comparison with alternative contamination strategies 41

J Alternative Noise Selection Strategies **42**
J.1 Disjoint-dataset hyperparameter selection 43
J.2 Kurtosis heterogeneity and the gain over DSM 44

K Noise Allocation and Inference Controls **45**
K.1 Matched controls for feature-wise noise allocation 45
K.2 Output parameterization and residual weighting 46
K.3 Clean-input and noisy reconstruction scoring 47
K.4 Paired analysis 48

L Full Unsupervised Results **49**

M Image Benchmark Results **56**
M.1 Per-class Image Benchmark Results 56

N Full Semi-supervised Results **58**
N.1 Synthetic anomalies defined only by feature interactions 61

A INTERPRETABILITY

Score matching provides a vector field whose coordinates can explain the model's output. For the parameterization used by K-DSM, the population field is $f_D^* = D\nabla \log q_D$, with $Df_D^*(x) = \mathbb{E}[X \mid Y = x] - x$. Thus the sign of coordinate $f_{D,j}^*$ agrees with the corresponding density gradient, and $f_{\theta,j}(x)^2$ is an exact additive contribution to the squared alarm score $a(x)^2 = \sum_j f_{\theta,j}(x)^2$.

These are directional descriptions of the learned smoothed distribution. For the exact field and sufficiently small steps, moving along f_D^* increases $\log q_D$ because $\nabla \log q_D(x)^\top f_D^*(x) = \nabla \log q_D(x)^\top D \nabla \log q_D(x) \geq 0$. This does not generally reduce the alarm score: the gradient of $a(x)^2/2$ is $J_{f_\theta}(x)^\top f_\theta(x)$, whose direction need not agree with $-f_\theta(x)$. For a finite network, $D^{-1} f_\theta$ need not be the gradient of any log density. Accordingly, the arrows below show the model's restoration directions, not guaranteed counterfactual interventions or causal explanations.

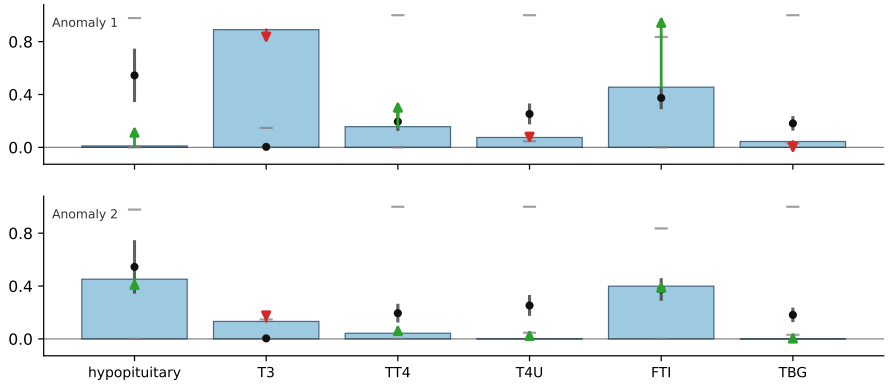
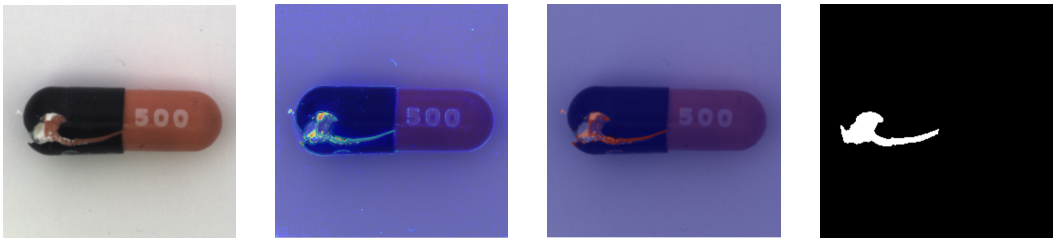


Figure 6: Feature values with directional attributions for two anomaly `thyroid` cases. Arrows encode suggested adjustments based on the estimated score (green/red; magnitude $\propto |score|$). Grey ticks denote normal min/max; black whiskers show mean ± 1 SD and dots the mean, estimated from normal training data.

Tabular case (UCI `thyroid`). We illustrate the interpretability of score matching on tabular data using the `thyroid` dataset in Figure 6. For each example, we plot raw feature values (bars) together with per-feature arrows derived from the estimated normalized field $f_\theta(x)$ (denoted s in the figure); the sign encodes the direction the model would adjust that feature to move x toward higher training-data density, and the arrow length is proportional to $|s_i(x)|$. To provide clinical context, grey ticks mark the empirical normal range (min/max) and black whiskers denote mean with standard deviation across normal training samples, with the mean shown as a dot. Two representative anomalies highlight how s(x) captures what the model "thinks is wrong." Anomaly 1 shows a low hypopituitary indicator alongside elevated T3, while its FTI is not correspondingly high; the score suggests a substantial increase in FTI (achievable via higher TT4 or uptake/lower TBG), making the profile more self-consistent with the hyperthyroid-like pattern in the training normals, where T3, TT4/uptake, and thus FTI tend to rise together. Anomaly 2 presents low T4U and low TBG, a physiologically discordant pair because T3 uptake (T4U) varies inversely with TBG; reduced TBG typically increases uptake, not lowers it. The score highlights these features as principal drivers of the anomaly, pushing them toward a configuration seen in normal data. We emphasize that s(x) explains movement toward the data manifold learned from training normals, which often (not always) aligns with textbook physiology. The qualitative interpretations above are consistent with standard thyroid-function test relationships (e.g., the Free Thyroxine Index reflecting TT4 and binding/uptake, and the inverse coupling of T3 uptake and TBG in binding-affecting states) as summarized in clinical endocrinology references (Spencer, 2025; Dunlap, 1990).

Vision case (`MVTec–capsule`). We also illustrate the interpretability of score-matching for image anomaly detection. As seen in Figure 7, the score field $s_\theta(x)$ exposes regions of the image that deviate from the learned normal manifold of defect-free capsules. Trained only on normals, the per-pixel magnitude $\|s_\theta(x)\|$ concentrates along the crack and spilled material near the seam, while remaining



(a) Input (b) $\|s_\theta(x)\|$ (c) Overlay (d) Mask

Figure 7: Score-based localization on MVTec–capsule. (a) Input image, (b) per-pixel score magnitude $\|s_\theta(x)\|$, (c) overlay of scores on the input, and (d) ground-truth defect mask. A DSM model trained only on normal images highlights the defect region.

low on intact areas such as the printed "500" and the uniform background. The overlay makes this concentration visually apparent, and the ground-truth mask confirms that the saliency peaks align with the annotated defect. No defect labels or auxiliary segmentation modules are involved: the signal arises directly from the score learned on normals. Although we display magnitude for clarity, the full vector field carries directional information: moving along an accurately estimated score increases the smoothed model density locally; whether this removes a visible defect requires a separate reconstruction evaluation. This is a simple example, single-scale score matching does not capture high-dimensional image structure particularly well; multi-scale formulations are better suited for such data, but the example illustrates the general interpretability argument for score-based methods.

B Coverage: Complete Proofs

We work first with a population marginal density p on $\mathbb{R}$. The coverage results require neither symmetry nor finite moments of p. The moment identity states its additional moment assumption explicitly. Appendix E treats the standardized histogram used for calibration.

For compactness, the appendix proofs use $D = \Sigma^{1/2}$ for the diagonal matrix of noise standard deviations. Fields and densities indexed by D correspond to those indexed by Σ in the main text.

B.1 Minimum-volume regions and rearranged kurtosis

For a density p on $\mathbb{R}$, define the minimum-volume curve

$$V_p(u) = \inf\{|A| : \Pr(X \in A) \geq u\}, \qquad 0 < u < 1,$$

where $|A|$ denotes Lebesgue measure. Let $p^\star$ be the symmetric decreasing rearrangement, characterized up to null sets by

$$|\{p > t\}| = |\{p^\star > t\}|, \qquad t > 0.$$

The Hardy–Littlewood inequality (Lieb & Loss, 2001; Burchard & Hajaiej, 2006) implies that for every measurable set A of finite length v,

$$\int_A p(x)\,dx \leq \int_{-v/2}^{v/2} p^\star(x)\,dx. \tag{8}$$

The upper bound is attainable by selecting the largest density values. In particular, for $u \in (0,1)$ a highest-density set of mass u can be chosen as $\{p > t\} \cup B$, with $B \subseteq \{p = t\}$ supplying any remaining mass. A positive threshold is available because the probability carried by $\{p > t\}$ approaches one as $t \downarrow 0$ and zero as $t \uparrow \infty$. Lebesgue nonatomicity permits the selection within a plateau. The selected set has finite length, and all such selections have the same minimum length $V_p(u)$.

The centered probability of $p^\star$ is continuous in its radius and strictly increasing until it reaches one. It follows from Eq. (8) that

$$Q_u(|X^\star|) = \tfrac{1}{2}V_p(u), \qquad X^\star \sim p^\star. \tag{9}$$

For each positive moment order k for which the integral is finite, the quantile representation of moments gives

$$\mathbb{E}|X^\star|^k = 2^{-k} \int_0^1 V_p(u)^k \, du.$$

Since $p^\star$ is centered, taking $k = 2, 4$ proves Eq. (6) whenever the fourth moment is finite. Its variance is positive for a density and is finite whenever the fourth moment is finite. The ratio is invariant to translations and rescalings of the original marginal, as well as to any rearrangement preserving its density values. This links our statistic to minimum-volume normal regions (Clémençon & Thomas, 2018). It summarizes the entire volume profile through two integrals; it does not determine every individual quantile.

B.2 Conservative coverage under rearrangement

Theorem 1 (Conservative coverage under rearrangement; restated)**.** *Fix $\tau \in (0, 1)$ and let A_τ be any highest-density set of a marginal density p with mass $1 - \tau$. Set $r_\tau = V_p(1 - \tau)/2$, let $X \sim p$, $X^\star \sim p^\star$, and let $Z \sim \mathcal{N}(0, 1)$ be independent. For every $\sigma > 0$,*

$$\Pr(X + \sigma Z \notin A_\tau) \geq C_{p^\star}(\sigma) := \Pr(|X^\star + \sigma Z| > r_\tau).$$

The bound is attained by $p^\star$ with its centered normal interval. Moreover, $C_{p^\star}$ is continuous and strictly increasing from τ to 1. For every $\alpha \in (\tau, 1)$, the unique root s_p of $C_{p^\star}(s_p) = \alpha$ is the smallest noise guaranteeing this coverage across all equimeasurable density arrangements and all their admissible highest-density sets.

Proof. Write ϕ_σ for the centered Gaussian density with standard deviation σ, and let $\mathcal{H}_\tau(p)$ denote all highest-density sets of p with probability $1 - \tau$. By Eq. (9), each such set has length $2r_\tau$, where $r_\tau = V_p(1 - \tau)/2 \in (0, \infty)$.

Apply the Riesz rearrangement inequality to the nonnegative functions p, ϕ_σ, and 1_{A_τ}:

$$\begin{aligned} \Pr(X + \sigma Z \in A_\tau) &= \int\int p(x)\phi_\sigma(y - x)1_{A_\tau}(y) \, dx \, dy \\ &\leq \int\int p^\star(x)\phi_\sigma(y - x)1_{[-r_\tau, r_\tau]}(y) \, dx \, dy. \end{aligned} \tag{10}$$

Here the Gaussian is already symmetric decreasing, and the rearranged set indicator is the centered interval indicator. Taking complements proves Eq. (5). Since all equimeasurable arrangements have the same minimum-volume curve, the precise worst-case identity is

$$\inf_{\substack{\widetilde{p}^\star = p^\star \\ A \in \mathcal{H}_\tau(\widetilde{p})}} \Pr_{\widetilde{p}}(X + \sigma Z \notin A) = C_{p^\star}(\sigma). \tag{11}$$

The infimum is attained by $\widetilde{p} = p^\star$ and $A = [-r_\tau, r_\tau]$. Allowing all admissible sets matters when the density has plateaus; the inequality holds for every such set, whereas equality uses the centered choice. The class in Eq. (11) fixes the density values in their original units and does not impose a common variance on different spatial arrangements.

To establish the unique calibration root, put $q_\sigma = p^\star * \phi_\sigma$. Differentiation of the Gaussian convolution is valid for $\sigma > 0$ because its derivatives are bounded and $p^\star$ is integrable. The heat equation yields

$$\partial_\sigma q_\sigma = \sigma q''_\sigma, \qquad C'_{p^\star}(\sigma) = -2\sigma q'_\sigma(r_\tau). \tag{12}$$

For completeness, strict negativity of $q'_\sigma(r)$ for every $r > 0$ holds even if $p^\star$ is discontinuous. Express its superlevel sets as centered intervals $[-a(t), a(t)]$ by the layer-cake representation. Then

$$q'_\sigma(r) = \int_0^\infty \{\phi_\sigma(r + a(t)) - \phi_\sigma(r - a(t))\} \, dt < 0.$$

The integrand is strictly negative whenever $a(t) > 0$, since $r + a(t) > |r - a(t)|$. These levels have positive measure for a probability density. Interchange of differentiation and integration can also be justified by $|\phi_\sigma(r + a) - \phi_\sigma(r - a)| \leq 2a\|\phi'_\sigma\|_\infty$ and $\int_0^\infty 2a(t) \, dt = 1$.

Thus $C_{p^\star}$ is continuous and strictly increasing on $(0, \infty)$. The approximate-identity property of Gaussian convolution gives $C_{p^\star}(0) = \tau$. Also,

$$\Pr(|X^\star + \sigma Z| \le r_\tau) \le \frac{2r_\tau}{\sigma\sqrt{2\pi}} \longrightarrow 0,$$

so $C_{p^\star}(\sigma) \to 1$. For $\tau < \alpha < 1$, the intermediate value theorem gives a unique $s_p > 0$ with $C_{p^\star}(s_p) = \alpha$. Strict monotonicity and Eq. (11) show that s_p is the minimum noise satisfying the robust constraint. □

B.3 Minimum-perturbation allocation

Corollary 1 (Least perturbation at prescribed marginal coverage)**.** *For an arbitrarily dependent joint distribution, let s_j be the required scale for marginal j from Theorem 1. Among diagonal perturbations satisfying the certificates $C_j(\sigma_j) \ge \alpha_j$, $\sigma_j = s_j$ uniquely minimizes $\mathbb{E}\|DZ\|^2 = \sum_j \sigma_j^2$. The smallest feasible shared scale is $\max_j s_j$, whose excess perturbation is*

$$d \max_j s_j^2 - \sum_j s_j^2 \ge 0, \tag{13}$$

with strict inequality unless all required scales coincide.

Proof. For $D = \mathrm{diag}(\sigma_1, \dots, \sigma_d)$ and $Z \sim \mathcal{N}(0, I_d)$, strict monotonicity makes each marginal constraint equivalent to $\sigma_j \ge s_j$, regardless of dependence among the clean coordinates. Thus $\mathbb{E}\|DZ\|^2 = \sum_j \sigma_j^2$ is uniquely minimized at $\sigma_j = s_j$. A shared scale must be at least $\max_j s_j$, giving Eq. (13). □

C First-Order Kurtosis Calibration

C.1 Why kurtosis alone does not order tail radii

Even within symmetric unimodal, unit-variance densities, kurtosis does not determine a tail quantile. For example,

$$P = \tfrac{3}{5}\,\mathrm{Unif}[-1, 1] + \tfrac{2}{5}\,\mathrm{Unif}[-\sqrt{6}, \sqrt{6}]$$

has $\mathbb{E}X^2 = 1$ and $\mathbb{E}X^4 = 3$, but its two-sided 5% tail radius is $7\sqrt{6}/8 \simeq 2.1433$, whereas the Gaussian radius is 1.9600. Thus even zero excess kurtosis does not justify replacing a general density by the Gaussian reference. Symmetry eliminates odd cumulants when they exist; it does not control higher even cumulants in terms of the fourth.

Nor does a pairwise small-tail asymptotic establish a positive tail threshold uniform over an entire distribution family. For the unit-variance generalized Gaussian, put $k = 1/\beta$ and let G_k^{-1} be the unit-scale gamma quantile. At any fixed $\tau \in (0, 1)$,

$$R_\tau(\beta) = \left\{\frac{\Gamma(k)}{\Gamma(3k)}\right\}^{1/2} \{G_k^{-1}(1-\tau)\}^k.$$

As $k \to \infty$, the gamma law gives $G_k^{-1}(1-\tau) = k + O(\sqrt{k})$. Stirling's formula therefore yields

$$\log R_\tau(\beta) = \left(1 - \tfrac{3}{2}\log 3\right) k + O(\sqrt{k} + \log k) \longrightarrow -\infty,$$

whereas $\log \kappa(\beta) = (5\log 5 - 6\log 3)k + O(\log k) \to \infty$. Consequently no fixed positive tail level orders all generalized Gaussians by kurtosis. A numerical grid cannot prove a family-wide threshold. The coverage theorem avoids such an ordering assumption; the affine approximation instead states its local shape assumption explicitly.

C.2 Assumptions and controlled quantile inversion

Let p_δ be a family of valid, symmetric decreasing densities with mean zero, variance one, and fourth moment $3 + \delta$. For sufficiently small $|\delta|$ in its parameter domain, assume

$$p_\delta(x) = \phi(x)\left[1 + \frac{\delta}{24}H_4(x)\right] + \rho_\delta(x), \qquad \|\rho_\delta\|_1 \le K\delta^2. \tag{14}$$

The family may be one-sided. The full p_δ must be a density; the leading polynomial expression need not be nonnegative for both signs of δ. The condition controls all residual shape changes in L^1 and is stronger than finite kurtosis or symmetry alone. The coverage theorem of Appendix B does not require this condition.

Conditionally on $X^\star = x$, the probability that $x + \sigma Z$ falls in $[-r_\tau, r_\tau]$ is at most the probability for $x = 0$: a Gaussian assigns maximal mass to a centered interval of a given length. Averaging over $X^\star$ gives

$$C_{p^\star}(\sigma) \geq 2\{1 - \Phi(r_\tau/\sigma)\}. \tag{15}$$

Therefore $\sigma \geq r_\tau/\Phi^{-1}(1-\alpha/2)$ guarantees coverage α. This simple radius calculation measures a displacement starting at the center. The root s_p measures escape of actual corrupted observations and is no larger than this sufficient scale.

Proposition 2 (Cornish–Fisher scale approximation)**.** *Fix $\tau, \alpha \in (0,1)$, write $z_\tau = \Phi^{-1}(1-\tau/2)$, and let $r_\tau(\delta)$ be the two-sided tail radius of p_δ. Define $\sigma_{\mathrm{suff}}(\delta) = r_\tau(\delta)/\Phi^{-1}(1-\alpha/2)$. Under Eq.* (14)*,*

$$\frac{\sigma_{\mathrm{suff}}(\delta)}{\sigma_{\mathrm{suff}}(0)} = \frac{r_\tau(\delta)}{z_\tau} = 1 + \frac{z_\tau^2 - 3}{24}\delta + O(\delta^2).$$

The coefficient is positive whenever $z_\tau^2 > 3$.

Proof. Write $H_3(x) = x^3 - 3x$. The Hermite identity $(\phi H_3)' = -\phi H_4$ gives the uniform CDF expansion

$$F_\delta(x) = \Phi(x) - \frac{\delta}{24} H_3(x)\phi(x) + O(\delta^2). \tag{16}$$

Fix $\tau \in (0,1)$ and $z = \Phi^{-1}(1-\tau/2)$. Substituting $z + \delta H_3(z)/24$ into the displayed approximation gives CDF error $O(\delta^2)$. The derivative of the approximating CDF is bounded below by a positive constant on a fixed neighborhood of z for small enough $|\delta|$. Bracketing the true CDF with its uniform error therefore proves

$$r_\tau(\delta) = z + \frac{\delta}{24} H_3(z) + O(\delta^2). \tag{17}$$

The scale ratio follows because the denominator $\Phi^{-1}(1-\alpha/2)$ cancels. This is controlled Cornish–Fisher inversion; general nonasymptotic CDF-to-quantile bounds are studied by Ulyanov et al. (2016). No differentiability of ρ_δ in δ is needed for this argument. □

C.3 Averaged-coverage calibration

We derive the local noise-scale correction needed to maintain the coverage of Theorem 1, completing the proof of Proposition 1. This calculation supports the affine form of K-DSM; the global defaults are selected empirically. The coefficient below calibrates averaged coverage, whereas Figure 5b evaluates the tail-radius mappings defined in Appendix C.5.

Proposition 1 (Noise-scale correction at fixed coverage; restated)**.** *Fix $0 < \tau < \alpha < 1$. Let $z = \Phi^{-1}(1-\tau/2)$, $w = \Phi^{-1}(1-\alpha/2)$, $v = (z/w)^2$, and $\sigma_G = \sqrt{v-1}$. Under Eq.* (14)*, the exact coverage root satisfies*

$$\frac{\sigma_*(\delta)}{\sigma_G} = 1 + c_{\mathrm{cov}}\delta + O(\delta^2), \quad c_{\mathrm{cov}} = \frac{z^2(1 + v^{-1} + v^{-2}) - 3(1 + v^{-1})}{24}. \tag{18}$$

In particular, $c_{\mathrm{cov}} > 0$ whenever $z^2 > 3$. In the notation of Proposition 1, $s(\delta) = \sigma_*(\delta)$.

Proof. Let $v_\sigma = 1 + \sigma^2$. Since $\phi H_4 = \phi^{(4)}$, convolution with a Gaussian commutes with the fourth derivative and attenuates the standardized fourth-order term. Gaussian convolution also contracts the L^1 remainder in Eq. (14). Integrating the convolved density gives, uniformly in y and for σ near a fixed positive reference,

$$F_{\delta,\sigma}(y) = \Phi(y/\sqrt{v_\sigma}) - \frac{\delta}{24 v_\sigma^2} H_3(y/\sqrt{v_\sigma})\phi(y/\sqrt{v_\sigma}) + O(\delta^2). \tag{19}$$

Fix $\tau < \alpha < 1$ and put $w = \Phi^{-1}(1-\alpha/2)$, $v = (z/w)^2 > 1$. At the Gaussian reference, the root is $\sigma_G = \sqrt{v-1}$. By symmetry the exact coverage constraint is

$$F_{\delta,\sigma_*(\delta)}(r_\tau(\delta)) = 1 - \alpha/2.$$

For a candidate variance expansion $1 + \sigma_*(\delta)^2 = v + v_1\delta + O(\delta^2)$, substitution of Eqs. (17)–(19) and cancellation of the first-order CDF terms gives

$$\frac{H_3(z)}{24\sqrt{v}} - \frac{wv_1}{2v} = \frac{H_3(w)}{24v^2}.$$

Using $w = z/\sqrt{v}$, this simplifies to

$$v_1 = \frac{v}{12}\{z^2(1 - v^{-3}) - 3(1 - v^{-2})\}. \tag{20}$$

The relative coefficient of σ_* is $v_1/[2(v-1)]$. Factoring $v - 1$ in Eq. (20) yields

$$c_{\text{cov}} = \frac{z^2(1 + v^{-1} + v^{-2}) - 3(1 + v^{-1})}{24}, \qquad \frac{\sigma_*(\delta)}{\sigma_G} = 1 + c_{\text{cov}}\delta + O(\delta^2). \tag{21}$$

For the implicit calculation in the proof sketch of Proposition 1, write $C(\delta, \sigma) = 2\{1 - F_{\delta,\sigma}(r_\tau(\delta))\}$. Equations (17) and (19) give

$$\begin{aligned}\partial_\delta C(0, \sigma_G) &= -\frac{\phi(w)}{12}\left\{\frac{H_3(z)}{\sqrt{v}} - \frac{H_3(w)}{v^2}\right\},\\ \partial_\sigma C(0, \sigma_G) &= \frac{2\phi(w)z\sigma_G}{v^{3/2}} > 0.\end{aligned} \tag{22}$$

Their negative ratio, divided by σ_G, gives c_{cov}, recovering the expansion in Proposition 1.

To justify the root expansion rather than just a formal substitution, the candidate root has coverage error $O(\delta^2)$ by the uniform remainder above. The Gaussian coverage derivative at $\sigma_G > 0$ is strictly positive. Convolution of the L^1 residual with the bounded Gaussian derivative, together with Eq. (17), preserves a positive lower bound in a neighborhood of this root. Since each p_δ is symmetric decreasing, Theorem 1 makes its exact coverage strictly increasing in σ, so bracketing gives a root error $O(\delta^2)$.

The coefficient is positive precisely when

$$z^2 > \frac{3(1 + v^{-1})}{1 + v^{-1} + v^{-2}},$$

and the right-hand side is smaller than three. This proves the stated sufficient condition. □

C.4 Illustrative calibration and numerical check

With $\sigma_G = 0.5$, $v = 1.25$ and $c_{\text{cov}} = (2.44z^2 - 5.4)/24$. Taking $\tau = 0.02$ gives

$$\alpha = 2\{1 - \Phi(z/\sqrt{1.25})\} \simeq 0.0374569, \qquad c_{\text{cov}} \simeq 0.3252093.$$

The coefficient $c = 0.33$ corresponds exactly in this calculation to $\tau \simeq 0.0194676$ and $\alpha \simeq 0.0366369$. This numerical correspondence depends on the chosen coverage levels and is not independent evidence for the empirical default or its detection performance.

As a check of the expansion, $p_\delta = \phi(1 + \delta H_4/24)$ is a valid symmetric decreasing density for $0 \le \delta < 2.4$. Indeed, $\min H_4 = -6$ ensures positivity, and

$$p'_\delta(x) = -x\phi(x)\left[1 + \frac{\delta}{24}(x^4 - 10x^2 + 15)\right]$$

is negative for $x > 0$ since the polynomial in the parentheses has minimum -10 before adding one. Hermite orthogonality gives mean zero, variance one, and excess kurtosis exactly δ. Numerical integration and root-finding at $\tau = 0.02$ give Table 2. This example checks the algebra; no named distribution family is needed for Theorem 1.

C.5 Generalized Gaussian calibration and ablation settings

Figure 5b compares two mappings from rearranged kurtosis to noise scale: the affine rule and a generalized Gaussian distribution (GGD) fitted by matching kurtosis. The GGD rule scales noise

Table 2: Numerical check of the local coverage expansion. The last column evaluates the affine rule with $\sigma_{\text{base}} = 0.5$ and $c = 0.33$. These are analytic-model calculations, not anomaly-detection experiments.

δ	Exact root	First-order c_{cov}	Affine $c = 0.33$
0.001	0.50016265	0.50016260	0.50016500
0.010	0.50163100	0.50162605	0.50165000
0.050	0.50825404	0.50813023	0.50825000
0.100	0.51675461	0.51626046	0.51650000
0.500	0.59282089	0.58130232	0.58250000

according to the width of a central interval of the fitted distribution relative to the Gaussian interval containing the same probability mass. This section specifies the fit, the resulting noise-scale formula, and how the two rules' settings are aligned. GGD points are aligned using the corresponding τ, rather than treating c as a GGD parameter. The mapping from tail level to affine slope comes from the Cornish–Fisher (CF) coefficient

$$c_{\text{CF}}(\tau) = \frac{z_\tau^2 - 3}{24}, \qquad z_\tau = \Phi^{-1}(1 - \tau/2),$$

from the first-order radius ratio $r_\tau(\delta)/r_\tau(0)$. In this parameterization, $c = 0.33$ is indexed by $\tau \simeq 10^{-3}$ (Table 12). Here τ specifies the same marginal tail probability as in the coverage analysis, but c_{CF} comes from the center-displacement radius rule. It differs from the coefficient c_{cov} for averaged noisy-observation coverage in Eq. (21).

For the center-displacement rule, the normalized ratio is exactly $R_\tau(\delta)/R_\tau(0)$: the coverage parameter α cancels at every order, not only at first order. Increasing c changes the factor $1 + c\delta$ and does not multiply all displacements by the ratio of the slopes. The coefficient c_{cov} instead retains α because it calibrates actual noisy observations.

For the GGD comparison, the symmetric density proportional to $\exp[-(|x|/a)^\beta]$ has

$$a(\beta)^2 = \frac{\Gamma(1/\beta)}{\Gamma(3/\beta)}, \qquad \kappa(\beta) = \frac{\Gamma(5/\beta)\Gamma(1/\beta)}{\Gamma(3/\beta)^2} \tag{23}$$

after enforcing unit variance. These identities follow by substituting $t = (|x|/a)^\beta$ in its even-moment integrals. If $G_{1/\beta}^{-1}$ is the unit-scale gamma quantile of shape $1/\beta$, its exact two-sided tail radius is

$$R_\tau(\beta) = a(\beta)\{G_{1/\beta}^{-1}(1 - \tau)\}^{1/\beta}.$$

For GGD calibration, we clip feature kurtosis to $[1.802, 200]$, fit the GGD shape to this value, and use $\sigma_{\text{base}} R_\tau(\beta)/R_\tau(2)$. Both mappings clip the resulting scale to $[0.1, 3.0]$. Figure 5b evaluates the detection performance of these calibration rules; it does not directly measure their quantile-approximation error. The coverage analysis does not require a GGD fit or a universal ordering of tail radii by kurtosis.

C.6 Affine calibration versus numerical coverage roots

We assess Eq. (3) against the coverage equation, without training a detector. We reconstruct the seed-0 training-normal histograms for all 121 task variants using the unchanged data generator, the 50,000-observation cap, and 200 equal-width bins. The resulting 44,951 coordinates reproduce the saved calibration inputs. We exclude 7,224 encoded-feature fallbacks and 4,256 additional coordinates with undefined rank kurtosis, leaving 33,471 valid comparisons. The comparison uses the measured rank kurtosis in Eq. (3); the root uses the exact continuous symmetric rearrangement of the same histogram, standardized to unit variance. It therefore includes the finite-bin and rank-embedding differences discussed in Appendix E.

We use $\tau = 0.0194676024$ and $\alpha = 0.0366368643$, which give $\sigma_G = 0.5$ and $c_{\text{cov}} = 0.33$ in the local expansion. If sorted bin b has probability m_b, its continuous rearrangement allocates this mass uniformly to a radial shell $[\ell_b, u_b]$. For radius r_τ, its coverage is

$$C(\sigma) = \sum_b \frac{m_b}{u_b - \ell_b} \int_{\ell_b}^{u_b} \left[\Phi\left(\frac{-r_\tau - t}{\sigma}\right) + \Phi\left(\frac{t - r_\tau}{\sigma}\right)\right] dt. \tag{24}$$

The Gaussian CDF antiderivative evaluates each integral analytically. We solve $C(s_j) = \alpha$ by 42 bisection steps, with the sufficient scale from Eq. (15) as the initial upper bound. The maximum absolute coverage residual is below 3×10^{-14}; independent numerical quadrature checks agree within 10^{-8}. These are numerical roots for the estimated histogram profiles, not exact calibrations of the unknown population marginals.

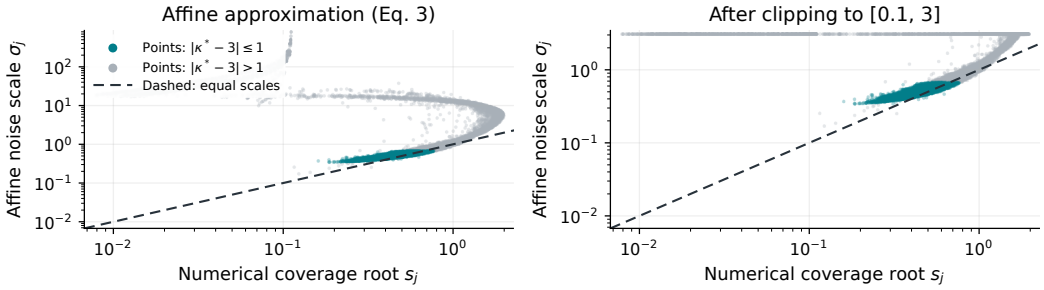


Figure 8: Affine calibration versus numerical coverage roots on 33,471 feature coordinates with defined rank kurtosis and no encoded-feature fallback. Left: Eq. (3) before clipping. Right: after clipping to $[0.1, 3.0]$. Dashed lines indicate equal scales; colored marks are individual feature coordinates, not fitted curves. Color distinguishes $|\kappa^\star - 3| \le 1$ from larger departures; proximity in kurtosis alone does not verify the proposition's local density assumption. Both axes are logarithmic.

Below the upper clip (26,697 coordinates), the affine scale tracks the root closely: the Spearman correlation is 0.96, the median affine-to-root ratio is 1.16, 64.1% of affine scales lie within 20% of the root, and 99.9% lie within a factor of two. Among the 17,694 coordinates with $|\kappa^\star - 3| \le 1$, the median ratio is 1.14 and 72.1% lie within 20%. Across all 33,471 valid coordinates, the median ratio is 1.20, with 51.1% within 20% and 81.1% within a factor of two. Agreement deteriorates at large kurtosis: the 6,774 valid coordinates reaching the upper clip have a median ratio of 4.61 before clipping and 1.85 afterward, with a median root of 1.62. Together with 4,256 undefined-kurtosis fallbacks, these account for the 11,030 upper-clipped coordinates reported in Appendix G.4.

The affine rule meets the specified histogram coverage target on 92.2% of valid coordinates, so it does not inherit the exact root's universal certificate. This diagnostic identifies where the local approximation is useful and where clipping limits a substantial overestimate. It does not evaluate the detection performance of training with numerical roots, or establish that the chosen coverage levels are optimal for anomaly detection.

D Local Coverage and Score Learnability

The preceding results concern exposure outside a normal region. We now connect local exposure to estimation of the actual noise-prediction field. Gaussian kernel score estimators have established theory, including minimax results under subgaussian and Lipschitz-score assumptions (Wibisono et al., 2024). The result here is a pointwise bound for the smoothed, noise-normalized target in terms of local kernel mass.

D.1 Population and empirical DSM solutions

Fix a positive diagonal matrix D independently of the sample, and let $Y = X + DZ$ with $Z \sim \mathcal{N}(0, I_d)$. Conditional squared-error minimization of Eq. (4) gives

$$f_D^*(x) = -\mathbb{E}[Z \mid Y = x] = D\nabla \log q_D(x) = \mathbb{E}[D^{-1}(X - x) \mid Y = x]. \tag{25}$$

The identity follows by differentiating the Gaussian convolution. At each query it remains well defined for any probability measure P, since polynomially weighted Gaussian kernels are bounded. It does not require a density or unweighted moments of P. Conditional squared-error projection also gives Eq. (39); its integrated error is weighted by q_D, so it does not by itself control errors at every clean or anomalous test point.

For independent clean samples $X_1, \dots, X_n$, let

$$U_i = D^{-1}(X_i - x), \quad k_i = e^{-\|U_i\|^2/2}, \quad m = m_D(x) = \mathbb{E}e^{-\|D^{-1}(X-x)\|^2/2} > 0.$$

The unrestricted minimizer of the empirical DSM loss with corruption integrated out is

$$\widehat{f}_D(x) = \frac{\sum_{i=1}^n k_i U_i}{\sum_{i=1}^n k_i}. \tag{26}$$

This follows by applying Eq. (25) to the empirical clean distribution. It identifies a statistical reference for the neural estimator, not an assertion that finite network training exactly computes this ratio.

Theorem 2 (Pointwise estimation from local training mass)**.** *For any probability measure P on $\mathbb{R}^d$, fixed $D \succ 0$, fixed query x, and $t > 0$,*

$$\Pr_{\text{train}}\{\|\widehat{f}_D(x) - f_D^*(x)\| > t\} \le e^{-nm/8} + \frac{8\log(1/m)}{nm\,t^2}. \tag{27}$$

No moment, density, symmetry, or coordinate-independence assumption is required.

Proof. Let $U = D^{-1}(X - x)$, $k = e^{-\|U\|^2/2}$, and define the Gaussian posterior Q_x by $dQ_x/dP = k/m$. Its moments of all positive integer orders are finite. Nonnegativity of relative entropy gives

$$0 \le \mathrm{KL}(Q_x\|P) = -\tfrac{1}{2}\mathbb{E}_{Q_x}\|U\|^2 - \log m,$$

so $\mathbb{E}_{Q_x}\|U\|^2 \le 2\log(1/m)$. Put $W_i = k_i(U_i - f_D^*(x))$. These independent vectors have mean zero, and $k_i^2 \le k_i$ implies

$$\mathbb{E}\|W_i\|^2 \le m\,\mathrm{tr}\,\mathrm{Cov}_{Q_x}(U) \le 2m\log(1/m).$$

The independent kernel weights lie in $[0,1]$, so the multiplicative Chernoff inequality gives $\Pr(n^{-1}\sum_i k_i < m/2) \le e^{-nm/8}$. On the complementary event,

$$\|\widehat{f}_D - f_D^*\| \le \frac{2}{m}\left\|\frac{1}{n}\sum_i W_i\right\|.$$

Apply Markov's inequality to the squared norm and use independence to obtain the second term of Eq. (27). If $m = 1$, P is concentrated at x and the estimation error is zero. □

The relevant support measure is $nm_D(x)$, with

$$m_D(x) = (2\pi)^{d/2}|D|q_D(x). \tag{28}$$

The determinant is part of this measure: a change in normalized density alone does not give the same estimation certificate. The bound can be loose, particularly for small $m_D(x)$ in high dimension, but explicitly identifies the statistical role of local coverage. It also fixes D before sampling; K-DSM selects D from the same training data, which the bound does not cover unless calibration uses an independent sample.

D.2 Transfer to anomaly ranking

Let X_0 and X_1 be independent normal and anomaly test draws, independent of training. Write $B_D(x) = \|f_D^*(x)\|$ and $A_D(x) = \|\widehat{f}_D(x)\|$.

Corollary 2 (Expected AUROC under a population margin)**.** *For $\gamma > 0$ and $m_0 \in (0,1]$, define*

$$\pi = \Pr\{B_D(X_1) - B_D(X_0) > \gamma,\ m_D(X_0) \ge m_0,\ m_D(X_1) \ge m_0\}.$$

Then

$$\mathbb{E}_{\text{train}}\,\mathrm{AUROC}(A_D) \ge \pi - 2e^{-nm_0/8} - \frac{64\log(1/m_0)}{nm_0\gamma^2}. \tag{29}$$

Proof. The reverse triangle inequality bounds norm-score error by vector error. Apply Theorem 2 with $t = \gamma/2$ to both test points. If both vector errors are at most $\gamma/2$, a strict population margin greater than γ preserves their ordering. A union bound, monotonicity of $\log(1/m)/m$ on $(0,1]$, and integration over the test pair give the result. Counting strict correct orderings lower-bounds AUROC even when ties receive half credit. □

For a neural output f_θ, assume additionally that $\|f_\theta - \widehat{f}_D\| \le e$ on both queries almost surely. This consumes $2e$ of the margin. If $\gamma > 2e$, replace γ by $\gamma - 2e$ in the error term. If this additional condition fails with probability at most η_{net}, subtract η_{net} from the ranking bound. The condition must account for finite corruption draws, network approximation, and optimization; the DSM objective alone does not establish it. The bound requires a useful population margin as well as local support, and does not identify every low-density point as an anomaly.

D.3 Gaussian example: local support and score separation

For $X \sim \mathcal{N}(0, I_d)$ and $D = \mathrm{diag}(\sigma_1, \dots, \sigma_d)$, Gaussian convolution gives $q_D = \mathcal{N}(0, I_d + D^2)$. Differentiating its log density proves

$$f_D^*(x) = -D(I_d + D^2)^{-1}x, \qquad m_D(x) = \prod_j \frac{\sigma_j}{\sqrt{1+\sigma_j^2}} \exp\left[-\frac{1}{2}\sum_j \frac{x_j^2}{1+\sigma_j^2}\right]. \tag{30}$$

The second identity follows by integrating the Gaussian kernel defining m_D. It increases in every σ_j, while each score weight $h(\sigma_j) = \sigma_j/(1+\sigma_j^2)$ has derivative $(1-\sigma_j^2)/(1+\sigma_j^2)^2$ and tends to zero as $\sigma_j \to \infty$.

To exhibit a change in ordering, take $d = 2$, $D = \mathrm{diag}(1, s)$, and test locations $x_0 = (1, 0)$ and $x_1 = (0, 2)$. Their ideal score gap is

$$\|f_D^*(x_1)\| - \|f_D^*(x_0)\| = \frac{2s}{1+s^2} - \frac{1}{2}. \tag{31}$$

It is positive precisely for $2 - \sqrt{3} < s < 2 + \sqrt{3}$, reaches $1/2$ at $s = 1$, and becomes negative for sufficiently large s. Local mass at both locations continues to increase with s. Thus increasing local support need not preserve a positive score gap. For pairs with a positive gap, the estimation penalty in Corollary 2 depends on both local mass and the gap through $1/(nm_0\gamma^2)$, up to its logarithmic factor. The gap assumption can fail entirely after the ordering reverses.

This is a pointwise illustration of the mechanism, not an anomaly-distribution model or an optimal-noise theorem. In this isotropic Gaussian example, a shared scale $D = sI_d$ instead multiplies every norm score by the same positive factor and leaves ideal AUROC and AUC-PR unchanged. Consequently shrinking absolute scores alone does not prove poorer detection. The example above uses unequal scales to demonstrate changing relative feature weights. For a general normal distribution, changing the noise also changes the shape of the smoothed score field. The coverage-calibration principle limits perturbation under marginal constraints; it does not directly optimize the ranking bound.

E From Rearranged Profiles to Histogram Calibration

E.1 Standardization preserves a conservative certificate

Let a marginal have finite variance and let $s_\star^2 = \mathrm{Var}(X^\star)$. Applying Hardy–Littlewood to any centered interval translated to a gives

$$\Pr(|X - a| \le r) \le \Pr(|X^\star| \le r).$$

Integration of the squared-distance tails and minimization over a imply

$$s_\star^2 \le \mathrm{Var}(X). \tag{32}$$

For an original unit-variance feature, put $W = X^\star/s_\star$ and $R_\tau = r_\tau/s_\star$. By Theorem 1 and monotonicity of its coverage function,

$$\begin{aligned}\Pr(X + \sigma Z \notin A_\tau) &\ge \Pr(|W + (\sigma/s_\star)Z| > R_\tau) \\ &\ge \Pr(|W + \sigma Z| > R_\tau).\end{aligned} \tag{33}$$

Thus calibrating against the standardized rearranged profile is conservative in standardized original coordinates. This normalization can increase the sufficient scale, so its optimum need not equal the smallest scale for a particular original arrangement.

E.2 FINITE-BIN APPROXIMATION

Let $m_1 \geq \cdots \geq m_B$ be the sorted masses of equal-width bins of width h. The nominal rank embedding places atoms at $0, +h, -h, +2h, -2h, \ldots$. It is exactly invariant to permutations of the original equal-width bins. Unlike the continuous rearrangement, a finite alternating embedding need not be exactly symmetric.

Let p_{alt} place each sorted mass uniformly over its rank bin and $p^{\star}_{\mathrm{hist}}$ be the exact symmetric decreasing rearrangement of the histogram. Padding absent masses by zero, direct comparison on the half-bin shells gives

$$\|p_{\mathrm{alt}} - p^{\star}_{\mathrm{hist}}\|_1 = \sum_{k\geq 1}(m_{2k} - m_{2k+1}) \leq m_2. \tag{34}$$

The inequality follows from $m_2 - m_3 + m_4 - m_5 + \cdots \leq m_2$. For the atoms actually used in the moment calculation, the rank mean lies in $[0, h/2]$. To prove this, express the decreasing mass sequence as a mixture of uniform prefixes with weights $a_l = l(m_l - m_{l+1})$. An odd prefix has mean zero and an even prefix has mean $h/2$.

Couple each atom with the exact rearranged radial shell of the same rank. Their absolute positions differ by at most $h/2$; centering the atoms adds at most another $h/2$. Let s^2_{rank} and $m_{4,\mathrm{rank}}$ be their centered second and fourth moments, and $s^2_{\star}, m_{4,\star}$ the corresponding exact histogram-rearrangement moments. The coupling gives

$$\begin{aligned} |s^2_{\mathrm{rank}} - s^2_{\star}| &\leq 2h\mathbb{E}|X^{\star}| + h^2 =: e_2, \\ |m_{4,\mathrm{rank}} - m_{4,\star}| &\leq 4h\mathbb{E}(|X^{\star}| + h)^3 =: e_4. \end{aligned} \tag{35}$$

Under a consistent histogram sequence, rearrangement contraction transfers L^1 consistency (Devroye, 1987; Lieb & Loss, 2001). Moment convergence additionally requires tail control, such as uniform integrability of fourth powers. The finite-bin bounds above quantify approximation for a fixed histogram without assuming that 200 bins resolves every marginal equally well.

E.3 CALIBRATION USED BY K-DSM

The implementation computes a single noise vector from the training data, then trains and scores on the original feature coordinates. For each feature it uses $B = 200$ equal-width bins on the observed values and computes Pearson kurtosis on the standardized rank embedding. Multiplying all rank locations by a common bin width leaves this kurtosis unchanged, so unit rank spacing is sufficient.

Algorithm 1 Histogram rank embedding for noise calibration

Require: Observed feature values $x_1, \ldots, x_n$; $B = 200$
1: Compute equal-width histogram counts and edges
2: Sort bins by decreasing count, obtaining permutation π
3: Assign $c_{\pi(1)} = 0$ and, for $r \geq 2$, $c_{\pi(r)} = (-1)^r \lfloor r/2 \rfloor$
4: Initialize $x^*_i = 0$ for all i
5: **for** each sample x_i **do**
6: Find its index b_i using the histogram edges
7: **if** b_i is a valid bin index **then**
8: $x^*_i \leftarrow c_{b_i}$
9: **end if**
10: **end for**
11: Standardize the transformed vector and compute its Pearson kurtosis

The digitization routine assigns an observation exactly at the final bin edge an out-of-range index, leaving its transformed value at zero. Equations (34)–(35) describe the nominal bin-mass embedding. If this endpoint convention changes a fraction ϵ of rank assignments, a coupling differs on at most probability ϵ, in addition to the nominal histogram approximation error.

With $\sigma_{\mathrm{base}} = 0.5$ and $c = 0.33$, the final scale is

$$\sigma_j = \mathrm{clip}_{[0.1,3.0]}\left\{\sigma_{\mathrm{base}}[1 + c(\kappa_j - 3)]\right\}. \tag{36}$$

Here κ_j is the rearranged histogram kurtosis, with the following exceptions. Features whose observed values are a subset of $\{-1, 0, 1\}$ use the fallback $\kappa_j = 0.9/\sigma_{\text{base}}$. At the default base scale this gives $\kappa_j = 1.8$ and $\sigma_j = 0.302$. Undefined or positive-infinite kurtosis values are replaced by 1000; negative infinity is replaced by 10^{-4} before scaling and clipping. The affine rule follows from a first-order expansion under the local shape assumption of Proposition 1; the discrete-feature fallback and clipping are empirical choices.

With $c = 0.33$, the unclipped rule is $\hat{\sigma}_j = \sigma_{\text{base}}(0.01+0.33\kappa_j^\star)$, close to $\sigma_{\text{base}}\kappa_j^\star/3$. The encoded and undefined-kurtosis fallbacks account for 7,224 and 4,256 of the 44,951 saved coordinates, respectively (25.5% together; Appendix G.4).

The exact root of Theorem 1 provides an interpretable reference against which this inexpensive approximation can be assessed. The reported detector uses Eq. (36); it does not numerically solve the root for every feature.

F Additional Related Work

Classical tabular detectors. Classical methods identify anomalies through deviations from a learned geometry or distribution. PCA (Shyu et al., 2003) measures standardized deviations along major and minor principal components, while KPCA (Hoffmann, 2007) uses reconstruction error in kernel feature space to describe nonlinear structure. OCSVM (Schölkopf et al., 2001) estimates the support of normal data through a kernel decision boundary. MCD (Rousseeuw & Driessen, 1999) estimates a robust center and covariance from a subset with a small covariance determinant, then scores observations by their Mahalanobis distance. Gaussian mixture models (GMMs) fit a mixture density and use negative log-likelihood as an anomaly score. Neighborhood methods use local comparisons: kNN (Ramaswamy et al., 2000) ranks observations by the distance to their kth nearest neighbor, whereas LOF (Breunig et al., 2000) compares an observation's local reachability density with those of its neighbors, allowing the reference density to vary across the data. Isolation Forest (Liu et al., 2008) instead uses random recursive partitions, assigning higher anomaly scores to observations isolated by shorter tree paths. Other methods aggregate evidence across individual features. HBOS (Goldstein & Dengel, 2012) combines univariate histogram densities under a feature-independence assumption. ECOD (Li et al., 2023) estimates marginal tail probabilities with empirical cumulative distribution functions and aggregates their negative logarithms, using skewness to guide tail selection. COPOD (Li et al., 2020) takes an empirical-copula approach to tail scoring, combining left-tail, right-tail, and skewness-adjusted evidence. These marginal density and tail scores differ from K-DSM's use of marginal statistics to calibrate perturbations for a network trained on the joint feature vector.

Deep tabular detectors. Deep methods learn representations or auxiliary prediction tasks that define normality. Deep SVDD (Ruff et al., 2018) contracts neural representations around a fixed center and scores their squared distance from that center. DAGMM (Zong et al., 2018) jointly learns an autoencoder and a Gaussian mixture model over latent representations augmented with reconstruction-error features; its anomaly score is the resulting sample energy. GOAD (Bergman & Hoshen, 2020) applies random affine transformations and learns distinct clusters for their representations, scoring test observations through distance-based probabilities of recovering the correct transformation. NeuTraL-AD (Qiu et al., 2021) learns the transformations themselves, using a contrastive objective that keeps transformed views close to their original observation while separating different views; the same objective supplies its anomaly score. ICL (Shenkar & Wolf, 2022) contrasts feature subsets and their complements within each observation, detecting departures from learned feature dependencies through the test-time contrastive loss. SLAD (Xu et al., 2023) predicts scale distributions associated with representations of sampled feature subspaces and scores disagreement with the target distributions; here, scale describes the relationship between subspace and representation dimensionality. MCM (Yin et al., 2024) learns multiple masks with a diversity penalty and averages reconstruction errors across the masked views to detect violations of feature relationships. LUNAR (Goodge et al., 2022) learns to aggregate nearest-neighbor distances within a graph message-passing formulation, using synthetic negative samples to train its anomaly scorer. NRDE (Fu et al., 2026) learns a Jacobian-regularized normalizing flow to separate latent data and nuisance noise sources, then uses the density obtained after suppressing the inferred noise sources for anomaly scoring. This targets inherent nuisance variation in the observations.

Methods evaluated. Under normal-only training, our benchmark includes KPCA, GMM, LOF, both kNN variants, LUNAR, ICL, SLAD, and NRDE, together with the noise-based detectors discussed in Section 3, including NEAD (Dai et al., 2025) and NCSBAD (Hirth & Kasneci, 2025); the contaminated-training comparison adds MCD and MCM and omits NRDE and kNN-mean (Appendix L). The broader discussion above also covers methods cited for context. Appendix G.2 specifies the implementations, nearest-neighbor scoring variants, and evaluation coverage.

G Implementation Details

G.1 Benchmark Description

We conduct experiments on the ADBench benchmark suite (Han et al., 2022), which provides a standardised evaluation framework across 57 datasets and 30 anomaly detection algorithms. The benchmark includes 47 established tabular datasets and 10 additional datasets introduced by the authors to extend coverage into computer vision and natural language processing. For these latter domains, features are obtained from pre-trained models such as ResNet for images and Transformer-based encoders for text, ensuring that all methods are compared on fixed representations rather than raw inputs. The 10 CV/NLP datasets are further organized into 74 subclasses to enable fine-grained evaluation. For the semi-supervised experiments, we follow Livernoche et al. (2024): after capping each task at 50,000 observations, we randomly split the normal observations equally between training and testing and include all anomalies in testing. All methods share these splits, with scalers fitted on training normals only (Section 4.1). Table 3 shows a description of all datasets as reported by Han et al. (2022).

Table 3: Description of all datasets in ADBench

Dataset Name	# Samples	# Features	# Anomaly	% Anomaly	Category
ALOI	49534	27	1508	3.04	Image
annthyroid	7200	6	534	7.42	Healthcare
backdoor	95329	196	2329	2.44	Network
breastw	683	9	239	34.99	Healthcare
campaign	41188	62	4640	11.27	Finance
cardio	1831	21	176	9.61	Healthcare
Cardiotocography	2114	21	466	22.04	Healthcare
celeba	202599	39	4547	2.24	Image
census	299285	500	18568	6.20	Sociology
cover	286048	10	2747	0.96	Botany
donors	619326	10	36710	5.93	Sociology
fault	1941	27	673	34.67	Physical
fraud	284807	29	492	0.17	Finance
glass	214	7	9	4.21	Forensic
Hepatitis	80	19	13	16.25	Healthcare
http	567498	3	2211	0.39	Web
InternetAds	1966	1555	368	18.72	Image
Ionosphere	351	32	126	35.90	Oryctognosy
landsat	6435	36	1333	20.71	Astronautics
letter	1600	32	100	6.25	Image
Lymphography	148	18	6	4.05	Healthcare
magic.gamma	19020	10	6688	35.16	Physical
mammography	11183	6	260	2.32	Healthcare
mnist	7603	100	700	9.21	Image
musk	3062	166	97	3.17	Chemistry
optdigits	5216	64	150	2.88	Image
PageBlocks	5393	10	510	9.46	Document
pendigits	6870	16	156	2.27	Image
Pima	768	8	268	34.90	Healthcare
satellite	6435	36	2036	31.64	Astronautics
satimage-2	5803	36	71	1.22	Astronautics
shuttle	49097	9	3511	7.15	Astronautics
skin	245057	3	50859	20.75	Image
smtp	95156	3	30	0.03	Web
SpamBase	4207	57	1679	39.91	Document
speech	3686	400	61	1.65	Linguistics
Stamps	340	9	31	9.12	Document
thyroid	3772	6	93	2.47	Healthcare
vertebral	240	6	30	12.50	Biology
vowels	1456	12	50	3.43	Linguistics
Waveform	3443	21	100	2.90	Physics
WBC	223	9	10	4.48	Healthcare
WDBC	367	30	10	2.72	Healthcare
Wilt	4819	5	257	5.33	Botany
wine	129	13	10	7.75	Chemistry
WPBC	198	33	47	23.74	Healthcare
yeast	1484	8	507	34.16	Biology
CIFAR10	5263	512	263	5.00	Image
FashionMNIST	6315	512	315	5.00	Image
MNIST-C	10000	512	500	5.00	Image
MVTec-AD	5354	512	1258	23.50	Image
SVHN	5208	512	260	5.00	Image
Agnews	10000	768	500	5.00	NLP
Amazon	10000	768	500	5.00	NLP
Imdb	10000	768	500	5.00	NLP
Yelp	10000	768	500	5.00	NLP
20newsgroups	11905	768	591	4.96	NLP

G.2 Baseline implementations and aggregation

All 121 tasks are aggregated into the same 57 datasets: variants are averaged within groups, groups within each seed, and then seeds. Standard errors for macro results are computed across the five seed-level macro means. A dataset group is complete only if all its constituent tasks are available in all reported seeds (five); missing values are never averaged into a benchmark score.

AUC-PR is computed as average precision. We prioritize it to emphasize the precision–recall trade-off when anomalies are rare (Davis & Goadrich, 2006; Saito & Rehmsmeier, 2015); class imbalance alone does not make it universally preferable to AUC-ROC (McDermott et al., 2024). We therefore report AUC-ROC and F1 alongside it.

For ordered Euclidean distances $d_1(x) \leq \cdots \leq d_5(x)$ to the five nearest training observations, kNN-mean uses $\frac{1}{5}\sum_{i=1}^{5} d_i(x)$ and kNN uses $d_5(x)$. The mean-distance variant is the nearest-neighbor rule associated with DTE-NP (Livernoche et al., 2024); the fifth-neighbor rule is a different aggregation. The neural DTE row uses DTE-C. NEAD is reimplemented, and NCSBAD uses an adaptation of released code. NRDE's implementation and evaluation differences are documented below. The combined K-DSM+DenoiserAD configuration is described in Appendix N.

NRDE evaluation and test-label leakage. The released NRDE evaluator selects the epoch with the highest test AUC-ROC and returns its AUC-PR. This uses test labels for model selection, even though the training objective itself does not use anomaly labels. We verified the selection in `NRDE_run`, lines 73–99 of the authors' released code (commit `786e676`). Our evaluations use the predefined final epoch.

To measure this effect, we ran the released algorithm for 100 epochs on the 47 classical tasks with one fixed seed and our data splits. Final-epoch evaluation gives 0.5398 mean AUC-PR and 0.7661 mean AUC-ROC. Selecting the best test-AUC-ROC epoch from the same runs raises these means to 0.6276 and 0.8435. These test-selected values are diagnostic only and are not used in the benchmark tables. The comparison demonstrates that test-label selection can substantially inflate performance in this setting; it does not identify the sole cause of the difference from the published results.

The clean-training NRDE benchmark entries use the released algorithm on all 121 tasks (57 dataset groups), with final-epoch evaluation and standard errors over five seeds. The algorithm selects latent coordinates from training-data Jacobians, zeros the remaining coordinates, reconstructs the input through the inverse flow, and evaluates its density. We use a persistent optimizer, 100 epochs, and batch size 512, except InternetAds (128 due to memory limits). NRDE is omitted from the contaminated-training comparison because the earlier adaptation differs from the released algorithm. The 47-task diagnostic above isolates test-label selection within the released algorithm, without attributing the full performance gap to that selection rule.

DDAE reproduction. Under our evaluation protocol, DDAE achieves approximately 0.594 AUC-PR and 0.819 AUC-ROC, compared with the published values of 0.614 and 0.832 (Sattarov et al., 2025). As a preprocessing diagnostic, fitting the feature scaler on the full dataset gives 0.593 AUC-PR, so this change does not explain the gap. All benchmark comparisons use the shared protocol, with scalers fitted on training normals only.

G.3 Binary and categorical features: scope and implementation

For a nondegenerate Bernoulli feature with probability p, Pearson kurtosis is

$$\kappa = \frac{1}{p(1-p)} - 3.$$

It equals 1 for a balanced binary variable and grows as the minority class becomes rare. This measures class balance rather than the decay of a continuous tail. One-hot indicators have the same limitation; ordinal or integer-valued features with more than two levels can still be processed by the binned marginal statistic. The continuous tail argument therefore does not directly justify scales for binary features.

To avoid this issue, we take columns with values in $\{0, 1\}$ or $\{-1, 0, 1\}$ and leave them unstandardized; other numeric columns are standardized using training statistics. K-DSM then checks the

observed values of each training coordinate. If they are contained in $\{-1, 0, 1\}$, including a binary or one-hot indicator, it bypasses histogram kurtosis and sets $\kappa_j = 0.9/\sigma_{\text{base}}$. With $\sigma_{\text{base}} = 0.5$ and $c = 0.33$, Eq. (36) assigns $\sigma_j = 0.302$ regardless of category frequency. Numeric categorical coordinates with more than three levels receive the usual histogram-based calibration; the implementation does not use categorical type metadata. Training adds independent Gaussian noise with standard deviation σ_j to every coordinate, including binary and categorical ones, so perturbed values need not remain valid category levels. The discrete-feature fallback is an empirical choice, outside the continuous-tail justification of the calibration rule.

Fallback-assigned scales and the gain over DSM. Two calibration rules bypass measured kurtosis (Appendix E.3): the encoded-feature fallback, and the replacement of undefined kurtosis by 1000, which places the feature at the upper clip. Together they set 11,480 of the 44,951 feature coordinates (25.5%); measured kurtosis above 1000 occurs on 102 coordinates and is not a fallback. To check whether the improvement over DSM depends on these cases, we compute for each dataset the fraction of its coordinates assigned by either fallback, using the seed-0 calibration of Appendix G.4, and stratify the same K-DSM minus DSM differences as Table 13.

Table 4: K-DSM minus DSM performance by the fraction of fallback-assigned noise scales: 56 groups with calibration summaries and five seeds for performance. n counts datasets; wins count datasets with positive ΔPR.

Fraction of fallback coordinates	n	ΔPR	Wins	ΔROC
0	33	+0.046	23/33	+0.006
$(0, 0.1]$	5	+0.003	3/5	+0.002
$(0.1, 0.5]$	12	+0.065	10/12	+0.033
> 0.5	6	+0.008	2/6	-0.008
All	56	+0.042	38/56	+0.010

The improvement does not require the fallbacks. On the 33 datasets where every scale comes from measured kurtosis, K-DSM improves mean AUC-PR by 0.046 and wins on 23, matching the overall average. On the six datasets where fallbacks assign more than half of the coordinates, the mean difference is 0.008 with two wins. Across datasets, the Spearman correlation between fallback fraction and ΔPR is -0.10. The largest gains in the intermediate stratum occur on image-derived datasets (FashionMNIST, MNIST-C, optdigits) whose constant coordinates (ResNet-embedding dimensions for FashionMNIST and MNIST-C, always-zero border pixels for optdigits) have undefined kurtosis and receive the upper-clip scale; on these datasets the fallback and the kurtosis rule act together, and this diagnostic does not separate their contributions.

G.4 Assigned noise scales across all ADBench features

To examine how the calibration allocates noise on real data, we use the feature statistics from seed 0 of the semi-supervised protocol, with the same 50,000-observation cap and training-normal subset. This covers all 121 task variants across the 57 benchmark groups and yields 44,951 feature coordinates. Figure 9 pairs each feature's kurtosis input with the scale computed by Eq. (36). This analysis requires only preprocessing and calibration, without training a network.

The assigned scales span approximately 0.173 to 3.0. A total of 11,030 coordinates (24.5%) reach the upper clip and none reach the lower clip; 7,224 coordinates use the encoded-feature fallback. Of the upper-clipped coordinates, 4,256 use an undefined-kurtosis fallback rather than a measured tail statistic; Appendix C.6 separates these cases when assessing the coverage approximation. The distribution shows the range of perturbations assigned by K-DSM; Appendix K evaluates their contribution to detection performance.

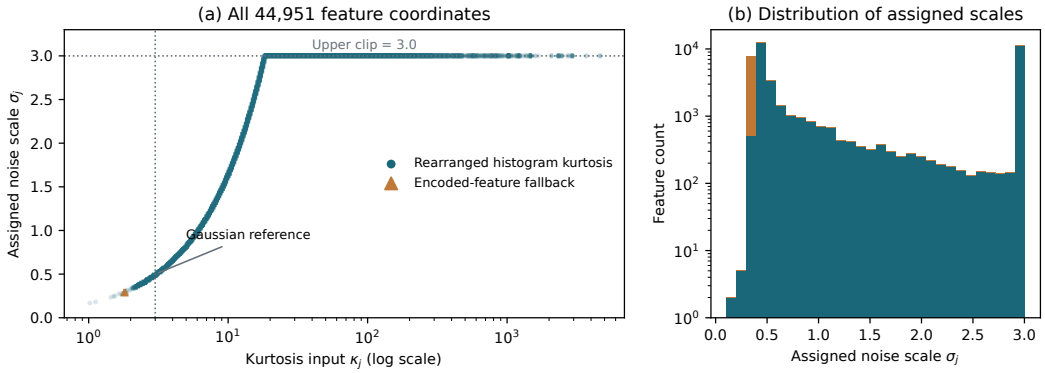


Figure 9: Feature-wise noise calibration across 57 ADBench groups (121 task variants, 44,951 coordinates, one preprocessing seed). Left: the scale from Eq. (36) versus its kurtosis input, on a logarithmic horizontal axis. Encoded features use the fallback defined in Appendix E.3 and are shown separately. Right: counts of assigned scales across all coordinates. These counts weight coordinates equally, rather than datasets.

G.5 Algorithm

Algorithm 2 K-DSM calibration, training, and inference

Require: Training data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{n}$ in preprocessed feature coordinates
Initialize network parameters θ
for each feature j **do**
  if observed values are contained in $\{-1, 0, 1\}$ **then**
    $k_j \leftarrow 0.9/\sigma_{\text{base}}$
  else
    $u_j \leftarrow$ rank embedding of the 200-bin histogram (Algorithm 1)
    $k_j \leftarrow \text{mean}[(u_j - \bar{u}_j)^4] / \text{mean}[(u_j - \bar{u}_j)^2]^2$
  end if
  Replace undefined or infinite k_j as in Appendix E.3
  $\sigma_j \leftarrow \text{clip}_{[0.1,3.0]}\{\sigma_{\text{base}}[1 + c(k_j - 3)]\}$
end for
$D \leftarrow \text{diag}(\sigma_1, \ldots, \sigma_d)$
for each minibatch $\mathcal{B}$ over the training epochs **do**
  Draw independent $\varepsilon_{ir} \sim \mathcal{N}(0, I_d)$ for $i \in \mathcal{B}$, $r = 1, 2, 3$
  $\mathcal{L} \leftarrow (6|\mathcal{B}|)^{-1} \sum_{i \in \mathcal{B}} \sum_{r=1}^{3} \| f_\theta(x^{(i)} + D\varepsilon_{ir}) + \varepsilon_{ir} \|_2^2$
  Update θ with Adam using $\nabla_\theta \mathcal{L}$
end for
Score a preprocessed test point by $a(x) = \| f_\theta(x) \|_2$

The defaults are $\sigma_{\text{base}} = 0.5$ and $c = 0.33$. The rank embedding is used only for calibration; the denoising objective uses the original joint samples in their preprocessed coordinates.

G.6 Model architecture

We used a ResNet-like tabular model as backbone for K-DSM, DSM, DTE and MSM (Gorishniy et al., 2021). We detail the used parameters for these models in Table 5. For DTE, this did not change the performance significantly compared to the original MLP implementation. MSM was originally made of image data, so we had to adapt it with a tabular backbone.

Table 5: Hyperparameters for Kurtosis-DSM and DSM models. We use the same hyperparameters for DTE and MSM for the backbone.

Hyperparameter	**Value**
Number of blocks	6
Main layer size	512
Hidden layer size	512
Optimizer	Adam
Learning rate	0.0005
Dropout layer 1	0.2
Dropout layer 2	0.1
Batch size	128
Number of epochs	500
Base sigma	0.5
c	0.33
Sigma clipping range (K-DSM)	$[0.1, 3.0]$

G.7 Loss parameterisation

For $D = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$ and $Y = X + D\varepsilon$, the conventional score target is $-D^{-1}\varepsilon$. Its unconditional second moment in coordinate j is $1/\sigma_j^2$. In score coordinates, rescaling each residual therefore gives the weighted objective

$$\mathcal{L}_{\mathrm{weighted}}(s_\theta) = \tfrac{1}{2}\mathbb{E}\|D(s_\theta(Y) + D^{-1}\varepsilon)\|_2^2. \tag{37}$$

Defining $f_\theta = Ds_\theta$ yields the parameterization used in training:

$$\mathcal{L}_D(f_\theta) = \tfrac{1}{2}\mathbb{E}\|f_\theta(X + D\varepsilon) + \varepsilon\|_2^2. \tag{38}$$

The unrestricted population minimizer is $f_D^* = D\nabla \log q_D$. For any square-integrable predictor f, conditional expectation also gives the exact excess-risk identity

$$\mathcal{L}_D(f) - \mathcal{L}_D(f_D^*) = \tfrac{1}{2}\mathbb{E}_{Y\sim q_D}\|f(Y) - f_D^*(Y)\|_2^2. \tag{39}$$

This normalizes unconditional target variances to one; it does not imply equal parameter-gradient magnitudes, which also depend on residuals and network Jacobians. Positive coordinate weighting preserves the unrestricted score target after converting back by D^{-1}, but can affect finite-network approximation and optimization. Inference uses $\|f_\theta(x)\|_2$, the norm of the normalized field. For a shared scale it differs from $\|s_\theta(x)\|_2$ only by a constant; unequal scales can change rankings. The absence of test-time noise does not make these two norms equivalent.

H Inference and Training Time

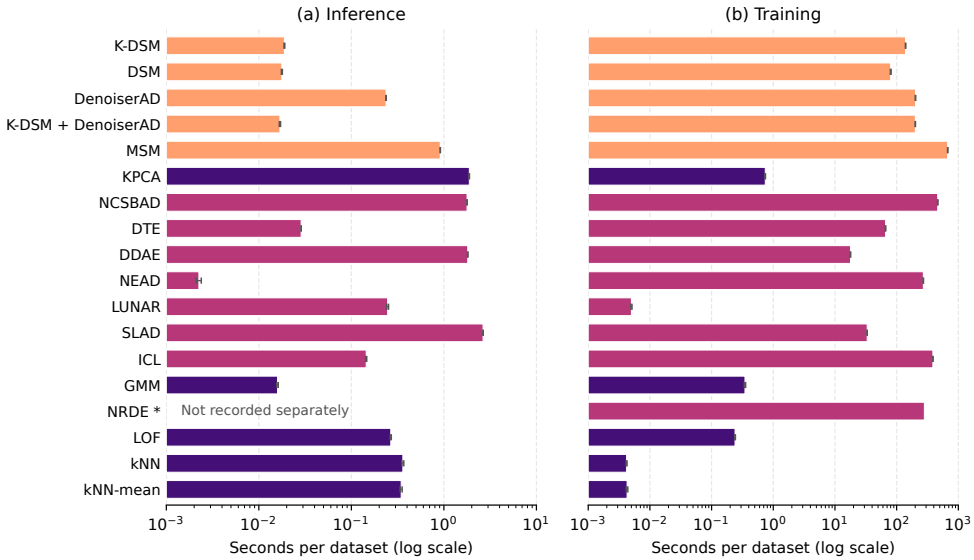


Figure 10: Mean inference and training time for all 18 configurations in Table 26, over 57 ADBench datasets with normal-only training. Task variants are averaged within each dataset, then datasets receive equal weight. Error bars are standard errors of five seed-level macro means. NRDE (*) uses the corrected run; inference time was not recorded separately. Orange denotes DSM, MSM, and DenoiserAD configurations; magenta denotes other neural models, and purple classical methods.

Inference time. DSM and K-DSM score each test point with one MLP forward pass, averaging 0.018 s for DSM and 0.019 s for K-DSM per test set under normal-only training. NEAD is faster in both measured settings ($\sim$0.002 s per test set with normal-only training). In the contaminated-training setting, DDAE takes about 7.9 s per test set, SLAD 5.3 s, and MSM 1.7 s; DDAE and MSM evaluate multiple noise levels at inference. DTE also uses one MLP pass and is of the same order as DSM. kNN and LOF have query costs that depend on the stored training set, while the EMA teacher adds work only during training and leaves K-DSM inference unchanged.

Practical importance of inference speed. For a fixed input dimension and network architecture, DSM and K-DSM have a per-query inference cost independent of training-set size. For kNN and LOF, exact neighbor search by comparison with all n stored training observations requires $O(nd)$ work per query in d dimensions, so this cost grows with the training set.

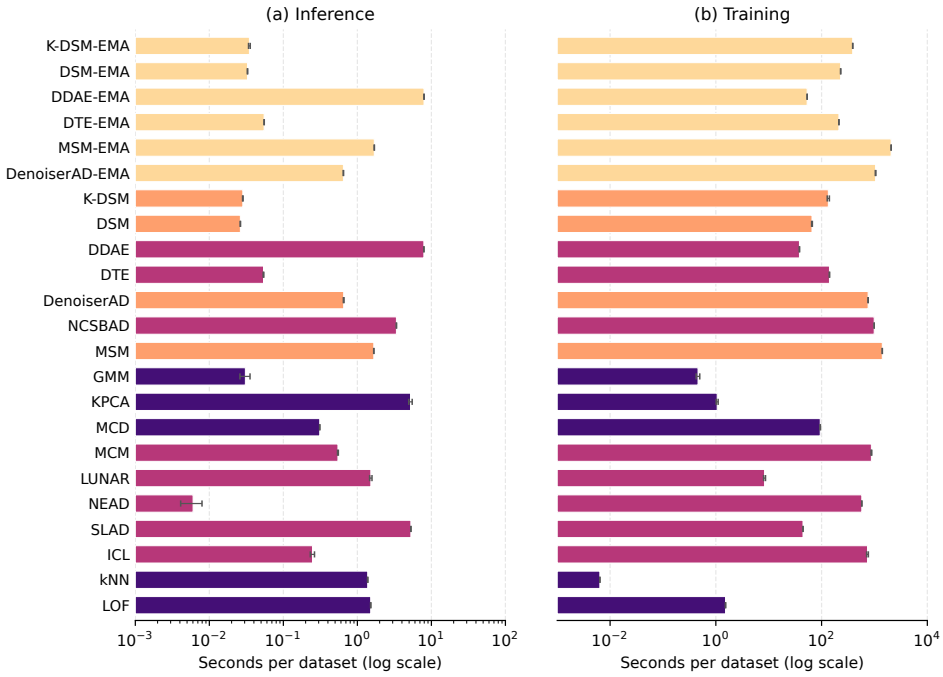


Figure 11: Mean inference and training time for all 23 configurations in Table 19, over the same 57 datasets with contaminated training. Aggregation and error bars follow Figure 10. K-DSM-EMA and DSM-EMA use the $\gamma = 80$ batch reported in the performance tables. Pale yellow denotes EMA-teacher variants; other colours follow Figure 10. Timings are from the recorded experiment runs, with different worker concurrency, rather than a dedicated throughput benchmark.

Training time. MSM and MSM-EMA are the most expensive configurations in this comparison (1,407 s and 2,070 s), since their training objectives evaluate multiple noise levels. K-DSM-EMA averages 388 s, compared with 134 s for K-DSM; DSM-EMA averages 229 s, compared with 66 s for DSM. Teacher scoring and sample filtering add work during training. DenoiserAD and DenoiserAD-EMA average 759 s and 1,051 s. These recorded times also reflect the worker concurrency of their respective runs and should not be interpreted as isolated measurements of teacher overhead. Among the classical methods, kNN takes less than 0.01 s to fit and LOF about 1.5 s. The timings cover the model's fit and scoring calls, excluding data loading and scheduler overhead; the separately logged NRDE optimization time is identified in Figure 10.

H.1 Compute Cost

Model size and computational cost. For input dimension D, the six-block, width-512 backbone has $3{,}159{,}552 + 1{,}025D$ trainable parameters. Counting a multiply-add as two operations, its linear layers and residual additions require $6{,}294{,}528 + 2{,}048D$ operations per forward pass; this count excludes normalization, activations, and bias additions. At $D = 100$, 512, and 1,024, the counts are respectively 3.26, 3.68, and 4.21 million parameters and 6.50, 7.34, and 8.39 million operations per sample (0.00650–0.00839 GFLOPs). Training evaluates three perturbed copies per original sample, multiplying the forward cost by three, with backpropagation adding further work. Inference uses one forward pass and, for fixed D, is independent of training-set size. The kurtosis computation adds no parameters and runs once before training. Measured training and inference times are reported in Figures 10 and 11.

All experiments were run on a custom cluster using NVIDIA L40S GPUs, with one GPU allocated per random seed and four CPU cores and 24 GB of RAM per worker. Since all five seeds for a given dataset run simultaneously, the wall time scales with the single-seed training time multiplied by the number of datasets.

Main ADBench experiments. We estimate GPU-hours as $\bar{t} \times 57 \times 5/3600$, where $\bar{t}$ is the mean training time (in seconds) reported across the 57 datasets. For the semi-supervised setting this yields approximately 11 GPU-hours for K-DSM, 6 for DSM, 53 for MSM, 5 for DTE, 1 for DDAE, 3 for SLAD, and 31 for ICL, for a total of roughly **110 GPU-hours**. The unsupervised setting requires running both base and EMA-teacher variants of each score-based method: the dominant costs are MSM-EMA ($\sim$164 GPU-hours), MSM ($\sim$111 GPU-hours), and ICL ($\sim$59 GPU-hours), with the remaining methods (K-DSM-EMA, DSM-EMA, DTE, DTE-EMA, K-DSM, DSM, DDAE, DDAE-EMA) contributing a further $\sim$100 GPU-hours, for a total of roughly **430 GPU-hours**. Classical methods (MCD, LOF, kNN) run on CPU and their cost is negligible by comparison.

Ablation studies. We sum the saved training times over all 121 task variants and five seeds, rather than estimating compute from dataset-group macro means. The base-scale batches total approximately 185 GPU-hours, including DSM, K-DSM, and the additional score-prediction sweep. The affine/GGD batches total 533 GPU-hours across both prediction parameterizations and all recorded settings, including settings omitted from the final figure. The filtering sweep evaluates five percentiles, $\{75, 80, 85, 90, 95\}$, for both K-DSM-EMA and DSM-EMA, totaling 229 and 134 GPU-hours, respectively; only the K-DSM-EMA sweep is reported (Appendix I.1). These recorded ablations sum to approximately **1,081 GPU-hours**; timings cover training and exclude inference and scheduler overhead.

Image benchmarks. VisA and MVTec used far fewer samples per category than the ADBench tabular datasets, so the training cost is modest. Scaling from the per-category timing data, the full method sweep across VisA (12 categories) and MVTec (15 categories) at five seeds each amounts to approximately 5.3 and 2.1 GPU-hours respectively. DINOv3 feature extraction was run once on a single L40S and took roughly 2 hours. The image benchmarks therefore account for approximately **9 GPU-hours** in total.

Development and exploration. Prior to converging on the final method design, we explored a range of approaches to the semi-supervised noise-scale problem, including entropy-based and IQR-based scaling (Appendix J), various loss formulations, and different MLP architectures. We estimate this exploratory phase at roughly **200 GPU-hours**.

Scope of the compute estimate. Combining these recorded ablation times with the preceding estimates gives approximately $110 + 430 + 1{,}081 + 9 + 200 = 1{,}830$ GPU-hours for the documented components. This is a subtotal: the additional baselines and later allocation controls are not included in these estimates.

I EMA TEACHER FOR UNSUPERVISED ANOMALY DETECTION

This appendix expands the description of the EMA-teacher filtering rule introduced in Section 4.3. The method is a training-loop companion that can be wrapped around any score- or reconstruction-based detector to improve robustness under contamination. We describe the score-based instantiation that we use for DSM and K-DSM, then summarise the analogous variants for DDAE, DTE, and MSM.

Motivation. Let the training law be $P_\eta = (1-\eta)P_0 + \eta P_1$, with P_0 normal, P_1 anomalous, and $0 < \eta < 1$. Writing $q_i = P_i * \mathcal{N}(0, D^2)$, the population noise-prediction field is

$$f^*_{\eta,D}(x) = [1 - w_\eta(x)]D\nabla \log q_0(x) + w_\eta(x)D\nabla \log q_1(x), \quad w_\eta(x) = \frac{\eta q_1(x)}{(1-\eta)q_0(x) + \eta q_1(x)}. \tag{40}$$

This follows by differentiating the mixture density. Near a contaminant mode where w_η is large, the learned field can be small, reducing the anomaly signal. However, DSM does not set the score to zero at every contaminated observation, and suppression is not universal. Filtering aims to reduce the contribution of such observations before fitting the denoising objective.

EMA teacher. We maintain a shadow copy $f_{\theta_{\text{ema}}}$ of the score network whose weights follow an exponential moving average of the online model, $\theta_{\text{ema}} \leftarrow \rho\,\theta_{\text{ema}} + (1-\rho)\,\theta$ with decay $\rho = 0.999$. For each minibatch, we score every sample with the teacher and compute its score norm $\|f_{\theta_{\text{ema}}}(x)\|$. Samples whose score norm exceeds the γ-th batch percentile are flagged as probable contaminants and dropped. The online model takes its gradient step only on the surviving $\gamma\%$ of the batch; the EMA is updated after the step. We set $\gamma = 80$, whose performance is close to the best tested value ($\gamma = 75$) in the ablation study (Appendix I.1). The teacher is intentionally slow: using the online model directly to filter its own batch creates a positive-feedback loop where whatever the model currently thinks is anomalous becomes more anomalous on the next step, while weight averaging slows changes in the selection signal. It does not by itself guarantee a correct density ranking or prevent feedback errors. This is the same decoupling principle used in Temporal Ensembling (Laine & Aila, 2017) and Mean Teacher (Tarvainen & Valpola, 2017) for semi-supervised classification, and later in momentum-encoder self-supervised representation learning such as MoCo (He et al., 2020), BYOL (Grill et al., 2020), and DINO (Caron et al., 2021); weight averaging itself dates back to Polyak–Ruppert iterate averaging (Polyak & Juditsky, 1992). Our contribution here is not the EMA mechanism but its repurposing as a *sample-selection* signal for contamination-robust density estimation: prior work uses the teacher to produce pseudo-labels or representation targets, whereas we use the teacher's score norm as a per-sample anomaly proxy and train only on the head of the batch.

Related contamination strategies. Latent Outlier Exposure (Qiu et al., 2022) jointly updates inferred anomaly labels and model parameters, while Self-supervise, Refine, Repeat (SRR) (Yoon et al., 2022) iteratively refines training data using an ensemble of one-class classifiers. These approaches also address contaminated training; we do not compare with either method here.

Variants for other detectors. The same filtering template works for the baselines, with the teacher's score replaced by a model-specific anomaly statistic:

- **DDAE-EMA**: reconstruction error $\|x_0 - \hat{x}_0^{\text{ema}}(x_t, t)\|$ at the sampled diffusion time, using larger reconstruction error as a heuristic signal of contamination.
- **DTE-EMA**: expected predicted diffusion-time bin $\sum_k k \cdot \text{softmax}(f_{\theta_{\text{ema}}}(x_t))_k$; using larger predicted diffusion times as a heuristic signal of contamination.
- **MSM-EMA**: score norm at the smallest training σ (the most density-sensitive scale in MSM's schedule).
- **DenoiserAD-EMA**: clean-input reconstruction error $\|x - r_{\theta_{\text{ema}}}(x)\|^2$ of the teacher denoiser, with DenoiserAD's training settings and native inference (15 noisy reconstructions using the online model) unchanged.

In all cases the filter cutoff is the γ-th batch percentile and the EMA decay is 0.999; Appendix I.1 reports the percentile ablation.

Algorithm 3 K-DSM training with EMA-teacher filtering

Require: contaminated loader $\mathcal{D}$, online network f_θ, per-feature noise vector $\boldsymbol{\sigma}$ from Algorithm 2, EMA decay ρ, filtering percentile $\gamma \in (0, 100)$

$\theta_{\text{ema}} \leftarrow \theta$ ▷ initialise teacher from a copy of the student
for each epoch **do**
  for each minibatch $\{x_i\}$ of size B from $\mathcal{D}$ **do**
    $n_i \leftarrow \|f_{\theta_{\text{ema}}}(x_i)\|_2$ for $i = 1, \ldots, B$ ▷ teacher anomaly proxy, no grad
    $t \leftarrow \text{quantile}(\{n_i\}, \gamma/100)$
    $\mathcal{B} \leftarrow \{\, i : n_i \le t \,\}$ ▷ drop top-$(100-\gamma)\%$ of batch
    Draw independent $\varepsilon_{ir} \sim \mathcal{N}(0, I_d)$ for $i \in \mathcal{B}$, $r = 1, 2, 3$
    $\tilde{x}_{ir} \leftarrow x_i + \boldsymbol{\sigma} \odot \varepsilon_{ir}$
    $\mathcal{L} \leftarrow \frac{1}{6|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{r=1}^{3} \|f_\theta(\tilde{x}_{ir}) + \varepsilon_{ir}\|_2^2$
    $\theta \leftarrow \theta - \lambda \nabla_\theta \mathcal{L}$
    $\theta_{\text{ema}} \leftarrow \rho\, \theta_{\text{ema}} + (1 - \rho)\, \theta$ ▷ EMA update after the step
  end for
end for

Design notes. Three design choices matter in practice. First, the teacher is updated *after* the student's step, not before, so the filter signal used in step k is based purely on state from step $k-1$ and earlier. Second, the filter acts batchwise on percentiles, not on absolute thresholds, so it is invariant to the overall scale of the score norms at any point during training. Third, if fewer than two samples survive the filter (an edge case on very small batches), we fall back to the whole batch rather than propagate an empty gradient.

Connection to contamination rate. The γ-th-percentile cutoff specifies the fraction retained, without estimating the contamination rate. With $\gamma = 80$, 20% of each batch is discarded, but this does not guarantee tolerance to 20% contamination; the outcome depends on which observations the teacher rejects. In our ADBench experiments the mean contamination across 57 datasets is 11.0%, so this is a conservative choice. On datasets with much higher contamination (e.g. SpamBase, Pima, magic.gamma, where anomaly rates exceed 30%), the fixed cutoff under-filters and the method's gains are correspondingly smaller. An adaptive cutoff tied to an estimate of the contamination rate is a natural extension.

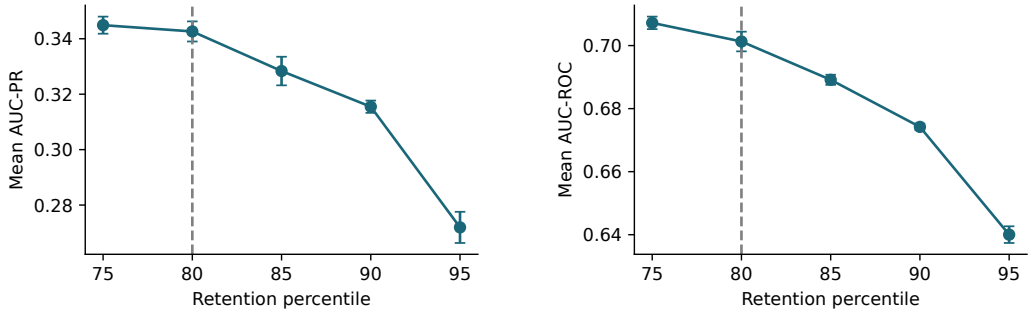
I.1 Filtering Percentile Ablation

We evaluate K-DSM-EMA with filtering percentiles $\gamma \in \{75, 80, 85, 90, 95\}$ and report mean AUC-PR and AUC-ROC across all 57 ADBench datasets (Figure 12). Performance is relatively stable over $\gamma \in [75, 85]$: $\gamma = 75$ gives the highest mean AUC-PR, with the default $\gamma = 80$ close behind. Above 85, AUC-PR decreases as more potentially contaminated observations are retained. The sweep supports the default within this tested range; the contamination-stratified analysis below examines how this choice varies across datasets.

Choosing the filtering percentile. We retain $\gamma = 80$ as the default without first estimating contamination. In the contamination-stratified analysis, the best tested percentile is either 75 or 80, and 75 is preferable in the higher-contamination strata (Appendix I.1.1). Values above 85 are less reliable on this benchmark. This is empirical guidance over the observed native anomaly rates, not a guarantee at arbitrary contamination levels. A one-seed comparison with simpler contamination strategies is reported separately in Appendix I.2.

I.1.1 Filtering sensitivity by native contamination rate

The contaminated protocol bootstraps training observations from the full dataset and evaluates the full dataset, so the native anomaly prevalence determines the expected training contamination. We stratify the 57 datasets using their benchmark anomaly rates, ranging from approximately 0.03% to 39.9%. This is an observational stratification across datasets; it does not vary contamination while holding a dataset fixed.



(a) Mean AUC-PR vs. filtering percentile γ.

(b) Mean AUC-ROC vs. filtering percentile γ.

Figure 12: Ablation over the filtering percentile γ for K-DSM-EMA on 57 ADBench datasets (unsupervised setting, 5 seeds). Shaded bands show ± 1 standard error. Performance is robust across a broad range; $\gamma = 80$ (vertical dashed line) is the default; $\gamma = 75$ gives a slightly higher mean AUC-PR in this sweep.

Table 6: K-DSM-EMA filtering-percentile sweep on all 57 datasets, five seeds; mean performance with standard error across seed-level macro means. The tested grid is $\{75, 80, 85, 90, 95\}$.

γ	AUC-PR	AUC-ROC
75	0.345 (0.003)	0.707 (0.002)
80	0.343 (0.004)	0.701 (0.003)
85	0.328 (0.005)	0.689 (0.002)
90	0.315 (0.002)	0.674 (0.001)
95	0.272 (0.006)	0.640 (0.003)

Table 7: Contamination-stratified K-DSM-EMA sensitivity, 57 datasets in total, five seeds. n counts datasets. The last two columns subtract performance at $\gamma = 75$ from performance at $\gamma = 95$; negative values indicate a cost from retaining more observations.

Anomaly rate (%)	n	Best γ (ROC)	Best γ (PR)	ΔROC	ΔPR
≤ 1	4	80	80	+0.002	-0.025
$(1, 5]$	25	75	80	-0.071	-0.070
$(5, 10]$	11	75	75	-0.086	-0.084
> 10	17	75	75	-0.066	-0.081

The best tested percentile is 75 or 80 in each stratum for both metrics. Relative to fixed $\gamma = 80$, selecting the best percentile separately for each stratum improves the overall dataset-weighted mean AUC-PR by approximately 0.0046. The differences vary across strata: 75 improves over 80 by approximately 0.013 in the 5–10% stratum, while 80 improves over 75 by approximately 0.023 in the lowest-rate stratum. A small overall gain therefore coexists with larger subgroup differences. The highest-rate stratum exceeds 20% contamination and still benefits from filtering, illustrating that performance depends on which observations the teacher excludes, rather than only on the fraction discarded.

I.2 Comparison with alternative contamination strategies

Table 8: Comparison of contamination-handling strategies on 17 datasets with seed 0. The single-seed results provide no estimate of seed uncertainty.

Strategy	AUC-ROC	AUC-PR
None	0.545	0.167
Loss trimming (retain 80%)	0.591	0.165
Mahalanobis ellipsoid (95%)	0.585	0.178
kNN density resampling	0.598	0.244
Iterative score reweighting	0.626	0.168
Adaptive per-sample σ	0.615	0.159
EMA teacher	0.715	0.335

We compare unfiltered training, retaining the lowest-loss 80% of each minibatch, a 95% Mahalanobis ellipsoid, kNN density resampling, iterative score reweighting, adaptive per-sample noise scales, and an EMA teacher. On this subset, the EMA teacher reaches 0.335 AUC-PR and 0.715 AUC-ROC. Density resampling is the strongest non-EMA alternative on AUC-PR (0.244), whereas iterative score reweighting is strongest on AUC-ROC (0.626). Loss trimming improves AUC-ROC over unfiltered training but does not improve AUC-PR. This limited comparison supports the teacher choice, while broader evaluation would be needed to establish its advantage across contamination strategies.

J ALTERNATIVE NOISE SELECTION STRATEGIES

We investigated Shannon entropy, variance of log density and the inter-quartile range (IQR) for selecting the per-feature noise scale. The performance is reported in Table 9:

Table 9: Comparison of noise-scale selection strategies on ADBench (57 datasets, five seeds, semi-supervised setting). K-DSM uses kurtosis-guided per-feature scaling. DSM uses a single global noise scale. Entropy, variance of log-density, and IQR are alternative marginal-statistic-based scaling rules described in Appendix J.

Metric	DSM	Entropy	Var (log-density)	IQR	K-DSM
AUC-ROC	0.813	0.805	0.8076	0.814	**0.823**
AUC-PR	0.585	0.597	0.5978	0.605	**0.627**
F1	0.574	0.578	0.5860	0.585	**0.608**

Shared preprocessing and postprocessing. The three alternative implementations use $\sigma_{\text{base}} = 0.5$ and remove the lowest and highest 2% of each marginal before calibration. Features whose retained nonmissing values lie in $\{-1, 0, 1\}$ receive $\sigma_j = 0.5$. For the remaining features, we apply histogram rearrangement with 100 equal-width bins and standardise the rearranged values. Only entropy estimation further restricts these values to $[-3, 3]$. The clipping bounds depend on the rule, as specified below.

Entropy-based scaling. We estimate $H_j = -\sum_i p_i \log p_i$ using natural logarithms and 40 equal-width bins over $[-3, 3]$, with probabilities normalised over the retained values. The Gaussian reference H_G uses the same estimator on 10,000 standard Gaussian draws, restricted to $[-3, 3]$ and generated with the training seed. We map entropy to noise by linear interpolation between

$$(H_G - 2.5,\ 1.3), \qquad (H_G,\ 0.5), \qquad (H_G + 1,\ 0.1),$$

and hold the endpoint values outside this interval, giving $\sigma_j \in [0.1, 1.3]$. Thus, lower entropy receives more noise. If fewer than six histogram bins are occupied, we set $H_j = H_G - 2.5$, which assigns the maximum scale 1.3; undefined entropy estimates are replaced by H_G.

IQR-based scaling. We compute the interquartile range of each standardised rearranged marginal and use the Gaussian reference value 1.3490:

$$\sigma_j = \text{clip}_{[0.1,2.0]}\left(\frac{0.5 \times 1.3490}{\max(\text{IQR}_j, 10^{-6})}\right).$$

Smaller central spread therefore gives larger noise. Undefined IQR estimates are replaced by 1.3490.

Variance of log-density scaling. For each standardised rearranged marginal Z_j, we compute the empirical variance v_j of $\log \phi(Z_j) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}Z_j^2$, using the sample count as the variance divisor. Its Gaussian reference is 0.5, giving

$$\sigma_j = \text{clip}_{[0.1,2.0]}\left(\frac{0.5 \times 0.5}{\max(v_j, 10^{-8})}\right).$$

Undefined variance estimates are replaced by 0.5. For any standardized W with a finite fourth moment,

$$\text{Var}[\log \phi(W)] = \tfrac{1}{4}\,\text{Var}(W^2) = \frac{\kappa(W) - 1}{4}, \tag{41}$$

so on the same standardized profile this variance equals $(\kappa_j - 1)/4$. Its inverse mapping therefore differs from K-DSM's increasing affine mapping; this baseline tests the noise-allocation rule rather than an independent source of distributional information.

Numerical base-scale ablation. Table 10 reports the base-scale sweep, using five ablation seeds independently of the main benchmark. Across the tested values, the AUC-PR range is 0.039 for K-DSM and 0.089 for DSM; excluding the smallest scale gives 0.018 and 0.020, respectively. Appendix J.1 examines slope sensitivity and parameter selection across disjoint sets of datasets. The base scale and slope are evaluated in separate one-dimensional sweeps.

Part of K-DSM's stability follows from the fallback rules: 25.5% of the saved feature coordinates use encoded-feature or undefined-kurtosis fallbacks, whose assigned scales respond little or not at all to σ_{base} (Appendix E.3). The sweep therefore measures the sensitivity of the full implemented calibration, including these rules.

Table 10: Base-scale sensitivity: mean AUC-PR (standard error), 57 datasets and five ablation seeds, with the backbone, loss, and evaluation protocol fixed.

σ_{base}	K-DSM AUC-PR	DSM AUC-PR
0.1	0.586 (0.002)	0.499 (0.003)
0.3	0.607 (0.003)	0.579 (0.001)
0.5	0.625 (0.002)	0.588 (0.002)
0.7	0.625 (0.001)	0.585 (0.003)
1.0	0.619 (0.002)	0.567 (0.002)

J.1 Disjoint-dataset hyperparameter selection

We assess cross-dataset selection using 200 random partitions of the 57 datasets into halves of sizes 28 and 29 (NumPy random seed 0). Within each group, we first average the five experimental seeds. For each partition, we select the parameter maximizing mean AUC-PR on one half and evaluate it on the other, then reverse the roles, yielding 400 directional selections. Regret is the difference between the best single setting on the evaluation half and the development-selected setting evaluated on that same half. The evaluation-half optimum is used only as a retrospective reference.

The base-scale grid is $\{0.1, 0.3, 0.5, 0.7, 1.0\}$. The direct-score slope experiment uses the eight τ values in Table 12, converted by the Cornish–Fisher radius map $c_{\text{CF}}(\tau) = [\Phi^{-1}(1-\tau/2)^2 - 3]/24$. The ε-prediction batch contains the same grid except $\tau = 0.07$, giving seven slope settings. The score-prediction row in Table 11 uses the same batch as Table 12; the ε-prediction row is computed separately from its own saved per-dataset results. These are separate one-dimensional experiment batches, not a joint $(\sigma_{\text{base}}, c)$ search. Selecting parameters across labeled benchmark datasets also differs from computing the feature scales without labels on a new dataset; the benchmark informed the global defaults, while no parameters are selected per test dataset.

Table 11: Disjoint-dataset selection, 57 datasets and five seeds per grid setting, with 200 random 28/29 partitions and both selection directions. Regret is in AUC-PR units; it is not a standard error.

Sweep	Random splits	Selections	Mean held-out PR regret
c, score prediction	200	400	0.0034
c, ε-prediction	200	400	0.0033
σ_{base}	200	400	0.0033

Base-scale selection has small regret across dataset partitions. Slope sensitivity depends on the parameterization. Under the deployed ε-prediction (Figure 5b), the affine sweep is flat, ranging from 0.617 to 0.625 mean AUC-PR with the setting nearest the default ($c = 0.326$) at its maximum; the score-prediction sweep in Table 12 instead ranges from 0.571 to 0.614 and peaks near $c = 0.035$. The weak slope sensitivity therefore applies to ε-prediction; the default is not optimal in every parameterization. Its mean held-out selection regret is 0.0033, compared with 0.0034 for score prediction (Table 11). Because the base-scale and slope sweeps use separate experiment batches, their absolute performance difference cannot be attributed to a single parameter change.

At $c \approx 0.035$, 69% of the saved feature coordinates receive scales in $[0.47, 0.62]$, while large scales are reserved for extreme kurtosis or undefined estimates. This setting reaches 0.617 mean AUC-PR with ε-prediction, compared with 0.625 near the default slope.

Table 12: CF-slope sensitivity with direct score prediction (the dashed affine curve in Figure 5b), on 57 datasets with five seeds per setting. Parentheses give standard errors across seed-level macro means. Coefficients follow $c_{\mathrm{CF}}(\tau) = [\Phi^{-1}(1-\tau/2)^2 - 3]/24$. The $c = 0.0118$ setting is included here but omitted from the figure. This batch is separate from Table 10.

τ	$c_{\mathrm{CF}}(\tau)$	AUC-PR (score prediction)
1e-05	0.6880	0.571 (0.004)
5e-05	0.5603	0.586 (0.002)
0.0001	0.5057	0.594 (0.002)
0.001	0.3261	0.606 (0.001)
0.005	0.2033	0.610 (0.002)
0.01	0.1515	0.611 (0.001)
0.05	0.0351	0.614 (0.001)
0.07	0.0118	0.611 (0.002)

J.2 Kurtosis heterogeneity and the gain over DSM

For each dataset, let $V_\kappa = \mathrm{Var}_j(\kappa_j)$ be the variance of the rearranged feature kurtoses. We group datasets by V_κ and subtract DSM from K-DSM using the same main-benchmark splits and five seeds. This analysis covers the 56 datasets with calibration summaries; census is excluded because its summary is unavailable. For groups containing several variants, we average the variant-level variances.

Table 13: K-DSM minus DSM performance by marginal-kurtosis heterogeneity: 56 groups with calibration summaries and five seeds for performance. n counts datasets.

Variance of feature kurtosis	n	ΔPR	ΔROC
< 0.1	4	+0.037	-0.005
$[0.1, 5)$	17	+0.028	+0.009
$[5, 100)$	16	+0.041	+0.003
≥ 100	19	+0.056	+0.021

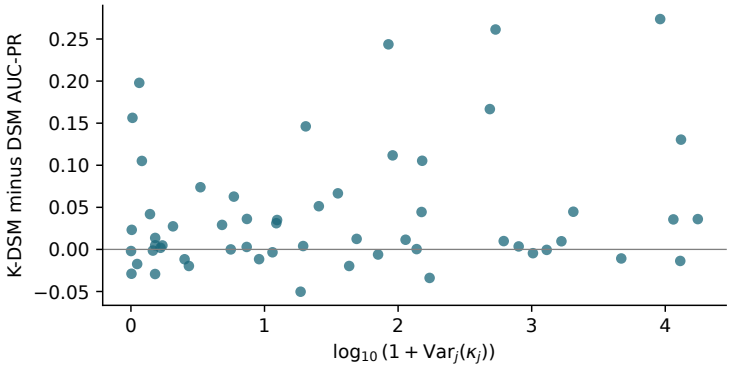


Figure 13: Dataset-level kurtosis heterogeneity and AUC-PR improvement over DSM for the 56 groups with calibration summaries. Each point is a dataset group, with performance averaged over five seeds. This is a dataset-level diagnostic, not a scatter of individual feature scales.

The average AUC-PR gain increases across the three larger strata, while the lowest stratum contains only four datasets. Across individual datasets the Spearman correlation is about 0.15, so the association is modest and descriptive. Near-identical kurtoses produce near-identical scales, but these need not equal the chosen global DSM scale unless the kurtoses are also near the Gaussian anchor.

K Noise Allocation and Inference Controls

We evaluate whether K-DSM benefits from assigning noise to particular features, beyond choosing an overall noise level, and how this benefit depends on the anomaly score. The main control changes only the noise vector within K-DSM's training procedure. Two further analyses rescore the original DSM and K-DSM checkpoints and compare clean-input scoring with noisy reconstruction scoring. All results use normal-only training on the same 121 benchmark tasks, grouped into 57 datasets, with five seeds. We average task variants within each dataset group before taking the mean across groups and seeds. AUC-PR denotes average precision throughout this section.

K.1 Matched controls for feature-wise noise allocation

Let $D = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$ contain the final K-DSM noise scales, after all calibration adjustments and clipping. We compare three training covariances: the original allocation D^2, a scalar allocation $b^2 I$, and a shuffled allocation D_π^2, where

$$b = \left(\frac{1}{d} \sum_{j=1}^{d} \sigma_j^2 \right)^{1/2}, \qquad D_\pi = \mathrm{diag}(\sigma_{\pi(1)}, \ldots, \sigma_{\pi(d)}). \tag{42}$$

The scalar control preserves the expected squared corruption magnitude, since $\mathbb{E}\|b\varepsilon\|_2^2 = \mathbb{E}\|D\varepsilon\|_2^2$. The shuffled control also preserves the complete set of scales, but randomly reassigns them to features. We draw one permutation independently for each task and seed and keep it fixed during training.

All three configurations use K-DSM's training procedure: 500 epochs, three noisy copies per observation, batch size 128, the same architecture, optimizer and learning-rate settings, and the final checkpoint. They also share initialization, minibatch order, dropout draws, and the underlying standard Gaussian noise draws. The scalar and shuffled controls replace only the calibrated noise vector; no hyperparameters are retuned. We retain all trained weights and reuse the same checkpoint for every scoring comparison within a configuration.

For any training scale matrix $A \in \{D, bI, D_\pi\}$, denote the fitted network by f_A. It is trained to predict $-\varepsilon$ from $x + A\varepsilon$, and its associated denoiser is

$$r_A(y) = y + A f_A(y). \tag{43}$$

All three configurations use the same native inference function, which evaluates $\|f_A(x)\|_2$ on a clean input. Table 14 therefore compares allocations without changing the training budget or adding a new inference operation.

Table 14: Matched allocation controls with native clean-input scoring, $\|f_A(x)\|_2$. All configurations use the K-DSM training procedure and have the same expected squared corruption magnitude. Parentheses give standard errors across the five seed-level macro means.

Training allocation	AUC-PR	AUC-ROC
Scalar, $A = bI$	0.5952 (0.0031)	0.8145 (0.0019)
Shuffled, $A = D_\pi$	0.6096 (0.0042)	0.8203 (0.0015)
Kurtosis, $A = D$	**0.6265** (0.0028)	**0.8229** (0.0021)

Kurtosis allocation improves mean AUC-PR by 0.0313 over the scalar control and by 0.0169 over the shuffled control. The scalar comparison supports using unequal scales beyond changing the total corruption strength. The shuffled comparison provides more specific evidence that assigning those scales to the corresponding features matters. Shuffled scales retain part of the improvement over scalar noise, so the gain cannot be attributed entirely to matching scales to features. In an exploratory paired bootstrap over datasets, the 95% intervals for the two native AUC-PR improvements are $[0.0114, 0.0558]$ and $[0.0046, 0.0301]$, respectively. These intervals were computed after inspecting the results and are not adjusted for multiple comparisons. The AUC-ROC difference from shuffled allocation is smaller (0.0026), with an interval spanning zero.

K.2 Output parameterization and residual weighting

DSM and K-DSM use the same clean-input inference code, but their training targets give their outputs different units. For the original scalar DSM model, let h denote the network output and $b_0 = 0.5$ its training noise scale. Its target is $-\varepsilon/b_0$, so its associated denoiser is $r_0(y) = y + b_0^2 h(y)$. K-DSM instead predicts $-\varepsilon$, giving the denoiser in Eq. (43). In particular,

$$\|f_D(x)\|_2 = \|D^{-1}(r_D(x) - x)\|_2. \tag{44}$$

Thus, the native K-DSM score measures the restoration residual in units of the training noise. This normalization is already encoded in the training target; inference does not divide the network output by D again. Multiplying the output by D gives the raw restoration residual, whereas dividing it by D gives the estimated unscaled score field (Appendix G.7). These are distinct scoring rules when feature scales differ.

Table 15: Rescoring the original DSM and K-DSM checkpoints on clean inputs. The DSM checkpoint comes from a separate run from the main benchmark, giving 0.5827 AUC-PR here versus 0.585 in Table 26. Training retains each method's original defaults. Rows within each model use identical weights. The raw DSM residual, $\|b_0^2 h(x)\|_2$, gives the same rankings and results as native DSM scoring.

Trained model	Anomaly score	AUC-PR	AUC-ROC
DSM	$\|h(x)\|_2$ (native)	0.5827	0.8094
DSM	$\|D^{-1} b_0^2 h(x)\|_2$	0.5735	0.7995
DSM	$\|Dh(x)\|_2$	0.5404	0.7983
K-DSM	$\|f_D(x)\|_2$ (native)	0.6265	0.8229
K-DSM	$\|Df_D(x)\|_2$ (raw residual)	0.5910	0.8207
K-DSM	$\|D^{-1} f_D(x)\|_2$ (unscaled score)	0.6077	0.8139

Table 15 shows that neither post-hoc weighting of the DSM residual nor weighting of its score field recovers native K-DSM performance. For K-DSM, both alternative scores reduce mean AUC-PR. Because each model is rescored with fixed weights, these comparisons isolate the effect of scoring within that model. The matched training controls below assess the contribution of noise allocation.

To isolate training differences under a common residual metric, we rescore the matched models from Table 14 using

$$a_{A,M}(x) = \|M^{-1}(r_A(x) - x)\|_2^2, \qquad M \in \{I, D\}. \tag{45}$$

Here D is always the original, unshuffled kurtosis allocation, including when scoring the scalar and shuffled models. The native score is equivalent in ranking to choosing $M = A$; consequently, $M = D$ reproduces native scoring only for the kurtosis-trained model. Squaring a norm preserves its ranking, apart from numerical tie effects.

Table 16: Matched training allocations evaluated with two fixed residual metrics. Each pair of columns holds the scoring metric constant across trained models. For shuffled allocation, the common D-weighted score differs from its native score in Table 14.

	Raw residual ($M = I$)		Common weighting ($M = D$)	
Training allocation	AUC-PR	AUC-ROC	AUC-PR	AUC-ROC
Scalar	0.5952	0.8145	0.5827	0.8006
Shuffled	0.5822	0.8104	0.5877	0.8059
Kurtosis	0.5910	0.8207	0.6265	0.8229

Under the common D-weighted metric, kurtosis training improves AUC-PR over scalar and shuffled training by 0.0439 and 0.0389, respectively (Table 16). Under the raw residual metric, the corresponding differences are -0.0042 and 0.0088, with bootstrap intervals spanning zero (Table 18). The observed advantage therefore depends on how residuals are measured. Applying the D-weighting after scalar training does not improve its mean AUC-PR, which supports using the allocation together with its training parameterization.

The matched controls also retain a coupling between corruption and loss weighting. For $y = x + A\varepsilon$, the training error satisfies

$$\|f_A(y) + \varepsilon\|_2^2 = \|A^{-1}(r_A(y) - x)\|_2^2. \tag{46}$$

Changing A therefore changes both the noisy training distribution and the equivalent weighting of reconstruction errors. The controls support the implemented allocation rule within K-DSM, but do not separate these two effects or isolate histogram rearrangement from the subsequent kurtosis calibration.

K.3 Clean-input and noisy reconstruction scoring

We next rescore the saved denoisers with common test-time perturbations. For each model, we evaluate

$$a_{A,B,M}(x) = \frac{1}{K}\sum_{k=1}^{K} \left\| M^{-1}\left[r_A(x + B\varepsilon_k) - x\right] \right\|_2^2, \qquad \varepsilon_k \sim \mathcal{N}(0, I_d). \tag{47}$$

We use $K = 15$ for $B \in \{0.2I, bI, D\}$ and one forward pass for clean scoring ($B = 0$). Models share the same standard Gaussian draws, and both B and M are held fixed within each comparison. The residual includes the injected noise: for a K-DSM model it is $B\varepsilon_k + Af_A(x + B\varepsilon_k)$. Evaluating only the network-output norm at a noisy input would give a different score.

Table 17: Mean AUC-PR under clean and common noisy-query scoring. Both D and b come from the original kurtosis allocation for every model.

Trained model	Metric M	Clean	$B = 0.2I$	$B = bI$	$B = D$
Kurtosis	I	0.5910	0.5902	0.5503	0.5802
Kurtosis	D	0.6265	0.6243	0.5210	0.6198
Scalar	I	0.5952	0.5943	0.5888	0.5688
Scalar	D	0.5827	0.5819	0.5719	0.5752
Shuffled	I	0.5822	0.5814	0.5452	0.5416
Shuffled	D	0.5877	0.5860	0.5131	0.5363

For kurtosis training, clean scoring reaches 0.6265 AUC-PR, compared with 0.6198 when query perturbations match its training allocation ($B = D$, $M = D$). Isotropic perturbations with the same total noise energy reduce AUC-PR to 0.5210. This larger drop concerns a change in the test-time covariance as well as the addition of noise. For the scalar model, adding its own training noise changes raw-residual AUC-PR from 0.5952 to 0.5888. Thus, the effect of noisy scoring depends on the covariance and cannot be summarized as a uniform advantage of clean inputs.

K.4 PAIRED ANALYSIS

We average paired differences over the five seeds within each dataset group, then use the 57 groups as the sampling units. Confidence intervals are percentile intervals from 10,000 paired group-bootstrap samples. Two-sided Wilcoxon tests receive Holm correction separately for AUC-PR and AUC-ROC. We report nine retained contrasts: four allocation comparisons under fixed residual metrics, two comparisons of residual weighting, and three clean-versus-noisy comparisons with $B = bI$ and $M = D$. The adjusted p-values retain the original ten-test correction and are therefore conservative for these nine contrasts. Bootstrap intervals are unadjusted; they need not agree with multiplicity-adjusted Wilcoxon decisions. Native-score intervals in Appendix K.1 are additional exploratory analyses.

Table 18: Paired AUC-PR differences for the four allocation comparisons under fixed residual metrics. Differences are kurtosis training minus the indicated control. Holm-adjusted p-values retain the original ten-test correction described above.

Control	Metric M	Mean difference	95% bootstrap interval	Holm p
Scalar	I	-0.0042	$[-0.0352, \ 0.0227]$	1.0000
Shuffled	I	$+0.0088$	$[-0.0181, \ 0.0331]$	0.7239
Scalar	D	$+0.0439$	$[\ 0.0190, \ 0.0759]$	0.0031
Shuffled	D	$+0.0389$	$[\ 0.0230, \ 0.0563]$	0.0005

Under common D-weighting, the kurtosis allocation improves both AUC-PR and AUC-ROC over scalar and shuffled training after correction. The corresponding AUC-ROC adjusted p-values are 0.0010 and 1.83×10^{-6}. The raw-residual comparisons do not pass the adjusted 0.05 threshold on either metric. Within the kurtosis-trained model, changing from raw to D-weighted residuals increases mean AUC-PR by 0.0355, but its adjusted p-value is 0.0717. We therefore interpret the allocation results jointly with the scoring definition, rather than treating every difference in the tables as a separate significant improvement.

L Full Unsupervised Results

We report the full per-dataset results for the unsupervised (contaminated) setting summarised in Section 4.3. Following the protocol of Livernoche et al. (2024), the training set is obtained by sampling the whole dataset with replacement (bootstrapping), so it contains anomalies at the dataset's native contamination rate; inference is performed on the full dataset. The "-EMA" columns correspond to the EMA-teacher variant of Appendix I. Best per row in bold; standard errors across seeds in parentheses. The per-dataset tables include all 23 configurations in two continued column panels. All configurations use five seeds. Each entry requires every constituent task for its reported seeds; incomplete groups are shown as dashes.

Benchmark comparison. Table 19 includes every evaluated configuration in one comparison. Across all 57 datasets, K-DSM-EMA leads mean AUC-PR and F1, while GMM leads AUC-ROC.

Table 19: Unified contaminated-training comparison. Results use five seeds. All methods are evaluated on the full benchmark of 57 datasets. Parentheses give standard errors of seed-level macro means. K-DSM-EMA and DSM-EMA use the five-seed $\gamma = 80$ ablation batch. Best in each metric is bold.

Method	AUC-PR	AUC-ROC	F1
K-DSM-EMA	**0.343 (0.004)**	0.701 (0.003)	**0.344 (0.005)**
DSM-EMA	0.321 (0.005)	0.710 (0.003)	0.308 (0.005)
DDAE-EMA	0.315 (0.004)	0.725 (0.001)	0.312 (0.004)
DTE-EMA	0.290 (0.002)	0.725 (0.003)	0.284 (0.003)
MSM-EMA	0.169 (0.001)	0.641 (0.003)	0.166 (0.001)
DenoiserAD-EMA	0.277 (0.008)	0.657 (0.003)	0.282 (0.007)
K-DSM	0.189 (0.003)	0.602 (0.002)	0.195 (0.003)
DSM	0.194 (0.002)	0.604 (0.001)	0.200 (0.003)
DDAE	0.228 (0.003)	0.685 (0.005)	0.241 (0.003)
DTE	0.281 (0.002)	0.726 (0.002)	0.279 (0.004)
DenoiserAD	0.146 (0.001)	0.534 (0.001)	0.150 (0.000)
NCSBAD	0.172 (0.002)	0.607 (0.003)	0.183 (0.001)
MSM	0.177 (0.002)	0.628 (0.005)	0.182 (0.002)
GMM	0.298 (0.002)	**0.742 (0.001)**	0.288 (0.001)
KPCA	0.205 (0.001)	0.598 (0.003)	0.220 (0.002)
MCD	0.321 (0.005)	0.730 (0.004)	0.308 (0.005)
MCM	0.285 (0.004)	0.726 (0.004)	0.280 (0.001)
LUNAR	0.193 (0.001)	0.607 (0.002)	0.204 (0.001)
NEAD	0.176 (0.001)	0.635 (0.002)	0.180 (0.002)
SLAD	0.245 (0.001)	0.673 (0.001)	0.238 (0.002)
ICL	0.240 (0.003)	0.695 (0.001)	0.245 (0.006)
kNN	0.239 (0.001)	0.665 (0.001)	0.250 (0.001)
LOF	0.175 (0.001)	0.608 (0.002)	0.184 (0.002)

DenoiserAD with contaminated training. DenoiserAD-EMA follows the filtering template of Appendix I. Across all 57 groups, filtering increases mean AUC-PR from 0.146 to 0.277, AUC-ROC from 0.534 to 0.657, and F1 from 0.150 to 0.282. These results support filtering under contamination.

Table 20: Average AUC-PR for the unsupervised (contaminated) setting on ADBench, column panel 1/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels. A dash denotes a group missing one or more constituent tasks for its reported seeds.

Dataset	K-DSM EMA	DSM EMA	DDAE EMA	DTE EMA	MSM EMA	K-DSM	DSM	DDAE	DTE	NCSBAD	DenoiserAD	DenoiserAD EMA
ALOI	0.040 (0.000)	0.041 (0.001)	0.048 (0.000)	0.033 (0.000)	0.061 (0.002)	0.040 (0.000)	0.042 (0.000)	0.051 (0.001)	0.033 (0.000)	0.068 (0.002)	0.090 (0.002)	0.061 (0.001)
annthyroid	0.192 (0.006)	0.276 (0.025)	0.219 (0.012)	**0.677 (0.007)**	0.272 (0.010)	0.130 (0.007)	0.201 (0.008)	0.225 (0.011)	0.650 (0.010)	0.266 (0.018)	0.184 (0.000)	0.403 (0.154)
backdoor	0.217 (0.096)	0.050 (0.000)	0.082 (0.002)	0.064 (0.000)	0.168 (0.025)	0.495 (0.010)	0.546 (0.010)	0.104 (0.004)	0.073 (0.004)	0.448 (0.008)	0.072 (0.001)	0.123 (0.043)
breastw	0.806 (0.014)	0.715 (0.009)	0.920 (0.003)	0.752 (0.007)	0.607 (0.029)	0.600 (0.014)	0.404 (0.009)	0.900 (0.005)	0.736 (0.012)	0.463 (0.007)	0.333 (0.003)	0.501 (0.010)
campaign	0.218 (0.008)	0.307 (0.008)	**0.393 (0.001)**	0.334 (0.009)	0.232 (0.009)	0.232 (0.009)	0.200 (0.003)	0.282 (0.006)	0.334 (0.006)	0.333 (0.003)	0.132 (0.003)	0.193 (0.012)
cardio	0.324 (0.021)	**0.518 (0.034)**	0.353 (0.014)	0.295 (0.007)	0.124 (0.003)	0.127 (0.004)	0.104 (0.003)	0.218 (0.013)	0.291 (0.016)	0.119 (0.004)	0.126 (0.000)	0.267 (0.068)
Cardiotocography	0.344 (0.018)	**0.398 (0.016)**	0.298 (0.006)	0.262 (0.007)	0.211 (0.003)	0.227 (0.005)	0.215 (0.004)	0.235 (0.007)	0.270 (0.005)	0.212 (0.004)	0.230 (0.001)	0.384 (0.007)
celeba	0.046 (0.006)	0.022 (0.000)	0.042 (0.002)	0.046 (0.005)	0.049 (0.003)	0.027 (0.001)	0.025 (0.001)	0.036 (0.001)	0.047 (0.003)	0.061 (0.002)	0.024 (0.000)	0.076 (0.002)
census	0.115 (0.004)	**0.214 (0.016)**	0.154 (0.003)	0.089 (0.003)	0.071 (0.003)	0.083 (0.001)	0.082 (0.001)	0.108 (0.001)	0.089 (0.002)	0.080 (0.001)	0.099 (0.001)	0.128 (0.002)
cover	0.332 (0.082)	0.097 (0.055)	0.028 (0.004)	0.026 (0.003)	0.034 (0.001)	0.012 (0.001)	0.012 (0.001)	0.031 (0.005)	0.033 (0.006)	0.021 (0.002)	0.009 (0.000)	**0.426 (0.048)**
donors	0.159 (0.012)	0.172 (0.029)	0.150 (0.006)	0.133 (0.017)	0.094 (0.004)	0.129 (0.001)	0.107 (0.003)	0.153 (0.008)	0.168 (0.011)	**0.219 (0.003)**	0.205 (0.001)	0.187 (0.011)
fault	0.406 (0.005)	0.400 (0.007)	0.506 (0.003)	0.425 (0.014)	0.404 (0.007)	0.459 (0.004)	0.452 (0.004)	0.502 (0.004)	0.430 (0.008)	0.410 (0.007)	0.390 (0.018)	0.404 (0.003)
fraud	**0.697 (0.031)**	0.589 (0.042)	0.648 (0.027)	0.458 (0.035)	0.023 (0.003)	0.285 (0.081)	0.134 (0.040)	0.155 (0.030)	0.633 (0.031)	0.044 (0.005)	0.015 (0.001)	0.479 (0.057)
glass	0.121 (0.029)	0.120 (0.022)	0.147 (0.013)	0.169 (0.013)	0.091 (0.010)	0.069 (0.012)	0.086 (0.015)	**0.195 (0.027)**	0.193 (0.019)	0.042 (0.004)	0.028 (0.002)	0.130 (0.006)
Hepatitis	0.299 (0.040)	0.311 (0.042)	0.228 (0.014)	0.280 (0.029)	0.201 (0.007)	0.141 (0.011)	0.152 (0.011)	0.152 (0.005)	0.223 (0.030)	0.151 (0.016)	0.138 (0.004)	0.258 (0.092)
http	0.830 (0.012)	0.415 (0.013)	0.396 (0.149)	0.382 (0.013)	0.041 (0.015)	0.071 (0.035)	0.334 (0.090)	0.008 (0.002)	0.463 (0.068)	0.157 (0.049)	0.010 (0.008)	0.233 (0.007)
InternetAds	0.542 (0.036)	**0.634 (0.003)**	0.532 (0.008)	0.240 (0.029)	0.325 (0.016)	0.392 (0.025)	0.324 (0.008)	0.317 (0.007)	0.246 (0.024)	0.337 (0.007)	0.249 (0.013)	0.283 (0.011)
Ionosphere	0.687 (0.013)	0.757 (0.011)	0.815 (0.010)	0.869 (0.011)	0.493 (0.022)	0.494 (0.008)	0.544 (0.006)	0.676 (0.012)	0.871 (0.012)	0.341 (0.010)	0.369 (0.016)	0.567 (0.019)
landsat	0.193 (0.005)	0.220 (0.006)	0.272 (0.003)	0.210 (0.006)	0.243 (0.005)	0.237 (0.002)	0.246 (0.001)	0.275 (0.006)	0.215 (0.003)	0.227 (0.002)	0.218 (0.003)	0.224 (0.005)
letter	0.210 (0.011)	0.271 (0.012)	0.316 (0.015)	0.260 (0.009)	0.087 (0.008)	0.396 (0.024)	**0.457 (0.023)**	0.299 (0.015)	0.253 (0.008)	0.177 (0.011)	0.211 (0.020)	0.214 (0.029)
Lymphography	0.663 (0.042)	**0.894 (0.014)**	0.868 (0.036)	0.752 (0.080)	0.052 (0.008)	0.045 (0.012)	0.043 (0.009)	0.132 (0.033)	0.368 (0.062)	0.141 (0.007)	0.215 (0.035)	0.543 (0.025)
magic.gamma	0.688 (0.002)	0.679 (0.005)	0.698 (0.002)	0.651 (0.011)	0.622 (0.010)	0.532 (0.002)	0.516 (0.002)	0.695 (0.004)	0.670 (0.010)	0.572 (0.003)	0.471 (0.003)	0.515 (0.003)
mammography	0.107 (0.001)	0.060 (0.004)	0.140 (0.012)	**0.190 (0.010)**	0.076 (0.007)	0.144 (0.010)	0.052 (0.001)	0.128 (0.012)	0.173 (0.008)	0.079 (0.007)	0.044 (0.006)	0.060 (0.008)
mnist	0.400 (0.008)	0.322 (0.008)	0.349 (0.003)	0.372 (0.010)	0.291 (0.008)	0.319 (0.010)	0.307 (0.006)	0.317 (0.007)	0.391 (0.016)	0.171 (0.007)	0.151 (0.008)	0.313 (0.006)
musk	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	0.775 (0.114)	0.112 (0.010)	0.292 (0.057)	0.468 (0.035)	0.323 (0.040)	0.493 (0.074)	0.088 (0.003)	0.116 (0.013)	**1.000 (0.000)**
optdigits	0.035 (0.002)	**0.037 (0.002)**	0.028 (0.000)	0.024 (0.001)	0.027 (0.001)	0.032 (0.001)	0.032 (0.000)	0.031 (0.001)	0.028 (0.002)	0.029 (0.001)	0.026 (0.001)	0.026 (0.001)
PageBlocks	**0.672 (0.015)**	0.443 (0.007)	0.555 (0.018)	0.585 (0.014)	0.470 (0.015)	0.272 (0.005)	0.299 (0.014)	0.516 (0.016)	0.538 (0.014)	0.358 (0.015)	0.164 (0.011)	0.312 (0.002)
pendigits	**0.426 (0.173)**	0.284 (0.120)	0.064 (0.003)	0.048 (0.004)	0.033 (0.003)	0.051 (0.004)	0.048 (0.003)	0.059 (0.005)	0.051 (0.006)	0.034 (0.002)	0.035 (0.004)	0.042 (0.002)
Pima	0.466 (0.024)	0.434 (0.011)	0.502 (0.009)	0.432 (0.014)	0.354 (0.009)	0.349 (0.007)	0.370 (0.008)	0.505 (0.009)	0.444 (0.005)	0.337 (0.003)	0.338 (0.008)	0.394 (0.009)
satellite	0.486 (0.009)	0.426 (0.014)	0.518 (0.007)	**0.578 (0.004)**	0.364 (0.007)	0.353 (0.004)	0.346 (0.003)	0.517 (0.007)	0.552 (0.010)	0.367 (0.003)	0.344 (0.008)	0.364 (0.000)
satimage-2	0.791 (0.157)	0.840 (0.046)	0.572 (0.083)	0.164 (0.003)	0.026 (0.003)	0.041 (0.011)	0.031 (0.007)	0.179 (0.026)	0.160 (0.012)	0.053 (0.008)	0.038 (0.007)	**0.843 (0.041)**
shuttle	**0.914 (0.002)**	0.592 (0.128)	0.091 (0.001)	0.627 (0.055)	0.211 (0.024)	0.085 (0.001)	0.116 (0.006)	0.096 (0.005)	0.685 (0.022)	0.141 (0.006)	0.058 (0.000)	0.503 (0.016)
skin	0.317 (0.020)	0.194 (0.008)	0.394 (0.007)	0.308 (0.007)	0.258 (0.004)	0.186 (0.001)	0.190 (0.001)	0.351 (0.005)	0.322 (0.005)	0.313 (0.001)	0.258 (0.002)	0.233 (0.005)
smtp	**0.657 (0.058)**	0.441 (0.058)	0.339 (0.038)	0.379 (0.028)	0.013 (0.008)	0.131 (0.052)	0.415 (0.026)	0.292 (0.066)	0.419 (0.017)	0.014 (0.012)	0.007 (0.006)	0.292 (0.112)
SpamBase	0.429 (0.002)	0.435 (0.006)	0.450 (0.004)	0.395 (0.006)	0.416 (0.004)	0.453 (0.003)	0.399 (0.002)	**0.457 (0.002)**	0.401 (0.007)	0.448 (0.003)	0.390 (0.003)	0.398 (0.007)
speech	**0.030 (0.005)**	0.025 (0.001)	0.024 (0.004)	0.018 (0.002)	0.019 (0.001)	0.023 (0.004)	0.022 (0.004)	0.020 (0.002)	0.019 (0.002)	0.019 (0.001)	0.026 (0.002)	0.026 (0.001)
Stamps	0.264 (0.020)	0.289 (0.034)	0.253 (0.021)	0.244 (0.024)	0.134 (0.015)	0.089 (0.005)	0.095 (0.006)	0.189 (0.011)	0.212 (0.018)	0.094 (0.008)	0.071 (0.000)	0.115 (0.028)
thyroid	0.336 (0.013)	0.414 (0.036)	0.421 (0.028)	**0.755 (0.007)**	0.166 (0.023)	0.068 (0.002)	0.045 (0.006)	0.189 (0.020)	0.705 (0.022)	0.099 (0.009)	0.081 (0.025)	0.508 (0.289)
vertebral	0.103 (0.005)	0.108 (0.005)	0.092 (0.006)	0.121 (0.006)	0.143 (0.010)	**0.166 (0.015)**	0.147 (0.010)	0.095 (0.004)	0.112 (0.005)	0.133 (0.006)	0.122 (0.003)	0.107 (0.000)
vowels	0.101 (0.025)	0.171 (0.011)	0.475 (0.017)	0.345 (0.071)	0.042 (0.002)	0.173 (0.010)	0.221 (0.031)	0.430 (0.020)	0.375 (0.060)	0.165 (0.022)	0.178 (0.004)	0.253 (0.059)
Waveform	0.038 (0.000)	0.040 (0.001)	0.052 (0.002)	0.040 (0.001)	0.029 (0.000)	0.036 (0.001)	0.041 (0.002)	0.050 (0.003)	0.043 (0.003)	0.035 (0.002)	0.039 (0.002)	0.040 (0.005)
WBC	0.839 (0.041)	0.843 (0.038)	0.852 (0.025)	0.235 (0.011)	0.095 (0.016)	0.044 (0.007)	0.048 (0.006)	0.202 (0.038)	0.194 (0.006)	0.057 (0.007)	0.040 (0.012)	**0.882 (0.036)**
WDBC	**0.561 (0.033)**	0.449 (0.026)	0.418 (0.036)	0.159 (0.033)	0.045 (0.009)	0.027 (0.004)	0.031 (0.005)	0.206 (0.085)	0.177 (0.034)	0.036 (0.004)	0.090 (0.021)	0.212 (0.070)
Wilt	0.041 (0.001)	0.051 (0.001)	0.042 (0.000)	0.174 (0.010)	0.106 (0.005)	0.042 (0.000)	0.067 (0.002)	0.043 (0.001)	0.148 (0.003)	**0.175 (0.029)**	0.102 (0.006)	0.128 (0.017)
wine	0.087 (0.013)	0.090 (0.009)	0.074 (0.009)	0.126 (0.029)	0.104 (0.015)	0.123 (0.012)	0.091 (0.016)	0.070 (0.011)	0.106 (0.016)	0.111 (0.014)	0.082 (0.001)	0.079 (0.001)
WPBC	0.235 (0.007)	0.239 (0.014)	0.226 (0.005)	0.228 (0.009)	0.254 (0.015)	0.255 (0.006)	0.252 (0.009)	0.247 (0.010)	0.247 (0.016)	0.235 (0.007)	**0.261 (0.029)**	0.250 (0.011)
yeast	0.329 (0.004)	0.307 (0.007)	0.295 (0.002)	0.290 (0.002)	0.322 (0.004)	0.313 (0.004)	0.333 (0.005)	0.297 (0.003)	0.301 (0.005)	0.323 (0.005)	0.317 (0.007)	0.322 (0.013)
20news	0.079 (0.004)	0.079 (0.002)	0.067 (0.002)	0.070 (0.002)	0.060 (0.001)	0.075 (0.004)	0.079 (0.003)	0.071 (0.002)	0.067 (0.002)	0.070 (0.003)	0.075 (0.004)	0.078 (0.001)
agnews	0.107 (0.000)	0.096 (0.000)	0.095 (0.001)	0.073 (0.003)	0.048 (0.003)	0.098 (0.004)	0.101 (0.002)	0.092 (0.002)	0.074 (0.002)	0.080 (0.001)	0.080 (0.003)	0.113 (0.001)
amazon	0.060 (0.001)	0.062 (0.000)	0.062 (0.001)	0.058 (0.001)	0.046 (0.003)	**0.065 (0.001)**	0.064 (0.001)	0.063 (0.001)	0.057 (0.001)	0.057 (0.001)	0.054 (0.002)	0.054 (0.002)
CIFAR10	0.095 (0.001)	0.095 (0.000)	0.100 (0.000)	0.095 (0.001)	0.068 (0.001)	0.100 (0.001)	0.088 (0.001)	0.097 (0.001)	0.093 (0.001)	0.075 (0.001)	0.084 (0.002)	0.092 (0.000)
FashionMNIST	**0.419 (0.010)**	0.362 (0.002)	0.358 (0.001)	0.315 (0.005)	0.136 (0.007)	0.315 (0.002)	0.228 (0.003)	0.292 (0.002)	0.266 (0.002)	0.159 (0.002)	0.174 (0.006)	0.325 (0.008)
imdb	0.045 (0.000)	0.046 (0.000)	0.047 (0.000)	0.048 (0.001)	0.051 (0.002)	0.046 (0.001)	0.046 (0.000)	0.047 (0.001)	0.046 (0.001)	0.050 (0.001)	0.049 (0.001)	0.047 (0.000)
MNIST-C	0.210 (0.015)	**0.226 (0.001)**	0.207 (0.002)	0.163 (0.005)	0.133 (0.004)	0.182 (0.001)	0.199 (0.001)	0.191 (0.001)	0.163 (0.001)	0.117 (0.001)	0.129 (0.001)	0.190 (0.003)
MVTec	0.554 (0.005)	0.547 (0.004)	0.509 (0.002)	0.534 (0.006)	0.354 (0.005)	0.419 (0.007)	0.386 (0.003)	0.419 (0.002)	0.524 (0.007)	0.276 (0.002)	0.351 (0.003)	0.458 (0.003)
SVHN	**0.083 (0.002)**	0.080 (0.000)	0.080 (0.000)	0.078 (0.000)	0.068 (0.000)	0.079 (0.000)	0.076 (0.000)	0.079 (0.000)	0.077 (0.001)	0.064 (0.001)	0.066 (0.001)	0.077 (0.001)
yelp	0.084 (0.001)	0.080 (0.000)	0.083 (0.001)	0.064 (0.006)	0.042 (0.002)	0.081 (0.003)	0.083 (0.001)	0.082 (0.001)	0.071 (0.004)	0.067 (0.002)	0.064 (0.001)	0.065 (0.002)
Overall	**0.343 (0.004)**	0.321 (0.005)	0.315 (0.004)	0.290 (0.002)	0.169 (0.001)	0.189 (0.003)	0.194 (0.002)	0.228 (0.003)	0.281 (0.002)	0.172 (0.002)	0.146 (0.001)	0.277 (0.008)

Table 20: Average AUC-PR for the unsupervised (contaminated) setting on ADBench, column panel 2/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels. A dash denotes a group missing one or more constituent tasks for its reported seeds.

Dataset	MSM	GMM	KPCA	MCD	MCM	LUNAR	NEAD	SLAD	ICL	*k*NN	LOF
ALOI	0.088 (0.001)	0.039 (0.000)	0.043 (0.001)	0.038 (0.000)	0.050 (0.001)	**0.099 (0.003)**	0.053 (0.001)	0.037 (0.000)	0.039 (0.001)	0.074 (0.000)	0.098 (0.001)
annthyroid	0.322 (0.013)	0.161 (0.005)	0.214 (0.005)	0.501 (0.002)	0.248 (0.039)	0.179 (0.004)	0.219 (0.004)	0.132 (0.014)	0.222 (0.011)	0.229 (0.001)	0.160 (0.004)
backdoor	0.210 (0.008)	0.541 (0.006)	0.141 (0.005)	0.129 (0.086)	0.305 (0.031)	0.132 (0.011)	0.096 (0.006)	0.043 (0.011)	**0.685 (0.028)**	0.068 (0.001)	0.069 (0.005)
breastw	0.674 (0.023)	0.933 (0.007)	0.445 (0.032)	**0.960 (0.007)**	0.909 (0.020)	0.736 (0.017)	0.565 (0.010)	0.663 (0.015)	0.830 (0.023)	0.834 (0.005)	0.292 (0.012)
campaign	0.201 (0.004)	0.312 (0.001)	0.363 (0.003)	0.311 (0.001)	0.364 (0.006)	0.191 (0.003)	0.176 (0.004)	0.243 (0.003)	0.253 (0.003)	0.335 (0.001)	0.127 (0.001)
cardio	0.125 (0.003)	0.463 (0.004)	0.177 (0.008)	0.348 (0.012)	0.366 (0.020)	0.209 (0.014)	0.118 (0.003)	0.187 (0.010)	0.124 (0.003)	0.305 (0.010)	0.165 (0.008)
Cardiotocography	0.213 (0.005)	0.347 (0.003)	0.239 (0.007)	0.309 (0.004)	0.317 (0.009)	0.261 (0.005)	0.233 (0.004)	0.223 (0.015)	0.192 (0.002)	0.277 (0.003)	0.259 (0.004)
celeba	0.041 (0.003)	0.078 (0.001)	0.047 (0.001)	**0.088 (0.002)**	0.062 (0.008)	0.027 (0.000)	0.035 (0.002)	0.056 (0.001)	0.046 (0.002)	0.033 (0.001)	0.034 (0.001)
census	0.069 (0.003)	0.085 (0.001)	0.163 (0.003)	0.060 (0.007)	0.162 (0.014)	0.082 (0.001)	0.080 (0.002)	0.103 (0.017)	0.102 (0.001)	0.129 (0.002)	0.058 (0.001)
cover	0.032 (0.003)	0.080 (0.002)	0.059 (0.001)	0.071 (0.001)	0.045 (0.012)	0.030 (0.002)	0.067 (0.005)	0.025 (0.004)	0.035 (0.012)	0.034 (0.002)	0.018 (0.001)
donors	0.127 (0.003)	0.141 (0.002)	0.168 (0.005)	0.184 (0.004)	0.171 (0.025)	0.144 (0.003)	0.088 (0.004)	0.079 (0.002)	0.108 (0.008)	0.156 (0.001)	0.107 (0.001)
fault	0.415 (0.007)	0.439 (0.004)	0.410 (0.008)	0.428 (0.005)	0.468 (0.031)	0.497 (0.006)	0.459 (0.006)	0.517 (0.006)	0.471 (0.004)	**0.519 (0.003)**	0.387 (0.005)
fraud	0.021 (0.006)	0.151 (0.017)	0.113 (0.020)	0.514 (0.030)	0.161 (0.016)	0.112 (0.017)	0.082 (0.014)	0.202 (0.030)	0.157 (0.018)	0.095 (0.009)	0.003 (0.000)
glass	0.093 (0.026)	0.134 (0.017)	0.054 (0.011)	0.116 (0.008)	0.178 (0.028)	0.107 (0.023)	0.130 (0.022)	0.147 (0.021)	0.148 (0.019)	0.163 (0.029)	0.169 (0.026)
Hepatitis	0.197 (0.022)	0.283 (0.013)	0.152 (0.010)	**0.316 (0.017)**	0.287 (0.038)	0.167 (0.003)	0.188 (0.019)	0.163 (0.006)	0.228 (0.011)	0.178 (0.008)	0.180 (0.022)
http	0.021 (0.003)	0.475 (0.011)	0.019 (0.005)	**0.861 (0.008)**	0.264 (0.076)	0.005 (0.001)	0.045 (0.007)	0.381 (0.017)	0.655 (0.008)	0.021 (0.004)	0.049 (0.009)
InternetAds	0.292 (0.005)	0.329 (0.007)	0.248 (0.005)	0.441 (0.043)	0.374 (0.016)	0.338 (0.014)	0.305 (0.009)	0.292 (0.006)	0.205 (0.010)	0.374 (0.013)	0.319 (0.013)
Ionosphere	0.522 (0.014)	0.915 (0.004)	0.415 (0.010)	**0.929 (0.005)**	0.770 (0.015)	0.555 (0.037)	0.675 (0.011)	0.808 (0.011)	0.536 (0.028)	0.715 (0.013)	0.798 (0.022)
landsat	0.242 (0.004)	0.190 (0.000)	0.232 (0.003)	0.194 (0.001)	0.182 (0.011)	0.241 (0.002)	0.219 (0.002)	0.306 (0.001)	**0.317 (0.008)**	0.250 (0.002)	0.250 (0.002)
letter	0.233 (0.013)	0.224 (0.004)	0.149 (0.009)	0.206 (0.005)	0.242 (0.036)	0.273 (0.012)	0.307 (0.016)	0.303 (0.016)	0.233 (0.012)	0.323 (0.011)	0.422 (0.011)
Lymphography	0.045 (0.010)	0.663 (0.045)	0.117 (0.039)	0.796 (0.044)	0.514 (0.079)	0.071 (0.012)	0.042 (0.007)	0.690 (0.012)	0.420 (0.053)	0.209 (0.045)	0.123 (0.042)
magic.gamma	0.637 (0.002)	0.660 (0.001)	0.667 (0.002)	0.673 (0.000)	0.617 (0.033)	0.676 (0.007)	0.585 (0.003)	0.515 (0.016)	0.572 (0.016)	**0.729 (0.001)**	0.519 (0.002)
mammography	0.100 (0.008)	0.182 (0.005)	0.179 (0.007)	0.037 (0.002)	0.137 (0.025)	0.136 (0.010)	0.105 (0.008)	0.078 (0.008)	0.067 (0.010)	0.158 (0.005)	0.085 (0.003)
mnist	0.210 (0.009)	0.375 (0.004)	0.305 (0.010)	0.379 (0.009)	**0.403 (0.012)**	0.231 (0.009)	0.219 (0.006)	0.148 (0.056)	0.255 (0.014)	0.376 (0.002)	0.226 (0.008)
musk	0.109 (0.007)	0.808 (0.015)	0.127 (0.022)	0.131 (0.022)	0.471 (0.009)	0.029 (0.004)	0.117 (0.012)	0.277 (0.032)	0.575 (0.088)	0.166 (0.017)	0.125 (0.028)
optdigits	0.034 (0.002)	0.027 (0.001)	0.028 (0.002)	0.028 (0.001)	0.029 (0.001)	0.024 (0.001)	0.025 (0.002)	0.030 (0.001)	0.035 (0.001)	0.022 (0.000)	0.034 (0.001)
PageBlocks	0.457 (0.022)	0.555 (0.010)	0.382 (0.010)	0.636 (0.004)	0.462 (0.021)	0.392 (0.018)	0.445 (0.009)	0.398 (0.011)	0.500 (0.007)	0.456 (0.006)	0.289 (0.011)
pendigits	0.035 (0.003)	0.055 (0.001)	0.092 (0.009)	0.054 (0.001)	0.105 (0.033)	0.056 (0.002)	0.047 (0.002)	0.044 (0.005)	0.098 (0.009)	0.072 (0.002)	0.045 (0.004)
Pima	0.380 (0.010)	**0.507 (0.011)**	0.392 (0.008)	0.502 (0.013)	0.462 (0.043)	0.446 (0.011)	0.432 (0.007)	0.354 (0.015)	0.420 (0.006)	0.504 (0.012)	0.404 (0.008)
satellite	0.388 (0.005)	0.515 (0.002)	0.523 (0.004)	0.529 (0.002)	0.543 (0.020)	0.448 (0.004)	0.383 (0.003)	0.533 (0.004)	0.550 (0.028)	0.518 (0.003)	0.375 (0.004)
satimage-2	0.033 (0.009)	0.368 (0.009)	0.420 (0.027)	0.372 (0.010)	0.571 (0.131)	0.199 (0.011)	0.045 (0.006)	0.339 (0.027)	0.378 (0.101)	0.330 (0.022)	0.040 (0.011)
shuttle	0.256 (0.010)	0.871 (0.002)	0.623 (0.011)	0.867 (0.003)	0.702 (0.124)	0.152 (0.001)	0.167 (0.007)	0.584 (0.118)	0.420 (0.029)	0.160 (0.001)	0.108 (0.002)
skin	0.271 (0.005)	0.286 (0.001)	0.267 (0.003)	**0.490 (0.001)**	0.190 (0.020)	0.281 (0.001)	0.288 (0.013)	0.354 (0.006)	0.215 (0.010)	0.266 (0.001)	0.221 (0.001)
smtp	0.019 (0.004)	0.277 (0.027)	0.437 (0.063)	0.006 (0.000)	0.248 (0.058)	0.026 (0.018)	0.004 (0.003)	0.425 (0.021)	0.175 (0.037)	0.210 (0.045)	0.033 (0.012)
SpamBase	0.428 (0.004)	0.397 (0.003)	0.383 (0.003)	0.399 (0.006)	0.398 (0.008)	0.387 (0.004)	0.376 (0.003)	0.384 (0.001)	0.379 (0.006)	0.397 (0.002)	0.358 (0.001)
speech	0.017 (0.001)	0.020 (0.001)	0.019 (0.001)	0.019 (0.001)	0.018 (0.000)	0.022 (0.001)	0.020 (0.002)	0.018 (0.001)	0.019 (0.001)	0.020 (0.001)	0.021 (0.001)
Stamps	0.115 (0.009)	0.301 (0.025)	0.114 (0.005)	0.265 (0.032)	**0.344 (0.029)**	0.137 (0.016)	0.129 (0.012)	0.156 (0.020)	0.141 (0.017)	0.180 (0.015)	0.152 (0.019)
thyroid	0.155 (0.008)	0.258 (0.013)	0.218 (0.018)	0.701 (0.003)	0.241 (0.069)	0.187 (0.012)	0.084 (0.005)	0.161 (0.014)	0.168 (0.027)	0.261 (0.010)	0.069 (0.007)
vertebral	0.143 (0.014)	0.100 (0.005)	0.153 (0.022)	0.106 (0.012)	0.114 (0.009)	0.130 (0.010)	0.117 (0.008)	0.102 (0.008)	0.112 (0.009)	0.112 (0.005)	0.133 (0.015)
vowels	0.160 (0.017)	0.354 (0.008)	0.243 (0.017)	0.294 (0.022)	0.448 (0.052)	0.402 (0.045)	0.240 (0.024)	0.320 (0.009)	0.239 (0.038)	**0.499 (0.031)**	0.303 (0.032)
Waveform	0.039 (0.006)	0.039 (0.000)	0.053 (0.005)	0.040 (0.000)	0.061 (0.007)	0.069 (0.006)	0.042 (0.003)	0.024 (0.000)	**0.120 (0.011)**	0.104 (0.004)	0.071 (0.004)
WBC	0.079 (0.014)	0.569 (0.055)	0.133 (0.025)	0.814 (0.044)	0.481 (0.101)	0.120 (0.016)	0.062 (0.006)	0.651 (0.078)	0.344 (0.033)	0.239 (0.046)	0.074 (0.008)
WDBC	0.042 (0.009)	0.340 (0.035)	0.071 (0.010)	0.446 (0.051)	0.424 (0.073)	0.056 (0.013)	0.047 (0.005)	0.243 (0.062)	0.087 (0.008)	0.309 (0.040)	0.084 (0.017)
Wilt	0.104 (0.003)	0.073 (0.001)	0.059 (0.001)	0.153 (0.001)	0.063 (0.006)	0.054 (0.003)	0.101 (0.003)	0.101 (0.003)	0.097 (0.004)	0.059 (0.001)	0.084 (0.002)
wine	0.106 (0.010)	0.116 (0.009)	0.062 (0.003)	**0.663 (0.124)**	0.079 (0.013)	0.082 (0.003)	0.071 (0.004)	0.067 (0.005)	0.085 (0.006)	0.087 (0.002)	0.068 (0.007)
WPBC	0.256 (0.012)	0.221 (0.007)	0.248 (0.010)	0.225 (0.006)	0.245 (0.007)	0.258 (0.014)	0.224 (0.003)	0.235 (0.003)	0.221 (0.004)	0.245 (0.011)	0.222 (0.012)
yeast	0.316 (0.003)	0.288 (0.001)	0.314 (0.005)	0.297 (0.002)	0.301 (0.004)	0.301 (0.005)	0.309 (0.003)	**0.342 (0.003)**	0.298 (0.005)	0.301 (0.005)	0.317 (0.004)
20news	0.070 (0.003)	0.074 (0.002)	0.066 (0.003)	0.065 (0.002)	0.080 (0.002)	0.069 (0.002)	0.077 (0.004)	0.064 (0.002)	0.061 (0.002)	0.074 (0.002)	**0.086 (0.002)**
agnews	0.041 (0.001)	0.105 (0.001)	0.081 (0.001)	0.055 (0.001)	0.102 (0.001)	0.082 (0.001)	0.092 (0.001)	0.063 (0.000)	0.062 (0.000)	0.089 (0.000)	**0.123 (0.001)**
amazon	0.043 (0.002)	0.063 (0.001)	0.061 (0.001)	0.053 (0.001)	0.063 (0.000)	0.059 (0.001)	0.061 (0.001)	0.051 (0.000)	0.052 (0.000)	0.063 (0.000)	0.058 (0.001)
CIFAR10	0.074 (0.001)	0.108 (0.001)	0.091 (0.002)	0.087 (0.003)	0.106 (0.000)	0.094 (0.001)	0.081 (0.002)	0.104 (0.000)	0.073 (0.001)	0.105 (0.000)	**0.115 (0.001)**
FashionMNIST	0.173 (0.003)	0.335 (0.002)	0.254 (0.003)	0.282 (0.012)	0.321 (0.002)	0.252 (0.008)	0.193 (0.003)	0.336 (0.003)	0.185 (0.002)	0.326 (0.001)	0.189 (0.001)
imdb	0.049 (0.001)	0.045 (0.000)	0.048 (0.000)	0.049 (0.001)	0.047 (0.001)	0.049 (0.000)	0.050 (0.001)	0.052 (0.000)	**0.054 (0.000)**	0.047 (0.000)	0.049 (0.000)
MNIST-C	0.136 (0.002)	0.201 (0.001)	0.162 (0.001)	0.164 (0.006)	0.171 (0.001)	0.170 (0.005)	0.135 (0.001)	0.169 (0.004)	0.110 (0.002)	0.194 (0.000)	0.126 (0.001)
MVTec	0.342 (0.005)	0.330 (0.003)	0.275 (0.001)	0.508 (0.002)	**0.568 (0.003)**	0.315 (0.005)	0.310 (0.006)	0.507 (0.009)	0.399 (0.004)	0.453 (0.004)	0.528 (0.005)
SVHN	0.073 (0.000)	**0.083 (0.000)**	0.074 (0.000)	0.068 (0.000)	0.080 (0.000)	0.077 (0.001)	0.070 (0.001)	0.079 (0.000)	0.069 (0.001)	0.081 (0.000)	0.082 (0.000)
yelp	0.039 (0.001)	0.085 (0.001)	0.082 (0.002)	0.056 (0.004)	0.086 (0.001)	0.084 (0.003)	0.068 (0.002)	0.052 (0.000)	0.051 (0.000)	**0.089 (0.001)**	0.084 (0.001)
Overall	0.177 (0.002)	0.298 (0.002)	0.205 (0.001)	0.321 (0.005)	0.285 (0.004)	0.193 (0.001)	0.176 (0.001)	0.245 (0.001)	0.240 (0.003)	0.239 (0.001)	0.175 (0.001)

Table 21: Average AUC-ROC for the unsupervised (contaminated) setting on ADBench, column panel 1/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels. A dash denotes a group missing one or more constituent tasks for its reported seeds.

Dataset	K-DSM EMA	DSM EMA	DDAE EMA	DTE EMA	MSM EMA	K-DSM	DSM	DDAE	DTE	NCSBAD	DenoiserAD	DenoiserAD EMA
ALOI	0.565 (0.001)	0.603 (0.003)	0.575 (0.001)	0.525 (0.001)	0.642 (0.002)	0.565 (0.001)	0.569 (0.001)	0.576 (0.001)	0.525 (0.001)	0.580 (0.001)	0.657 (0.000)	0.682 (0.004)
annthyroid	0.707 (0.003)	0.777 (0.008)	0.736 (0.004)	**0.959 (0.003)**	0.848 (0.009)	0.642 (0.010)	0.713 (0.006)	0.737 (0.004)	0.956 (0.006)	0.841 (0.010)	0.637 (0.007)	0.815 (0.087)
backdoor	0.732 (0.052)	0.727 (0.007)	0.813 (0.009)	0.785 (0.004)	0.832 (0.029)	0.851 (0.046)	0.924 (0.010)	0.819 (0.005)	0.802 (0.012)	**0.936 (0.006)**	0.468 (0.000)	0.789 (0.129)
breastw	0.833 (0.010)	0.775 (0.004)	0.971 (0.001)	0.913 (0.004)	0.825 (0.019)	0.807 (0.008)	0.623 (0.004)	0.966 (0.001)	0.908 (0.005)	0.620 (0.011)	0.185 (0.001)	0.371 (0.010)
campaign	0.706 (0.009)	0.670 (0.012)	0.787 (0.001)	0.786 (0.009)	0.689 (0.003)	0.720 (0.012)	0.653 (0.003)	0.762 (0.005)	0.785 (0.004)	0.752 (0.004)	0.495 (0.002)	0.561 (0.004)
cardio	0.758 (0.024)	0.847 (0.016)	0.700 (0.018)	0.740 (0.010)	0.574 (0.002)	0.510 (0.005)	0.450 (0.010)	0.626 (0.011)	0.755 (0.026)	0.488 (0.002)	0.585 (0.012)	0.676 (0.065)
Cardiotocography	0.610 (0.018)	**0.652 (0.009)**	0.469 (0.005)	0.511 (0.017)	0.470 (0.013)	0.460 (0.006)	0.438 (0.005)	0.447 (0.012)	0.525 (0.009)	0.463 (0.005)	0.537 (0.007)	0.621 (0.004)
celeba	0.706 (0.035)	0.501 (0.004)	0.683 (0.007)	0.740 (0.025)	0.729 (0.010)	0.601 (0.011)	0.563 (0.006)	0.653 (0.007)	0.751 (0.017)	0.785 (0.007)	0.534 (0.002)	0.697 (0.023)
census	0.691 (0.006)	**0.721 (0.011)**	0.714 (0.002)	0.612 (0.010)	0.577 (0.008)	0.652 (0.006)	0.640 (0.002)	0.696 (0.001)	0.586 (0.007)	0.644 (0.003)	0.671 (0.000)	0.660 (0.001)
cover	0.947 (0.019)	0.777 (0.065)	0.747 (0.032)	0.726 (0.030)	0.864 (0.006)	0.492 (0.020)	0.499 (0.025)	0.746 (0.032)	0.746 (0.036)	0.594 (0.005)	0.347 (0.019)	**0.954 (0.003)**
donors	0.791 (0.010)	0.803 (0.027)	0.824 (0.007)	0.775 (0.029)	0.721 (0.013)	0.758 (0.003)	0.752 (0.004)	0.822 (0.007)	0.831 (0.016)	**0.886 (0.002)**	0.853 (0.000)	0.841 (0.002)
fault	0.555 (0.006)	0.553 (0.009)	**0.705 (0.002)**	0.590 (0.015)	0.572 (0.008)	0.638 (0.006)	0.630 (0.004)	0.698 (0.003)	0.592 (0.014)	0.558 (0.006)	0.482 (0.016)	0.513 (0.007)
fraud	0.950 (0.006)	**0.961 (0.007)**	0.958 (0.007)	0.938 (0.010)	0.926 (0.007)	0.918 (0.008)	0.910 (0.011)	0.951 (0.006)	0.947 (0.007)	0.931 (0.006)	0.762 (0.004)	0.941 (0.026)
glass	0.480 (0.056)	0.655 (0.039)	0.872 (0.009)	0.861 (0.012)	0.581 (0.029)	0.565 (0.055)	0.606 (0.029)	**0.903 (0.014)**	0.879 (0.014)	0.421 (0.018)	0.208 (0.007)	0.560 (0.006)
Hepatitis	0.616 (0.043)	0.664 (0.047)	0.555 (0.021)	0.641 (0.038)	0.532 (0.026)	0.357 (0.038)	0.378 (0.028)	0.469 (0.012)	0.580 (0.028)	0.471 (0.049)	0.431 (0.023)	0.600 (0.126)
http	**0.999 (0.000)**	0.995 (0.000)	0.996 (0.001)	0.994 (0.000)	0.737 (0.097)	0.887 (0.039)	0.987 (0.004)	0.379 (0.146)	0.995 (0.001)	0.970 (0.011)	0.024 (0.003)	0.994 (0.000)
InternetAds	0.752 (0.018)	**0.795 (0.002)**	0.745 (0.004)	0.538 (0.034)	0.686 (0.012)	0.709 (0.015)	0.664 (0.005)	0.659 (0.005)	0.540 (0.022)	0.677 (0.005)	0.523 (0.013)	0.640 (0.005)
Ionosphere	0.670 (0.020)	0.779 (0.017)	0.868 (0.003)	0.906 (0.007)	0.690 (0.019)	0.620 (0.005)	0.724 (0.004)	0.812 (0.006)	0.909 (0.008)	0.400 (0.013)	0.469 (0.010)	0.591 (0.014)
landsat	0.463 (0.010)	0.518 (0.009)	0.625 (0.004)	0.520 (0.011)	0.552 (0.008)	0.547 (0.001)	0.559 (0.002)	0.622 (0.007)	0.522 (0.005)	0.507 (0.004)	0.491 (0.006)	0.504 (0.005)
letter	0.731 (0.011)	0.769 (0.006)	0.834 (0.007)	0.792 (0.004)	0.633 (0.011)	0.854 (0.012)	**0.896 (0.006)**	0.826 (0.010)	0.784 (0.009)	0.488 (0.014)	0.600 (0.006)	0.665 (0.021)
Lymphography	0.880 (0.049)	**0.996 (0.001)**	0.995 (0.001)	0.943 (0.032)	0.339 (0.042)	0.135 (0.021)	0.194 (0.034)	0.514 (0.029)	0.789 (0.064)	0.877 (0.007)	0.918 (0.018)	0.962 (0.001)
magic.gamma	0.752 (0.002)	0.755 (0.004)	0.768 (0.002)	0.759 (0.005)	0.745 (0.006)	0.679 (0.003)	0.666 (0.003)	0.769 (0.005)	0.773 (0.009)	0.675 (0.002)	0.485 (0.001)	0.545 (0.000)
mammography	0.838 (0.001)	0.738 (0.018)	0.807 (0.014)	0.834 (0.007)	0.784 (0.011)	0.838 (0.003)	0.695 (0.012)	0.820 (0.009)	0.808 (0.012)	0.791 (0.005)	0.666 (0.006)	0.688 (0.009)
mnist	0.756 (0.006)	0.693 (0.009)	0.779 (0.004)	0.823 (0.011)	0.682 (0.008)	0.801 (0.006)	0.745 (0.006)	0.775 (0.003)	0.843 (0.011)	0.596 (0.007)	0.599 (0.010)	0.695 (0.010)
musk	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	0.975 (0.016)	0.876 (0.014)	0.859 (0.039)	0.919 (0.015)	0.916 (0.011)	0.969 (0.007)	0.720 (0.006)	0.858 (0.035)	**1.000 (0.000)**
optdigits	0.530 (0.016)	0.577 (0.015)	0.497 (0.005)	0.443 (0.019)	0.519 (0.008)	0.538 (0.005)	0.533 (0.003)	0.522 (0.016)	0.512 (0.031)	0.481 (0.011)	0.446 (0.012)	0.471 (0.012)
PageBlocks	**0.940 (0.006)**	0.865 (0.007)	0.918 (0.006)	0.911 (0.007)	0.913 (0.006)	0.663 (0.003)	0.711 (0.012)	0.901 (0.006)	0.915 (0.003)	0.807 (0.015)	0.445 (0.013)	0.603 (0.024)
pendigits	0.842 (0.072)	0.827 (0.081)	0.711 (0.015)	0.756 (0.020)	0.579 (0.021)	0.536 (0.017)	0.541 (0.019)	0.706 (0.031)	0.759 (0.034)	0.529 (0.008)	0.450 (0.035)	0.497 (0.026)
Pima	0.588 (0.020)	0.569 (0.007)	**0.691 (0.009)**	0.607 (0.015)	0.512 (0.011)	0.465 (0.015)	0.509 (0.016)	0.685 (0.013)	0.626 (0.009)	0.421 (0.012)	0.386 (0.015)	0.452 (0.031)
satellite	0.582 (0.007)	0.560 (0.009)	0.715 (0.006)	0.729 (0.006)	0.586 (0.008)	0.545 (0.004)	0.528 (0.004)	0.706 (0.006)	0.706 (0.010)	0.490 (0.002)	0.494 (0.011)	0.514 (0.003)
satimage-2	0.934 (0.065)	**0.998 (0.001)**	0.983 (0.005)	0.959 (0.001)	0.701 (0.012)	0.498 (0.024)	0.470 (0.006)	0.884 (0.026)	0.952 (0.006)	0.474 (0.013)	0.658 (0.018)	0.986 (0.000)
shuttle	**0.994 (0.000)**	0.831 (0.073)	0.373 (0.004)	0.975 (0.008)	0.742 (0.057)	0.333 (0.014)	0.438 (0.029)	0.399 (0.033)	0.984 (0.002)	0.636 (0.008)	0.251 (0.008)	0.666 (0.003)
skin	0.684 (0.023)	0.495 (0.020)	0.831 (0.006)	0.748 (0.008)	0.672 (0.005)	0.497 (0.004)	0.513 (0.003)	0.793 (0.006)	0.764 (0.006)	0.743 (0.001)	0.646 (0.001)	0.611 (0.006)
smtp	0.841 (0.025)	0.921 (0.011)	0.922 (0.013)	**0.953 (0.007)**	0.930 (0.017)	0.803 (0.007)	0.933 (0.007)	0.921 (0.007)	0.952 (0.007)	0.824 (0.013)	0.748 (0.032)	0.932 (0.011)
SpamBase	0.544 (0.002)	0.528 (0.006)	0.593 (0.005)	0.517 (0.007)	0.531 (0.005)	0.599 (0.004)	0.520 (0.004)	**0.611 (0.003)**	0.519 (0.009)	0.566 (0.002)	0.456 (0.001)	0.477 (0.007)
speech	0.564 (0.012)	**0.567 (0.011)**	0.495 (0.010)	0.475 (0.012)	0.520 (0.015)	0.467 (0.015)	0.474 (0.017)	0.494 (0.009)	0.490 (0.005)	0.504 (0.013)	0.545 (0.009)	0.544 (0.010)
Stamps	0.802 (0.035)	0.812 (0.047)	0.822 (0.011)	0.759 (0.015)	0.573 (0.040)	0.388 (0.020)	0.419 (0.030)	0.734 (0.019)	0.704 (0.011)	0.415 (0.013)	0.374 (0.007)	0.442 (0.061)
thyroid	0.963 (0.002)	0.961 (0.004)	0.966 (0.003)	0.988 (0.001)	0.903 (0.006)	0.620 (0.004)	0.536 (0.020)	0.928 (0.007)	**0.989 (0.001)**	0.845 (0.010)	0.393 (0.032)	0.793 (0.195)
vertebral	0.411 (0.021)	0.445 (0.010)	0.383 (0.011)	0.507 (0.018)	0.539 (0.035)	**0.620 (0.012)**	0.576 (0.005)	0.400 (0.013)	0.452 (0.016)	0.547 (0.013)	0.557 (0.035)	0.518 (0.044)
vowels	0.514 (0.026)	0.712 (0.039)	**0.955 (0.005)**	0.884 (0.020)	0.593 (0.006)	0.707 (0.028)	0.808 (0.035)	0.935 (0.007)	0.913 (0.023)	0.465 (0.012)	0.421 (0.044)	0.606 (0.127)
Waveform	0.553 (0.011)	0.565 (0.012)	0.656 (0.006)	0.591 (0.006)	0.510 (0.005)	0.563 (0.004)	0.604 (0.005)	0.656 (0.008)	0.602 (0.014)	0.514 (0.007)	0.560 (0.011)	0.543 (0.001)
WBC	0.970 (0.017)	0.985 (0.005)	**0.989 (0.001)**	0.867 (0.007)	0.616 (0.028)	0.299 (0.018)	0.226 (0.030)	0.815 (0.026)	0.846 (0.006)	0.564 (0.036)	0.357 (0.061)	0.958 (0.033)
WDBC	**0.986 (0.001)**	0.981 (0.003)	0.979 (0.002)	0.820 (0.086)	0.546 (0.096)	0.253 (0.036)	0.286 (0.057)	0.795 (0.084)	0.857 (0.026)	0.553 (0.050)	0.856 (0.051)	0.913 (0.057)
Wilt	0.411 (0.008)	0.523 (0.007)	0.427 (0.005)	0.850 (0.011)	0.766 (0.013)	0.412 (0.005)	0.592 (0.010)	0.431 (0.012)	0.834 (0.006)	0.783 (0.025)	0.715 (0.012)	0.772 (0.025)
wine	0.498 (0.060)	0.514 (0.029)	0.458 (0.050)	0.584 (0.118)	0.500 (0.073)	0.594 (0.033)	0.403 (0.014)	0.336 (0.016)	0.557 (0.065)	0.593 (0.038)	0.510 (0.039)	0.505 (0.030)
WPBC	0.491 (0.018)	0.505 (0.017)	0.488 (0.012)	0.468 (0.014)	0.503 (0.018)	0.527 (0.009)	0.517 (0.014)	0.519 (0.016)	0.494 (0.029)	0.497 (0.015)	0.529 (0.035)	0.512 (0.019)
yeast	0.452 (0.010)	0.433 (0.012)	0.405 (0.004)	0.391 (0.003)	0.459 (0.007)	0.459 (0.007)	0.465 (0.007)	0.408 (0.006)	0.412 (0.010)	0.477 (0.006)	0.485 (0.009)	0.479 (0.019)
20news	0.585 (0.006)	0.603 (0.002)	0.560 (0.007)	0.580 (0.008)	0.539 (0.005)	0.546 (0.008)	0.580 (0.006)	0.559 (0.008)	0.574 (0.007)	0.515 (0.005)	0.552 (0.013)	0.575 (0.008)
agnews	0.702 (0.001)	0.701 (0.001)	0.678 (0.002)	0.616 (0.015)	0.464 (0.011)	0.683 (0.007)	0.703 (0.001)	0.680 (0.003)	0.625 (0.007)	0.560 (0.004)	0.553 (0.005)	0.641 (0.004)
amazon	0.576 (0.006)	0.599 (0.001)	0.596 (0.007)	0.567 (0.002)	0.458 (0.020)	0.601 (0.005)	0.601 (0.004)	0.607 (0.004)	0.557 (0.008)	0.532 (0.010)	0.514 (0.013)	0.514 (0.016)
CIFAR10	0.585 (0.004)	0.610 (0.003)	0.644 (0.001)	0.643 (0.003)	0.588 (0.004)	0.662 (0.000)	0.640 (0.002)	0.649 (0.001)	0.639 (0.003)	0.508 (0.002)	0.575 (0.003)	0.598 (0.000)
FashionMNIST	0.828 (0.006)	0.839 (0.003)	0.865 (0.001)	0.845 (0.002)	0.794 (0.002)	0.865 (0.000)	0.798 (0.006)	0.858 (0.001)	0.843 (0.003)	0.565 (0.004)	0.686 (0.001)	0.806 (0.004)
imdb	0.480 (0.002)	0.485 (0.001)	0.492 (0.001)	0.485 (0.011)	0.507 (0.008)	0.482 (0.004)	0.484 (0.002)	0.489 (0.006)	0.485 (0.010)	0.514 (0.004)	0.495 (0.001)	0.484 (0.001)
MNIST-C	0.727 (0.007)	0.752 (0.002)	0.782 (0.002)	0.743 (0.003)	0.747 (0.003)	0.772 (0.001)	0.760 (0.004)	0.787 (0.002)	0.751 (0.001)	0.532 (0.002)	0.628 (0.001)	0.713 (0.004)
MVTec	0.667 (0.006)	0.712 (0.006)	0.678 (0.002)	0.724 (0.006)	0.613 (0.006)	0.624 (0.007)	0.585 (0.004)	0.624 (0.002)	0.735 (0.005)	0.438 (0.005)	0.611 (0.001)	0.648 (0.002)
SVHN	0.589 (0.004)	0.596 (0.001)	0.609 (0.001)	0.605 (0.002)	0.586 (0.003)	0.606 (0.000)	0.617 (0.001)	0.613 (0.001)	0.601 (0.003)	0.514 (0.002)	0.544 (0.006)	0.575 (0.007)
yelp	0.651 (0.004)	0.661 (0.002)	0.653 (0.004)	0.573 (0.023)	0.428 (0.022)	0.643 (0.003)	0.653 (0.004)	0.650 (0.003)	0.612 (0.010)	0.541 (0.006)	0.544 (0.001)	0.555 (0.002)
Overall	0.701 (0.003)	0.710 (0.003)	0.725 (0.001)	0.725 (0.003)	0.641 (0.003)	0.602 (0.002)	0.604 (0.001)	0.685 (0.005)	0.726 (0.002)	0.607 (0.003)	0.534 (0.001)	0.657 (0.003)

Table 21: Average AUC-ROC for the unsupervised (contaminated) setting on ADBench, column panel 2/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels. A dash denotes a group missing one or more constituent tasks for its reported seeds.

Dataset	MSM	GMM	KPCA	MCD	MCM	LUNAR	NEAD	SLAD	ICL	kNN	LOF
ALOI	0.642 (0.003)	0.559 (0.000)	0.580 (0.002)	0.553 (0.000)	0.587 (0.002)	0.693 (0.007)	0.592 (0.002)	0.550 (0.001)	0.517 (0.003)	0.703 (0.001)	**0.768 (0.001)**
annthyroid	0.884 (0.006)	0.644 (0.002)	0.676 (0.006)	0.917 (0.001)	0.755 (0.026)	0.708 (0.006)	0.803 (0.010)	0.579 (0.029)	0.750 (0.022)	0.802 (0.002)	0.704 (0.001)
backdoor	0.821 (0.017)	0.904 (0.003)	0.819 (0.011)	0.484 (0.120)	0.864 (0.004)	0.604 (0.010)	0.824 (0.014)	0.620 (0.074)	0.913 (0.005)	0.709 (0.005)	0.683 (0.014)
breastw	0.846 (0.015)	0.971 (0.003)	0.503 (0.063)	**0.985 (0.002)**	0.969 (0.006)	0.810 (0.015)	0.801 (0.006)	0.806 (0.008)	0.941 (0.010)	0.875 (0.007)	0.449 (0.013)
campaign	0.641 (0.003)	0.766 (0.000)	0.786 (0.002)	0.762 (0.000)	**0.828 (0.006)**	0.623 (0.002)	0.629 (0.004)	0.724 (0.003)	0.731 (0.003)	0.787 (0.001)	0.549 (0.002)
cardio	0.571 (0.016)	**0.892 (0.004)**	0.555 (0.020)	0.813 (0.009)	0.771 (0.034)	0.597 (0.012)	0.517 (0.011)	0.499 (0.018)	0.495 (0.013)	0.670 (0.012)	0.546 (0.013)
Cardiotocography	0.470 (0.009)	0.593 (0.002)	0.496 (0.015)	0.498 (0.002)	0.514 (0.009)	0.502 (0.005)	0.476 (0.004)	0.383 (0.018)	0.374 (0.005)	0.474 (0.006)	0.530 (0.007)
celeba	0.704 (0.014)	0.794 (0.002)	0.690 (0.005)	**0.820 (0.002)**	0.746 (0.018)	0.565 (0.004)	0.663 (0.015)	0.775 (0.004)	0.728 (0.012)	0.589 (0.005)	0.590 (0.006)
census	0.578 (0.016)	0.659 (0.001)	0.691 (0.002)	0.507 (0.045)	**0.721 (0.009)**	0.643 (0.002)	0.610 (0.009)	0.600 (0.041)	0.651 (0.002)	0.707 (0.002)	0.515 (0.003)
cover	0.801 (0.013)	0.931 (0.002)	0.866 (0.005)	0.923 (0.002)	0.817 (0.038)	0.690 (0.009)	0.819 (0.026)	0.742 (0.023)	0.783 (0.041)	0.751 (0.013)	0.569 (0.008)
donors	0.799 (0.005)	0.799 (0.004)	0.812 (0.011)	0.863 (0.004)	0.841 (0.021)	0.735 (0.003)	0.647 (0.020)	0.658 (0.009)	0.695 (0.023)	0.732 (0.003)	0.629 (0.006)
fault	0.580 (0.008)	0.619 (0.004)	0.520 (0.009)	0.604 (0.006)	0.652 (0.036)	0.672 (0.005)	0.645 (0.005)	0.704 (0.004)	0.657 (0.004)	0.703 (0.002)	0.579 (0.006)
fraud	0.801 (0.010)	0.953 (0.003)	0.921 (0.008)	0.957 (0.003)	0.951 (0.003)	0.919 (0.014)	0.923 (0.006)	0.942 (0.007)	0.951 (0.004)	0.946 (0.006)	0.550 (0.033)
glass	0.550 (0.053)	0.768 (0.005)	0.472 (0.077)	0.799 (0.005)	0.741 (0.061)	0.596 (0.049)	0.737 (0.031)	0.746 (0.013)	0.838 (0.018)	0.629 (0.046)	0.659 (0.036)
Hepatitis	0.466 (0.028)	**0.690 (0.010)**	0.409 (0.028)	0.644 (0.037)	0.687 (0.027)	0.462 (0.014)	0.436 (0.027)	0.455 (0.016)	0.642 (0.020)	0.475 (0.020)	0.431 (0.046)
http	0.400 (0.159)	0.996 (0.000)	0.045 (0.007)	**0.999 (0.000)**	0.972 (0.017)	0.103 (0.007)	0.914 (0.012)	0.994 (0.000)	0.998 (0.000)	0.096 (0.007)	0.337 (0.005)
InternetAds	0.634 (0.011)	0.706 (0.005)	0.515 (0.006)	0.714 (0.029)	0.643 (0.008)	0.637 (0.008)	0.634 (0.007)	0.609 (0.006)	0.568 (0.020)	0.681 (0.007)	0.635 (0.008)
Ionosphere	0.661 (0.018)	0.924 (0.006)	0.496 (0.016)	**0.939 (0.006)**	0.825 (0.006)	0.564 (0.024)	0.810 (0.007)	0.885 (0.005)	0.682 (0.033)	0.656 (0.017)	0.866 (0.009)
landsat	0.556 (0.006)	0.464 (0.001)	0.543 (0.005)	0.477 (0.002)	0.423 (0.035)	0.552 (0.005)	0.504 (0.003)	**0.674 (0.002)**	0.617 (0.005)	0.579 (0.003)	0.544 (0.007)
letter	0.645 (0.015)	0.800 (0.001)	0.544 (0.013)	0.797 (0.003)	0.782 (0.053)	0.766 (0.005)	0.828 (0.009)	0.863 (0.002)	0.786 (0.006)	0.881 (0.005)	0.876 (0.002)
Lymphography	0.353 (0.066)	0.984 (0.003)	0.558 (0.058)	0.989 (0.003)	0.897 (0.055)	0.530 (0.043)	0.292 (0.021)	0.969 (0.003)	0.948 (0.007)	0.544 (0.063)	0.618 (0.077)
magic.gamma	0.760 (0.002)	0.736 (0.001)	0.717 (0.002)	0.746 (0.000)	0.723 (0.019)	0.736 (0.006)	0.710 (0.002)	0.640 (0.008)	0.687 (0.010)	**0.797 (0.001)**	0.679 (0.002)
mammography	0.811 (0.010)	**0.861 (0.001)**	0.782 (0.008)	0.696 (0.008)	0.824 (0.021)	0.809 (0.006)	0.813 (0.010)	0.722 (0.008)	0.705 (0.016)	0.829 (0.002)	0.703 (0.009)
mnist	0.657 (0.015)	0.843 (0.002)	0.695 (0.011)	0.846 (0.005)	**0.866 (0.009)**	0.658 (0.010)	0.633 (0.004)	0.554 (0.054)	0.681 (0.020)	0.824 (0.003)	0.658 (0.009)
musk	0.793 (0.011)	0.990 (0.003)	0.528 (0.016)	0.615 (0.043)	0.895 (0.010)	0.333 (0.014)	0.601 (0.069)	0.836 (0.020)	0.979 (0.006)	0.597 (0.031)	0.558 (0.029)
optdigits	0.527 (0.009)	0.516 (0.013)	0.492 (0.021)	0.529 (0.021)	0.546 (0.024)	0.432 (0.016)	0.460 (0.039)	0.557 (0.017)	**0.612 (0.014)**	0.399 (0.012)	0.529 (0.016)
PageBlocks	0.910 (0.008)	0.919 (0.002)	0.654 (0.013)	0.930 (0.001)	0.859 (0.013)	0.746 (0.012)	0.878 (0.009)	0.735 (0.012)	0.891 (0.008)	0.817 (0.006)	0.703 (0.007)
pendigits	0.564 (0.015)	0.762 (0.005)	0.686 (0.019)	0.761 (0.006)	0.812 (0.044)	0.636 (0.018)	0.590 (0.011)	0.664 (0.020)	**0.847 (0.008)**	0.716 (0.012)	0.546 (0.018)
Pima	0.529 (0.019)	0.684 (0.009)	0.528 (0.014)	0.690 (0.008)	0.596 (0.051)	0.560 (0.019)	0.599 (0.014)	0.474 (0.020)	0.572 (0.019)	0.636 (0.017)	0.561 (0.016)
satellite	0.597 (0.007)	0.637 (0.001)	0.608 (0.004)	0.648 (0.002)	0.630 (0.011)	0.600 (0.004)	0.550 (0.001)	**0.751 (0.004)**	0.705 (0.016)	0.662 (0.002)	0.541 (0.006)
satimage-2	0.678 (0.011)	0.984 (0.001)	0.767 (0.019)	0.984 (0.000)	0.970 (0.017)	0.787 (0.015)	0.648 (0.013)	0.965 (0.007)	0.976 (0.011)	0.919 (0.011)	0.529 (0.036)
shuttle	0.845 (0.011)	0.983 (0.000)	0.923 (0.003)	0.983 (0.000)	0.959 (0.022)	0.604 (0.001)	0.694 (0.022)	0.928 (0.053)	0.942 (0.009)	0.630 (0.002)	0.524 (0.005)
skin	0.698 (0.008)	0.713 (0.002)	0.689 (0.005)	**0.891 (0.001)**	0.468 (0.055)	0.675 (0.001)	0.702 (0.025)	0.750 (0.003)	0.550 (0.024)	0.672 (0.001)	0.549 (0.001)
smtp	0.936 (0.010)	0.817 (0.026)	0.904 (0.022)	0.951 (0.005)	0.838 (0.029)	0.800 (0.041)	0.787 (0.011)	0.919 (0.006)	0.893 (0.018)	0.894 (0.038)	0.891 (0.022)
SpamBase	0.531 (0.005)	0.528 (0.003)	0.483 (0.003)	0.531 (0.007)	0.532 (0.013)	0.471 (0.009)	0.488 (0.007)	0.491 (0.002)	0.506 (0.010)	0.494 (0.005)	0.451 (0.002)
speech	0.486 (0.016)	0.487 (0.007)	0.496 (0.016)	0.476 (0.005)	0.471 (0.001)	0.527 (0.010)	0.489 (0.011)	0.512 (0.011)	0.516 (0.015)	0.515 (0.005)	0.509 (0.004)
Stamps	0.504 (0.030)	0.853 (0.011)	0.465 (0.041)	0.833 (0.014)	**0.888 (0.015)**	0.515 (0.035)	0.538 (0.022)	0.505 (0.020)	0.558 (0.049)	0.567 (0.030)	0.482 (0.034)
thyroid	0.917 (0.005)	0.933 (0.002)	0.612 (0.030)	0.986 (0.000)	0.894 (0.035)	0.879 (0.013)	0.820 (0.014)	0.777 (0.010)	0.898 (0.022)	0.949 (0.006)	0.651 (0.012)
vertebral	0.542 (0.036)	0.412 (0.006)	0.536 (0.044)	0.412 (0.020)	0.470 (0.050)	0.498 (0.024)	0.483 (0.014)	0.399 (0.013)	0.463 (0.025)	0.483 (0.033)	0.503 (0.028)
vowels	0.711 (0.009)	0.910 (0.002)	0.579 (0.022)	0.906 (0.005)	0.946 (0.010)	0.814 (0.033)	0.827 (0.019)	0.931 (0.002)	0.879 (0.022)	0.943 (0.016)	0.926 (0.007)
Waveform	0.505 (0.013)	0.571 (0.001)	0.551 (0.017)	0.574 (0.002)	0.673 (0.003)	0.591 (0.016)	0.618 (0.008)	0.438 (0.004)	**0.728 (0.015)**	0.716 (0.003)	0.694 (0.007)
WBC	0.539 (0.068)	0.969 (0.004)	0.605 (0.036)	0.984 (0.004)	0.893 (0.055)	0.486 (0.021)	0.571 (0.013)	0.954 (0.017)	0.935 (0.007)	0.525 (0.042)	0.577 (0.049)
WDBC	0.519 (0.098)	0.955 (0.006)	0.556 (0.069)	0.976 (0.005)	0.961 (0.017)	0.520 (0.044)	0.527 (0.023)	0.817 (0.059)	0.814 (0.028)	0.809 (0.046)	0.739 (0.090)
Wilt	0.767 (0.007)	0.664 (0.004)	0.549 (0.008)	**0.858 (0.001)**	0.562 (0.033)	0.532 (0.018)	0.756 (0.008)	0.681 (0.007)	0.716 (0.010)	0.589 (0.005)	0.684 (0.008)
wine	0.485 (0.014)	0.637 (0.022)	0.290 (0.038)	**0.949 (0.034)**	0.423 (0.061)	0.462 (0.018)	0.357 (0.020)	0.364 (0.036)	0.490 (0.028)	0.484 (0.020)	0.355 (0.061)
WPBC	0.521 (0.025)	0.482 (0.011)	0.515 (0.019)	0.488 (0.009)	0.510 (0.010)	**0.539 (0.017)**	0.478 (0.015)	0.481 (0.009)	0.462 (0.009)	0.513 (0.017)	0.461 (0.018)
yeast	0.448 (0.008)	0.402 (0.001)	0.467 (0.010)	0.404 (0.004)	0.428 (0.008)	0.426 (0.005)	0.440 (0.002)	**0.497 (0.005)**	0.420 (0.006)	0.411 (0.008)	0.455 (0.010)
20news	0.547 (0.006)	0.567 (0.007)	0.529 (0.007)	0.543 (0.006)	0.597 (0.005)	0.548 (0.005)	0.572 (0.005)	0.552 (0.004)	0.543 (0.003)	0.571 (0.008)	**0.608 (0.003)**
agnews	0.413 (0.007)	**0.715 (0.001)**	0.615 (0.003)	0.518 (0.008)	0.692 (0.005)	0.606 (0.006)	0.658 (0.005)	0.574 (0.000)	0.559 (0.001)	0.660 (0.001)	0.712 (0.001)
amazon	0.439 (0.020)	0.604 (0.002)	0.572 (0.006)	0.519 (0.006)	**0.610 (0.002)**	0.556 (0.011)	0.573 (0.009)	0.518 (0.001)	0.529 (0.001)	0.605 (0.002)	0.570 (0.002)
CIFAR10	0.596 (0.002)	0.681 (0.000)	0.585 (0.006)	0.618 (0.008)	0.669 (0.000)	0.608 (0.006)	0.581 (0.005)	0.648 (0.001)	0.545 (0.002)	0.662 (0.000)	**0.687 (0.001)**
FashionMNIST	0.805 (0.004)	**0.886 (0.000)**	0.704 (0.003)	0.760 (0.007)	0.872 (0.001)	0.756 (0.010)	0.761 (0.005)	0.862 (0.006)	0.764 (0.001)	0.863 (0.001)	0.739 (0.001)
imdb	0.499 (0.006)	0.483 (0.001)	0.503 (0.004)	0.495 (0.009)	0.491 (0.003)	0.495 (0.004)	0.512 (0.006)	0.506 (0.001)	**0.521 (0.001)**	0.496 (0.001)	0.501 (0.002)
MNIST-C	0.730 (0.002)	**0.800 (0.001)**	0.690 (0.001)	0.684 (0.007)	0.770 (0.002)	0.697 (0.010)	0.708 (0.004)	0.739 (0.005)	0.683 (0.004)	0.787 (0.001)	0.699 (0.002)
MVTec	0.582 (0.009)	0.576 (0.006)	0.507 (0.004)	0.693 (0.003)	**0.759 (0.001)**	0.531 (0.007)	0.529 (0.007)	0.697 (0.008)	0.656 (0.005)	0.624 (0.004)	0.740 (0.002)
SVHN	0.600 (0.002)	0.626 (0.000)	0.563 (0.001)	0.552 (0.002)	0.610 (0.000)	0.585 (0.004)	0.575 (0.004)	0.596 (0.001)	0.567 (0.002)	0.611 (0.001)	**0.628 (0.000)**
yelp	0.392 (0.014)	0.666 (0.001)	0.625 (0.004)	0.515 (0.018)	0.672 (0.002)	0.632 (0.015)	0.580 (0.011)	0.541 (0.001)	0.533 (0.004)	**0.678 (0.002)**	0.659 (0.002)
Overall	0.628 (0.005)	**0.742 (0.001)**	0.598 (0.003)	0.730 (0.004)	0.726 (0.004)	0.607 (0.002)	0.635 (0.002)	0.673 (0.001)	0.695 (0.001)	0.665 (0.001)	0.608 (0.002)

Table 22: Average F1 for the unsupervised (contaminated) setting on ADBench, column panel 1/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels. A dash denotes a group missing one or more constituent tasks for its reported seeds.

Dataset	K-DSM EMA	DSM EMA	DDAE EMA	DTE EMA	MSM EMA	K-DSM	DSM	DDAE	DTE	NCSBAD	DenoiserAD	DenoiserAD EMA
ALOI	0.049 (0.001)	0.028 (0.002)	0.092 (0.002)	0.043 (0.001)	0.098 (0.006)	0.060 (0.001)	0.053 (0.002)	0.091 (0.003)	0.045 (0.001)	0.114 (0.002)	0.150 (0.003)	0.048 (0.003)
annthyroid	0.252 (0.004)	0.297 (0.018)	0.294 (0.014)	**0.670 (0.005)**	0.331 (0.018)	0.163 (0.009)	0.261 (0.009)	0.291 (0.010)	0.648 (0.007)	0.332 (0.020)	0.281 (0.007)	0.414 (0.114)
backdoor	0.312 (0.138)	0.001 (0.001)	0.003 (0.000)	0.011 (0.002)	0.240 (0.017)	0.477 (0.008)	0.515 (0.004)	0.127 (0.014)	0.028 (0.006)	0.524 (0.006)	0.139 (0.001)	0.002 (0.002)
breastw	0.674 (0.013)	0.603 (0.007)	0.878 (0.004)	0.751 (0.016)	0.621 (0.030)	0.614 (0.010)	0.396 (0.021)	0.874 (0.009)	0.736 (0.019)	0.381 (0.009)	0.151 (0.004)	0.343 (0.007)
campaign	0.243 (0.010)	0.335 (0.012)	0.402 (0.002)	0.397 (0.007)	0.281 (0.007)	0.279 (0.011)	0.240 (0.003)	0.328 (0.006)	0.398 (0.005)	**0.431 (0.004)**	0.166 (0.005)	0.240 (0.016)
cardio	0.393 (0.027)	**0.527 (0.033)**	0.362 (0.015)	0.301 (0.010)	0.107 (0.017)	0.153 (0.014)	0.106 (0.014)	0.252 (0.014)	0.275 (0.011)	0.144 (0.010)	0.139 (0.003)	0.295 (0.085)
Cardiotocography	0.336 (0.016)	**0.415 (0.014)**	0.307 (0.006)	0.276 (0.009)	0.182 (0.006)	0.227 (0.005)	0.218 (0.005)	0.241 (0.011)	0.296 (0.010)	0.192 (0.005)	0.174 (0.004)	0.310 (0.014)
celeba	0.044 (0.008)	0.013 (0.002)	0.054 (0.003)	0.042 (0.006)	0.061 (0.005)	0.022 (0.002)	0.022 (0.002)	0.042 (0.002)	0.045 (0.004)	0.069 (0.004)	0.026 (0.002)	**0.142 (0.003)**
census	0.150 (0.010)	**0.264 (0.019)**	0.179 (0.004)	0.095 (0.004)	0.065 (0.008)	0.054 (0.002)	0.061 (0.001)	0.114 (0.003)	0.097 (0.002)	0.054 (0.001)	0.113 (0.006)	0.134 (0.002)
cover	0.414 (0.073)	0.134 (0.066)	0.057 (0.005)	0.044 (0.005)	0.017 (0.004)	0.031 (0.006)	0.031 (0.002)	0.067 (0.009)	0.063 (0.015)	0.051 (0.007)	0.032 (0.004)	**0.553 (0.019)**
donors	0.172 (0.033)	0.188 (0.053)	0.133 (0.013)	0.115 (0.016)	0.043 (0.009)	0.145 (0.002)	0.077 (0.009)	0.150 (0.016)	0.156 (0.010)	0.202 (0.009)	**0.242 (0.004)**	0.212 (0.014)
fault	0.411 (0.005)	0.405 (0.011)	0.514 (0.005)	0.445 (0.018)	0.434 (0.003)	0.459 (0.006)	0.456 (0.006)	0.506 (0.005)	0.453 (0.014)	0.385 (0.006)	0.370 (0.013)	0.417 (0.010)
fraud	**0.784 (0.029)**	0.635 (0.058)	0.715 (0.015)	0.471 (0.035)	0.025 (0.009)	0.349 (0.074)	0.225 (0.055)	0.232 (0.048)	0.750 (0.025)	0.087 (0.017)	0.057 (0.019)	0.513 (0.052)
glass	0.110 (0.012)	0.139 (0.028)	0.169 (0.021)	0.145 (0.014)	0.109 (0.029)	0.019 (0.019)	0.075 (0.028)	0.158 (0.055)	0.155 (0.012)	0.043 (0.013)	0.000 (0.000)	**0.211 (0.051)**
Hepatitis	0.299 (0.034)	**0.354 (0.071)**	0.311 (0.027)	0.270 (0.057)	0.166 (0.026)	0.072 (0.022)	0.062 (0.017)	0.152 (0.017)	0.166 (0.057)	0.114 (0.015)	0.085 (0.020)	0.262 (0.015)
http	0.828 (0.014)	0.009 (0.003)	0.201 (0.194)	0.011 (0.005)	0.044 (0.008)	0.009 (0.005)	0.106 (0.100)	0.027 (0.004)	0.149 (0.125)	0.140 (0.061)	0.015 (0.005)	0.003 (0.003)
InternetAds	0.470 (0.024)	**0.539 (0.008)**	0.465 (0.009)	0.226 (0.028)	0.367 (0.021)	0.446 (0.029)	0.371 (0.006)	0.359 (0.009)	0.218 (0.023)	0.390 (0.005)	0.264 (0.011)	0.330 (0.010)
Ionosphere	0.518 (0.017)	0.604 (0.014)	0.711 (0.009)	0.719 (0.013)	0.535 (0.028)	0.429 (0.011)	0.523 (0.013)	0.625 (0.009)	0.725 (0.021)	0.262 (0.020)	0.283 (0.023)	0.389 (0.012)
landsat	0.206 (0.012)	0.230 (0.010)	0.294 (0.004)	0.174 (0.017)	0.253 (0.004)	0.262 (0.002)	0.269 (0.004)	0.303 (0.008)	0.179 (0.004)	0.253 (0.005)	0.228 (0.005)	0.241 (0.008)
letter	0.300 (0.023)	0.338 (0.022)	0.324 (0.012)	0.334 (0.016)	0.022 (0.006)	0.396 (0.022)	**0.452 (0.024)**	0.306 (0.004)	0.326 (0.016)	0.270 (0.010)	0.275 (0.005)	0.315 (0.065)
Lymphography	0.720 (0.056)	**0.826 (0.033)**	0.810 (0.024)	0.769 (0.062)	0.021 (0.021)	0.058 (0.038)	0.021 (0.021)	0.217 (0.055)	0.377 (0.038)	0.044 (0.027)	0.161 (0.002)	0.562 (0.221)
magic.gamma	0.595 (0.002)	0.594 (0.003)	0.610 (0.002)	0.617 (0.009)	0.597 (0.008)	0.521 (0.003)	0.511 (0.003)	0.611 (0.004)	0.633 (0.013)	0.514 (0.002)	0.408 (0.000)	0.467 (0.001)
mammography	0.178 (0.003)	0.079 (0.008)	0.188 (0.009)	0.221 (0.010)	0.098 (0.013)	0.258 (0.022)	0.082 (0.003)	0.173 (0.012)	0.208 (0.008)	0.098 (0.015)	0.052 (0.013)	0.094 (0.013)
mnist	**0.484 (0.009)**	0.390 (0.010)	0.385 (0.006)	0.381 (0.007)	0.335 (0.004)	0.371 (0.006)	0.376 (0.007)	0.354 (0.006)	0.393 (0.016)	0.213 (0.009)	0.172 (0.015)	0.377 (0.001)
musk	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	0.740 (0.120)	0.049 (0.021)	0.303 (0.052)	0.454 (0.018)	0.334 (0.031)	0.470 (0.065)	0.109 (0.008)	0.088 (0.005)	**1.000 (0.000)**
optdigits	0.017 (0.003)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.001 (0.001)	0.025 (0.003)	0.025 (0.003)	0.020 (0.006)	0.000 (0.000)	0.021 (0.003)	0.020 (0.013)	0.000 (0.000)
PageBlocks	**0.633 (0.016)**	0.362 (0.007)	0.525 (0.012)	0.573 (0.011)	0.471 (0.010)	0.335 (0.008)	0.340 (0.010)	0.483 (0.019)	0.527 (0.011)	0.387 (0.010)	0.158 (0.011)	0.357 (0.002)
pendigits	**0.449 (0.156)**	0.319 (0.129)	0.069 (0.007)	0.046 (0.006)	0.042 (0.010)	0.076 (0.007)	0.067 (0.004)	0.069 (0.004)	0.049 (0.006)	0.042 (0.006)	0.042 (0.003)	0.051 (0.019)
Pima	0.470 (0.021)	0.444 (0.009)	0.520 (0.005)	0.425 (0.013)	0.367 (0.017)	0.323 (0.010)	0.350 (0.010)	0.514 (0.004)	0.448 (0.001)	0.289 (0.006)	0.257 (0.006)	0.318 (0.022)
satellite	0.424 (0.007)	0.384 (0.008)	0.503 (0.005)	**0.602 (0.009)**	0.376 (0.008)	0.365 (0.003)	0.355 (0.003)	0.506 (0.006)	0.573 (0.012)	0.357 (0.003)	0.328 (0.006)	0.346 (0.002)
satimage-2	0.744 (0.130)	0.775 (0.031)	0.563 (0.059)	0.206 (0.011)	0.028 (0.009)	0.059 (0.015)	0.045 (0.014)	0.217 (0.020)	0.197 (0.018)	0.110 (0.012)	0.070 (0.000)	**0.810 (0.035)**
shuttle	**0.957 (0.001)**	0.635 (0.132)	0.091 (0.003)	0.676 (0.066)	0.227 (0.028)	0.093 (0.002)	0.141 (0.008)	0.103 (0.007)	0.744 (0.025)	0.187 (0.008)	0.054 (0.000)	0.569 (0.014)
skin	0.408 (0.019)	0.195 (0.033)	0.421 (0.007)	0.199 (0.006)	0.164 (0.011)	0.110 (0.008)	0.115 (0.002)	0.355 (0.007)	0.194 (0.011)	0.295 (0.003)	0.267 (0.005)	0.197 (0.017)
smtp	0.656 (0.058)	0.656 (0.058)	0.546 (0.056)	0.614 (0.053)	0.000 (0.000)	0.193 (0.083)	0.686 (0.027)	0.495 (0.072)	**0.696 (0.020)**	0.012 (0.012)	0.024 (0.024)	0.520 (0.186)
SpamBase	0.420 (0.001)	0.405 (0.005)	0.464 (0.006)	0.380 (0.007)	0.413 (0.005)	0.471 (0.003)	0.387 (0.002)	**0.479 (0.004)**	0.385 (0.010)	0.427 (0.003)	0.349 (0.005)	0.368 (0.004)
speech	0.043 (0.008)	0.039 (0.011)	0.026 (0.011)	0.023 (0.012)	0.026 (0.008)	0.033 (0.009)	0.026 (0.008)	0.036 (0.012)	0.030 (0.010)	0.020 (0.003)	0.041 (0.008)	**0.057 (0.025)**
Stamps	0.210 (0.035)	0.239 (0.057)	0.229 (0.037)	0.222 (0.022)	0.104 (0.016)	0.056 (0.015)	0.063 (0.018)	0.176 (0.013)	0.213 (0.019)	0.091 (0.015)	0.053 (0.002)	0.180 (0.053)
thyroid	0.383 (0.015)	0.409 (0.043)	0.417 (0.050)	**0.703 (0.014)**	0.228 (0.036)	0.148 (0.005)	0.095 (0.020)	0.202 (0.032)	0.701 (0.004)	0.097 (0.021)	0.210 (0.027)	0.532 (0.220)
vertebral	0.000 (0.000)	0.000 (0.000)	0.043 (0.007)	0.084 (0.014)	0.116 (0.020)	**0.155 (0.031)**	0.133 (0.019)	0.039 (0.010)	0.078 (0.015)	0.122 (0.015)	0.097 (0.008)	0.043 (0.027)
vowels	0.132 (0.019)	0.252 (0.019)	0.472 (0.016)	0.392 (0.054)	0.008 (0.005)	0.212 (0.008)	0.272 (0.026)	0.424 (0.007)	0.416 (0.047)	0.244 (0.019)	0.250 (0.010)	0.360 (0.080)
Waveform	0.052 (0.007)	0.048 (0.004)	0.064 (0.012)	0.048 (0.002)	0.012 (0.005)	0.032 (0.012)	0.042 (0.009)	0.058 (0.010)	0.064 (0.007)	0.042 (0.010)	0.050 (0.000)	0.050 (0.010)
WBC	0.751 (0.058)	0.739 (0.059)	0.782 (0.032)	0.313 (0.016)	0.060 (0.030)	0.040 (0.027)	0.064 (0.027)	0.244 (0.036)	0.222 (0.018)	0.022 (0.014)	0.011 (0.011)	**0.827 (0.088)**
WDBC	**0.531 (0.027)**	0.433 (0.036)	0.417 (0.068)	0.183 (0.047)	0.015 (0.009)	0.016 (0.010)	0.016 (0.010)	0.246 (0.081)	0.208 (0.050)	0.023 (0.010)	0.041 (0.003)	0.125 (0.048)
Wilt	0.004 (0.000)	0.012 (0.001)	0.003 (0.001)	0.105 (0.031)	0.079 (0.011)	0.013 (0.002)	0.074 (0.002)	0.003 (0.001)	0.055 (0.013)	**0.239 (0.036)**	0.107 (0.021)	0.091 (0.025)
wine	0.000 (0.000)	0.035 (0.035)	0.005 (0.005)	0.043 (0.027)	0.051 (0.028)	0.066 (0.027)	0.070 (0.020)	0.029 (0.020)	0.044 (0.027)	0.098 (0.041)	0.060 (0.060)	0.075 (0.075)
WPBC	0.182 (0.025)	0.192 (0.013)	0.203 (0.009)	0.197 (0.027)	0.246 (0.025)	0.223 (0.022)	0.254 (0.028)	0.254 (0.018)	0.226 (0.027)	0.216 (0.016)	0.251 (0.033)	**0.272 (0.008)**
yeast	0.293 (0.007)	0.281 (0.013)	0.282 (0.005)	0.272 (0.005)	0.303 (0.008)	0.321 (0.003)	0.331 (0.006)	0.284 (0.008)	0.295 (0.010)	0.320 (0.008)	0.310 (0.000)	0.297 (0.017)
20news	0.086 (0.003)	0.081 (0.003)	0.070 (0.004)	0.064 (0.007)	0.058 (0.002)	0.087 (0.007)	0.092 (0.004)	0.090 (0.005)	0.070 (0.005)	0.102 (0.006)	**0.103 (0.004)**	0.095 (0.002)
agnews	0.137 (0.000)	0.105 (0.001)	0.125 (0.001)	0.085 (0.007)	0.039 (0.004)	0.114 (0.006)	0.123 (0.003)	0.115 (0.004)	0.084 (0.003)	0.131 (0.002)	0.129 (0.003)	**0.197 (0.000)**
amazon	0.045 (0.000)	0.050 (0.001)	0.046 (0.001)	0.056 (0.003)	0.039 (0.008)	0.057 (0.004)	0.050 (0.001)	0.051 (0.002)	0.057 (0.003)	0.058 (0.002)	**0.063 (0.001)**	0.050 (0.002)
CIFAR10	0.140 (0.004)	0.123 (0.001)	0.128 (0.001)	0.130 (0.001)	0.075 (0.003)	0.127 (0.001)	0.108 (0.002)	0.125 (0.002)	0.126 (0.002)	0.116 (0.002)	0.120 (0.006)	0.134 (0.001)
FashionMNIST	**0.481 (0.008)**	0.404 (0.002)	0.394 (0.001)	0.361 (0.006)	0.125 (0.013)	0.347 (0.002)	0.267 (0.001)	0.316 (0.002)	0.321 (0.002)	0.248 (0.003)	0.247 (0.007)	0.388 (0.010)
imdb	0.018 (0.001)	0.022 (0.001)	0.023 (0.001)	0.040 (0.004)	0.045 (0.003)	0.029 (0.002)	0.028 (0.002)	0.025 (0.002)	0.026 (0.003)	0.032 (0.002)	0.038 (0.008)	0.037 (0.003)
MNIST-C	**0.264 (0.016)**	0.255 (0.002)	0.232 (0.001)	0.202 (0.006)	0.139 (0.004)	0.212 (0.002)	0.233 (0.003)	0.217 (0.001)	0.195 (0.003)	0.193 (0.001)	0.193 (0.000)	0.243 (0.003)
MVTec	0.497 (0.005)	0.514 (0.005)	0.454 (0.005)	0.488 (0.007)	0.371 (0.007)	0.394 (0.007)	0.364 (0.005)	0.393 (0.002)	0.493 (0.006)	0.246 (0.003)	0.363 (0.010)	0.395 (0.001)
SVHN	**0.124 (0.004)**	0.115 (0.001)	0.113 (0.000)	0.113 (0.001)	0.078 (0.000)	0.108 (0.002)	0.091 (0.002)	0.105 (0.001)	0.111 (0.001)	0.097 (0.002)	0.096 (0.003)	0.115 (0.002)
yelp	0.114 (0.002)	0.098 (0.002)	0.109 (0.004)	0.074 (0.014)	0.030 (0.004)	0.097 (0.008)	0.109 (0.003)	0.102 (0.005)	0.076 (0.012)	0.106 (0.003)	0.101 (0.005)	0.089 (0.005)
Overall	**0.344 (0.005)**	0.308 (0.005)	0.312 (0.004)	0.284 (0.003)	0.166 (0.001)	0.195 (0.003)	0.200 (0.003)	0.241 (0.003)	0.279 (0.004)	0.183 (0.001)	0.150 (0.000)	0.282 (0.007)

Table 22: Average F1 for the unsupervised (contaminated) setting on ADBench, column panel 2/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels. A dash denotes a group missing one or more constituent tasks for its reported seeds.

Dataset	MSM	GMM	KPCA	MCD	MCM	LUNAR	NEAD	SLAD	ICL	kNN	LOF
ALOI	0.141 (0.003)	0.045 (0.000)	0.066 (0.002)	0.046 (0.001)	0.089 (0.004)	**0.157 (0.005)**	0.079 (0.003)	0.049 (0.001)	0.070 (0.002)	0.118 (0.001)	0.140 (0.001)
annthyroid	0.390 (0.016)	0.205 (0.003)	0.290 (0.003)	0.455 (0.002)	0.277 (0.040)	0.238 (0.003)	0.255 (0.005)	0.128 (0.017)	0.313 (0.008)	0.306 (0.004)	0.200 (0.002)
backdoor	0.300 (0.011)	0.510 (0.003)	0.228 (0.017)	0.119 (0.083)	0.420 (0.021)	0.249 (0.014)	0.116 (0.005)	0.039 (0.010)	**0.716 (0.063)**	0.058 (0.003)	0.130 (0.015)
breastw	0.687 (0.011)	0.909 (0.003)	0.325 (0.065)	**0.918 (0.011)**	0.890 (0.015)	0.637 (0.032)	0.586 (0.008)	0.590 (0.012)	0.815 (0.024)	0.837 (0.011)	0.187 (0.031)
campaign	0.255 (0.005)	0.404 (0.001)	0.420 (0.002)	0.399 (0.001)	0.424 (0.005)	0.202 (0.006)	0.222 (0.006)	0.298 (0.003)	0.314 (0.003)	0.385 (0.001)	0.133 (0.001)
cardio	0.117 (0.009)	0.455 (0.004)	0.223 (0.009)	0.460 (0.017)	0.384 (0.017)	0.242 (0.020)	0.127 (0.008)	0.184 (0.016)	0.159 (0.006)	0.345 (0.010)	0.201 (0.017)
Cardiotocography	0.175 (0.006)	0.348 (0.002)	0.185 (0.006)	0.282 (0.001)	0.322 (0.006)	0.249 (0.005)	0.250 (0.006)	0.208 (0.019)	0.200 (0.007)	0.276 (0.004)	0.273 (0.003)
celeba	0.040 (0.004)	0.107 (0.002)	0.067 (0.002)	0.121 (0.003)	0.079 (0.016)	0.033 (0.002)	0.032 (0.003)	0.066 (0.004)	0.054 (0.004)	0.062 (0.003)	0.052 (0.005)
census	0.054 (0.010)	0.062 (0.001)	0.186 (0.004)	0.021 (0.013)	0.182 (0.016)	0.070 (0.001)	0.078 (0.006)	0.103 (0.018)	0.128 (0.002)	0.139 (0.002)	0.004 (0.001)
cover	0.050 (0.007)	0.057 (0.005)	0.067 (0.010)	0.054 (0.005)	0.051 (0.016)	0.069 (0.009)	0.129 (0.006)	0.026 (0.008)	0.048 (0.018)	0.075 (0.003)	0.048 (0.004)
donors	0.088 (0.005)	0.134 (0.004)	0.141 (0.007)	0.148 (0.003)	0.123 (0.024)	0.187 (0.005)	0.055 (0.005)	0.004 (0.001)	0.137 (0.019)	0.212 (0.004)	0.100 (0.004)
fault	0.413 (0.007)	0.457 (0.004)	0.357 (0.006)	0.444 (0.006)	0.475 (0.033)	0.484 (0.005)	0.477 (0.009)	0.514 (0.003)	0.481 (0.005)	**0.521 (0.003)**	0.414 (0.006)
fraud	0.073 (0.019)	0.240 (0.031)	0.181 (0.039)	0.526 (0.021)	0.243 (0.031)	0.176 (0.018)	0.145 (0.032)	0.261 (0.036)	0.203 (0.033)	0.121 (0.019)	0.000 (0.000)
glass	0.095 (0.045)	0.140 (0.017)	0.025 (0.018)	0.039 (0.024)	0.183 (0.047)	0.129 (0.029)	0.100 (0.033)	0.155 (0.041)	0.105 (0.025)	0.205 (0.030)	0.189 (0.029)
Hepatitis	0.182 (0.063)	0.244 (0.021)	0.071 (0.033)	0.259 (0.029)	0.235 (0.026)	0.086 (0.018)	0.136 (0.036)	0.067 (0.020)	0.135 (0.038)	0.180 (0.030)	0.135 (0.044)
http	0.048 (0.009)	0.035 (0.016)	0.034 (0.006)	**0.861 (0.011)**	0.029 (0.008)	0.018 (0.001)	0.028 (0.004)	0.025 (0.006)	0.503 (0.021)	0.041 (0.006)	0.049 (0.009)
InternetAds	0.298 (0.013)	0.360 (0.008)	0.278 (0.009)	0.431 (0.041)	0.358 (0.012)	0.334 (0.009)	0.322 (0.005)	0.276 (0.007)	0.191 (0.013)	0.359 (0.012)	0.359 (0.011)
Ionosphere	0.525 (0.011)	0.839 (0.004)	0.362 (0.018)	**0.854 (0.010)**	0.634 (0.016)	0.423 (0.039)	0.650 (0.013)	0.717 (0.016)	0.491 (0.021)	0.577 (0.018)	0.758 (0.014)
landsat	0.257 (0.005)	0.192 (0.001)	0.279 (0.006)	0.200 (0.002)	0.205 (0.014)	0.264 (0.007)	0.215 (0.001)	0.368 (0.003)	**0.414 (0.016)**	0.289 (0.005)	0.268 (0.005)
letter	0.268 (0.017)	0.272 (0.009)	0.244 (0.013)	0.252 (0.011)	0.266 (0.049)	0.324 (0.015)	0.348 (0.015)	0.342 (0.013)	0.272 (0.015)	0.356 (0.005)	0.396 (0.016)
Lymphography	0.034 (0.034)	0.680 (0.038)	0.152 (0.071)	0.758 (0.055)	0.492 (0.057)	0.092 (0.040)	0.021 (0.021)	0.605 (0.036)	0.483 (0.076)	0.228 (0.067)	0.112 (0.047)
magic.gamma	0.617 (0.004)	0.583 (0.000)	0.577 (0.003)	0.586 (0.000)	0.558 (0.020)	0.595 (0.008)	0.572 (0.002)	0.483 (0.009)	0.531 (0.014)	**0.639 (0.001)**	0.513 (0.001)
mammography	0.145 (0.016)	**0.278 (0.009)**	0.241 (0.010)	0.012 (0.002)	0.197 (0.021)	0.179 (0.010)	0.142 (0.007)	0.125 (0.010)	0.109 (0.009)	0.219 (0.006)	0.182 (0.005)
mnist	0.251 (0.009)	0.390 (0.003)	0.359 (0.006)	0.397 (0.005)	0.408 (0.010)	0.296 (0.011)	0.267 (0.007)	0.087 (0.087)	0.295 (0.011)	0.405 (0.001)	0.268 (0.006)
musk	0.099 (0.008)	0.746 (0.020)	0.144 (0.018)	0.173 (0.008)	0.377 (0.003)	0.035 (0.013)	0.113 (0.014)	0.260 (0.026)	0.540 (0.083)	0.155 (0.023)	0.144 (0.035)
optdigits	**0.047 (0.011)**	0.000 (0.000)	0.003 (0.002)	0.000 (0.000)	0.000 (0.000)	0.013 (0.003)	0.001 (0.001)	0.000 (0.000)	0.012 (0.004)	0.000 (0.000)	**0.047 (0.005)**
PageBlocks	0.469 (0.013)	0.510 (0.006)	0.402 (0.010)	0.598 (0.005)	0.434 (0.017)	0.391 (0.017)	0.456 (0.008)	0.388 (0.009)	0.504 (0.007)	0.462 (0.005)	0.320 (0.012)
pendigits	0.029 (0.006)	0.045 (0.002)	0.117 (0.014)	0.044 (0.003)	0.128 (0.049)	0.073 (0.006)	0.071 (0.007)	0.051 (0.012)	0.122 (0.016)	0.096 (0.005)	0.090 (0.008)
Pima	0.391 (0.017)	0.521 (0.017)	0.374 (0.017)	**0.533 (0.014)**	0.465 (0.048)	0.432 (0.015)	0.444 (0.009)	0.340 (0.022)	0.427 (0.011)	0.525 (0.011)	0.406 (0.012)
satellite	0.370 (0.004)	0.456 (0.002)	0.491 (0.001)	0.468 (0.003)	0.471 (0.008)	0.415 (0.004)	0.386 (0.003)	0.534 (0.006)	0.564 (0.029)	0.478 (0.003)	0.371 (0.005)
satimage-2	0.037 (0.016)	0.372 (0.007)	0.476 (0.028)	0.355 (0.010)	0.648 (0.123)	0.239 (0.006)	0.068 (0.012)	0.341 (0.026)	0.400 (0.103)	0.355 (0.022)	0.099 (0.027)
shuttle	0.250 (0.010)	0.866 (0.002)	0.688 (0.011)	0.875 (0.002)	0.681 (0.138)	0.185 (0.003)	0.174 (0.006)	0.611 (0.139)	0.441 (0.023)	0.211 (0.003)	0.128 (0.004)
skin	0.180 (0.006)	0.201 (0.004)	0.201 (0.007)	**0.588 (0.002)**	0.073 (0.050)	0.292 (0.003)	0.242 (0.006)	0.415 (0.006)	0.189 (0.036)	0.282 (0.004)	0.210 (0.003)
smtp	0.000 (0.000)	0.369 (0.041)	0.646 (0.064)	0.000 (0.000)	0.434 (0.102)	0.034 (0.021)	0.012 (0.012)	**0.696 (0.020)**	0.206 (0.079)	0.324 (0.078)	0.000 (0.000)
SpamBase	0.395 (0.005)	0.410 (0.003)	0.386 (0.003)	0.410 (0.007)	0.410 (0.013)	0.402 (0.008)	0.373 (0.005)	0.388 (0.003)	0.379 (0.012)	0.417 (0.003)	0.355 (0.003)
speech	0.007 (0.004)	0.026 (0.008)	0.036 (0.006)	0.030 (0.006)	0.016 (0.000)	0.043 (0.007)	0.026 (0.008)	0.013 (0.006)	0.026 (0.011)	0.030 (0.003)	0.030 (0.008)
Stamps	0.126 (0.014)	0.235 (0.030)	0.079 (0.017)	0.124 (0.030)	**0.298 (0.025)**	0.130 (0.023)	0.094 (0.019)	0.164 (0.027)	0.159 (0.035)	0.206 (0.020)	0.187 (0.033)
thyroid	0.178 (0.025)	0.271 (0.014)	0.359 (0.015)	0.656 (0.000)	0.273 (0.067)	0.252 (0.010)	0.075 (0.010)	0.168 (0.021)	0.194 (0.035)	0.297 (0.012)	0.097 (0.009)
vertebral	0.130 (0.023)	0.040 (0.007)	0.136 (0.036)	0.068 (0.044)	0.057 (0.026)	0.137 (0.030)	0.071 (0.018)	0.065 (0.010)	0.052 (0.011)	0.046 (0.012)	0.105 (0.034)
vowels	0.208 (0.022)	0.380 (0.006)	0.320 (0.023)	0.356 (0.028)	0.464 (0.044)	0.444 (0.036)	0.284 (0.019)	0.344 (0.025)	0.244 (0.024)	**0.504 (0.013)**	0.324 (0.021)
Waveform	0.050 (0.011)	0.064 (0.006)	0.080 (0.012)	0.064 (0.006)	0.088 (0.009)	0.108 (0.010)	0.042 (0.011)	0.008 (0.004)	**0.198 (0.022)**	0.156 (0.009)	0.102 (0.010)
WBC	0.049 (0.013)	0.584 (0.025)	0.227 (0.061)	0.667 (0.056)	0.505 (0.082)	0.185 (0.028)	0.022 (0.014)	0.540 (0.088)	0.329 (0.071)	0.250 (0.025)	0.082 (0.029)
WDBC	0.015 (0.009)	0.355 (0.040)	0.031 (0.008)	0.438 (0.070)	0.406 (0.050)	0.074 (0.031)	0.034 (0.009)	0.303 (0.077)	0.023 (0.010)	0.279 (0.053)	0.058 (0.027)
Wilt	0.066 (0.006)	0.008 (0.000)	0.003 (0.001)	0.000 (0.000)	0.045 (0.013)	0.005 (0.001)	0.068 (0.007)	0.107 (0.011)	0.075 (0.011)	0.001 (0.001)	0.060 (0.006)
wine	0.060 (0.026)	0.000 (0.000)	0.014 (0.014)	**0.550 (0.128)**	0.000 (0.000)	0.032 (0.032)	0.032 (0.021)	0.000 (0.000)	0.021 (0.021)	0.076 (0.025)	0.000 (0.000)
WPBC	**0.272 (0.016)**	0.172 (0.021)	0.239 (0.024)	0.158 (0.016)	0.191 (0.023)	0.256 (0.033)	0.202 (0.016)	0.218 (0.016)	0.178 (0.010)	0.261 (0.015)	0.187 (0.032)
yeast	0.302 (0.007)	0.262 (0.001)	0.325 (0.009)	0.268 (0.007)	0.275 (0.012)	0.275 (0.004)	0.297 (0.005)	**0.336 (0.005)**	0.277 (0.009)	0.295 (0.005)	0.306 (0.004)
20news	0.092 (0.005)	0.093 (0.005)	0.082 (0.004)	0.069 (0.004)	0.086 (0.005)	0.081 (0.005)	0.101 (0.010)	0.060 (0.004)	0.066 (0.002)	0.081 (0.004)	**0.103 (0.003)**
agnews	0.030 (0.002)	0.129 (0.002)	0.100 (0.001)	0.062 (0.004)	0.134 (0.002)	0.114 (0.001)	0.125 (0.003)	0.075 (0.001)	0.074 (0.002)	0.110 (0.001)	0.169 (0.002)
amazon	0.027 (0.007)	0.048 (0.001)	0.055 (0.000)	0.057 (0.004)	0.047 (0.002)	0.049 (0.003)	0.061 (0.002)	0.043 (0.001)	0.044 (0.003)	0.046 (0.001)	0.053 (0.002)
CIFAR10	0.095 (0.002)	0.137 (0.002)	0.125 (0.001)	0.106 (0.004)	0.133 (0.000)	0.131 (0.002)	0.116 (0.002)	0.138 (0.001)	0.105 (0.002)	0.130 (0.000)	**0.159 (0.001)**
FashionMNIST	0.212 (0.004)	0.365 (0.002)	0.327 (0.003)	0.327 (0.012)	0.347 (0.002)	0.305 (0.007)	0.236 (0.002)	0.360 (0.003)	0.223 (0.003)	0.354 (0.001)	0.257 (0.002)
imdb	0.044 (0.003)	0.024 (0.001)	0.026 (0.001)	0.034 (0.006)	0.026 (0.002)	0.024 (0.002)	0.030 (0.003)	**0.057 (0.001)**	0.055 (0.002)	0.019 (0.001)	0.040 (0.002)
MNIST-C	0.168 (0.002)	0.232 (0.001)	0.208 (0.001)	0.187 (0.006)	0.200 (0.001)	0.223 (0.004)	0.169 (0.001)	0.214 (0.004)	0.139 (0.003)	0.222 (0.001)	0.159 (0.001)
MVTec	0.334 (0.008)	0.308 (0.006)	0.258 (0.005)	0.462 (0.004)	**0.529 (0.001)**	0.284 (0.007)	0.299 (0.007)	0.469 (0.007)	0.389 (0.005)	0.422 (0.003)	0.495 (0.005)
SVHN	0.094 (0.001)	0.113 (0.002)	0.106 (0.001)	0.096 (0.001)	0.112 (0.000)	0.109 (0.002)	0.096 (0.002)	0.111 (0.001)	0.093 (0.002)	0.113 (0.001)	0.110 (0.001)
yelp	0.025 (0.004)	0.104 (0.001)	0.112 (0.003)	0.072 (0.011)	0.102 (0.002)	**0.116 (0.006)**	0.086 (0.007)	0.023 (0.001)	0.031 (0.002)	0.115 (0.002)	0.102 (0.004)
Overall	0.182 (0.002)	0.288 (0.001)	0.220 (0.002)	0.308 (0.005)	0.280 (0.001)	0.204 (0.001)	0.180 (0.002)	0.238 (0.002)	0.245 (0.006)	0.250 (0.001)	0.184 (0.002)

M IMAGE BENCHMARK RESULTS

To assess how K-DSM behaves on dense embedding features with small per-class training sets, we evaluate on two standard industrial image anomaly detection benchmarks, MVTec-AD (Bergmann et al., 2019) and VisA (Zou et al., 2022), using frozen DINOv3 (Siméoni et al., 2025) feature embeddings followed by standardisation. All nine methods evaluated on these image benchmarks share the same features and preprocessing.

Table 23: Mean AUC-PR and AUC-ROC on MVTec-AD and VisA with standardised DINOv3 embeddings (5 seeds, standard error in parentheses). Best displayed mean per row in bold, including ties.

		*k*NN	LOF	ICL	DTE	DSM	K-DSM	MSM	DDAE	SLAD
AUC-PR	VISA	0.904 (0.000)	0.900 (0.000)	0.785 (0.001)	0.857 (0.001)	**0.939 (0.000)**	0.918 (0.000)	0.936 (0.000)	0.926 (0.000)	0.829 (0.000)
	MVTec	0.971 (0.000)	0.970 (0.000)	0.954 (0.000)	0.963 (0.000)	**0.983 (0.000)**	0.962 (0.000)	0.958 (0.000)	**0.983 (0.000)**	0.970 (0.000)
AUC-ROC	VISA	0.887 (0.000)	0.878 (0.000)	0.735 (0.001)	0.815 (0.001)	**0.928 (0.000)**	0.881 (0.000)	0.926 (0.000)	0.913 (0.000)	0.784 (0.000)
	MVTec	0.943 (0.000)	0.938 (0.000)	0.911 (0.000)	0.930 (0.000)	0.960 (0.000)	0.900 (0.000)	0.919 (0.001)	**0.965 (0.000)**	0.939 (0.000)

K-DSM does not reproduce its ADBench advantage on these benchmarks: single-scale DSM reaches 0.983 mean AUC-PR on MVTec-AD and 0.939 on VisA, compared with 0.962 and 0.918 for K-DSM. The per-class results show that the loss is concentrated in a few classes (Tables 24 and 25). K-DSM is within 0.025 AUC-PR of DSM on 23 of 27 classes (14 of 15 on MVTec-AD and 9 of 12 on VisA). On these classes, the K-DSM and DSM means are 0.980 versus 0.982 on MVTec-AD and 0.919 versus 0.923 on VisA. Four classes account for most of the overall difference: tile ($1.000 \to 0.714$), cashew ($0.977 \to 0.875$), pcb4 ($0.990 \to 0.922$), and pipe_fryum ($0.992 \to 0.952$), where each pair gives DSM followed by K-DSM.

These failures do not extend to all embedding features. On the ten ADBench groups built from image and text embeddings, K-DSM improves over DSM on nine, including FashionMNIST ($0.428 \to 0.702$) and MNIST-C ($0.526 \to 0.656$; Table 29). Two differences may help explain the image-benchmark results. MVTec-AD provides only 60–400 normal training images per class, compared with thousands of observations in many ADBench groups, while we use 200 histogram bins in both cases. DSM also performs close to perfectly on many image classes, leaving little room for gains. We have not established whether either factor explains the four failures. Limiting the fraction of features assigned the upper noise clip is one possible safeguard to test in future work.

M.1 PER-CLASS IMAGE BENCHMARK RESULTS

We report per-class AUC-PR on MVTec-AD and VisA with standardised DINOv3 embeddings. All methods use identical features, preprocessing, and splits. Results are mean over 5 seeds with standard error in parentheses; best displayed mean per row in bold, including ties.

Table 24: Per-class AUC-PR on MVTec-AD (standardised DINOv3 embeddings). Best displayed mean per row in bold, including ties.

	*k*NN	LOF	ICL	DTE	DSM	K-DSM	MSM	DDAE	SLAD
bottle	0.999 (0.000)	0.999 (0.000)	0.987 (0.000)	0.999 (0.000)	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	0.999 (0.000)
cable	0.936 (0.000)	0.946 (0.000)	0.894 (0.000)	0.896 (0.000)	0.959 (0.000)	**0.974 (0.000)**	0.956 (0.002)	0.963 (0.001)	0.935 (0.000)
capsule	0.986 (0.000)	**0.988 (0.000)**	0.979 (0.000)	**0.988 (0.000)**	0.987 (0.000)	0.985 (0.000)	0.987 (0.001)	0.985 (0.000)	0.978 (0.000)
carpet	0.996 (0.000)	0.996 (0.000)	0.996 (0.000)	0.995 (0.000)	**0.997 (0.000)**	0.985 (0.000)	0.996 (0.000)	0.995 (0.000)	0.995 (0.000)
grid	**0.999 (0.000)**	0.998 (0.000)	0.987 (0.000)	0.996 (0.000)	**0.999 (0.000)**	**0.999 (0.000)**	**0.999 (0.000)**	0.998 (0.000)	0.997 (0.000)
hazelnut	0.945 (0.000)	0.960 (0.000)	0.937 (0.000)	0.970 (0.000)	**0.995 (0.000)**	0.975 (0.000)	**0.995 (0.000)**	0.983 (0.001)	0.952 (0.000)
leather	**1.000 (0.000)**	**1.000 (0.000)**	0.997 (0.000)	0.999 (0.000)	0.999 (0.000)	0.992 (0.000)	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**
metal_nut	0.986 (0.000)	0.986 (0.000)	0.970 (0.000)	0.985 (0.000)	**0.998 (0.000)**	0.978 (0.000)	0.995 (0.001)	0.996 (0.000)	0.991 (0.000)
pill	0.992 (0.000)	0.992 (0.000)	0.971 (0.000)	0.976 (0.000)	0.995 (0.000)	**0.996 (0.000)**	**0.996 (0.000)**	0.995 (0.000)	0.989 (0.000)
screw	0.916 (0.000)	0.900 (0.000)	0.908 (0.000)	0.914 (0.000)	0.951 (0.000)	**0.954 (0.000)**	0.927 (0.002)	0.934 (0.001)	0.919 (0.000)
tile	0.999 (0.000)	**1.000 (0.000)**	0.997 (0.000)	0.997 (0.000)	**1.000 (0.000)**	0.714 (0.000)	**1.000 (0.000)**	**1.000 (0.000)**	0.999 (0.000)
toothbrush	0.959 (0.000)	0.925 (0.000)	0.960 (0.000)	0.922 (0.000)	0.941 (0.000)	0.965 (0.000)	0.579 (0.001)	**0.970 (0.001)**	0.954 (0.000)
transistor	0.908 (0.000)	0.914 (0.000)	0.776 (0.000)	0.829 (0.000)	0.936 (0.000)	0.934 (0.000)	**0.959 (0.005)**	0.956 (0.002)	0.908 (0.000)
wood	0.944 (0.000)	0.953 (0.000)	0.979 (0.000)	0.979 (0.000)	0.985 (0.000)	**0.988 (0.000)**	0.983 (0.000)	0.971 (0.001)	0.946 (0.000)
zipper	0.995 (0.000)	0.997 (0.000)	0.979 (0.000)	0.997 (0.000)	0.998 (0.000)	0.992 (0.000)	**0.999 (0.000)**	0.998 (0.000)	0.992 (0.000)
Mean	0.971 (0.000)	0.970 (0.000)	0.954 (0.000)	0.963 (0.000)	**0.983 (0.000)**	0.962 (0.000)	0.958 (0.000)	**0.983 (0.000)**	0.970 (0.000)

Table 25: Per-class AUC-PR on VisA (standardised DINOv3 embeddings). Best displayed mean per row in bold, including ties.

	kNN	LOF	ICL	DTE	DSM	K-DSM	MSM	DDAE	SLAD
candle	0.878 (0.000)	0.887 (0.000)	0.785 (0.010)	0.851 (0.017)	0.912 (0.001)	**0.924 (0.004)**	0.909 (0.001)	0.896 (0.001)	0.797 (0.000)
capsules	0.910 (0.000)	0.913 (0.000)	0.811 (0.000)	0.891 (0.000)	0.931 (0.000)	0.931 (0.000)	**0.936 (0.002)**	0.921 (0.000)	0.878 (0.000)
cashew	0.967 (0.000)	0.967 (0.000)	0.882 (0.000)	0.867 (0.000)	0.977 (0.000)	0.875 (0.000)	**0.979 (0.001)**	0.969 (0.002)	0.919 (0.000)
chewinggum	0.991 (0.000)	0.990 (0.000)	0.969 (0.000)	0.988 (0.000)	**0.995 (0.000)**	0.988 (0.000)	0.994 (0.000)	0.994 (0.000)	0.987 (0.000)
fryum	0.950 (0.000)	0.956 (0.000)	0.897 (0.000)	0.924 (0.000)	0.980 (0.000)	**0.983 (0.000)**	0.981 (0.000)	0.969 (0.001)	0.938 (0.000)
macaroni1	0.846 (0.000)	0.879 (0.000)	0.715 (0.000)	0.811 (0.000)	**0.939 (0.000)**	0.918 (0.000)	0.933 (0.002)	0.897 (0.004)	0.739 (0.000)
macaroni2	0.708 (0.000)	0.716 (0.000)	0.588 (0.000)	0.673 (0.000)	**0.801 (0.000)**	0.791 (0.000)	0.783 (0.005)	0.769 (0.006)	0.712 (0.000)
pcb1	0.967 (0.000)	0.907 (0.000)	0.562 (0.000)	0.926 (0.000)	**0.983 (0.000)**	0.977 (0.000)	0.971 (0.001)	0.980 (0.001)	0.552 (0.000)
pcb2	0.867 (0.000)	0.873 (0.000)	0.721 (0.000)	0.815 (0.000)	0.939 (0.000)	**0.940 (0.000)**	0.937 (0.002)	0.909 (0.003)	0.750 (0.000)
pcb3	0.803 (0.000)	0.812 (0.000)	0.721 (0.000)	0.694 (0.000)	0.829 (0.000)	0.819 (0.000)	0.825 (0.001)	**0.830 (0.002)**	0.765 (0.000)
pcb4	0.987 (0.000)	0.921 (0.000)	0.827 (0.000)	0.899 (0.000)	0.990 (0.000)	0.922 (0.000)	**0.992 (0.001)**	**0.992 (0.001)**	0.951 (0.000)
pipe_fryum	0.976 (0.000)	0.979 (0.000)	0.937 (0.000)	0.946 (0.000)	**0.992 (0.000)**	0.952 (0.000)	0.988 (0.001)	0.985 (0.001)	0.959 (0.000)
Mean	0.904 (0.000)	0.900 (0.000)	0.785 (0.001)	0.857 (0.001)	**0.939 (0.000)**	0.918 (0.000)	0.936 (0.000)	0.926 (0.000)	0.829 (0.000)

N Full Semi-supervised Results

We report per-dataset AUC-PR, AUC-ROC, and F1 for the semi-supervised setting summarised in Section 4.2. Best per row in bold; standard errors over 5 seeds in parentheses. Each metric table includes all 18 configurations in two continued column panels.

Benchmark comparison. Table 26 reports all evaluated configurations on the same 57 datasets, with five seeds. K-DSM obtains 0.627 mean AUC-PR, compared with 0.620 for DenoiserAD and 0.596 for KPCA; the combined denoiser reaches 0.630. KPCA leads mean AUC-ROC at 0.833, compared with 0.823 for K-DSM. The per-dataset tables below report AUC-PR, AUC-ROC, and F1 for the same methods. NRDE entries use the released algorithm described in Appendix G.2.

Table 26: Semi-supervised comparison on all 57 ADBench datasets, five seeds. Each dataset receives equal weight after averaging its constituent variants. Parentheses are standard errors of the five seed-level macro means; best mean per metric is bold. The combined denoiser is a separate configuration. NRDE uses the released algorithm discussed in Appendix G.2.

Method	AUC-PR	AUC-ROC
K-DSM+DenoiserAD	**0.630 (0.001)**	0.832 (0.001)
K-DSM	0.627 (0.003)	0.823 (0.002)
DenoiserAD	0.620 (0.005)	0.831 (0.001)
KPCA	0.596 (0.002)	**0.833 (0.000)**
DDAE	0.594 (0.001)	0.819 (0.001)
NCSBAD	0.592 (0.005)	0.827 (0.001)
kNN-mean	0.592 (0.002)	0.824 (0.000)
DSM	0.585 (0.001)	0.813 (0.001)
ICL	0.572 (0.004)	0.812 (0.001)
NEAD	0.566 (0.004)	0.812 (0.001)
kNN	0.565 (0.001)	0.808 (0.000)
LUNAR	0.560 (0.001)	0.808 (0.000)
MSM	0.550 (0.004)	0.813 (0.001)
GMM	0.528 (0.001)	0.798 (0.001)
DTE	0.526 (0.003)	0.805 (0.001)
LOF	0.524 (0.001)	0.786 (0.001)
SLAD	0.516 (0.002)	0.773 (0.001)
NRDE	0.486 (0.003)	0.741 (0.002)

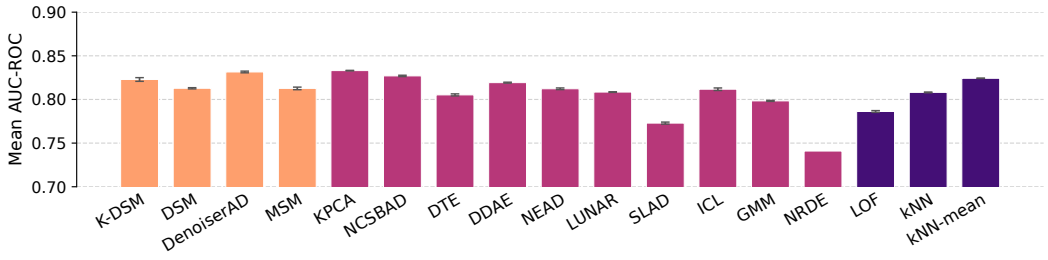


Figure 14: Mean AUC-ROC on 57 ADBench datasets over five seeds. Error bars are standard errors across seed-level macro means; NRDE's standard error is reported in Table 26. This is the AUC-ROC counterpart of the AUC-PR comparison in Figure 4.

Combining noise allocation with a denoiser. We apply the same kurtosis calibration to DenoiserAD (Hirth et al., 2026), retaining its residual MLP of width 2,048 with three blocks, 1,000 training epochs, one noisy copy per observation, and AdamW settings (learning rate 10^{-4}, weight decay 10^{-3}, batch size 128). The combined configuration replaces the scalar training noise of 0.2 with the K-DSM feature scales, using $\sigma_{\text{base}} = 0.5$, $c = 0.33$, and clipping to $[0.1, 3.0]$. The denoiser's preconditioning and reconstruction-loss weights also become feature-specific. With the data-scale

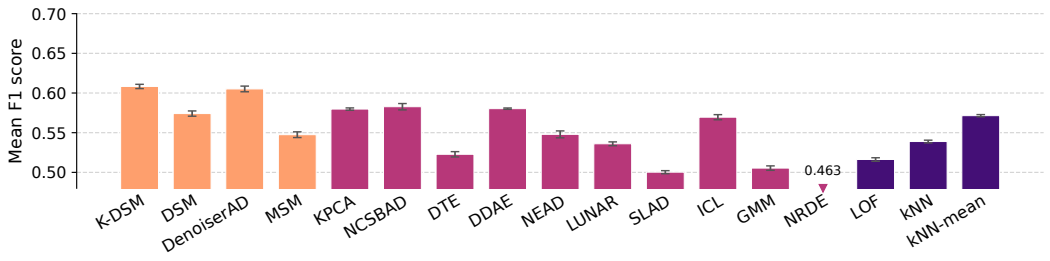


Figure 15: Mean F1 score on 57 ADBench datasets over five seeds. Error bars are standard errors across seed-level macro means; NRDE's standard error is reported in Table 31. Methods and ordering match the AUC-PR comparison in Figure 4.

parameter fixed to one, these are

$$c_{\text{skip},j} = \frac{1}{1+\sigma_j^2}, \qquad c_{\text{out},j} = \frac{\sigma_j}{\sqrt{1+\sigma_j^2}}, \qquad c_{\text{in},j} = \frac{1}{\sqrt{1+\sigma_j^2}}. \tag{48}$$

For $y = x + D\varepsilon$, where $D = \operatorname{diag}(\boldsymbol{\sigma})$, the model reconstructs $r_\theta(y) = c_{\text{skip}} \odot y + c_{\text{out}} \odot F_\theta(c_{\text{in}} \odot y)$ and minimizes $\mathbb{E}\sum_j (r_{\theta,j}(y) - x_j)^2/(d\, c_{\text{out},j}^2)$. At inference, it evaluates the clean input once and returns $\|D^{-1}(r_\theta(x) - x)\|_2$, replacing DenoiserAD's repeated noisy reconstruction scoring.

The combined model reaches 0.630 mean AUC-PR and 0.832 mean AUC-ROC, compared with 0.620 and 0.831 for DenoiserAD and 0.627 and 0.823 for K-DSM (Table 26). This shows that the calibration and clean-input residual score can also be used with a preconditioned denoiser. Several components change together, however: relative to DenoiserAD, the configuration changes corruption, preconditioning, loss weights, and scoring; relative to K-DSM, it also changes the architecture and training budget. These results therefore assess the combined configuration. The matched controls in Appendix K examine allocation and scoring separately within K-DSM's training procedure; they are a different experiment.

Dataset-level comparisons and interpretation. Table 27 reports K-DSM's wins, ties, and losses against every comparator in Table 26. A tie means an absolute difference of at most 0.001 between the dataset means (five seeds). These counts are descriptive, not significance tests. For example, K-DSM wins/ties/loses against DenoiserAD on 29/2/26 datasets in AUC-PR; the higher macro mean does not imply improvement on every dataset. Figure 16 reports paired Wilcoxon signed-rank comparisons of K-DSM with the other configurations whose complete five-seed source results are available. The per-detection thyroid visualization is in Appendix A; it shows the feature contributions behind individual alarms. Appendices J.2 and N.1 examine marginal heterogeneity and purely relational anomalies, respectively. Figure 9 additionally plots the measured kurtosis–scale pairs across every ADBench feature (Appendix G.4).

Paired Wilcoxon comparisons. We apply two-sided Wilcoxon signed-rank tests to K-DSM against 16 comparators on the primary AUC-PR metric (Demšar, 2006). For each method, we average constituent tasks within each dataset and seed, then average five seeds; the 57 paired dataset differences are the test observations. We use full-precision results, omit zero differences, assign average ranks to tied absolute differences, and use the tie-adjusted normal approximation with continuity correction. Holm correction controls family-wise error across the 16 comparisons at $\alpha = 0.05$. The test assumes independent dataset groups and symmetric distributions of paired differences.

The tests favor K-DSM over 12 comparators: DSM, MSM, DDAE, NCSBAD, ICL, NEAD, kNN, LUNAR, GMM, DTE, LOF, and SLAD (all adjusted $p \le 0.0152$). They do not establish a difference from DenoiserAD ($p_{\text{adj}} = 0.659$), K-DSM+DenoiserAD (0.659), KPCA (0.597), or kNN-mean (0.134). Non-rejection does not establish equivalence. The mean differences in Figure 16 describe effect magnitude; the test decisions depend on signed ranks, not on the macro means alone.

Table 27: K-DSM versus each comparator on the same 57 datasets, five seeds. W/T/L counts use a 0.001 tolerance on seed-averaged performance. Average PR ranks use exact ties and lower is better. The combined denoiser is included as a separate configuration.

Comparator	PR W/T/L	ROC W/T/L	PR rank
K-DSM+DenoiserAD	26/4/27	16/12/29	5.94
DenoiserAD	29/2/26	21/8/28	6.85
KPCA	29/3/25	23/6/28	8.10
DDAE	35/3/19	31/7/19	8.22
NCSBAD	35/4/18	29/8/20	7.89
kNN-mean	34/4/19	29/7/21	8.33
DSM	38/3/16	30/9/18	9.08
ICL	36/1/20	31/5/21	9.82
NEAD	35/2/20	37/4/16	9.02
kNN	39/2/16	33/6/18	10.40
LUNAR	40/2/15	32/6/19	10.89
MSM	38/1/18	30/6/21	10.15
GMM	42/5/10	37/4/16	11.42
DTE	42/4/11	40/3/14	11.96
LOF	43/1/13	37/7/13	10.99
SLAD	39/1/17	34/8/15	11.19
NRDE	48/2/7	46/4/7	14.18

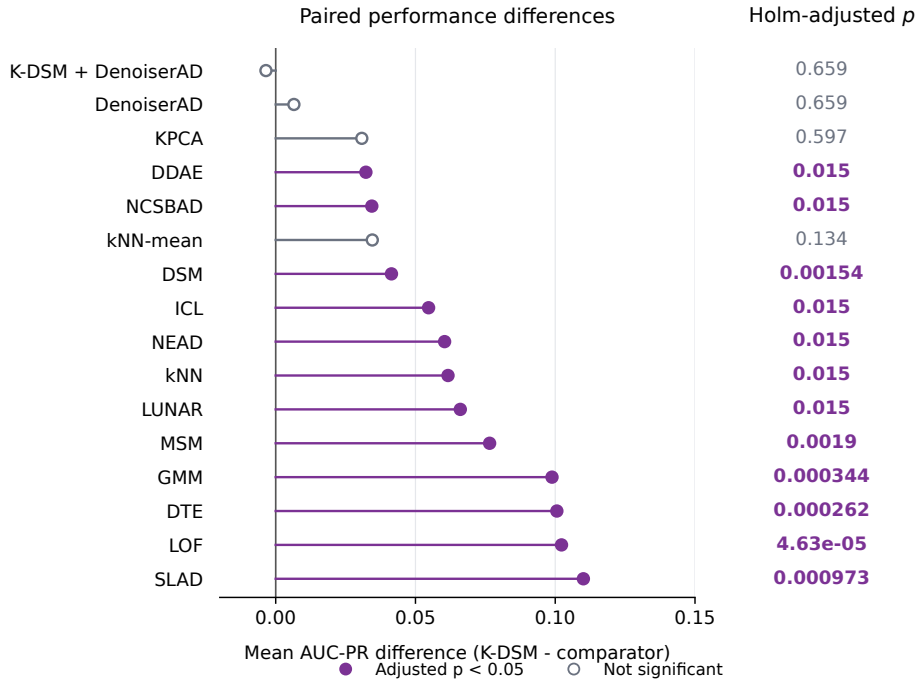


Figure 16: Paired AUC-PR comparisons under normal-only training on 57 ADBench datasets, after averaging five seeds within each dataset. Points show the mean difference (K-DSM minus comparator); positive values favor K-DSM. The right column gives two-sided Wilcoxon signed-rank p-values with Holm correction over 16 comparisons. Filled purple points indicate adjusted $p < 0.05$; open grey points indicate non-rejection.

N.1 SYNTHETIC ANOMALIES DEFINED ONLY BY FEATURE INTERACTIONS

We construct six-dimensional observations from three correlated Gaussian pairs. Anomalies are formed by independently deranging the response column within each pair, preserving each column's empirical multiset exactly while breaking its pairing with the other coordinate. The experiment checks this equality at runtime. Regime A retains Gaussian marginals. Regime B applies coordinate-wise monotone maps to obtain uniform, Laplace, Student-t_5, and lognormal marginals while preserving the normal copula before derangement; the reported per-feature scales span approximately $[0.34, 0.62]$. Both regimes use balanced test sets and ten seeds.

Table 28: Detection of anomalies that preserve marginal values but break feature dependencies. Results are ten-seed means on balanced synthetic test sets in two marginal regimes. Only aggregate means are available for this experiment; no uncertainty estimates or significance tests are reported.

Method	A: ROC	A: PR	B: ROC	B: PR
HBOS	0.541	0.497	0.526	0.497
ECOD	0.531	0.488	0.531	0.488
kNN	0.927	0.935	0.903	0.896
DSM, $\sigma = 0.5$	0.967	0.976	0.965	0.975
DSM, $\sigma = \mathrm{mean}_j\, \sigma_j$	0.972	0.980	0.968	0.977
K-DSM, per-feature σ_j	0.972	0.980	0.967	0.976

The marginal baselines remain close to chance, while the DSM variants reach approximately 0.97 AUC-ROC. K-DSM and DSM at the matched average scale have similar mean performance in both regimes. This is a controlled mechanism test showing that diagonal perturbations do not prevent the network from learning joint anomalies; it is not evidence of a benchmark-wide advantage on relational anomalies.

Table 29: Average AUC-PR for the semi-supervised setting on ADBench, column panel 1/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels.

Dataset	K-DSM	DSM	MSM	DTE	DDAE	DenoiserAD	NCSBAD	NRDE	K-DSM+ DenoiserAD
ALOI	**0.071 (0.001)**	0.062 (0.000)	0.061 (0.001)	0.058 (0.000)	0.061 (0.000)	0.062 (0.000)	0.062 (0.001)	0.060 (0.000)	0.069 (0.000)
annthyroid	0.517 (0.018)	0.406 (0.033)	0.354 (0.007)	**0.828 (0.001)**	0.686 (0.008)	0.521 (0.031)	0.318 (0.012)	0.331 (0.009)	0.604 (0.010)
backdoor	0.887 (0.005)	**0.893 (0.004)**	0.889 (0.005)	0.606 (0.013)	0.749 (0.012)	0.870 (0.009)	0.884 (0.008)	0.788 (0.085)	0.884 (0.007)
breastw	0.991 (0.001)	0.987 (0.002)	0.824 (0.028)	0.895 (0.008)	**0.995 (0.001)**	0.990 (0.002)	0.982 (0.002)	0.955 (0.005)	0.994 (0.001)
campaign	0.357 (0.007)	0.371 (0.008)	0.432 (0.006)	0.486 (0.008)	0.517 (0.006)	0.325 (0.005)	0.516 (0.002)	0.409 (0.010)	0.415 (0.002)
cardio	0.705 (0.008)	0.724 (0.003)	0.259 (0.005)	0.704 (0.003)	0.726 (0.004)	0.675 (0.007)	0.712 (0.005)	0.643 (0.025)	0.688 (0.008)
Cardiotocography	0.544 (0.002)	0.531 (0.007)	0.338 (0.008)	0.547 (0.009)	0.530 (0.005)	0.510 (0.004)	0.523 (0.001)	0.470 (0.032)	0.523 (0.003)
celeba	0.085 (0.024)	0.119 (0.007)	0.083 (0.002)	0.139 (0.006)	0.048 (0.003)	0.050 (0.002)	0.123 (0.004)	0.143 (0.002)	0.090 (0.008)
census	0.197 (0.003)	0.189 (0.003)	0.185 (0.008)	0.184 (0.003)	0.204 (0.005)	0.224 (0.004)	0.182 (0.001)	0.175 (0.008)	0.301 (0.008)
cover	0.882 (0.020)	**0.911 (0.008)**	0.849 (0.018)	0.676 (0.020)	0.705 (0.025)	0.795 (0.020)	0.857 (0.024)	0.079 (0.000)	0.852 (0.016)
donors	0.991 (0.003)	0.959 (0.012)	**0.995 (0.001)**	0.733 (0.018)	0.948 (0.007)	0.984 (0.004)	0.988 (0.002)	0.693 (0.006)	0.988 (0.004)
fault	0.598 (0.002)	0.609 (0.003)	0.581 (0.008)	0.637 (0.003)	0.609 (0.003)	0.622 (0.005)	0.600 (0.003)	0.593 (0.029)	0.562 (0.001)
fraud	0.624 (0.060)	0.561 (0.040)	0.515 (0.056)	0.625 (0.020)	0.532 (0.032)	0.517 (0.051)	0.147 (0.032)	0.653 (0.068)	0.496 (0.036)
glass	0.780 (0.058)	0.706 (0.032)	0.849 (0.041)	0.395 (0.025)	0.628 (0.045)	0.838 (0.042)	0.767 (0.038)	0.400 (0.059)	0.807 (0.046)
Hepatitis	**1.000 (0.000)**	0.997 (0.003)	0.997 (0.003)	0.970 (0.023)	0.997 (0.003)	0.997 (0.003)	0.997 (0.003)	0.560 (0.026)	0.999 (0.001)
http	**1.000 (0.000)**	0.895 (0.012)	0.966 (0.022)	0.578 (0.050)	0.951 (0.042)	0.783 (0.169)	0.621 (0.150)	0.911 (0.001)	0.996 (0.002)
InternetAds	0.767 (0.002)	0.662 (0.005)	0.691 (0.013)	0.557 (0.027)	0.731 (0.006)	**0.772 (0.002)**	0.768 (0.001)	0.412 (0.019)	0.743 (0.005)
Ionosphere	0.991 (0.001)	0.986 (0.001)	0.975 (0.005)	0.966 (0.006)	0.985 (0.002)	**0.993 (0.001)**	0.990 (0.001)	0.972 (0.013)	0.991 (0.001)
landsat	0.497 (0.003)	0.484 (0.007)	0.422 (0.006)	0.374 (0.011)	0.519 (0.008)	0.475 (0.004)	0.432 (0.005)	0.265 (0.005)	0.516 (0.007)
letter	0.093 (0.000)	0.095 (0.000)	**0.119 (0.009)**	0.090 (0.001)	0.095 (0.001)	0.103 (0.004)	0.099 (0.000)	0.091 (0.003)	0.091 (0.000)
Lymphography	0.996 (0.003)	**1.000 (0.000)**	0.900 (0.062)	0.890 (0.041)	**1.000 (0.000)**	**1.000 (0.000)**	0.993 (0.004)	0.950 (0.014)	0.994 (0.004)
magic.gamma	**0.904 (0.001)**	0.875 (0.002)	0.893 (0.002)	0.895 (0.002)	0.871 (0.002)	0.887 (0.000)	0.877 (0.005)	0.826 (0.005)	0.884 (0.001)
mammography	**0.496 (0.010)**	0.349 (0.019)	0.360 (0.016)	0.399 (0.024)	0.341 (0.017)	0.406 (0.010)	0.287 (0.015)	0.055 (0.019)	0.475 (0.005)
mnist	**0.823 (0.003)**	0.771 (0.008)	0.654 (0.004)	0.587 (0.005)	0.749 (0.006)	0.751 (0.003)	0.725 (0.004)	0.662 (0.035)	0.787 (0.003)
musk	**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 (0.033)**	**1.000 (0.000)**
optdigits	0.648 (0.018)	0.481 (0.020)	0.537 (0.081)	0.179 (0.012)	0.397 (0.017)	0.714 (0.019)	0.501 (0.012)	0.418 (0.030)	**0.722 (0.038)**
PageBlocks	0.618 (0.007)	0.551 (0.002)	0.594 (0.011)	0.675 (0.006)	0.678 (0.006)	0.641 (0.002)	0.570 (0.006)	0.704 (0.006)	0.683 (0.006)
pendigits	0.966 (0.008)	0.924 (0.014)	0.848 (0.028)	0.518 (0.032)	0.943 (0.003)	0.934 (0.014)	0.926 (0.013)	0.744 (0.042)	0.914 (0.019)
Pima	0.795 (0.008)	0.759 (0.010)	0.668 (0.014)	0.675 (0.011)	0.768 (0.014)	0.813 (0.005)	0.751 (0.004)	0.783 (0.016)	0.813 (0.006)
satellite	0.849 (0.002)	0.844 (0.002)	0.813 (0.008)	0.846 (0.002)	0.858 (0.002)	0.852 (0.001)	0.855 (0.001)	0.827 (0.007)	0.857 (0.001)
satimage-2	0.931 (0.002)	0.908 (0.004)	0.162 (0.041)	0.716 (0.024)	0.917 (0.007)	0.889 (0.009)	0.932 (0.003)	0.925 (0.106)	0.935 (0.000)
shuttle	0.989 (0.002)	0.993 (0.002)	**1.000 (0.000)**	0.942 (0.000)	0.995 (0.001)	0.998 (0.001)	0.944 (0.048)	0.973 (0.004)	0.999 (0.000)
skin	0.854 (0.013)	0.697 (0.016)	0.959 (0.007)	0.693 (0.003)	0.792 (0.009)	0.954 (0.003)	0.740 (0.012)	0.708 (0.032)	0.832 (0.010)
smtp	0.622 (0.048)	0.360 (0.032)	0.255 (0.083)	0.499 (0.028)	0.525 (0.042)	**0.630 (0.064)**	0.528 (0.072)	0.005 (0.045)	0.592 (0.063)
SpamBase	0.836 (0.001)	0.847 (0.001)	**0.860 (0.003)**	0.838 (0.002)	0.851 (0.001)	0.831 (0.001)	0.844 (0.001)	0.823 (0.004)	0.833 (0.001)
speech	0.030 (0.001)	0.032 (0.002)	**0.041 (0.003)**	0.029 (0.001)	0.030 (0.001)	0.037 (0.002)	0.034 (0.002)	0.026 (0.001)	0.036 (0.001)
Stamps	0.889 (0.031)	0.854 (0.010)	0.827 (0.036)	0.598 (0.040)	0.828 (0.036)	0.916 (0.027)	0.862 (0.028)	0.402 (0.039)	0.917 (0.027)
thyroid	0.727 (0.015)	0.484 (0.030)	0.126 (0.014)	**0.832 (0.004)**	0.756 (0.034)	0.776 (0.015)	0.573 (0.044)	0.822 (0.008)	0.774 (0.006)
vertebral	0.718 (0.032)	0.520 (0.013)	**0.763 (0.013)**	0.344 (0.023)	0.263 (0.016)	0.684 (0.021)	0.578 (0.033)	0.126 (0.010)	0.683 (0.017)
vowels	0.418 (0.008)	0.436 (0.023)	0.337 (0.049)	0.383 (0.026)	0.349 (0.011)	0.341 (0.029)	**0.450 (0.005)**	0.048 (0.002)	0.369 (0.024)
Waveform	0.080 (0.002)	0.109 (0.003)	0.093 (0.007)	0.103 (0.005)	0.292 (0.015)	0.134 (0.008)	0.088 (0.002)	0.098 (0.018)	0.129 (0.011)
WBC	0.986 (0.008)	0.941 (0.033)	0.803 (0.044)	0.319 (0.014)	0.970 (0.017)	0.979 (0.013)	0.914 (0.057)	0.691 (0.043)	0.990 (0.005)
WDBC	**0.980 (0.018)**	0.953 (0.023)	0.662 (0.088)	0.747 (0.060)	0.925 (0.043)	0.928 (0.026)	0.968 (0.029)	0.837 (0.046)	0.964 (0.025)
Wilt	0.090 (0.001)	0.140 (0.003)	0.446 (0.028)	0.256 (0.006)	0.104 (0.001)	0.237 (0.003)	**0.483 (0.006)**	0.224 (0.003)	0.088 (0.000)
wine	0.996 (0.004)	0.994 (0.006)	0.998 (0.002)	0.995 (0.002)	0.999 (0.001)	0.990 (0.010)	0.996 (0.004)	0.987 (0.080)	0.992 (0.008)
WPBC	0.880 (0.024)	0.891 (0.026)	**0.899 (0.021)**	0.603 (0.020)	0.805 (0.020)	0.872 (0.029)	0.886 (0.026)	0.610 (0.018)	0.874 (0.026)
yeast	0.488 (0.003)	0.508 (0.003)	**0.546 (0.007)**	0.499 (0.004)	0.487 (0.003)	0.491 (0.002)	0.522 (0.002)	0.487 (0.008)	0.479 (0.003)
20news	0.217 (0.007)	0.207 (0.006)	0.208 (0.011)	0.163 (0.006)	0.192 (0.008)	0.220 (0.005)	0.213 (0.006)	0.128 (0.005)	**0.235 (0.005)**
agnews	0.278 (0.003)	0.233 (0.001)	0.279 (0.010)	0.173 (0.005)	0.228 (0.002)	**0.350 (0.002)**	0.290 (0.000)	0.128 (0.001)	0.336 (0.003)
amazon	0.119 (0.001)	0.118 (0.000)	0.107 (0.003)	0.105 (0.002)	0.121 (0.000)	0.125 (0.001)	0.122 (0.000)	0.111 (0.006)	**0.127 (0.000)**
CIFAR10	0.232 (0.000)	0.196 (0.001)	0.144 (0.001)	0.198 (0.002)	0.208 (0.000)	0.211 (0.000)	0.212 (0.000)	0.153 (0.002)	**0.236 (0.000)**
FashionMNIST	0.702 (0.000)	0.428 (0.008)	0.448 (0.003)	0.557 (0.001)	0.619 (0.000)	0.666 (0.001)	0.638 (0.000)	0.385 (0.004)	**0.710 (0.000)**
imdb	0.088 (0.000)	0.089 (0.000)	0.097 (0.002)	0.087 (0.003)	0.089 (0.000)	0.091 (0.000)	0.089 (0.000)	0.086 (0.003)	0.092 (0.000)
MNIST-C	0.656 (0.001)	0.526 (0.009)	0.471 (0.004)	0.472 (0.001)	0.507 (0.000)	0.653 (0.001)	0.571 (0.000)	0.393 (0.004)	**0.682 (0.000)**
MVTec	0.875 (0.004)	0.872 (0.004)	0.884 (0.005)	0.838 (0.006)	0.882 (0.005)	0.888 (0.005)	0.883 (0.005)	0.714 (0.004)	**0.906 (0.006)**
SVHN	**0.185 (0.000)**	0.149 (0.001)	0.144 (0.001)	0.156 (0.001)	0.160 (0.000)	0.174 (0.000)	0.165 (0.000)	0.139 (0.002)	**0.185 (0.000)**
yelp	0.173 (0.002)	0.162 (0.000)	0.139 (0.003)	0.123 (0.006)	0.162 (0.002)	0.173 (0.001)	0.175 (0.000)	0.121 (0.012)	**0.179 (0.001)**
Overall	0.627 (0.003)	0.585 (0.001)	0.550 (0.004)	0.526 (0.003)	0.594 (0.001)	0.620 (0.005)	0.592 (0.005)	0.486 (0.003)	**0.630 (0.001)**

Table 29: Average AUC-PR for the semi-supervised setting on ADBench, column panel 2/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels.

Dataset	GMM	KPCA	LUNAR	NEAD	SLAD	ICL	kNN-mean	kNN	LOF
ALOI	0.065 (0.000)	0.060 (0.000)	0.061 (0.000)	0.062 (0.000)	0.060 (0.000)	0.055 (0.001)	0.060 (0.000)	0.060 (0.000)	0.065 (0.000)
annthyroid	0.560 (0.000)	0.658 (0.002)	0.680 (0.000)	0.674 (0.010)	0.707 (0.000)	0.527 (0.013)	0.683 (0.000)	0.681 (0.000)	0.535 (0.000)
backdoor	0.869 (0.007)	0.473 (0.011)	0.238 (0.003)	0.826 (0.008)	0.048 (0.000)	0.880 (0.005)	0.346 (0.005)	0.285 (0.004)	0.537 (0.012)
breastw	0.985 (0.003)	0.994 (0.001)	0.991 (0.001)	0.964 (0.005)	0.994 (0.001)	0.983 (0.002)	0.992 (0.001)	0.990 (0.001)	0.828 (0.032)
campaign	0.514 (0.000)	0.586 (0.002)	**0.595 (0.000)**	0.342 (0.004)	0.484 (0.000)	0.466 (0.004)	0.576 (0.000)	0.587 (0.000)	0.402 (0.000)
cardio	0.727 (0.000)	0.776 (0.000)	**0.784 (0.000)**	0.522 (0.018)	0.692 (0.000)	0.631 (0.021)	0.770 (0.000)	0.772 (0.000)	0.702 (0.000)
Cardiotocography	0.558 (0.000)	0.555 (0.000)	0.571 (0.000)	0.516 (0.004)	0.494 (0.000)	0.489 (0.011)	0.548 (0.000)	0.561 (0.000)	**0.573 (0.000)**
celeba	**0.165 (0.002)**	0.109 (0.005)	0.075 (0.002)	0.077 (0.002)	0.094 (0.003)	0.101 (0.002)	0.062 (0.002)	0.059 (0.002)	0.036 (0.001)
census	0.196 (0.002)	**0.370 (0.009)**	0.347 (0.006)	0.181 (0.005)	0.132 (0.014)	0.258 (0.005)	0.326 (0.007)	0.334 (0.008)	0.136 (0.002)
cover	0.189 (0.003)	0.316 (0.012)	0.516 (0.016)	0.894 (0.014)	0.062 (0.021)	0.418 (0.072)	0.638 (0.016)	0.552 (0.017)	0.829 (0.010)
donors	0.431 (0.003)	0.764 (0.009)	0.748 (0.008)	0.750 (0.017)	0.460 (0.064)	0.793 (0.034)	0.846 (0.006)	0.799 (0.008)	0.636 (0.007)
fault	0.631 (0.000)	0.643 (0.000)	0.614 (0.000)	0.628 (0.017)	**0.668 (0.000)**	0.644 (0.002)	0.622 (0.000)	0.620 (0.000)	0.504 (0.000)
fraud	0.602 (0.039)	0.422 (0.043)	0.431 (0.020)	**0.701 (0.021)**	0.463 (0.026)	0.585 (0.050)	0.408 (0.033)	0.423 (0.028)	0.598 (0.017)
glass	0.289 (0.026)	0.823 (0.044)	0.372 (0.025)	0.651 (0.014)	0.415 (0.023)	**0.857 (0.047)**	0.601 (0.035)	0.441 (0.032)	0.423 (0.036)
Hepatitis	0.884 (0.013)	0.996 (0.004)	0.917 (0.011)	0.992 (0.005)	0.998 (0.002)	0.990 (0.010)	0.995 (0.004)	0.911 (0.022)	0.458 (0.048)
http	0.906 (0.006)	0.973 (0.014)	0.985 (0.015)	0.767 (0.122)	0.886 (0.034)	0.962 (0.009)	**1.000 (0.000)**	**1.000 (0.000)**	0.971 (0.017)
InternetAds	0.731 (0.000)	0.732 (0.000)	0.689 (0.000)	0.769 (0.002)	0.593 (0.000)	0.596 (0.005)	0.738 (0.000)	0.705 (0.000)	0.504 (0.000)
Ionosphere	0.978 (0.003)	**0.993 (0.001)**	0.983 (0.002)	0.980 (0.002)	0.985 (0.003)	0.989 (0.002)	0.987 (0.002)	0.979 (0.003)	0.945 (0.007)
landsat	0.317 (0.000)	0.474 (0.003)	0.509 (0.000)	0.346 (0.002)	0.446 (0.000)	0.535 (0.005)	0.563 (0.000)	0.548 (0.000)	**0.614 (0.000)**
letter	0.087 (0.000)	0.089 (0.000)	0.086 (0.000)	0.100 (0.004)	0.091 (0.000)	0.099 (0.003)	0.088 (0.000)	0.087 (0.000)	0.113 (0.000)
Lymphography	0.995 (0.005)	**1.000 (0.000)**	0.994 (0.004)	**1.000 (0.000)**	0.999 (0.001)	**1.000 (0.000)**	**1.000 (0.000)**	0.986 (0.005)	0.865 (0.021)
magic.gamma	0.833 (0.000)	0.848 (0.000)	0.854 (0.000)	0.894 (0.001)	0.774 (0.000)	0.841 (0.002)	0.865 (0.000)	0.859 (0.000)	0.864 (0.000)
mammography	0.410 (0.000)	0.439 (0.002)	0.409 (0.000)	0.405 (0.006)	0.177 (0.000)	0.233 (0.005)	0.425 (0.000)	0.413 (0.000)	0.341 (0.000)
mnist	0.699 (0.000)	0.732 (0.003)	0.717 (0.000)	0.662 (0.011)	0.678 (0.000)	0.555 (0.006)	0.743 (0.000)	0.727 (0.000)	0.710 (0.000)
musk	0.985 (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)**	**1.000 (0.000)**
optdigits	0.134 (0.000)	0.229 (0.002)	0.264 (0.000)	0.190 (0.021)	0.334 (0.000)	0.286 (0.021)	0.352 (0.000)	0.300 (0.000)	0.436 (0.000)
PageBlocks	0.628 (0.000)	0.647 (0.003)	0.678 (0.000)	0.691 (0.002)	0.640 (0.000)	0.642 (0.017)	0.674 (0.000)	0.676 (0.000)	**0.711 (0.000)**
pendigits	0.159 (0.000)	0.918 (0.003)	0.964 (0.000)	0.776 (0.027)	0.353 (0.000)	0.724 (0.057)	**0.978 (0.000)**	0.970 (0.000)	0.786 (0.000)
Pima	0.721 (0.012)	**0.839 (0.007)**	0.759 (0.012)	0.739 (0.008)	0.624 (0.023)	0.765 (0.015)	0.804 (0.008)	0.757 (0.015)	0.674 (0.016)
satellite	0.846 (0.000)	0.844 (0.000)	0.853 (0.000)	0.829 (0.002)	**0.887 (0.000)**	0.886 (0.003)	0.862 (0.000)	0.860 (0.000)	0.859 (0.000)
satimage-2	0.851 (0.000)	0.966 (0.001)	**0.971 (0.000)**	0.526 (0.047)	0.956 (0.000)	0.966 (0.004)	0.962 (0.000)	0.967 (0.000)	0.885 (0.000)
shuttle	0.968 (0.000)	0.975 (0.001)	0.974 (0.000)	0.997 (0.000)	0.981 (0.000)	0.999 (0.000)	0.984 (0.000)	0.979 (0.000)	0.997 (0.000)
skin	0.645 (0.002)	0.883 (0.006)	0.967 (0.002)	0.959 (0.004)	0.789 (0.037)	0.657 (0.047)	**0.989 (0.001)**	0.983 (0.002)	0.618 (0.009)
smtp	0.379 (0.038)	0.567 (0.042)	0.502 (0.029)	0.225 (0.051)	0.500 (0.028)	0.440 (0.076)	0.502 (0.029)	0.503 (0.029)	0.492 (0.023)
SpamBase	0.813 (0.000)	0.841 (0.000)	0.827 (0.000)	0.850 (0.002)	0.856 (0.000)	0.856 (0.004)	0.837 (0.000)	0.831 (0.000)	0.727 (0.000)
speech	0.030 (0.000)	0.031 (0.000)	0.029 (0.000)	0.030 (0.001)	0.033 (0.000)	0.037 (0.003)	0.032 (0.000)	0.028 (0.000)	0.032 (0.000)
Stamps	0.640 (0.036)	**0.933 (0.028)**	0.708 (0.028)	0.861 (0.031)	0.496 (0.047)	0.819 (0.016)	0.825 (0.018)	0.733 (0.039)	0.665 (0.030)
thyroid	0.802 (0.000)	0.793 (0.000)	0.813 (0.000)	0.375 (0.020)	0.740 (0.000)	0.535 (0.021)	0.809 (0.000)	0.809 (0.000)	0.606 (0.000)
vertebral	0.209 (0.009)	0.672 (0.030)	0.251 (0.014)	0.381 (0.027)	0.198 (0.020)	0.480 (0.043)	0.441 (0.028)	0.251 (0.016)	0.327 (0.018)
vowels	0.416 (0.000)	0.302 (0.000)	0.297 (0.000)	0.421 (0.009)	0.387 (0.000)	0.281 (0.058)	0.323 (0.000)	0.302 (0.000)	0.331 (0.000)
Waveform	0.080 (0.000)	0.205 (0.000)	0.264 (0.000)	0.154 (0.006)	0.053 (0.000)	0.255 (0.017)	0.279 (0.000)	0.270 (0.000)	**0.307 (0.000)**
WBC	0.854 (0.032)	**0.992 (0.007)**	0.914 (0.024)	0.839 (0.048)	0.978 (0.009)	**0.992 (0.007)**	0.966 (0.011)	0.895 (0.028)	0.251 (0.017)
WDBC	0.915 (0.045)	0.948 (0.023)	0.861 (0.029)	0.922 (0.030)	0.882 (0.030)	0.881 (0.038)	0.924 (0.034)	0.853 (0.027)	0.955 (0.016)
Wilt	0.276 (0.000)	0.167 (0.001)	0.113 (0.000)	0.268 (0.005)	0.121 (0.000)	0.344 (0.040)	0.134 (0.000)	0.122 (0.000)	0.157 (0.000)
wine	0.963 (0.011)	0.997 (0.003)	0.960 (0.006)	**1.000 (0.000)**	**1.000 (0.000)**	0.980 (0.019)	0.994 (0.004)	0.952 (0.008)	0.910 (0.009)
WPBC	0.486 (0.009)	0.884 (0.028)	0.517 (0.011)	0.713 (0.009)	0.871 (0.029)	0.846 (0.025)	0.753 (0.010)	0.471 (0.013)	0.420 (0.010)
yeast	0.468 (0.000)	0.479 (0.000)	0.475 (0.000)	0.511 (0.001)	0.507 (0.000)	0.486 (0.003)	0.483 (0.000)	0.483 (0.000)	0.489 (0.000)
20news	0.195 (0.007)	0.175 (0.007)	0.134 (0.004)	0.228 (0.002)	0.137 (0.002)	0.136 (0.001)	0.156 (0.006)	0.138 (0.005)	0.150 (0.003)
agnews	0.232 (0.000)	0.176 (0.000)	0.161 (0.000)	0.303 (0.002)	0.122 (0.000)	0.130 (0.001)	0.174 (0.000)	0.167 (0.000)	0.259 (0.000)
amazon	0.119 (0.000)	0.119 (0.000)	0.117 (0.000)	0.113 (0.002)	0.098 (0.000)	0.100 (0.000)	0.117 (0.000)	0.117 (0.000)	0.110 (0.000)
CIFAR10	0.212 (0.000)	0.201 (0.000)	0.196 (0.000)	0.212 (0.000)	0.200 (0.000)	0.191 (0.002)	0.199 (0.000)	0.196 (0.000)	0.222 (0.000)
FashionMNIST	0.616 (0.000)	0.592 (0.001)	0.588 (0.000)	0.619 (0.000)	0.594 (0.000)	0.647 (0.001)	0.598 (0.000)	0.591 (0.000)	0.636 (0.000)
imdb	0.089 (0.000)	0.090 (0.000)	0.090 (0.000)	0.093 (0.001)	0.098 (0.000)	**0.101 (0.000)**	0.090 (0.000)	0.089 (0.000)	0.090 (0.000)
MNIST-C	0.514 (0.000)	0.463 (0.000)	0.455 (0.000)	0.541 (0.001)	0.469 (0.000)	0.501 (0.001)	0.474 (0.000)	0.462 (0.000)	0.519 (0.000)
MVTec	0.884 (0.005)	0.888 (0.004)	0.766 (0.005)	0.885 (0.004)	0.879 (0.004)	0.889 (0.004)	0.829 (0.005)	0.754 (0.004)	0.754 (0.005)
SVHN	0.162 (0.000)	0.155 (0.000)	0.153 (0.000)	0.163 (0.000)	0.154 (0.000)	0.157 (0.001)	0.155 (0.000)	0.153 (0.000)	0.160 (0.000)
yelp	0.163 (0.000)	0.160 (0.001)	0.157 (0.000)	0.153 (0.002)	0.100 (0.000)	0.098 (0.000)	0.163 (0.000)	0.160 (0.000)	0.161 (0.000)
Overall	0.528 (0.001)	0.596 (0.002)	0.560 (0.001)	0.566 (0.004)	0.516 (0.002)	0.572 (0.004)	0.592 (0.002)	0.565 (0.001)	0.524 (0.001)

Table 30: Average AUC-ROC for the semi-supervised setting on ADBench, column panel 1/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels.

Dataset	K-DSM	DSM	MSM	DTE	DDAE	DenoiserAD	NCSBAD	NRDE	K-DSM+ DenoiserAD
ALOI	0.541 (0.000)	0.494 (0.001)	0.488 (0.001)	0.506 (0.002)	0.518 (0.000)	0.491 (0.001)	0.490 (0.001)	0.502 (0.001)	0.542 (0.000)
annthyroid	0.816 (0.012)	0.748 (0.026)	0.824 (0.004)	**0.975 (0.000)**	0.931 (0.004)	0.852 (0.012)	0.689 (0.005)	0.692 (0.019)	0.894 (0.004)
backdoor	0.949 (0.002)	0.972 (0.002)	0.967 (0.001)	0.902 (0.006)	0.955 (0.002)	0.968 (0.002)	**0.977 (0.001)**	0.926 (0.066)	0.950 (0.002)
breastw	0.992 (0.001)	0.990 (0.001)	0.900 (0.018)	0.938 (0.007)	**0.995 (0.001)**	0.992 (0.001)	0.987 (0.002)	0.981 (0.002)	0.994 (0.001)
campaign	0.706 (0.006)	0.703 (0.009)	0.748 (0.006)	0.791 (0.002)	0.810 (0.004)	0.646 (0.003)	0.772 (0.003)	0.707 (0.009)	0.731 (0.003)
cardio	0.874 (0.006)	0.895 (0.004)	0.623 (0.006)	0.878 (0.003)	0.871 (0.006)	0.857 (0.004)	0.889 (0.003)	0.857 (0.018)	0.846 (0.007)
Cardiotocography	0.595 (0.004)	0.611 (0.009)	0.448 (0.007)	0.620 (0.015)	0.562 (0.010)	0.552 (0.006)	0.587 (0.001)	0.513 (0.033)	0.571 (0.003)
celeba	0.617 (0.098)	0.743 (0.018)	0.714 (0.004)	**0.823 (0.007)**	0.528 (0.014)	0.543 (0.007)	0.783 (0.006)	0.813 (0.002)	0.699 (0.020)
census	0.700 (0.003)	0.686 (0.003)	0.663 (0.007)	0.701 (0.003)	0.716 (0.005)	0.719 (0.001)	0.687 (0.002)	0.674 (0.033)	**0.764 (0.002)**
cover	0.992 (0.002)	**0.997 (0.000)**	0.985 (0.002)	0.982 (0.001)	0.972 (0.004)	0.965 (0.005)	0.974 (0.004)	0.836 (0.022)	0.989 (0.003)
donors	**1.000 (0.000)**	0.994 (0.002)	**1.000 (0.000)**	0.983 (0.002)	0.996 (0.001)	0.997 (0.001)	0.999 (0.000)	0.971 (0.001)	0.999 (0.000)
fault	0.569 (0.002)	0.576 (0.003)	0.553 (0.002)	0.588 (0.005)	0.579 (0.003)	0.567 (0.005)	0.565 (0.003)	0.569 (0.027)	0.535 (0.003)
fraud	0.945 (0.007)	0.942 (0.006)	0.949 (0.007)	0.944 (0.007)	0.955 (0.007)	0.945 (0.007)	0.938 (0.002)	0.950 (0.010)	0.949 (0.007)
glass	0.969 (0.013)	0.938 (0.017)	0.991 (0.003)	0.918 (0.008)	0.964 (0.003)	0.991 (0.002)	0.959 (0.014)	0.811 (0.016)	0.975 (0.013)
Hepatitis	**1.000 (0.000)**	0.999 (0.001)	0.999 (0.001)	0.993 (0.005)	0.999 (0.001)	0.999 (0.001)	0.999 (0.001)	0.770 (0.015)	**1.000 (0.000)**
http	**1.000 (0.000)**	0.999 (0.000)	**1.000 (0.000)**	0.996 (0.001)	**1.000 (0.000)**	0.988 (0.012)	0.994 (0.003)	0.999 (0.000)	**1.000 (0.000)**
InternetAds	0.826 (0.001)	0.692 (0.011)	0.801 (0.004)	0.771 (0.010)	0.816 (0.002)	0.809 (0.001)	0.802 (0.001)	0.613 (0.004)	**0.842 (0.003)**
Ionosphere	0.989 (0.001)	0.983 (0.002)	0.977 (0.004)	0.946 (0.013)	0.981 (0.002)	**0.992 (0.001)**	0.988 (0.001)	0.966 (0.012)	0.990 (0.001)
landsat	0.605 (0.002)	0.610 (0.004)	0.617 (0.005)	0.521 (0.008)	0.667 (0.002)	0.653 (0.002)	0.594 (0.005)	0.329 (0.009)	0.634 (0.005)
letter	0.413 (0.002)	0.412 (0.001)	**0.481 (0.015)**	0.371 (0.007)	0.406 (0.001)	0.406 (0.001)	0.422 (0.001)	0.398 (0.020)	0.393 (0.003)
Lymphography	**1.000 (0.000)**	**1.000 (0.000)**	0.993 (0.004)	0.992 (0.001)	**1.000 (0.000)**	**1.000 (0.000)**	0.999 (0.000)	0.996 (0.001)	0.999 (0.000)
magic.gamma	**0.879 (0.002)**	0.849 (0.003)	0.870 (0.002)	0.873 (0.003)	0.839 (0.003)	0.860 (0.001)	0.842 (0.006)	0.772 (0.007)	0.862 (0.001)
mammography	0.883 (0.002)	0.859 (0.008)	0.857 (0.004)	0.865 (0.007)	0.844 (0.006)	0.843 (0.003)	0.830 (0.007)	0.572 (0.004)	0.877 (0.001)
mnist	**0.956 (0.002)**	0.936 (0.004)	0.880 (0.004)	0.892 (0.002)	0.943 (0.002)	0.914 (0.001)	0.923 (0.001)	0.907 (0.040)	0.952 (0.001)
musk	**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 (0.026)**	**1.000 (0.000)**
optdigits	0.979 (0.002)	0.965 (0.003)	0.947 (0.014)	0.857 (0.011)	0.960 (0.003)	**0.984 (0.001)**	0.965 (0.001)	0.932 (0.023)	**0.984 (0.003)**
PageBlocks	0.831 (0.004)	0.832 (0.003)	0.885 (0.001)	0.901 (0.004)	0.900 (0.002)	0.846 (0.002)	0.859 (0.002)	0.885 (0.008)	0.865 (0.003)
pendigits	**0.999 (0.000)**	0.997 (0.001)	0.994 (0.001)	0.981 (0.003)	0.998 (0.000)	0.997 (0.000)	0.997 (0.000)	0.989 (0.009)	0.997 (0.001)
Pima	0.816 (0.003)	0.753 (0.005)	0.693 (0.014)	0.699 (0.009)	0.772 (0.011)	0.833 (0.008)	0.753 (0.015)	0.749 (0.010)	0.837 (0.009)
satellite	0.798 (0.002)	0.799 (0.003)	0.784 (0.009)	0.786 (0.004)	0.818 (0.003)	0.818 (0.001)	0.811 (0.002)	0.750 (0.008)	0.814 (0.002)
satimage-2	0.993 (0.001)	0.990 (0.001)	0.883 (0.021)	0.994 (0.000)	0.994 (0.001)	0.987 (0.001)	0.993 (0.001)	0.994 (0.015)	0.993 (0.000)
shuttle	**1.000 (0.000)**	0.999 (0.000)	**1.000 (0.000)**	0.998 (0.000)	**1.000 (0.000)**	0.999 (0.000)	0.993 (0.005)	0.995 (0.000)	**1.000 (0.000)**
skin	0.954 (0.005)	0.889 (0.011)	0.984 (0.003)	0.918 (0.001)	0.946 (0.002)	0.979 (0.001)	0.913 (0.004)	0.923 (0.028)	0.951 (0.003)
smtp	0.848 (0.018)	0.923 (0.009)	0.943 (0.010)	**0.954 (0.007)**	0.917 (0.013)	0.934 (0.006)	0.874 (0.018)	0.516 (0.005)	0.852 (0.017)
SpamBase	0.833 (0.001)	0.841 (0.001)	**0.853 (0.002)**	0.831 (0.002)	0.848 (0.001)	0.835 (0.001)	0.841 (0.001)	0.826 (0.002)	0.836 (0.001)
speech	0.442 (0.006)	0.435 (0.004)	**0.560 (0.020)**	0.381 (0.010)	0.388 (0.002)	0.519 (0.003)	0.441 (0.006)	0.364 (0.011)	0.517 (0.001)
Stamps	0.985 (0.004)	0.971 (0.006)	0.969 (0.007)	0.922 (0.008)	0.977 (0.004)	0.989 (0.003)	0.980 (0.004)	0.827 (0.010)	0.989 (0.003)
thyroid	0.963 (0.003)	0.884 (0.015)	0.746 (0.037)	**0.989 (0.000)**	0.976 (0.005)	0.981 (0.002)	0.935 (0.007)	0.988 (0.003)	0.976 (0.001)
vertebral	0.893 (0.016)	0.776 (0.005)	0.916 (0.004)	0.677 (0.012)	0.550 (0.019)	0.920 (0.008)	0.802 (0.014)	0.269 (0.021)	0.911 (0.004)
vowels	0.855 (0.003)	**0.871 (0.008)**	0.864 (0.008)	0.869 (0.009)	0.854 (0.002)	0.859 (0.007)	0.863 (0.003)	0.337 (0.023)	0.837 (0.006)
Waveform	0.596 (0.007)	0.693 (0.006)	0.628 (0.016)	0.650 (0.005)	**0.781 (0.007)**	0.680 (0.011)	0.625 (0.004)	0.683 (0.013)	0.660 (0.009)
WBC	0.998 (0.001)	0.994 (0.003)	0.983 (0.004)	0.844 (0.027)	0.997 (0.002)	0.997 (0.002)	0.993 (0.005)	0.971 (0.003)	**0.999 (0.001)**
WDBC	**0.999 (0.001)**	0.998 (0.001)	0.977 (0.007)	0.989 (0.003)	0.997 (0.001)	0.997 (0.001)	**0.999 (0.001)**	0.988 (0.020)	0.998 (0.001)
Wilt	0.483 (0.003)	0.650 (0.007)	0.919 (0.006)	0.849 (0.004)	0.557 (0.004)	0.807 (0.002)	**0.941 (0.002)**	0.776 (0.015)	0.475 (0.003)
wine	0.999 (0.001)	0.999 (0.001)	**1.000 (0.000)**	0.999 (0.000)	**1.000 (0.000)**	0.999 (0.001)	0.999 (0.001)	0.998 (0.023)	0.999 (0.001)
WPBC	0.961 (0.009)	0.966 (0.008)	0.964 (0.008)	0.699 (0.020)	0.910 (0.008)	0.962 (0.009)	**0.967 (0.007)**	0.667 (0.019)	0.959 (0.008)
yeast	0.461 (0.005)	0.476 (0.007)	**0.526 (0.009)**	0.471 (0.007)	0.449 (0.004)	0.464 (0.002)	0.503 (0.001)	0.454 (0.017)	0.450 (0.005)
20news	0.688 (0.003)	0.688 (0.002)	0.682 (0.007)	0.640 (0.007)	0.669 (0.004)	**0.701 (0.002)**	0.697 (0.002)	0.536 (0.008)	0.700 (0.002)
agnews	0.763 (0.004)	0.760 (0.001)	0.712 (0.010)	0.676 (0.006)	0.758 (0.002)	0.781 (0.001)	**0.786 (0.000)**	0.554 (0.003)	0.783 (0.002)
amazon	0.604 (0.006)	0.607 (0.001)	0.536 (0.012)	0.540 (0.009)	0.618 (0.001)	0.623 (0.001)	0.621 (0.001)	0.567 (0.019)	**0.631 (0.001)**
CIFAR10	**0.721 (0.000)**	0.683 (0.001)	0.609 (0.002)	0.689 (0.003)	0.692 (0.000)	0.694 (0.000)	0.696 (0.000)	0.614 (0.005)	0.718 (0.001)
FashionMNIST	**0.937 (0.001)**	0.711 (0.006)	0.864 (0.001)	0.904 (0.001)	0.913 (0.000)	0.931 (0.000)	0.922 (0.000)	0.773 (0.003)	**0.937 (0.000)**
imdb	0.492 (0.001)	0.494 (0.001)	0.508 (0.007)	0.473 (0.015)	0.497 (0.003)	0.508 (0.001)	0.500 (0.001)	0.474 (0.016)	0.511 (0.001)
MNIST-C	0.914 (0.000)	0.837 (0.009)	0.868 (0.001)	0.856 (0.000)	0.867 (0.000)	0.913 (0.000)	0.894 (0.000)	0.745 (0.005)	**0.918 (0.000)**
MVTec	0.913 (0.003)	0.922 (0.002)	0.941 (0.001)	0.881 (0.004)	0.938 (0.002)	0.947 (0.002)	0.939 (0.001)	0.759 (0.004)	**0.951 (0.003)**
SVHN	**0.685 (0.000)**	0.609 (0.001)	0.626 (0.001)	0.630 (0.001)	0.630 (0.000)	0.657 (0.000)	0.641 (0.000)	0.593 (0.004)	0.682 (0.000)
yelp	0.685 (0.003)	0.678 (0.001)	0.629 (0.004)	0.580 (0.019)	0.682 (0.002)	0.703 (0.001)	0.701 (0.001)	0.609 (0.025)	**0.708 (0.002)**
Overall	0.823 (0.002)	0.813 (0.001)	0.813 (0.001)	0.805 (0.001)	0.819 (0.001)	0.831 (0.001)	0.827 (0.001)	0.741 (0.002)	0.832 (0.001)

Table 30: Average AUC-ROC for the semi-supervised setting on ADBench, column panel 2/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels.

Dataset	GMM	KPCA	LUNAR	NEAD	SLAD	ICL	kNN-mean	kNN	LOF
ALOI	**0.546 (0.000)**	0.509 (0.000)	0.513 (0.000)	0.522 (0.001)	0.512 (0.000)	0.474 (0.004)	0.507 (0.000)	0.510 (0.000)	0.488 (0.000)
annthyroid	0.849 (0.000)	0.922 (0.001)	0.927 (0.000)	0.939 (0.003)	0.934 (0.000)	0.843 (0.013)	0.930 (0.000)	0.928 (0.000)	0.886 (0.000)
backdoor	0.928 (0.003)	0.943 (0.003)	0.892 (0.003)	0.941 (0.003)	0.500 (0.000)	0.921 (0.003)	0.921 (0.002)	0.907 (0.002)	0.954 (0.001)
breastw	0.988 (0.002)	**0.995 (0.001)**	0.992 (0.001)	0.970 (0.003)	**0.995 (0.000)**	0.989 (0.001)	0.993 (0.001)	0.991 (0.001)	0.916 (0.016)
campaign	0.799 (0.000)	0.846 (0.001)	**0.853 (0.000)**	0.661 (0.005)	0.762 (0.000)	0.770 (0.003)	0.837 (0.000)	0.848 (0.000)	0.705 (0.000)
cardio	0.911 (0.000)	0.914 (0.000)	**0.925 (0.000)**	0.849 (0.008)	0.819 (0.000)	0.820 (0.015)	0.915 (0.000)	0.920 (0.000)	0.922 (0.000)
Cardiotocography	0.644 (0.000)	0.614 (0.000)	0.630 (0.000)	0.600 (0.002)	0.477 (0.000)	0.523 (0.011)	0.598 (0.000)	0.619 (0.000)	**0.645 (0.000)**
celeba	0.812 (0.002)	0.742 (0.006)	0.661 (0.003)	0.702 (0.009)	0.675 (0.006)	0.761 (0.005)	0.576 (0.007)	0.570 (0.007)	0.437 (0.003)
census	0.704 (0.001)	0.759 (0.003)	0.759 (0.002)	0.648 (0.005)	0.537 (0.037)	0.712 (0.003)	0.757 (0.002)	0.759 (0.002)	0.585 (0.005)
cover	0.948 (0.001)	0.960 (0.003)	0.974 (0.001)	0.996 (0.001)	0.737 (0.048)	0.937 (0.021)	0.979 (0.001)	0.975 (0.001)	0.992 (0.000)
donors	0.925 (0.001)	0.986 (0.001)	0.984 (0.001)	0.975 (0.002)	0.871 (0.031)	0.983 (0.003)	0.992 (0.000)	0.988 (0.001)	0.970 (0.001)
fault	0.611 (0.000)	0.586 (0.000)	0.587 (0.000)	0.576 (0.015)	**0.640 (0.000)**	0.619 (0.006)	0.583 (0.000)	0.587 (0.000)	0.474 (0.000)
fraud	**0.958 (0.003)**	**0.958 (0.003)**	0.956 (0.005)	0.942 (0.007)	0.944 (0.006)	0.950 (0.003)	0.957 (0.005)	0.956 (0.005)	0.945 (0.007)
glass	0.784 (0.006)	0.988 (0.003)	0.898 (0.006)	0.959 (0.006)	0.859 (0.027)	**0.992 (0.002)**	0.961 (0.002)	0.921 (0.005)	0.898 (0.010)
Hepatitis	0.949 (0.006)	0.999 (0.001)	0.969 (0.005)	0.998 (0.001)	0.999 (0.001)	0.999 (0.001)	0.999 (0.001)	0.968 (0.007)	0.693 (0.035)
http	0.999 (0.000)	**1.000 (0.000)**	**1.000 (0.000)**	0.996 (0.002)	0.999 (0.000)	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**
InternetAds	0.813 (0.000)	0.806 (0.000)	0.712 (0.000)	0.803 (0.001)	0.760 (0.000)	0.785 (0.001)	0.764 (0.000)	0.737 (0.000)	0.717 (0.000)
Ionosphere	0.969 (0.004)	**0.992 (0.001)**	0.979 (0.003)	0.971 (0.004)	0.982 (0.004)	0.989 (0.002)	0.985 (0.003)	0.973 (0.004)	0.943 (0.010)
landsat	0.452 (0.000)	0.612 (0.001)	0.665 (0.000)	0.490 (0.004)	0.647 (0.000)	0.643 (0.001)	**0.690 (0.000)**	0.683 (0.000)	0.666 (0.000)
letter	0.351 (0.000)	0.364 (0.000)	0.348 (0.000)	0.419 (0.015)	0.382 (0.000)	0.422 (0.013)	0.361 (0.000)	0.354 (0.000)	0.448 (0.000)
Lymphography	**1.000 (0.000)**	**1.000 (0.000)**	0.999 (0.000)	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	0.999 (0.000)	0.985 (0.004)
magic.gamma	0.801 (0.000)	0.820 (0.001)	0.829 (0.000)	0.871 (0.001)	0.720 (0.000)	0.793 (0.003)	0.839 (0.000)	0.833 (0.000)	0.834 (0.000)
mammography	0.881 (0.000)	**0.884 (0.001)**	0.877 (0.000)	0.872 (0.005)	0.736 (0.000)	0.779 (0.005)	0.877 (0.000)	0.876 (0.000)	0.855 (0.000)
mnist	0.921 (0.000)	0.938 (0.000)	0.936 (0.000)	0.843 (0.009)	0.896 (0.000)	0.800 (0.005)	0.942 (0.000)	0.938 (0.000)	0.929 (0.000)
musk	0.999 (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)**	**1.000 (0.000)**
optdigits	0.815 (0.000)	0.912 (0.001)	0.928 (0.000)	0.835 (0.025)	0.946 (0.000)	0.913 (0.012)	0.953 (0.000)	0.940 (0.000)	0.967 (0.000)
PageBlocks	0.885 (0.000)	0.890 (0.001)	0.898 (0.000)	0.905 (0.001)	0.869 (0.000)	0.890 (0.005)	0.892 (0.000)	0.897 (0.000)	**0.913 (0.000)**
pendigits	0.850 (0.000)	0.996 (0.000)	0.998 (0.000)	0.990 (0.003)	0.946 (0.000)	0.980 (0.006)	**0.999 (0.000)**	**0.999 (0.000)**	0.991 (0.000)
Pima	0.738 (0.006)	**0.865 (0.007)**	0.780 (0.007)	0.732 (0.006)	0.602 (0.020)	0.774 (0.020)	0.822 (0.009)	0.774 (0.010)	0.701 (0.010)
satellite	0.787 (0.000)	0.791 (0.000)	0.815 (0.000)	0.787 (0.002)	**0.875 (0.000)**	0.860 (0.006)	0.824 (0.000)	0.822 (0.000)	0.803 (0.000)
satimage-2	0.997 (0.000)	0.997 (0.000)	**0.998 (0.000)**	0.976 (0.003)	**0.998 (0.000)**	0.997 (0.001)	0.997 (0.000)	0.997 (0.000)	0.994 (0.000)
shuttle	0.994 (0.000)	0.999 (0.000)	0.999 (0.000)	**1.000 (0.000)**	0.999 (0.000)	**1.000 (0.000)**	0.999 (0.000)	0.999 (0.000)	**1.000 (0.000)**
skin	0.888 (0.001)	0.973 (0.001)	0.993 (0.000)	0.990 (0.001)	0.911 (0.012)	0.863 (0.029)	**0.997 (0.000)**	0.995 (0.000)	0.865 (0.008)
smtp	0.817 (0.026)	0.911 (0.014)	0.932 (0.011)	0.830 (0.035)	0.919 (0.008)	0.919 (0.007)	0.929 (0.013)	0.929 (0.012)	0.929 (0.015)
SpamBase	0.807 (0.000)	0.845 (0.000)	0.828 (0.000)	0.842 (0.002)	0.848 (0.000)	0.838 (0.002)	0.839 (0.000)	0.832 (0.000)	0.732 (0.000)
speech	0.378 (0.000)	0.408 (0.000)	0.375 (0.000)	0.385 (0.009)	0.440 (0.000)	0.512 (0.014)	0.414 (0.000)	0.364 (0.000)	0.375 (0.000)
Stamps	0.933 (0.008)	**0.992 (0.003)**	0.956 (0.004)	0.975 (0.007)	0.818 (0.029)	0.974 (0.002)	0.979 (0.002)	0.962 (0.006)	0.942 (0.007)
thyroid	0.982 (0.000)	0.987 (0.000)	0.987 (0.000)	0.932 (0.004)	0.952 (0.000)	0.955 (0.008)	0.986 (0.000)	0.987 (0.000)	0.927 (0.000)
vertebral	0.485 (0.013)	**0.937 (0.009)**	0.571 (0.010)	0.686 (0.012)	0.449 (0.041)	0.743 (0.029)	0.796 (0.011)	0.573 (0.019)	0.638 (0.008)
vowels	0.832 (0.000)	0.798 (0.000)	0.805 (0.000)	0.849 (0.002)	0.855 (0.000)	0.829 (0.022)	0.821 (0.000)	0.822 (0.000)	0.863 (0.000)
Waveform	0.585 (0.000)	0.774 (0.000)	0.762 (0.000)	0.710 (0.005)	0.489 (0.000)	0.725 (0.006)	0.745 (0.000)	0.752 (0.000)	0.760 (0.000)
WBC	0.985 (0.003)	**0.999 (0.001)**	0.992 (0.002)	0.983 (0.005)	0.997 (0.001)	**0.999 (0.001)**	0.996 (0.001)	0.989 (0.002)	0.793 (0.019)
WDBC	0.996 (0.002)	0.998 (0.001)	0.993 (0.001)	0.996 (0.001)	0.994 (0.001)	0.993 (0.002)	0.996 (0.001)	0.992 (0.001)	0.998 (0.001)
Wilt	0.852 (0.000)	0.748 (0.002)	0.602 (0.000)	0.858 (0.004)	0.616 (0.000)	0.846 (0.019)	0.670 (0.000)	0.637 (0.000)	0.688 (0.000)
wine	0.994 (0.002)	**1.000 (0.000)**	0.993 (0.001)	**1.000 (0.000)**	**1.000 (0.000)**	0.997 (0.002)	0.999 (0.001)	0.992 (0.001)	0.985 (0.002)
WPBC	0.617 (0.009)	0.963 (0.009)	0.696 (0.006)	0.737 (0.016)	0.955 (0.010)	0.943 (0.011)	0.893 (0.005)	0.643 (0.012)	0.581 (0.009)
yeast	0.440 (0.000)	0.437 (0.000)	0.443 (0.000)	0.474 (0.001)	0.488 (0.000)	0.463 (0.008)	0.447 (0.000)	0.447 (0.000)	0.458 (0.000)
20news	0.678 (0.002)	0.642 (0.002)	0.576 (0.004)	**0.701 (0.001)**	0.595 (0.003)	0.596 (0.004)	0.600 (0.003)	0.575 (0.004)	0.603 (0.004)
agnews	0.762 (0.000)	0.699 (0.000)	0.665 (0.000)	0.774 (0.002)	0.585 (0.000)	0.577 (0.001)	0.680 (0.000)	0.670 (0.000)	0.746 (0.000)
amazon	0.613 (0.000)	0.610 (0.001)	0.606 (0.000)	0.580 (0.009)	0.521 (0.000)	0.534 (0.001)	0.608 (0.000)	0.606 (0.000)	0.579 (0.000)
CIFAR10	0.702 (0.000)	0.685 (0.000)	0.675 (0.000)	0.697 (0.001)	0.666 (0.000)	0.658 (0.001)	0.678 (0.000)	0.675 (0.000)	0.703 (0.000)
FashionMNIST	0.919 (0.000)	0.901 (0.000)	0.897 (0.000)	0.912 (0.000)	0.900 (0.000)	0.908 (0.001)	0.901 (0.000)	0.899 (0.000)	0.916 (0.000)
imdb	0.494 (0.000)	0.503 (0.001)	0.502 (0.000)	0.506 (0.006)	0.513 (0.000)	**0.522 (0.001)**	0.504 (0.000)	0.501 (0.000)	0.496 (0.000)
MNIST-C	0.871 (0.000)	0.840 (0.000)	0.836 (0.000)	0.886 (0.000)	0.836 (0.000)	0.850 (0.001)	0.847 (0.000)	0.841 (0.000)	0.872 (0.000)
MVTec	0.944 (0.001)	0.945 (0.002)	0.828 (0.002)	0.936 (0.001)	0.932 (0.002)	0.943 (0.002)	0.897 (0.003)	0.813 (0.002)	0.802 (0.003)
SVHN	0.639 (0.000)	0.622 (0.000)	0.615 (0.000)	0.639 (0.001)	0.609 (0.000)	0.617 (0.001)	0.621 (0.000)	0.617 (0.000)	0.638 (0.000)
yelp	0.685 (0.000)	0.679 (0.001)	0.678 (0.000)	0.652 (0.005)	0.548 (0.000)	0.541 (0.001)	0.687 (0.000)	0.681 (0.000)	0.672 (0.000)
Overall	0.798 (0.001)	**0.833 (0.000)**	0.808 (0.000)	0.812 (0.001)	0.773 (0.001)	0.812 (0.001)	0.824 (0.000)	0.808 (0.000)	0.786 (0.001)

Table 31: Average F1 for the semi-supervised setting on ADBench, column panel 1/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels.

Dataset	K-DSM	DSM	MSM	DTE	DDAE	DenoiserAD	NCSBAD	NRDE	K-DSM+ DenoiserAD
ALOI	0.071 (0.002)	0.065 (0.001)	0.061 (0.001)	0.043 (0.000)	0.064 (0.002)	0.074 (0.002)	0.066 (0.002)	0.071 (0.001)	0.072 (0.001)
annthyroid	0.529 (0.020)	0.436 (0.025)	0.424 (0.010)	**0.773 (0.002)**	0.628 (0.015)	0.522 (0.029)	0.327 (0.008)	0.345 (0.013)	0.595 (0.011)
backdoor	0.859 (0.006)	**0.886 (0.004)**	0.884 (0.006)	0.788 (0.020)	0.865 (0.005)	0.879 (0.006)	0.883 (0.004)	0.779 (0.116)	0.859 (0.005)
breastw	0.962 (0.005)	0.959 (0.004)	0.855 (0.021)	0.884 (0.010)	0.969 (0.002)	0.969 (0.002)	0.955 (0.004)	0.955 (0.002)	0.968 (0.003)
campaign	0.362 (0.008)	0.397 (0.006)	0.478 (0.005)	0.522 (0.004)	0.529 (0.005)	0.349 (0.003)	0.546 (0.002)	0.377 (0.010)	0.436 (0.004)
cardio	0.645 (0.004)	0.650 (0.004)	0.226 (0.013)	0.580 (0.002)	0.620 (0.001)	0.613 (0.003)	0.632 (0.004)	**0.670 (0.030)**	0.617 (0.004)
Cardiotocography	0.461 (0.006)	0.453 (0.011)	0.304 (0.009)	0.407 (0.015)	0.368 (0.009)	0.371 (0.007)	0.427 (0.003)	0.361 (0.024)	0.420 (0.010)
celeba	0.103 (0.038)	0.183 (0.012)	0.091 (0.007)	0.170 (0.006)	0.049 (0.007)	0.057 (0.003)	0.162 (0.005)	0.179 (0.004)	0.116 (0.013)
census	0.219 (0.006)	0.205 (0.004)	0.206 (0.017)	0.196 (0.004)	0.213 (0.009)	0.233 (0.002)	0.185 (0.003)	0.170 (0.013)	0.335 (0.006)
cover	0.863 (0.016)	0.874 (0.005)	0.826 (0.011)	0.734 (0.014)	0.728 (0.015)	0.796 (0.012)	0.851 (0.009)	0.108 (0.001)	0.840 (0.012)
donors	0.984 (0.004)	0.904 (0.021)	0.980 (0.005)	0.852 (0.015)	0.920 (0.008)	0.963 (0.008)	0.973 (0.008)	0.734 (0.006)	**0.991 (0.001)**
fault	0.559 (0.002)	0.554 (0.001)	0.547 (0.001)	0.551 (0.005)	0.555 (0.005)	0.539 (0.005)	0.550 (0.004)	0.559 (0.020)	0.527 (0.003)
fraud	0.659 (0.056)	0.595 (0.051)	0.561 (0.052)	0.684 (0.035)	0.572 (0.019)	0.569 (0.052)	0.190 (0.035)	0.620 (0.065)	0.535 (0.031)
glass	0.758 (0.055)	0.675 (0.030)	0.838 (0.048)	0.350 (0.027)	0.554 (0.036)	0.857 (0.031)	0.772 (0.039)	0.240 (0.058)	0.756 (0.046)
Hepatitis	**1.000 (0.000)**	0.996 (0.004)	0.996 (0.004)	0.957 (0.018)	0.995 (0.003)	0.996 (0.004)	0.996 (0.004)	0.537 (0.030)	0.996 (0.004)
http	0.999 (0.001)	0.925 (0.005)	0.974 (0.019)	0.392 (0.082)	0.980 (0.017)	0.778 (0.155)	0.665 (0.134)	0.929 (0.005)	0.992 (0.004)
InternetAds	0.671 (0.003)	0.549 (0.003)	0.661 (0.008)	0.646 (0.017)	**0.695 (0.003)**	0.684 (0.002)	0.676 (0.002)	0.421 (0.013)	0.642 (0.005)
Ionosphere	0.936 (0.006)	0.929 (0.006)	0.921 (0.005)	0.899 (0.007)	0.924 (0.007)	0.943 (0.007)	0.931 (0.005)	0.914 (0.012)	0.935 (0.007)
landsat	0.486 (0.003)	0.492 (0.007)	0.450 (0.008)	0.377 (0.012)	0.519 (0.004)	0.472 (0.002)	0.441 (0.006)	0.206 (0.008)	0.502 (0.003)
letter	0.018 (0.004)	0.042 (0.002)	0.056 (0.006)	0.026 (0.007)	0.036 (0.002)	0.054 (0.004)	0.058 (0.002)	0.050 (0.005)	0.010 (0.000)
Lymphography	0.969 (0.021)	**1.000 (0.000)**	0.893 (0.070)	0.798 (0.008)	**1.000 (0.000)**	**1.000 (0.000)**	0.969 (0.021)	0.865 (0.005)	0.969 (0.021)
magic.gamma	**0.810 (0.002)**	0.780 (0.003)	0.802 (0.001)	0.807 (0.003)	0.764 (0.002)	0.790 (0.001)	0.771 (0.005)	0.702 (0.008)	0.794 (0.001)
mammography	**0.499 (0.007)**	0.359 (0.016)	0.405 (0.006)	0.372 (0.024)	0.351 (0.007)	0.398 (0.009)	0.303 (0.013)	0.054 (0.017)	0.483 (0.005)
mnist	**0.785 (0.003)**	0.720 (0.011)	0.614 (0.002)	0.600 (0.007)	0.725 (0.009)	0.697 (0.005)	0.686 (0.006)	0.650 (0.047)	0.766 (0.004)
musk	**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 (0.018)**	**1.000 (0.000)**
optdigits	0.652 (0.017)	0.551 (0.020)	0.573 (0.058)	0.135 (0.014)	0.441 (0.027)	0.700 (0.020)	0.543 (0.006)	0.440 (0.040)	**0.705 (0.030)**
PageBlocks	0.585 (0.013)	0.513 (0.010)	0.602 (0.008)	0.630 (0.005)	0.610 (0.002)	0.624 (0.002)	0.549 (0.005)	0.618 (0.006)	0.645 (0.007)
pendigits	0.936 (0.006)	0.850 (0.020)	0.829 (0.018)	0.613 (0.043)	0.910 (0.008)	0.885 (0.011)	0.882 (0.014)	0.750 (0.057)	0.876 (0.015)
Pima	0.744 (0.008)	0.707 (0.009)	0.670 (0.012)	0.656 (0.012)	0.713 (0.014)	0.752 (0.011)	0.705 (0.009)	0.707 (0.016)	0.757 (0.007)
satellite	0.722 (0.001)	0.726 (0.002)	0.693 (0.009)	0.722 (0.001)	0.732 (0.002)	0.731 (0.001)	0.726 (0.002)	0.717 (0.009)	0.723 (0.001)
satimage-2	0.907 (0.003)	0.870 (0.009)	0.189 (0.034)	0.699 (0.021)	0.831 (0.015)	0.868 (0.006)	0.899 (0.008)	0.845 (0.107)	0.904 (0.003)
shuttle	0.986 (0.001)	**0.989 (0.000)**	**0.989 (0.001)**	0.980 (0.000)	0.985 (0.000)	0.986 (0.000)	0.930 (0.040)	0.964 (0.007)	0.987 (0.000)
skin	0.840 (0.012)	0.762 (0.011)	0.910 (0.010)	0.821 (0.002)	0.846 (0.005)	0.899 (0.003)	0.766 (0.005)	0.830 (0.059)	0.824 (0.008)
smtp	0.637 (0.056)	0.512 (0.045)	0.297 (0.104)	**0.696 (0.020)**	0.672 (0.012)	**0.696 (0.020)**	0.557 (0.073)	0.000 (0.026)	0.648 (0.058)
SpamBase	0.805 (0.002)	0.806 (0.001)	0.820 (0.002)	0.800 (0.001)	**0.821 (0.001)**	0.812 (0.001)	0.808 (0.002)	0.797 (0.003)	0.809 (0.001)
speech	0.033 (0.000)	0.036 (0.006)	0.039 (0.013)	0.036 (0.012)	0.033 (0.009)	0.039 (0.011)	0.046 (0.008)	0.033 (0.004)	0.039 (0.004)
Stamps	0.865 (0.023)	0.845 (0.022)	0.784 (0.031)	0.620 (0.038)	0.823 (0.028)	0.908 (0.011)	0.850 (0.021)	0.468 (0.041)	0.893 (0.010)
thyroid	0.667 (0.020)	0.477 (0.027)	0.168 (0.009)	**0.766 (0.006)**	0.690 (0.032)	0.697 (0.016)	0.596 (0.036)	0.763 (0.011)	0.718 (0.010)
vertebral	0.728 (0.029)	0.512 (0.011)	**0.741 (0.020)**	0.403 (0.051)	0.273 (0.032)	0.697 (0.023)	0.579 (0.021)	0.050 (0.019)	0.710 (0.030)
vowels	0.384 (0.010)	0.392 (0.027)	0.348 (0.048)	0.388 (0.026)	0.360 (0.013)	0.364 (0.013)	0.400 (0.006)	0.020 (0.008)	0.356 (0.007)
Waveform	0.100 (0.003)	0.136 (0.004)	0.128 (0.012)	0.126 (0.009)	0.326 (0.009)	0.154 (0.005)	0.104 (0.009)	0.130 (0.027)	0.176 (0.015)
WBC	0.948 (0.024)	0.867 (0.038)	0.723 (0.050)	0.340 (0.023)	0.904 (0.032)	0.924 (0.028)	0.851 (0.061)	0.696 (0.020)	0.944 (0.022)
WDBC	**0.953 (0.030)**	0.908 (0.031)	0.619 (0.066)	0.731 (0.052)	0.899 (0.032)	0.898 (0.022)	0.945 (0.034)	0.609 (0.086)	0.942 (0.028)
Wilt	0.022 (0.002)	0.142 (0.008)	0.541 (0.033)	0.175 (0.013)	0.030 (0.002)	0.250 (0.005)	**0.592 (0.004)**	0.268 (0.003)	0.016 (0.001)
wine	0.993 (0.007)	0.982 (0.018)	0.993 (0.007)	0.979 (0.010)	0.993 (0.007)	0.982 (0.018)	0.993 (0.007)	0.959 (0.066)	0.982 (0.018)
WPBC	0.887 (0.016)	0.899 (0.015)	0.887 (0.019)	0.577 (0.023)	0.818 (0.011)	0.887 (0.012)	0.894 (0.013)	0.496 (0.027)	0.887 (0.011)
yeast	0.499 (0.006)	0.498 (0.007)	**0.523 (0.008)**	0.497 (0.006)	0.470 (0.003)	0.478 (0.003)	0.512 (0.002)	0.467 (0.011)	0.478 (0.004)
20news	0.238 (0.006)	0.231 (0.006)	0.243 (0.014)	0.181 (0.010)	0.229 (0.008)	0.259 (0.006)	0.249 (0.005)	0.125 (0.009)	**0.268 (0.010)**
agnews	0.309 (0.004)	0.288 (0.002)	0.318 (0.010)	0.210 (0.008)	0.288 (0.005)	**0.372 (0.001)**	0.350 (0.001)	0.139 (0.004)	0.370 (0.003)
amazon	0.107 (0.002)	0.106 (0.003)	0.112 (0.007)	0.104 (0.004)	0.103 (0.001)	0.118 (0.002)	0.108 (0.001)	0.114 (0.012)	**0.130 (0.004)**
CIFAR10	0.272 (0.001)	0.232 (0.002)	0.166 (0.002)	0.238 (0.002)	0.245 (0.001)	0.242 (0.001)	0.246 (0.001)	0.183 (0.005)	**0.281 (0.000)**
FashionMNIST	0.688 (0.001)	0.468 (0.007)	0.471 (0.004)	0.594 (0.002)	0.620 (0.001)	0.665 (0.001)	0.641 (0.001)	0.389 (0.003)	**0.694 (0.001)**
imdb	0.058 (0.001)	0.063 (0.000)	0.101 (0.004)	0.072 (0.008)	0.057 (0.001)	0.072 (0.003)	0.060 (0.001)	0.048 (0.004)	0.071 (0.002)
MNIST-C	0.667 (0.001)	0.532 (0.007)	0.506 (0.003)	0.492 (0.001)	0.509 (0.001)	0.670 (0.001)	0.580 (0.001)	0.393 (0.004)	**0.690 (0.001)**
MVTec	0.807 (0.004)	0.816 (0.003)	0.825 (0.004)	0.776 (0.006)	0.822 (0.005)	0.835 (0.004)	0.827 (0.004)	0.625 (0.006)	**0.842 (0.004)**
SVHN	0.225 (0.001)	0.183 (0.001)	0.173 (0.002)	0.195 (0.000)	0.201 (0.000)	0.220 (0.001)	0.207 (0.000)	0.169 (0.005)	**0.229 (0.001)**
yelp	0.199 (0.001)	0.189 (0.002)	0.169 (0.006)	0.140 (0.009)	0.195 (0.003)	0.207 (0.003)	0.207 (0.002)	0.104 (0.020)	**0.214 (0.003)**
Overall	0.608 (0.003)	0.574 (0.003)	0.547 (0.004)	0.523 (0.003)	0.580 (0.001)	0.605 (0.004)	0.583 (0.004)	0.463 (0.002)	**0.613 (0.001)**

Table 31: Average F1 for the semi-supervised setting on ADBench, column panel 2/2. Parentheses give standard errors over five seeds. Bold marks the best displayed mean across both panels.

Dataset	GMM	KPCA	LUNAR	NEAD	SLAD	ICL	kNN-mean	kNN	LOF
ALOI	0.068 (0.000)	0.059 (0.000)	0.058 (0.000)	0.046 (0.002)	0.051 (0.000)	0.051 (0.002)	0.059 (0.000)	0.059 (0.000)	**0.082 (0.000)**
annthyroid	0.504 (0.000)	0.582 (0.002)	0.607 (0.000)	0.640 (0.008)	0.661 (0.000)	0.560 (0.013)	0.625 (0.000)	0.620 (0.000)	0.496 (0.000)
backdoor	0.856 (0.005)	0.659 (0.017)	0.278 (0.004)	0.834 (0.009)	0.000 (0.000)	0.861 (0.006)	0.381 (0.008)	0.322 (0.006)	0.721 (0.011)
breastw	0.958 (0.002)	**0.976 (0.002)**	0.962 (0.002)	0.912 (0.005)	0.969 (0.003)	0.972 (0.003)	0.967 (0.003)	0.960 (0.002)	0.878 (0.012)
campaign	0.537 (0.000)	0.581 (0.002)	**0.583 (0.000)**	0.375 (0.004)	0.497 (0.000)	0.487 (0.003)	0.569 (0.000)	0.578 (0.000)	0.422 (0.000)
cardio	0.614 (0.000)	0.648 (0.000)	0.619 (0.000)	0.514 (0.020)	0.614 (0.000)	0.581 (0.006)	0.631 (0.000)	0.619 (0.000)	0.625 (0.000)
Cardiotocography	0.461 (0.000)	0.448 (0.000)	0.472 (0.000)	0.412 (0.004)	0.352 (0.000)	0.378 (0.012)	0.446 (0.000)	0.468 (0.000)	**0.483 (0.000)**
celeba	**0.214 (0.005)**	0.138 (0.006)	0.108 (0.004)	0.073 (0.004)	0.141 (0.005)	0.130 (0.004)	0.084 (0.007)	0.080 (0.006)	0.019 (0.002)
census	0.203 (0.003)	**0.354 (0.009)**	0.316 (0.003)	0.206 (0.004)	0.042 (0.042)	0.249 (0.007)	0.308 (0.004)	0.313 (0.003)	0.127 (0.004)
cover	0.182 (0.006)	0.398 (0.015)	0.631 (0.011)	**0.900 (0.007)**	0.075 (0.038)	0.476 (0.059)	0.694 (0.010)	0.650 (0.010)	0.821 (0.010)
donors	0.443 (0.005)	0.856 (0.006)	0.837 (0.006)	0.757 (0.014)	0.539 (0.063)	0.808 (0.027)	0.922 (0.004)	0.876 (0.006)	0.748 (0.008)
fault	0.579 (0.000)	0.542 (0.000)	0.557 (0.000)	0.551 (0.011)	**0.599 (0.000)**	0.586 (0.007)	0.560 (0.000)	0.556 (0.000)	0.507 (0.000)
fraud	0.604 (0.031)	0.474 (0.044)	0.490 (0.020)	**0.754 (0.024)**	0.490 (0.018)	0.614 (0.046)	0.484 (0.023)	0.478 (0.015)	0.615 (0.021)
glass	0.209 (0.011)	0.764 (0.062)	0.280 (0.021)	0.535 (0.036)	0.334 (0.027)	**0.860 (0.016)**	0.478 (0.029)	0.299 (0.048)	0.250 (0.037)
Hepatitis	0.796 (0.019)	0.996 (0.004)	0.817 (0.021)	0.984 (0.013)	0.996 (0.004)	0.996 (0.004)	0.992 (0.005)	0.820 (0.023)	0.400 (0.047)
http	0.919 (0.006)	0.973 (0.014)	0.985 (0.014)	0.596 (0.224)	0.886 (0.041)	0.968 (0.008)	**1.000 (0.000)**	**1.000 (0.000)**	0.966 (0.013)
InternetAds	0.609 (0.000)	0.617 (0.000)	0.549 (0.000)	0.680 (0.004)	0.582 (0.000)	0.624 (0.002)	0.617 (0.000)	0.598 (0.000)	0.546 (0.000)
Ionosphere	0.916 (0.008)	**0.954 (0.007)**	0.917 (0.006)	0.920 (0.008)	0.923 (0.005)	0.936 (0.007)	0.925 (0.006)	0.902 (0.009)	0.873 (0.011)
landsat	0.305 (0.000)	0.486 (0.002)	0.492 (0.000)	0.308 (0.004)	0.467 (0.000)	**0.542 (0.003)**	0.524 (0.000)	0.515 (0.000)	0.536 (0.000)
letter	0.000 (0.000)	0.010 (0.000)	0.010 (0.000)	0.054 (0.009)	0.020 (0.000)	0.046 (0.007)	0.010 (0.000)	0.010 (0.000)	**0.100 (0.000)**
Lymphography	0.973 (0.027)	**1.000 (0.000)**	0.968 (0.026)	**1.000 (0.000)**	0.995 (0.005)	**1.000 (0.000)**	**1.000 (0.000)**	0.916 (0.030)	0.767 (0.030)
magic.gamma	0.730 (0.000)	0.747 (0.000)	0.756 (0.000)	0.803 (0.001)	0.660 (0.000)	0.716 (0.003)	0.768 (0.000)	0.762 (0.000)	0.761 (0.000)
mammography	0.446 (0.000)	0.454 (0.004)	0.438 (0.000)	0.386 (0.004)	0.212 (0.000)	0.252 (0.009)	0.419 (0.000)	0.404 (0.000)	0.385 (0.000)
mnist	0.689 (0.000)	0.721 (0.001)	0.716 (0.000)	0.647 (0.011)	0.664 (0.000)	0.510 (0.005)	0.731 (0.000)	0.719 (0.000)	0.714 (0.000)
musk	0.928 (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)**	**1.000 (0.000)**
optdigits	0.020 (0.000)	0.103 (0.003)	0.140 (0.000)	0.223 (0.026)	0.360 (0.000)	0.339 (0.023)	0.340 (0.000)	0.240 (0.000)	0.533 (0.000)
PageBlocks	0.559 (0.000)	0.567 (0.003)	0.596 (0.000)	0.647 (0.004)	0.586 (0.000)	0.639 (0.012)	0.594 (0.000)	0.590 (0.000)	**0.659 (0.000)**
pendigits	0.167 (0.000)	0.835 (0.005)	0.917 (0.000)	0.773 (0.026)	0.442 (0.000)	0.692 (0.053)	**0.942 (0.000)**	0.904 (0.000)	0.763 (0.000)
Pima	0.702 (0.012)	**0.779 (0.006)**	0.714 (0.010)	0.695 (0.009)	0.582 (0.013)	0.714 (0.013)	0.748 (0.010)	0.710 (0.013)	0.667 (0.014)
satellite	0.714 (0.000)	0.696 (0.000)	0.709 (0.000)	0.709 (0.002)	**0.785 (0.000)**	0.763 (0.007)	0.723 (0.000)	0.718 (0.000)	0.726 (0.000)
satimage-2	0.789 (0.000)	0.910 (0.003)	0.901 (0.000)	0.566 (0.044)	0.887 (0.000)	**0.921 (0.008)**	0.901 (0.000)	0.901 (0.000)	0.817 (0.000)
shuttle	0.968 (0.000)	0.984 (0.000)	0.981 (0.000)	0.983 (0.000)	0.985 (0.000)	0.988 (0.000)	0.983 (0.000)	0.982 (0.000)	0.984 (0.000)
skin	0.782 (0.001)	0.903 (0.002)	0.961 (0.001)	0.943 (0.002)	0.744 (0.012)	0.691 (0.042)	**0.971 (0.002)**	0.967 (0.002)	0.707 (0.009)
smtp	0.403 (0.060)	**0.696 (0.020)**	**0.696 (0.020)**	0.259 (0.051)	**0.696 (0.020)**	0.562 (0.065)	**0.696 (0.020)**	**0.696 (0.020)**	0.684 (0.011)
SpamBase	0.779 (0.000)	0.816 (0.000)	0.802 (0.000)	0.812 (0.003)	0.818 (0.000)	0.810 (0.002)	0.808 (0.000)	0.805 (0.000)	0.740 (0.000)
speech	**0.066 (0.000)**	0.033 (0.000)	0.033 (0.000)	0.043 (0.004)	**0.066 (0.000)**	0.033 (0.009)	0.049 (0.000)	0.033 (0.000)	0.033 (0.000)
Stamps	0.638 (0.056)	**0.943 (0.014)**	0.740 (0.022)	0.833 (0.039)	0.467 (0.042)	0.840 (0.011)	0.859 (0.013)	0.786 (0.042)	0.659 (0.044)
thyroid	0.710 (0.000)	0.763 (0.000)	0.742 (0.000)	0.413 (0.018)	0.720 (0.000)	0.587 (0.017)	0.753 (0.000)	0.753 (0.000)	0.527 (0.000)
vertebral	0.116 (0.032)	0.686 (0.034)	0.241 (0.020)	0.375 (0.041)	0.143 (0.049)	0.531 (0.045)	0.492 (0.025)	0.230 (0.026)	0.332 (0.025)
vowels	**0.460 (0.000)**	0.300 (0.000)	0.280 (0.000)	0.392 (0.010)	0.400 (0.000)	0.292 (0.063)	0.300 (0.000)	0.260 (0.000)	0.340 (0.000)
Waveform	0.090 (0.000)	0.250 (0.000)	0.270 (0.000)	0.214 (0.008)	0.030 (0.000)	**0.352 (0.014)**	0.260 (0.000)	0.270 (0.000)	0.280 (0.000)
WBC	0.794 (0.032)	**0.973 (0.018)**	0.880 (0.017)	0.716 (0.052)	0.914 (0.032)	**0.973 (0.022)**	0.894 (0.015)	0.836 (0.023)	0.212 (0.049)
WDBC	0.866 (0.039)	0.911 (0.019)	0.810 (0.020)	0.900 (0.035)	0.797 (0.047)	0.831 (0.049)	0.851 (0.031)	0.809 (0.022)	0.901 (0.033)
Wilt	0.323 (0.000)	0.035 (0.001)	0.016 (0.000)	0.238 (0.011)	0.070 (0.000)	0.401 (0.047)	0.035 (0.000)	0.023 (0.000)	0.167 (0.000)
wine	0.905 (0.030)	0.993 (0.007)	0.864 (0.018)	**1.000 (0.000)**	**1.000 (0.000)**	0.971 (0.019)	0.977 (0.018)	0.859 (0.017)	0.816 (0.021)
WPBC	0.451 (0.013)	**0.900 (0.013)**	0.529 (0.011)	0.675 (0.022)	0.887 (0.015)	0.863 (0.023)	0.747 (0.012)	0.502 (0.014)	0.417 (0.019)
yeast	0.465 (0.000)	0.460 (0.000)	0.464 (0.000)	0.490 (0.001)	0.497 (0.000)	0.486 (0.008)	0.460 (0.000)	0.467 (0.000)	0.477 (0.000)
20news	0.216 (0.006)	0.201 (0.005)	0.145 (0.008)	0.259 (0.005)	0.136 (0.006)	0.155 (0.007)	0.178 (0.007)	0.151 (0.010)	0.169 (0.007)
agnews	0.294 (0.000)	0.202 (0.001)	0.196 (0.000)	0.364 (0.005)	0.129 (0.000)	0.144 (0.001)	0.207 (0.000)	0.201 (0.000)	0.306 (0.000)
amazon	0.090 (0.000)	0.111 (0.003)	0.106 (0.000)	0.100 (0.004)	0.102 (0.000)	0.104 (0.002)	0.108 (0.000)	0.114 (0.000)	0.100 (0.000)
CIFAR10	0.254 (0.000)	0.237 (0.000)	0.229 (0.000)	0.252 (0.001)	0.244 (0.000)	0.225 (0.001)	0.232 (0.000)	0.229 (0.000)	0.271 (0.000)
FashionMNIST	0.632 (0.000)	0.598 (0.001)	0.587 (0.000)	0.623 (0.001)	0.597 (0.000)	0.641 (0.001)	0.597 (0.000)	0.590 (0.000)	0.633 (0.000)
imdb	0.064 (0.000)	0.053 (0.001)	0.054 (0.000)	0.074 (0.004)	**0.102 (0.000)**	0.100 (0.001)	0.052 (0.000)	0.054 (0.000)	0.064 (0.000)
MNIST-C	0.523 (0.000)	0.466 (0.001)	0.456 (0.000)	0.550 (0.001)	0.478 (0.000)	0.514 (0.001)	0.475 (0.000)	0.463 (0.000)	0.530 (0.000)
MVTec	0.826 (0.003)	0.831 (0.004)	0.684 (0.006)	0.834 (0.004)	0.819 (0.004)	0.821 (0.003)	0.757 (0.005)	0.671 (0.005)	0.671 (0.003)
SVHN	0.204 (0.000)	0.192 (0.000)	0.189 (0.000)	0.203 (0.001)	0.192 (0.000)	0.197 (0.001)	0.192 (0.000)	0.190 (0.000)	0.192 (0.000)
yelp	0.192 (0.000)	0.183 (0.001)	0.176 (0.000)	0.186 (0.004)	0.070 (0.000)	0.078 (0.001)	0.196 (0.000)	0.188 (0.000)	0.206 (0.000)
Overall	0.505 (0.003)	0.580 (0.001)	0.536 (0.002)	0.548 (0.004)	0.500 (0.002)	0.570 (0.003)	0.571 (0.001)	0.539 (0.002)	0.516 (0.002)