RGLD: RANDOMIZED GLOBAL-LOCAL DENSITY ESTIMATION FOR TABULAR ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised tabular anomaly detection requires methods that are accurate, robust across heterogeneous datasets, and computationally efficient. Classical statistical detectors are often efficient, but they usually rely on a fixed data view and a single notion of abnormality. Deep anomaly detectors can learn more flexible scoring functions, but they are substantially slower and difficult to tune in unsupervised settings due to the lack of a reliable supervisory signal. We propose **RGLD**, a randomized global-local density estimator for efficient unsupervised tabular anomaly detection. RGLD combines a global random-feature density branch, which identifies samples in broadly low-density regions, with a local neighbor branch, which detects samples that are weakly supported by nearby observations. Both branches operate over feature-bagged randomized views, allowing RGLD to expose anomaly evidence that may be hidden in any single representation. We evaluate RGLD on 47 tabular datasets against 23 statistical and deep anomaly detection baselines in the fully unsupervised setting. RGLD achieves competitive detection accuracy, with 77.50 AUROC and 38.59 AUPRC, while obtaining the most AUROC dataset wins. It runs 4.9x faster than ADERH and 50x–580x faster than the evaluated deep detectors, yielding a strong accuracy–efficiency tradeoff.

1 INTRODUCTION

Unsupervised anomaly detection is a fundamental problem in machine learning with wide-ranging applications (Han et al., 2022; Jiang et al., 2023; Gungor et al., 2024; Chen et al., 2025; Chatterjee & Ahmed, 2022). Given a dataset dominated by normal samples, the goal is to assign high anomaly scores to rare, irregular, or structurally inconsistent observations, without access to labels. As datasets continue to grow and anomaly detection is increasingly deployed in time-sensitive applications, efficiency and scalability become central requirements.

A long-standing principle in anomaly detection is that anomalies are associated with low-density or weakly supported regions of the data distribution (Chandola et al., 2009; Ruff et al., 2021). This idea motivates many statistical methods, including nearest-neighbor detectors (Ramaswamy et al., 2000), local density estimators (Breunig et al., 2000), kernel methods (Schölkopf et al., 2001), isolation-based methods (Liu et al., 2008), and histogram-based approaches (Goldstein & Dengel, 2012). These methods are attractive because they are often simple, interpretable, and applicable without supervision. However, many statistical detectors are limited by two fixed choices: how abnormality is defined and where it is measured. First, each method typically adopts a particular notion of abnormality. For example, density-based methods treat anomalies as points in low-density regions, while nearest-neighbor methods treat anomalies as points with weak local support. These criteria are complementary: a point may be globally rare but not locally isolated, or locally isolated without being globally extreme. Second, a given notion of abnormality is often evaluated in a fixed view of the data, such as the full feature space, a chosen distance metric, and a scale. This can fail when the anomaly signal is visible only in a subset of features or under a different projection. In such cases, irrelevant or high-variance features can dilute the anomaly signal, making a point appear well supported in the full space even though it is clearly isolated in an informative subspace (Aggarwal & Yu, 2001; Kriegel et al., 2009). Thus, robust tabular anomaly detection requires both diverse notions of abnormality and diverse views in which those notions are evaluated.

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.

Deep anomaly detectors offer another route by learning nonlinear representations or anomaly scoring functions (Ruff et al., 2018; Zong et al., 2018; Pang et al., 2021). This flexibility can be valuable when anomalies are not well captured by a rigid notion of abnormality. However, in fully unsupervised settings, the same flexibility also creates a practical challenge as there is no reliable supervisory signal for training or choosing objectives. As a result, deep detectors often rely on proxy learning tasks whose effectiveness can vary substantially across datasets. They also introduce additional training cost and optimization variability, and benchmark studies have found that they can be orders of magnitude slower than lightweight statistical methods (Han et al., 2022; Jiang et al., 2023).

This motivates a different question: can we obtain diverse multi-view anomaly signals without the computational burden of deep anomaly detectors? We argue that for tabular anomaly detection, a promising path is to revisit classical density principles through randomized computation. Instead of relying on a single view or a single notion of abnormality, we can generate many inexpensive randomized views of the data and aggregate their anomaly signals. Each view provides a partial estimate of abnormality, but the diversity across views and anomaly mechanisms can expose different anomalies and reduce dependence on any single notion of abnormality.

Built around this idea, we propose **RGLD: Randomized Global-Local Density Estimation for Tabular Anomaly Detection**, an efficient detector for fully unsupervised tabular anomaly detection. RGLD combines two complementary scoring branches. The global branch uses random-feature density estimation to identify samples in broadly low-density regions, capturing global rarity while avoiding expensive pairwise density computation. The local neighbor branch measures neighborhood support in randomized projected spaces, capturing locally isolated samples that may not be globally extreme. RGLD further ensembles these scores over feature-bagged randomized views, revealing anomalies hidden in the full feature space. Finally, rank-based aggregation combines the resulting heterogeneous scores into a stable anomaly ranking. This design yields a simple, parallelizable, and highly efficient unsupervised detector for settings where both detection accuracy and runtime matter.

- We propose **RGLD**, a fully unsupervised framework that turns classical density and neighborhood principles into scalable randomized estimators for tabular anomaly detection.
- We provide explanatory analyses of RGLD's randomized mechanisms, covering feature-bagged anomaly contrast, KDE approximation, and sampled-reference local support.
- We evaluate RGLD on 47 ADBench datasets against 23 statistical and deep baselines. RGLD achieves 77.50 mean AUROC and 38.59 mean AUPRC, obtains the most AUROC dataset wins, and runs approximately 4.9× faster than the strongest statistical baseline and 50×–580× faster than the evaluated deep detectors.

2 Related Work

Statistical and Randomized Anomaly Detection: Statistical detectors define anomaly scores from explicit assumptions about distance, density, support, isolation, or distributional tails. Distance- and neighborhood-based methods such as kNN outlier detection (Ramaswamy et al., 2000), LOF (Breunig et al., 2000), COF (Tang et al., 2002), and subspace outlier scoring (Kriegel et al., 2009) identify samples with weak local support or abnormal behavior in selected subspaces. Other methods rely on cluster structure, as in CBLOF (He et al., 2003), one-class support estimation, as in one-class SVM (Schölkopf et al., 2001), or low-dimensional projection and reconstruction, as in PCA-based detection (Shyu et al., 2003). Efficient distributional detectors such as HBOS (Goldstein & Dengel, 2012), COPOD (Li et al., 2020), and ECOD (Li et al., 2022) score deviations from marginal, copula-based, or empirical feature distributions. Randomized and ensemble methods reduce dependence on a single detector or data view: Feature Bagging (Lazarevic & Kumar, 2005) ensembles detectors over feature subsets, Isolation Forest (Liu et al., 2008) uses randomized partitions, LODA (Pevnỳ, 2016) aggregates random-projection histograms, and ADERH (Durani et al., 2025) builds an ensemble from randomized hypersphere pairs. Zimek et al. (Zimek et al., 2014) provide a broader discussion of ensemble methods for unsupervised outlier detection. These methods demonstrate the strength of lightweight statistical scoring and randomization. However, most emphasize a primary scoring principle, such as local density, isolation, marginal tail behavior, or projection-based histograms. RGLD instead explicitly combines global density support and local neighborhood support, and evaluates both across feature-bagged randomized views and multiple density scales.

Deep Unsupervised Anomaly Detection: Deep anomaly detectors learn representations or proxy objectives for anomaly scoring, including one-class, reconstruction-based, transformation-based, and contrastive approaches (Ruff et al., 2018; Pang et al., 2018; Zong et al., 2018; Liu et al., 2021; Wang et al., 2019; Bergman & Hoshen, 2020; Qiu et al., 2021; Shenkar & Wolf, 2022; Xu et al., 2023b;a). While these methods can learn flexible anomaly scores, fully unsupervised settings provide limited guidance for model and objective selection and introduce substantial computational cost. RGLD instead seeks flexible anomaly evidence through randomized statistical computation without requiring a learned representation.

Positioning: RGLD builds on several established ingredients, including feature bagging, ensembling, and kernel/neighborhood-based scoring. Our contribution is not any one of these primitives in isolation. Rather, RGLD provides a unified global–local formulation that efficiently combines complementary signals. The empirical question studied here is whether this particular combination provides a useful accuracy–efficiency operating point for heterogeneous tabular anomaly detection.

3 Why Randomized Global-Local Density Estimation?

We first use simple synthetic examples to motivate why RGLD is built around randomized global-local density estimation. In fully unsupervised anomaly detection, labels are unavailable, so one of the most general principles is to measure how well each sample is supported by the observed data distribution. Under this view, normal samples tend to lie in high-support regions, while anomalies are weakly supported, either because they occur in globally low-density regions or because they are locally isolated from nearby samples (Chandola et al., 2009; Ruff et al., 2021). This support-based perspective is attractive for tabular data because it is label-free, model-agnostic, and closely connected to many statistical detectors. However, turning this principle into a robust detector is nontrivial: the relevant density scale is unknown, global and local notions of support capture different anomaly mechanisms, and the feature view in which the anomaly is visible is usually unknown. The examples below illustrate these three challenges and motivate the main design choices of RGLD.

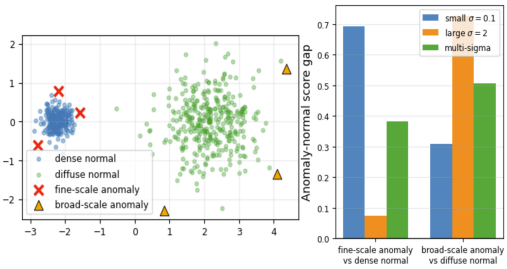


Figure 1: **Unknown anomaly scale. Left:** Fine-scale anomalies occur near a dense cluster, while broad-scale anomalies occur around a diffuse cluster. **Right:** Small and large bandwidths favor different anomaly scales, while multi-sigma scoring provides a more balanced signal when the correct scale is unknown.

Unknown Anomaly Scale: A central difficulty in density-based anomaly detection is that the appropriate density scale is unknown. As shown in Figure 1, KDE-style density scoring requires a bandwidth σ that controls the resolution of the density estimate (Silverman, 2018). However, anomalies may appear at different resolutions. Fine-scale anomalies may occur near a dense normal cluster, while broad-scale anomalies may be unusual only relative to a more diffuse normal structure. A single bandwidth can therefore emphasize one anomaly type while suppressing another. This motivates the multi-σ design of RGLD's global density branch, which aggregates density evidence across several bandwidths instead of committing to one fixed resolution.

Global Rarity and Local Isolation Are Complementary: A second challenge is that abnormality can take different forms. Some anomalies are globally rare and lie in broadly low-density regions of the data distribution. Other anomalies are locally isolated as they may not be globally extreme, but they are poorly supported relative to nearby samples. A detector based only on global density or only on local isolation can miss one of these cases. Here, local isolation is measured using a kNN isolation score (Ramaswamy et al., 2000), which assigns higher anomaly scores to points with larger average distances to their k nearest neighbors. Figure 2A shows this complementarity. The global density score assigns high anomaly values to points far from the main distribution, capturing broad low-density structure. In contrast, the local kNN isolation score emphasizes points that are poorly supported by nearby samples, including anomalies between or near normal clusters.

Anomalies Are View-Dependent: A third challenge is that anomalies may be visible only under certain views. Relevant abnormal structures often lie in a subset of features or along a particular

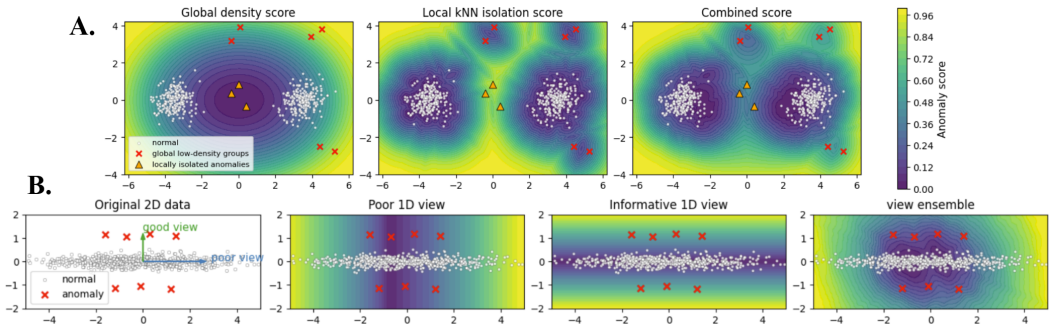


Figure 2: **Complementary evidence and view dependence. A.** Global density scoring detects broadly low-density anomalies, while local kNN isolation detects samples that are weakly supported by nearby observations. Their combination captures complementary anomaly mechanisms. **B.** Anomalies can be hidden in a poor view but become apparent in an informative projected view. Randomized view ensembling improves robustness by aggregating evidence across many perspectives.

direction. Irrelevant features or high-variance nuisance directions can obscure the anomaly signal in the full space (Aggarwal & Yu, 2001; Kriegel et al., 2009). Thus, a fixed representation can make an anomaly appear normal even when another view separates it clearly. Figure 2B illustrates this issue. In the original two-dimensional data, the anomaly signal depends strongly on the projection direction. A poor one-dimensional view largely hides the anomalies, while an informative view separates them more clearly. An ensemble over randomized views increases the chance that some views expose the relevant abnormal structure, even when the correct view is unknown in advance.

4 RGLD Algorithm

The previous section motivates three design principles: collect density evidence across multiple scales, combine global rarity with local isolation, and evaluate anomaly scores across randomized views. We now describe **RGLD**, a fully unsupervised anomaly detector that implements these principles through randomized global-local density estimation. Figure 3 summarizes the RGLD pipeline. Given an unlabeled dataset $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$, RGLD produces an anomaly score $s(x)$, where larger values indicate more anomalous samples. RGLD consists of two complementary branches. The global random-feature density branch estimates broad distributional support using randomized kernel density surrogates. The local neighbor branch estimates local support using sampled nearest-neighbor distances in randomized projected spaces. Following prior work on feature bagging for outlier detection (Lazarevic & Kumar, 2005), randomized feature subsets increase view diversity and help expose subspace-specific anomalies. Both branches operate over independently sampled randomized feature subsets and projections, and their scores are combined through rank-based aggregation.

Randomized Views: For each view t, RGLD samples a feature subset $S_t \subseteq \{1, \dots, d\}$ and a random projection matrix $R_t \in \mathbb{R}^{|S_t| \times r}$, representing a sample x as $z_t(x) = x_{S_t} R_t$. This feature-bagged projection serves two roles: feature subsets help expose subspace-specific anomalies that may be diluted in the full feature space, and random projections reduce dimensionality for high-dimensional data while approximately preserving the geometry needed for anomaly detection (Johnson et al., 1984; Dasgupta & Gupta, 2003). The global branch uses each view to define a random-feature density estimator, whereas the local neighbor branch uses independently sampled views to define local isolation estimators. Because any single view may be incomplete or noisy, RGLD aggregates scores across many views rather than relying on one fixed representation.

Global Density Branch: The global branch measures whether a sample lies in a broadly supported region of the data distribution. A natural nonparametric density-based score is kernel density estimation (KDE) (Rosenblatt, 1956; Parzen, 1962): for bandwidth σ, one may score a sample x by $\widehat{p}_\sigma(x) = \frac{1}{n} \sum_{i=1}^n k_\sigma(x, x_i)$, where k_σ is a Gaussian kernel. Low values of $\widehat{p}_\sigma(x)$ indicate weak distributional support and therefore high anomaly likelihood. However, evaluating this score for all samples requires pairwise kernel comparisons, leading to $O(n^2)$ cost per view and bandwidth. This becomes expensive when density scoring is repeated across many randomized views and multiple density scales. RGLD avoids this pairwise computation by replacing explicit KDE with random-

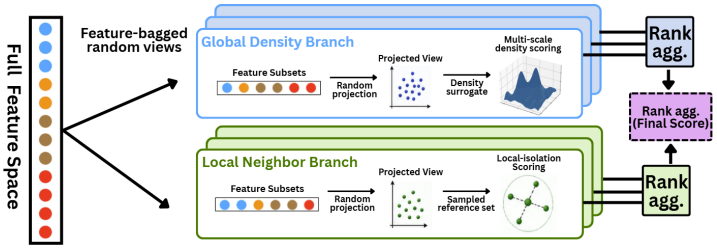


Figure 3: **RGLD Overview.** RGLD constructs feature-bagged random views and processes them through two complementary branches. The global density branch uses random projections and density surrogates for multi-scale density scoring, capturing global rarity. The local neighbor branch uses projected views and sampled reference sets to measure local isolation. Scores are rank-aggregated within each branch and then rank-aggregated again to produce the final anomaly score.

feature density estimation. For a randomized view t, let $z_i^{(t)} = z_t(x_i)$. Random features approximate shift-invariant kernels by finite-dimensional inner products (Rahimi & Recht, 2007) $k_\sigma(z, z') \approx \langle \phi_{t,\sigma}(z), \phi_{t,\sigma}(z') \rangle$. Substituting this approximation into KDE yields the density estimator:

$$\widehat{p}_{t,\sigma}(x) \approx \widetilde{p}_{t,\sigma}(x) = \langle \phi_{t,\sigma}(z_t(x)), \mu_{t,\sigma} \rangle \,, \qquad \mu_{t,\sigma} = \frac{1}{n} \sum_{i=1}^{n} \phi_{t,\sigma}(z_i^{(t)})$$

Thus, the full training set is compressed into one mean embedding $\mu_{t,\sigma}$ (Muandet et al., 2016) per randomized view and bandwidth. Scoring a sample only requires an inner product with this stored vector, rather than comparisons to all training samples, reducing the complexity of density estimation over all data points from $O(n^2)$ to $O(n)$. This mean-embedding form is the main efficiency gain of the global branch. To achieve the kernel sum approximation required for KDE, RGLD uses cosine random features. For a view t, let W_t be a random frequency matrix, b_t a random phase vector, and D the random-feature dimension. The random frequencies are sampled using quasi-Monte Carlo (QMC) (Yang et al., 2014) to reduce approximation variance at small D. Then, the random-feature map at bandwidth σ is:

$$\phi_{t,\sigma}(z) = \sqrt{\frac{2}{D}} \cos\left(\frac{W_t z}{\sigma} + b_t \right)$$

This form also makes multi-scale density scoring cheap. For each randomized view, RGLD first computes the linear random-feature projection $P_t(z) = W_t z$. Different bandwidths only rescale this same projection inside the cosine map. Therefore, for bandwidths $\sigma_{t,m} = \alpha_m \widehat{\sigma}_t$, where $\widehat{\sigma}_t$ is a base bandwidth estimated from sampled projected distances, RGLD reuses $P_t(z)$ and changes only $P_t(z)/\sigma_{t,m}$. The resulting density support score is:

$$d_{t,m}(x) = \left\langle \sqrt{\frac{2}{D}} \cos\left(\frac{P_t(z_t(x))}{\sigma_{t,m}} + b_t \right), \mu_{t,m} \right\rangle, \qquad \sigma_{t,m} = \alpha_m \widehat{\sigma}_t$$

The support score $d_{t,m}(x)$ is large when x is well supported under view t and scale m. RGLD converts it to an anomaly score by negation, $a_{t,m}^{\text{glob}}(x) = -d_{t,m}(x)$. Within each global view, scores from different bandwidths can have different scales. RGLD therefore standardizes the negative density scores for each bandwidth and averages them:

$$a_t^{\text{glob}}(x) = \frac{1}{M} \sum_{m=1}^{M} \mathcal{Z}\left(a_{t,m}^{\text{glob}}(x) \right)$$

where $\mathcal{Z}$ denotes standardization across samples for a fixed bandwidth. The view-level scores a_t^{glob} are then rank-aggregated across randomized global views to obtain s_{glob}. In this way, the global branch acts as an efficient multi-view, multi-scale approximation to KDE.

Local Neighbor Branch: The global branch captures broad low-density structure, but some anomalies are better characterized by local isolation. RGLD, therefore, includes a local neighbor branch that measures whether a sample is weakly supported by nearby reference points in randomized projected spaces. For each local view t, RGLD samples a feature subset and random projection, producing

$z_i^{(t)} = z_t(x_i)$. The reference set C_t is sampled uniformly without replacement from the projected training samples $\{z_i^{(t)}\}_{i=1}^n$. We use $\mathcal{N}_k(z_t(x); C_t)$ to denote the set of the k nearest reference points to $z_t(x)$ within C_t. For a sample x, the projected-neighbor score is the root mean squared distance from $z_t(x)$ to its k nearest points in C_t:

$$a_t^{\mathrm{loc}}(x) = \left(\frac{1}{k} \sum_{u \in \mathcal{N}_k(z_t(x); C_t)} \|z_t(x) - u\|_2^2 \right)^{1/2}$$

Larger values indicate weaker local support and therefore higher anomaly likelihood. This branch differs from a standard full-space kNN detector (Ramaswamy et al., 2000) in three ways. First, distances are computed in feature-bagged projected spaces rather than in the original feature space. Second, each local view uses a sampled reference set rather than the full training set, reducing the cost of neighbor scoring. Third, RGLD aggregates local isolation scores across randomized views, reducing dependence on any single projection, feature subset, or reference sample. The local neighbor branch score s_{loc} is obtained by rank-aggregating a_t^{loc} across local views.

Rank-Based Aggregation: The global and local branches produce scores with different meanings and expose anomalies of different mechanisms. The global branch produces negative density-support scores, while the local neighbor branch produces distance-based isolation scores. Directly averaging raw scores can therefore be unstable. RGLD instead aggregates scores by rank. For a score vector $v \in \mathbb{R}^n$, RGLD maps each score to its normalized rank $\rho_i(v) = \frac{\mathrm{rank}(v_i)-1}{n-1}$ where larger ranks correspond to more anomalous samples. Given score vectors $v^{(1)}, \ldots, v^{(T)}$, rank aggregation is

$$\mathrm{RankAgg}_i\left(v^{(1)}, \ldots, v^{(T)}\right) = \frac{1}{T} \sum_{t=1}^{T} \rho_i(v^{(t)})$$

This converts heterogeneous scores into a common scale and emphasizes consistency in anomaly ranking rather than raw score magnitude. The global branch score is s_{glob}, and the local neighbor branch score is s_{loc}. The final RGLD score is:

$$s(x_i) = \mathrm{RankAgg}_i\left(s_{\mathrm{glob}}, s_{\mathrm{loc}}\right)$$
$$\text{where} \quad s_{\mathrm{glob}}(x_i) = \mathrm{RankAgg}_i(a_1^{\mathrm{glob}}, \ldots, a_{T_g}^{\mathrm{glob}}), \quad s_{\mathrm{loc}}(x_i) = \mathrm{RankAgg}_i(a_1^{\mathrm{loc}}, \ldots, a_{T_\ell}^{\mathrm{loc}})$$

A sample receives a high RGLD score if it is consistently low-density across randomized global density views, locally isolated across projected-neighbor views, or both. This final aggregation treats global rarity and local isolation as complementary evidence while remaining insensitive to the score-scale mismatch between the two branches. Complete pseudo-code and implementation details are provided in Appendix G.

5 Analysis: Subspace Contrast, Approximation, and Complexity

RGLD relies on randomized views and efficient approximations to obtain global-local anomaly scores. We provide an analysis of why feature-bagged views can increase anomaly contrast when the anomaly signal is sparse, and bound the approximation gaps introduced by the two main computational shortcuts. These results are not intended as end-to-end anomaly detection guarantees, which would require assumptions on the anomaly-generating process. Instead, they clarify the conditions under which RGLD preserves useful density and locality signals while avoiding quadratic computation.

Feature-Bagged Subspace Contrast: We first formalize the mechanism by which feature-bagged views used by RGLD can yield stronger anomaly contrast than the full feature space when the anomaly signal is concentrated in a small set of informative coordinates. Consider an anomaly-normal pair (x_a, x_n). Suppose the anomaly signal lies in an unknown informative feature set $S^\star \subseteq [d]$ with $|S^\star| = s$, while the remaining features are nuisance coordinates.

Proposition 1 (Feature-bagged subspace contrast)**.** *Let $S^\star \subseteq [d]$ contain s informative features. Assume that the expected squared difference between an anomaly–normal pair is μ_{sig} on each feature in $S^\star$ and μ_{noise} otherwise, where $\mu_{\mathrm{sig}} > \mu_{\mathrm{noise}}$. A feature subset S has greater average anomaly contrast than the full feature space whenever $|S \cap S^\star|/|S| > s/d$ (See Appendix I for proof)*

Proposition 1 motivates the randomized-view design of RGLD. A fixed full-space representation can dilute sparse anomaly signals with nuisance features, while feature bagging increases the chance of seeing a view where the relevant coordinates are more concentrated. Random projection then provides an efficient alternative geometry within each selected subspace, preserving useful distance-based density and neighborhood signals while reducing cost.

Global Density Approximation Gap: Here, we analyze the approximation gap introduced by the global random-feature density branch. The method section defines the exact projected KDE score $\widehat{p}_{t,\sigma}(x)$ and its random-feature approximation $\widetilde{p}_{t,\sigma}(x)$. Here, we ask how far the surrogate can deviate from the exact KDE score for a fixed randomized view t and bandwidth σ. The following proposition is a direct consequence of the classical random Fourier feature approximation bound (Rahimi & Recht, 2007; Sutherland & Schneider, 2015):

Proposition 2 (RFF approximation of KDE scores)**.** *Fix a randomized view t, bandwidth σ, and training set $\{z_i^{(t)}\}_{i=1}^N$. Using D independently sampled Gaussian random Fourier features, Then for any fixed evaluation point x, with probability at least $1-\delta$ over the random-feature draw:*

$$|\widetilde{p}_{t,\sigma}(x) - \widehat{p}_{t,\sigma}(x)| \le O\left(\sqrt{\frac{\log(1/\delta)}{D}}\right) \quad \textit{(See Appendix J for proof)}$$

RGLD uses QMC by default as a practical variance-reduction choice. The proposition provides a conservative approximation bound for the corresponding i.i.d. RFF construction. Proposition 2 shows that the global branch remains close to the exact KDE score in each fixed view and bandwidth, with error decreasing as $D^{-1/2}$. Thus, RGLD can use many inexpensive random-feature density estimators while retaining a controlled approximation to KDE density scoring.

Local Support Approximation Gap: We next analyze the approximation introduced by the local neighbor branch. In a fixed projected view t, full kNN scoring compares each point against all n projected training samples. RGLD instead samples a reference set C_t and computes kNN distances only against this subset. This reduces computation but introduces a sampling gap: nearby training samples may be missed if they are not included in C_t.

Proposition 3 (Reference coverage for local support)**.** *Fix a projected view t, an evaluation point x, and radius $\rho > 0$. Let $B_t(x,\rho) = \{z_i^{(t)} : \|z_t(x) - z_i^{(t)}\|_2 \le \rho\}$ be the set of projected training samples within radius ρ, and let $q_t(x,\rho) = |B_t(x,\rho)|/n$ be its empirical mass. If C_t is sampled uniformly without replacement from the projected training samples, then*

$$\Pr\left[C_t \cap B_t(x,\rho) = \emptyset\right] \le \exp\left(-|C_t| q_t(x,\rho)\right) \quad \textit{(See Appendix K for proof)}$$

Equivalently, with probability at least $1 - \exp(-|C_t| q_t(x,\rho))$, the reference set contains at least one point within distance ρ of $z_t(x)$.

Proposition 3 shows that the probability of missing a local neighborhood decreases exponentially with the reference size $|C_t|$ and the empirical local mass $q_t(x,\rho)$. Thus, sampled-reference kNN is likely to capture local support when nearby samples are not too rare, while assigning large scores to points with weak local support. For $k > 1$, the same argument extends by considering the probability that C_t contains at least k points in $B_t(x,\rho)$, which follows a hypergeometric tail bound (Hoeffding, 1963; Serfling, 1974). Aggregating across multiple projected-neighbor views and independently sampled reference sets can further reduce sensitivity on any single reference sample.

Computational Complexity: Let n be the number of samples, d_t the number of selected features in a view, r the projection dimension, D the random-feature dimension, M the number of bandwidths, T_g the number of global views, T_ℓ the number of projected-neighbor views, and $|C_t|$ the reference-set size. For one global view, exact multi-scale KDE requires $O(Mn^2 r)$ kernel computations. RGLD replaces this with random-feature estimation: feature-bagged projection costs $O(nd_t r)$, the random-feature projection costs $O(nrD)$, and evaluating M bandwidths costs $O(MnD)$. Across global views, the cost is $O\left(T_g(nd_t r + nrD + MnD)\right)$. The key saving is avoiding the n^2 KDE term. For projected-neighbor views, full kNN costs $O(n^2 r)$. RGLD instead compares each point to a sampled reference set of size $|C_t|$, giving $O\left(T_\ell(nd_t r + n|C_t|r)\right)$ across local views. Rank aggregation adds $O(T_g n \log n + T_\ell n \log n)$ sorting cost. Overall, RGLD replaces the quadratic costs of exact KDE and full kNN with near linear-in-n randomized approximations, up to rank-sorting overhead, controlled by D, $|C_t|$, r. Since randomized views are independent, both branches are naturally parallelizable.

Table 1: **Main benchmark results.** On 47 ADBench tabular datasets across RGLD and 23 statistical and deep learning methods, all evaluated under the same fully unsupervised protocol. Marks 1st, 2nd and 3rd place respectively.

Method	Type	AUROC	AUROC Wins	AUPRC	AUPRC Wins	Avg. Rank	Time (s)	Relative Speed (×)
RGLD (Ours)	Statistical	77.50	**8**	38.59	5	9.72	0.22	1
SLAD (Xu et al., 2023b)	DL	73.39	7	35.23	**10**	9.76	39.78	0.01
ADERH (Durani et al., 2025)	Statistical	**78.18**	4	**39.72**	3	**8.22**	1.06	0.21
REPEN (Pang et al., 2018)	DL	74.72	4	37.44	4	10.52	47.88	0.005
COPOD (Li et al., 2020)	Statistical	74.39	4	35.22	3	11.52	0.07	3.3
NeuTraL (Qiu et al., 2021)	DL	70.76	4	28.12	4	12.04	54.21	0.004
*k*NN (Ramaswamy et al., 2000)	Statistical	69.87	4	31.54	2	11.96	0.84	0.26
ICL (Shenkar & Wolf, 2022)	DL	58.85	4	21.97	3	16.63	128.95	0.002
PCA (Shyu et al., 2003)	Statistical	73.60	3	38.27	1	10.92	0.66	0.33
HBOS (Goldstein & Dengel, 2012)	Statistical	74.04	3	36.96	2	10.84	**0.04**	**5.41**
GOAD (Bergman & Hoshen, 2020)	DL	68.73	2	33.25	3	12.54	80.31	0.003
IForest (Liu et al., 2008)	Statistical	76.17	1	38.78	0	9.91	0.52	0.42
DIF (Xu et al., 2023a)	DL	76.13	1	36.62	0	10.43	15.88	0.01
CBLOF (He et al., 2003)	Statistical	74.47	1	37.32	2	10.45	0.30	0.73
ECOD (Li et al., 2022)	Statistical	73.95	1	35.80	2	10.84	0.10	2.26
RDP (Wang et al., 2019)	DL	74.15	1	30.99	2	11.43	29.41	0.01
OCSVM (Schölkopf et al., 2001)	Statistical	69.98	1	33.23	1	13.63	1.80	0.12
LOF (Breunig et al., 2000)	Statistical	61.50	1	21.47	2	15.57	0.39	0.56
COF (Tang et al., 2002)	Statistical	62.75	1	23.07	0	15.98	18.53	0.01
DeepSVDD (Ruff et al., 2018)	DL	53.61	1	18.56	1	18.28	11.23	0.02
RCA (Liu et al., 2021)	DL	72.82	0	34.00	1	12.49	39.85	0.01
SOD(Kriegel et al., 2009)	Statistical	68.73	0	26.19	0	13.78	2.64	0.08
LODA (Pevnỳ, 2016)	Statistical	64.93	0	30.52	1	15.29	0.05	4.84
DAGMM (Zong et al., 2018)	DL	64.51	0	24.58	3	16.07	13.25	0.02

6 Experimental Analysis

Experimental Setup: We evaluate RGLD on 47 tabular datasets from ADBench (Han et al., 2022), which span diverse real-world anomaly detection settings including healthcare, finance, network/web security, image-derived features, documents, and scientific measurements. We compare against 23 baselines from ADBench (Han et al., 2022), DeepOD (Xu et al., 2023a; 2024), and the recently proposed ADERH (Durani et al., 2025), covering both statistical and deep anomaly detectors. We follow the standard ADBench data-generation protocol for all benchmark methods. All methods are evaluated in the fully unsupervised setting without using labels for model selection or hyperparameter tuning. Baselines use the official ADBench/DeepOD default configurations. We report Area Under the Receiver Operating Characteristic Curve (AUROC), Area Under the Precision-Recall Curve (AUPRC), and average runtime across datasets. Statistical methods are run on an Intel Core i7-155H CPU, while deep methods are run on an NVIDIA RTX 4070 Laptop GPU. RGLD uses fixed configuration across all datasets. Full RGLD configurations are provided in Appendix H.

Anomaly Detection Results: Table 1 summarizes the main results on 47 ADBench tabular datasets against 23 baselines. RGLD shows strong robustness across datasets, obtaining the most AUROC wins with 8 first-place results. It also ranks second in mean AUROC with 77.50, third in mean AUPRC with 38.59, and second in AUPRC wins with 5 wins. These results show that RGLD is consistently competitive across both ranking-based anomaly detection metrics. RGLD is also substantially more efficient than deep anomaly detectors, achieving $50\times$–$580\times$ speedups. While ADERH obtains the best mean AUROC and AUPRC, RGLD is approximately $5\times$ faster and achieves twice as many AUROC wins. Compared with SLAD, which has the most AUPRC wins, RGLD is roughly two orders of magnitude faster. Although a few simple statistical methods are faster, they generally sacrifice accuracy and dataset-level wins. Overall, RGLD provides one of the strongest accuracy-efficiency tradeoffs among all evaluated methods, combining top-tier detection performance with lightweight runtime. Because ADERH is the strongest baseline in mean AUROC and AUPRC, we provide a detailed paired comparison in Appendix M. The comparison shows that ADERH achieves slightly higher mean AUROC and AUPRC, while the paired differences are not statistically significant in this benchmark. RGLD trades this small observed accuracy gap for a 4.86x reduction in runtime. RGLD also remains stable across multiple seeds as shown in Appendix D.

Accuracy-Efficiency Tradeoff: Figure 4 plots mean AUROC and mean AUPRC against average runtime across the 47 datasets, with runtime on a logarithmic scale. The figure compares the baselines with two RGLD variants (RGLD-Fast variant uses a reduced number of views). RGLD occupies a favorable accuracy-efficiency region in both metrics. In AUROC, RGLD performs close to the strongest method, ADERH, while running substantially faster. RGLD-Fast further shifts the tradeoff

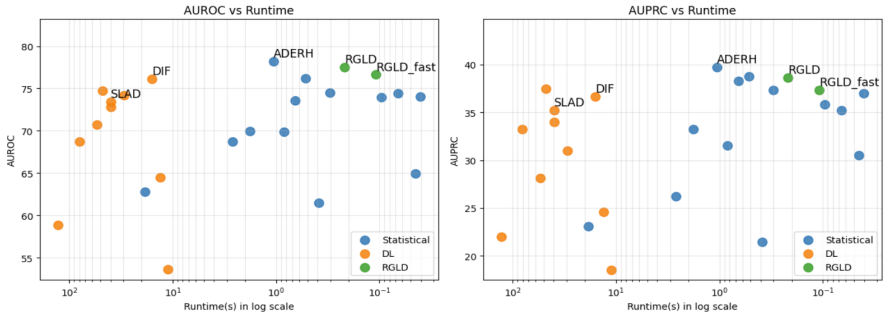


Figure 4: **Accuracy-efficiency tradeoff.** Mean AUROC and AUPRC versus average runtime on 47 ADBench datasets. Runtime is logarithmic; RGLD-Fast halves the view, random-feature, and reference-set budgets. Full plots are in Appendix N.

Table 2: **Ablation study on different RGLD components.** We report mean AUROC, mean AUPRC, average runtime, and changes relative to full RGLD. Removing any major component degrades performance, with the largest drop coming from removing the randomized-view ensemble.

Variant	AUROC ↑	AUPRC ↑	Time (s) ↓	ΔAUROC	ΔAUPRC	ΔTime
RGLD	**77.50**	**38.59**	0.22	-	-	-
w/o multi-σ	77.24	38.46	0.19	-0.25	-0.13	-0.02
w/o feature bagging	77.02	36.07	0.22	-0.48	-2.51	0.00
w/o local-neighbor	75.52	38.07	0.15	-1.98	-0.51	-0.07
w/o view ensemble	73.16	32.29	**0.06**	-4.33	-6.30	-0.16

toward lower runtime while retaining high AUROC. In contrast, deep methods such as SLAD, REPEN, NeuTraL, GOAD, and ICL require much longer runtime while giving lower or comparable AUROC. Fast statistical methods are efficient, but many fall below RGLD in accuracy. The AUPRC plot shows a similar pattern, where RGLD remains among the top-performing methods and ahead of most statistical and deep baselines, while being highly efficient. RGLD-Fast again provides a faster variant with only a modest AUPRC reduction. These results support RGLD's central efficiency claim: randomized global-local density estimation can achieve competitive detection accuracy without the high training cost of deep anomaly detectors.

Ablation Study on Different RGLD Components: Table 2 studies the contribution of RGLD's main components. The full model achieves the best AUROC and AUPRC, indicating that the proposed components are complementary rather than redundant. Removing multi-scale density scoring causes a modest drop, suggesting that aggregating multiple bandwidths improves robustness to unknown anomaly scale. Removing feature bagging leads to a larger AUPRC degradation, supporting the importance of subspace diversity. The local neighbor branch also contributes substantially: removing it reduces AUROC by 1.98, showing that local isolation provides useful information beyond global density estimation. The largest degradation occurs when removing the randomized-view ensemble, which reduces AUROC by 4.33 and AUPRC by 6.3. Although this variant is faster, the large accuracy drop suggests the central design principle of RGLD: many inexpensive randomized views are important for producing a robust anomaly ranking.

7 Conclusion

We introduced **RGLD**, a randomized global-local density estimator for fully unsupervised tabular anomaly detection. RGLD estimates complementary global and local anomaly signals from feature-bagged randomized views, allowing it to capture multiple anomaly mechanisms with efficiency. Across 47 datasets and 23 statistical and deep baselines, RGLD ranks first in AUROC wins and second in AUPRC wins while running $50\times$–$580\times$ faster than the evaluated deep detectors, yielding a favorable accuracy-efficiency tradeoff. Ablations show that randomized views, feature bagging, multi-scale density scoring, and projected-neighbor scoring each contribute to performance. These results suggest that the gains come not from a single component, but from combining many weak randomized density and locality signals into a stable ensemble. More broadly, RGLD shows that well-designed statistical methods can remain highly competitive for anomaly detection when equipped with the right randomized inductive biases.

AI USE STATEMENT

In this work, we used generative AI tools for generate synthetic data sets, formulate mathematical claims and assist in the writing of proofs, design or provide feedback on research methodology or experiments, implement methods, interpret results and support qualitative and thematic data analysis. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, assist with translation, clean and reformat datasets. Additionally, we used generative AI tools for summarize or analyse existing literature, discover research topics or identify gaps, brainstorming, identify relevant literature. We reviewed all AI-assisted information. Claims, and related-work comparisons were checked against the relevant literature, and citations were verified against their sources. The authors made all final methodological, experimental, and presentation decisions. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We value the reproducibility and accessibility of our work. The supplementary material includes the source code, complete experimental configurations, all hyperparameters used in the experiments, and the raw experimental results and execution logs. The full RGLD algorithm, pseudocode, and implementation details are provided in the Appendix G, H, together with the complete proofs of the theoretical analysis in Appendix I, J, K. We will make the code publicly available upon acceptance.

REFERENCES

Charu C Aggarwal and Philip S Yu. Outlier detection for high dimensional data. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pp. 37–46, 2001.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*, 2020.

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.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

Ayan Chatterjee and Bestoun S Ahmed. Iot anomaly detection methods and applications: A survey. *Internet of Things*, 19:100568, 2022.

Sihan Chen, Zhuangzhuang Qian, Wingchun Siu, Xingcan Hu, Jiaqi Li, Shawn Li, Yuehan Qin, Tiankai Yang, Zhuo Xiao, Wanghao Ye, et al. Pyod 2: A python library for outlier detection with llm-powered model selection. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 2807–2810, 2025.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

Walid Durani, Collin Leiber, Khalid Durani, Claudia Plant, and Christian Böhm. Anomaly detection by an ensemble of random pairs of hyperspheres. *Advances in Neural Information Processing Systems*, 38:166168–166207, 2025.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 1:59–63, 2012.

Onat Gungor, Amanda Rios, Priyanka Mudgal, Nilesh Ahuja, and Tajana Rosing. A robust framework for evaluation of unsupervised time-series anomaly detection. In *International Conference on Pattern Recognition*, pp. 48–64. Springer, 2024.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. *Advances in neural information processing systems*, 35:32142–32159, 2022.

Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10):1641–1650, 2003.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

Minqi Jiang, Chaochuan Hou, Ao Zheng, Songqiao Han, Hailiang Huang, Qingsong Wen, Xiyang Hu, and Yue Zhao. Adgym: Design choices for deep anomaly detection. *Advances in Neural Information Processing Systems*, 36:70179–70207, 2023.

William B Johnson, Joram Lindenstrauss, et al. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Pacific-asia conference on knowledge discovery and data mining*, pp. 831–838. Springer, 2009.

Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 157–166, 2005.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, pp. 1118–1123. IEEE, 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, 2022.

Boyang Liu, Ding Wang, Kaixiang Lin, Pang-Ning Tan, and Jiayu Zhou. Rca: A deep collaborative autoencoder approach for anomaly detection. In *IJCAI: proceedings of the conference*, volume 2021, pp. 1505, 2021.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. Ieee, 2008.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *arXiv preprint arXiv:1605.09522*, 2016.

Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2041–2050, 2018.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38, 2021.

Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

Tomáš Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.

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.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438, 2000.

Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pp. 832–837, 1956.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018.

Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

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.

Robert J Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, pp. 39–48, 1974.

Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International conference on learning representations*, 2022.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. 2003.

Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

Danica J Sutherland and Jeff Schneider. On the error of random fourier features. *arXiv preprint arXiv:1506.02785*, 2015.

Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 535–548. Springer, 2002.

Hu Wang, Guansong Pang, Chunhua Shen, and Congbo Ma. Unsupervised representation learning by predicting random distances. *arXiv preprint arXiv:1912.12186*, 2019.

Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12591–12604, 2023a. doi: 10.1109/TKDE.2023.3270293.

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, 2023b.

Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang, and Guansong Pang. Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Jiyan Yang, Vikas Sindhwani, Haim Avron, and Michael Mahoney. Quasi-monte carlo feature maps for shift-invariant kernels. In *International Conference on Machine Learning*, pp. 485–493. PMLR, 2014.

Arthur Zimek, Ricardo JGB Campello, and Jörg Sander. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *Acm Sigkdd Explorations Newsletter*, 15(1): 11–22, 2014.

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.

The appendix here provides additional details for the submission titled: "RGLD: Randomized Global-Local Density Estimation for Tabular Anomaly Detection". The appendix is organized as follows:

A. **List of Notation**

B. **Discussion**

C. **Limitations**

D. **Stability Across Randomized Seeds**

E. **Benefits of Randomized Views**

F. **Sensitivity to Ensemble Size and Random-Feature Dimension**

G. **Full RGLD Pseudocode and Implementation Detail**

H. **Hyperparameters for Two RGLD Variants**

I. **Proof of Proposition 1**

J. **Proof of Proposition 2**

K. **Proof of Proposition 3**

L. **Scaling in Dataset Size and Dataset Dimension**

M. **Detailed Comparison with ADERH**

N. **Full Pareto Frontier Graph**

O. **Full Benchmark Results**

A List of Notation

We hereby provide a list of notations used in this paper:

Table 3: List of notations.

Symbol	Meaning
$X = \{x_i\}_{i=1}^n$	Unlabeled dataset with n samples.
d	Ambient data dimension.
$s(x)$	Final RGLD anomaly score; larger values indicate more anomalous samples.
S_t	Feature subset sampled for randomized view t.
r	Projected dimension after random projection.
R_t	Random projection matrix for view t.
$z_t(x) = x_{S_t} R_t$	Projected representation of x in randomized view t.
T_g, T_ℓ	Number of global density views and projected-neighbor views.
$\sigma_{t,m}$	Bandwidth for global view t and scale m.
D	Random-feature dimension.
$\phi_{t,\sigma}(z)$	Random-feature map for projected sample z.
$\mu_{t,\sigma}$	Empirical random-feature mean embedding.
$\widehat{p}_{t,\sigma}(x), \widetilde{p}_{t,\sigma}(x)$	Exact projected KDE score and its random-feature approximation.
$s_{\mathrm{glob}}(x)$	Rank-aggregated global density branch score.
C_t	Sampled reference set for projected-neighbor view t.
k	Number of nearest reference points.
$a_t^{\mathrm{loc}}(x)$	Projected-neighbor local isolation score.
$s_{\mathrm{loc}}(x)$	Rank-aggregated local neighbor branch score.
$\mathrm{RankAgg}(\cdot)$	Rank-based aggregation operator.
$B_t(x,\rho)$, $q_t(x,\rho)$	Projected radius-ρ neighborhood of x, and its empirical mass.

B Discussion

Practical Guide for Hyperparameter Selection: RGLD's hyperparameters mainly control the tradeoff between approximation fidelity, ensemble diversity, and runtime. The random-feature dimension D controls the global density approximation: by Proposition 2, the random-feature KDE error decreases as $O(D^{-1/2})$, so larger D improves approximation quality with diminishing returns. In our experiments, a modest value such as 10 was sufficient to provide a strong accuracy-efficiency tradeoff. The reference set size $|C_t|$ controls the local-support approximation: by Proposition 3, the probability of missing a radius-ρ neighborhood is at most $\exp(-|C_t| q_t(x,\rho))$, so larger $|C_t|$ improves local coverage exponentially while increasing neighbor-scoring cost linearly. Thus, within a computational budget, larger D and $|C_t|$ generally improve the fidelity of the global and local branch approximations. The number of global and local views, T_g and T_ℓ, increases ensemble diversity and the chance of observing informative views, with runtime growing approximately linearly in the number of views. The bandwidth multipliers α_m define the density scales used by the global branch; a small grid such as $\{0.25, 0.5, 1.0, 2.0\}$ covers both fine and coarse anomaly scales without high cost. Finally, the number of neighbors k controls the locality of the projected-neighbor score, smaller k emphasizes fine local isolation, while a larger k gives smoother but less local support estimates.

The Need for Random Projection: Random projection is used in RGLD to make both scoring branches cheaper while retaining useful geometric information. By the Johnson-Lindenstrauss principle, random projections can approximately preserve pairwise distances in a lower-dimensional space (Johnson et al., 1984), which is important because both density and neighbor-based anomaly scores depend on this geometry. In the global branch, projecting the data reduces the cost of random-feature density estimation and helps avoid directly applying KDE-style scoring in the original high-dimensional space, where density estimates are sensitive to dimension and bandwidth choice (Silverman, 2018). In the local branch, projection reduces the cost of neighbor-distance computation by replacing full-dimensional distances with low-dimensional projected distances. RGLD therefore uses random projection as a lightweight mechanism for producing inexpensive density and neighborhood estimates across multiple randomized views.

C LIMITATIONS

Potential Failure Mode: RGLD relies on the assumption that useful anomaly evidence can be exposed through randomized views of the observed tabular feature space. This makes the method efficient and fully unsupervised, but it may be less effective when anomaly signals are extremely weak, highly diffuse across many features, or only visible through complex nonlinear representations. Feature bagging is most helpful when some subsets of features provide stronger anomaly contrast than the full space; when the raw features do not contain meaningful density or neighborhood structure, representation-learning methods or domain-specific features may be more suitable.

Anomaly Signals: RGLD focuses on scalable approximations to density-based and neighborhood-based anomaly signals. This covers a broad and effective class of unsupervised detectors, but abnormality can also appear through other mechanisms. For example, anomalies may be characterized by reconstruction error, violation of feature dependencies, abnormal marginal tail behavior, or inconsistency with learned semantic representations. RGLD's randomized ensemble framework is compatible with these signals in principle, but this paper focuses on global density and local neighbor support. Future work could extend RGLD by incorporating additional scalable score approximations into the same randomized rank-aggregation framework.

Extension to Broader Data Modalities or Settings: Our experiments focus on static tabular anomaly detection. Extending RGLD to time series, graphs, images, streaming data, or semi-supervised settings would require additional design choices, such as temporal modeling, graph-aware decisions, and online updates. We view these directions as complementary to the present work and as promising paths toward more broadly applicable, scalable anomaly detection.

D STABILITY ACROSS RANDOMIZED SEEDS

Table 4: **Stability across randomized runs.** Aggregate performance is computed across five 47-dataset averages. The last two columns summarize the per-dataset standard deviations across the 47 datasets. All values are percentage points.

	Aggregate performance	Per-dataset standard deviation	
Metric	Mean $\pm$ std.	Median	90th pct.
AUROC	77.81 ± 0.23	1.34	3.30
AUPRC	38.40 ± 0.78	1.87	8.83

The main benchmark tables report results from the designated benchmark seed used for the full per-dataset comparison, while this section repeats the complete benchmark with five independent method seeds to assess sensitivity to RGLD's internal randomness.

RGLD uses randomized feature subsets, projections, quasi-random feature maps, bandwidth-estimation samples, and local reference sets. We evaluate its sensitivity to these components by repeating the complete 47-dataset benchmark with five independent method seeds. The data split, preprocessing rule, and all hyperparameters are fixed across runs, so the measured variation reflects only the internal randomness of RGLD.

Table 4 reports both benchmark-level variation. For each seed, we first average performance over the 47 datasets and then compute the mean and standard deviation across seeds. RGLD achieves an aggregate AUROC of 77.81 ± 0.23 and an aggregate AUPRC of 38.40 ± 0.78. The median per-dataset standard deviations are 1.34 AUROC points and 1.87 AUPRC points. Although AUPRC is more variable on a small number of datasets, the aggregate performance remains consistent, indicating that the main results do not depend on a favorable random realization.

E BENEFITS OF RANDOMIZED VIEWS

To isolate the contribution of randomized subspace and projection views, we compare RGLD with a matched full-space global–local baseline. The baseline retains the same preprocessing, numbers of

global and local estimators, random-feature dimensions, bandwidths, sampled-reference sizes, and aggregation rules as RGLD. The only difference is that each estimator operates directly on the full preprocessed feature space rather than on a feature-bagged random projection. Both methods are evaluated on all 47 datasets using the same fixed data split and five independent method seeds.

Table 5: **Comparison with a matched full-space global–local detector.** Values report the mean and standard deviation across five complete 47-dataset runs. Runtime is averaged over datasets and seeds.

Method	AUROC	AUPRC	Runtime (s)
Full-space global–local	77.54 ± 0.15	36.37 ± 0.83	0.401 ± 0.006
RGLD	$\mathbf{77.81 \pm 0.23}$	$\mathbf{38.40 \pm 0.78}$	$\mathbf{0.227 \pm 0.005}$
RGLD improvement	$+0.27$	$+2.03$	$1.76\times$ faster

Table 5 shows that randomized subspace and projection views provide a clear advantage over applying the same global–local detector directly in the full feature space. RGLD improves aggregate AUROC from 77.54 to 77.81 and, more substantially, improves AUPRC from 36.37 to 38.40, a gain of 2.03 percentage points. The stronger AUPRC improvement indicates that randomized views are particularly effective at improving the ranking of rare anomalies, which is important for highly imbalanced anomaly-detection tasks. At the same time, RGLD reduces average runtime from 0.401 to 0.227 seconds, yielding a $1.76\times$ speedup. These results demonstrate that randomized views contribute beyond the global–local combination itself: they improve detection accuracy while also reducing the cost of both density estimation and local-neighborhood scoring.

F Sensitivity to Ensemble Size and Random-Feature Dimension

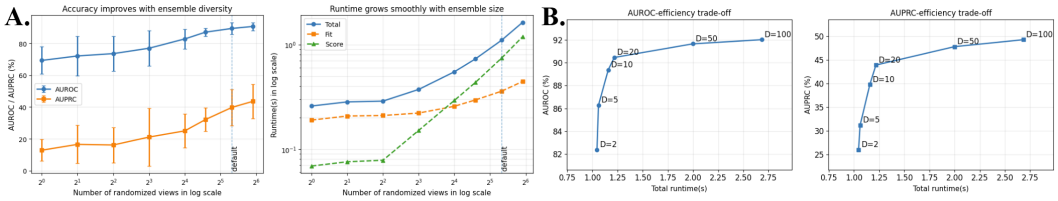


Figure 5: **Sensitivity to randomized views and random-feature dimension. A.** Increasing the number of randomized views improves AUROC and AUPRC for sparse subspace anomalies. **B.** Increasing the random-feature dimension D improves accuracy with diminishing returns, illustrating the accuracy-efficiency tradeoff of the global density branch.

To isolate the mechanisms behind RGLD, we conduct controlled synthetic experiments in addition to the ADBench. The synthetic data contain sparse subspace anomalies: normal samples are drawn from an isotropic Gaussian distribution, while anomalies are shifted along a small subset of informative dimensions. This setup allows us to test whether increasing the number of randomized views helps expose subspace-specific anomaly signals. Figure 5 studies two key efficiency parameters in RGLD: the number of randomized views and the random-feature dimension D. Panel A shows that increasing the number of randomized views improves both AUROC and AUPRC, indicating that the ensemble contributes complementary anomaly evidence rather than merely adding redundant detectors. Runtime grows smoothly with the number of views, and the default configuration lies near the high-performance region of the curve. This supports the central design principle of RGLD: aggregating many inexpensive randomized views yields a stronger anomaly ranking. Panel B studies the accuracy-efficiency tradeoff controlled by the random-feature dimension D. Larger D improves both AUROC and AUPRC, consistent with the random-feature approximation analysis in Section 5. The improvement is largest when increasing D from very small values and then gradually saturates, indicating diminishing returns at larger feature dimensions. Together, these results show that RGLD provides a controllable accuracy-efficiency tradeoff through both the number of randomized views and the random-feature dimension.

G RGLD PSEUDOCODE AND IMPLEMENTATION DETAIL

Algorithm 1 summarizes RGLD. The global branch fits randomized density estimators using random-feature mean embeddings, while the local neighbor branch fits sampled-reference local isolation estimators. Scores are rank-aggregated within each branch and then across branches.

Algorithm 1: RGLD

Input: Unlabeled data $X = \{x_i\}_{i=1}^{n}$; global views T_g; local views T_ℓ; bandwidth multipliers $\{\alpha_m\}_{m=1}^{M}$; reference size c; neighbors k; Random feature dimension D

Output: Anomaly scores $s(x_i)$

Preprocess X with dataset statistics;

Global density branch

for $t = 1, \ldots, T_g$ **do**

- Sample feature subset and random projection; obtain projected data Z_t
- Estimate base bandwidth $\widehat{\sigma}_t$ from sampled pairwise distances in Z_t
- Construct multi-scale bandwidths $\sigma_{t,m} = \alpha_m \widehat{\sigma}_t$
- Fit random-feature mean embeddings $\{\mu_{t,m}\}_{m=1}^{M}$
- Score samples by negative random-feature density at each scale
- Standardize and average multi-scale scores to obtain a_t^{glob}

end

Aggregate global scores: $s_{\mathrm{glob}} = \mathrm{RankAgg}(a_1^{\mathrm{glob}}, \ldots, a_{T_g}^{\mathrm{glob}})$

local neighbor branch

for $t = 1, \ldots, T_\ell$ **do**

- Sample feature subset and random projection; obtain projected data Z_t^ℓ
- Sample reference set $C_t \subset Z_t^\ell$ with effective size $|C_t| = min(c, n)$
- Score each sample by the root mean squared distance to the k nearest points in C_t
- Store the resulting local score vector a_t^{loc}

end

Aggregate local scores: $s_{\mathrm{loc}} = \mathrm{RankAgg}(a_1^{\mathrm{loc}}, \ldots, a_{T_\ell}^{\mathrm{loc}})$

Final score

Return $s = \mathrm{RankAgg}(s_{\mathrm{glob}}, s_{\mathrm{loc}})$

When a scored sample belongs to the reference set C_t, its zero self-distance is retained. Its effect is limited because it contributes only one of the k distance terms and scores are aggregated across randomized views.

Label-free preprocessing selection: RGLD selects preprocessing using only training-set statistics. It chooses among standard scaling, robust scaling, and quantile transformation based on scale heterogeneity, tail heaviness, skewness, sparsity, and discreteness. Robust scaling is used as the safe default when feature geometry may be distorted by outliers or scale mismatch; quantile transformation is used for strongly skewed continuous data; otherwise, standard scaling is used. The exact preprocessing rules are provided in the supplementary code.

Fully unsupervised evaluation setting: RGLD operates without access to anomaly labels. Labels are not used for preprocessing selection, bandwidth estimation, hyperparameter tuning, model selection, or construction of the global and local scoring branches. The same fixed RGLD configuration is applied across all benchmark datasets, while preprocessing is selected using only unlabeled dataset statistics. Ground-truth labels are accessed only after anomaly scores have been produced, for computing evaluation metrics such as AUROC and AUPRC.

Automatic projection dimension: RGLD uses a fixed rule for random projection dimension based only on input dimension d: $r = 2$ for $d \leq 3$, $r = 5$ for $d \leq 20$, $r = 8$ for $d \leq 40$, $r = 12$ for $d \leq 100$, and $r = 16$ otherwise. The local neighbor branch uses the same rule.

H HYPERPARAMETERS FOR TWO RGLD VARIANTS

We use two RGLD variants in the experiments: **RGLD** and **RGLD-Fast**. RGLD is the default configuration used for the main benchmark, while RGLD-Fast reduces the number of randomized views, random-feature dimension, and reference size to improve runtime. The hyperparameters used for each variant are shown in Table 6.

Table 6: **Hyperparameters for RGLD and RGLD-Fast.** RGLD-Fast keeps the same design as RGLD but reduces the ensemble and random-feature budgets for lower runtime.

Hyperparameter	RGLD	RGLD-Fast
Random-feature dimension D	10	5
Bandwidth multipliers α_m	$\{0.25, 0.5, 1.0, 2.0\}$	$\{0.25, 0.5, 1.0, 2.0\}$
Global randomized views T_g	40	20
Projected-neighbor views T_ℓ	10	5
Reference-set budget c	400	200
Number of neighbors k	10	10

For both variants, each randomized view samples approximately 30% of the input features, subject to a minimum of 3 selected features, before applying a random projection. For each global randomized view, the base bandwidth is estimated as the median Euclidean distance in the projected space over 8192 sampled pairs for RGLD and 4096 sampled pairs for RGLD-Fast, and the final set of bandwidths is obtained by multiplying this view-specific base bandwidth by the fixed scale multipliers.

I PROOF OF PROPOSITION 1

For any nonempty feature set $A \subseteq [d]$, define its average anomaly contrast as

$$C(A) = \frac{1}{|A|}\mathbb{E}\left[\|(x^a - x^n)_A\|_2^2\right].$$

Let $S^\star$ be the informative feature set, with $|S^\star| = s$, and let S be a feature subset with $|S| = m$. By the assumptions of the proposition,

$$\mathbb{E}\left[\|(x^a - x^n)_S\|_2^2\right] = |S \cap S^\star|\mu_{\text{sig}} + |S \setminus S^\star|\mu_{\text{noise}}.$$

Hence,

$$C(S) = \frac{|S \cap S^\star|}{m}\mu_{\text{sig}} + \left(1 - \frac{|S \cap S^\star|}{m}\right)\mu_{\text{noise}}.$$

For the full feature space,

$$C([d]) = \frac{s}{d}\mu_{\text{sig}} + \left(1 - \frac{s}{d}\right)\mu_{\text{noise}}.$$

Therefore,

$$C(S) - C([d]) = \left(\frac{|S \cap S^\star|}{m} - \frac{s}{d}\right)(\mu_{\text{sig}} - \mu_{\text{noise}}).$$

Since $\mu_{\text{sig}} > \mu_{\text{noise}}$, the condition $|S \cap S^\star|/m > s/d$ implies $C(S) > C([d])$.

■

J PROOF OF PROPOSITION 2

We prove the result for a fixed randomized view t and bandwidth σ. Conditioning on the sampled feature subset and projection, the projected samples $z_i^{(t)}$ are fixed. Thus, the only randomness comes from the random features used to approximate the Gaussian kernel.

For simplicity, write $z = z_t(x)$ and $z_i = z_i^{(t)}$. The exact projected KDE score is

$$\widehat{p}_{t,\sigma}(x) = \frac{1}{N}\sum_{i=1}^{N} k_\sigma(z, z_i)$$

Using D i.i.d. cosine random features, the RGLD approximation can be written as

$$\widetilde{p}_{t,\sigma}(x) = \frac{1}{D}\sum_{\ell=1}^{D} h_\ell(x)$$

where

$$h_\ell(x) = \frac{2}{N}\sum_{i=1}^{N} \cos\left(\frac{\omega_\ell^\top z}{\sigma} + b_\ell\right)\cos\left(\frac{\omega_\ell^\top z_i}{\sigma} + b_\ell\right)$$

where the pairs (ω_ℓ, b_ℓ) are sampled independently, with Gaussian frequencies and uniform random phases. By the standard random Fourier feature theory, each feature gives an unbiased estimate of the Gaussian kernel, so

$$\mathbb{E}[h_\ell(x)] = \frac{1}{N}\sum_{i=1}^{N} k_\sigma(z, z_i) = \widehat{p}_{t,\sigma}(x)$$

Therefore, $\widetilde{p}_{t,\sigma}(x)$ is an unbiased estimator of $\widehat{p}_{t,\sigma}(x)$.

It remains to bound its deviation. Since each cosine term lies in $[-1, 1]$, we have $h_\ell(x) \in [-2, 2]$. The variables $\{h_\ell(x)\}_{\ell=1}^{D}$ are independent across random features. Applying Hoeffding's inequality to the average $\frac{1}{D}\sum_{\ell=1}^{D} h_\ell(x)$, for any $\epsilon > 0$,

$$\Pr\left(|\widetilde{p}_{t,\sigma}(x) - \widehat{p}_{t,\sigma}(x)| \geq \epsilon\right) \leq 2\exp\left(-\frac{D\epsilon^2}{8}\right)$$

Setting the right-hand side equal to δ gives, with probability at least $1 - \delta$,

$$|\widetilde{p}_{t,\sigma}(x) - \widehat{p}_{t,\sigma}(x)| \leq \sqrt{\frac{8\log(2/\delta)}{D}} = O\left(\sqrt{\frac{\log(1/\delta)}{D}}\right)$$

■

The proposition is stated for original cosine random Fourier features; in our implementation, QMC features are used as a variance-reduced alternative, while the bound serves as the standard RFF approximation upper bound.

K Proof of Proposition 3

Fix a projected view t, an evaluation point x, and radius $\rho > 0$. Let $B_t(x, \rho)$ be the set of projected training samples within distance ρ of $z_t(x)$, and let $b = |B_t(x, \rho)|$. By definition, $q_t(x, \rho) = b/n$. Let $m = |C_t|$ be the sampled reference-set size.

Since C_t is sampled uniformly without replacement from the n projected training samples, the probability that it contains no point from $B_t(x, \rho)$ is

$$\Pr\left[C_t \cap B_t(x, \rho) = \emptyset\right] = \frac{\binom{n-b}{m}}{\binom{n}{m}}$$

If $m > n - b$, the probability is 0, and the desired bound holds trivially. Assuming $m \leq n - b$, expanding the ratio gives

$$\frac{\binom{n-b}{m}}{\binom{n}{m}} = \prod_{j=0}^{m-1} \frac{n-b-j}{n-j}$$

For each $j \geq 0$, we have

$$\frac{n-b-j}{n-j} \leq \frac{n-b}{n} = 1 - \frac{b}{n} = 1 - q_t(x, \rho)$$

Therefore,

$$\Pr\left[C_t \cap B_t(x,\rho) = \emptyset\right] \le (1 - q_t(x,\rho))^m$$

Using $1 - u \le e^{-u}$ for $u \in [0,1]$, we obtain

$$\Pr\left[C_t \cap B_t(x,\rho) = \emptyset\right] \le \exp\left(-m q_t(x,\rho)\right) = \exp\left(-|C_t| q_t(x,\rho)\right)$$

Thus, with probability at least $1 - \exp(-|C_t| q_t(x,\rho))$, the reference set contains at least one projected training sample within distance ρ of $z_t(x)$. This proves the proposition.

■

Proposition 3 characterizes the sampling gap introduced by replacing full-reference kNN with sampled-reference kNN. In the full kNN score, a point x can use all training samples to find nearby support. In RGLD, x only searches within the sampled reference set C_t. The bound shows that if the projected neighborhood $B_t(x,\rho)$ contains non-negligible empirical mass $q_t(x,\rho)$, then C_t is unlikely to miss this neighborhood; the miss probability decays exponentially in $|C_t| q_t(x,\rho)$. Thus, when x is well supported in a projected view, the sampled-reference score is likely to find nearby points and remain small. Conversely, if the sampled-reference score is large, this suggests either that x has genuinely weak local support in that view or that the reference sample missed a very small neighborhood. RGLD mitigates the latter failure mode by aggregating multiple independently sampled projected-neighbor views and reference sets. Therefore, the proposition justifies sampled-reference kNN as a scalable approximation to local support.

L Scaling in Dataset Size and Dataset Dimension

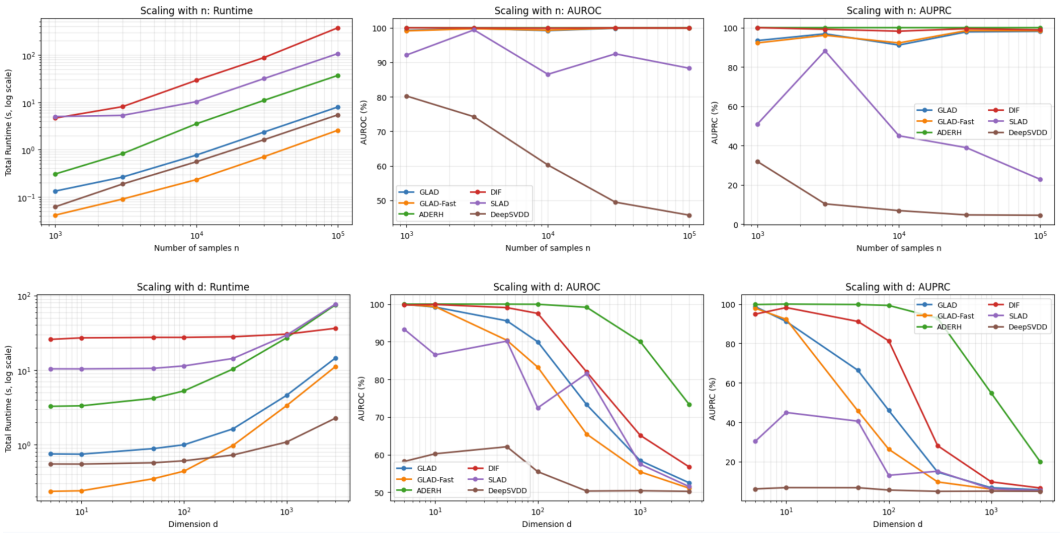


Figure 6: **Scaling with dataset size and ambient dimension.** Top: runtime, AUROC, and AUPRC as the number of samples n increases. Bottom: runtime, AUROC, and AUPRC as the ambient dimension d increases. RGLD and RGLD-Fast maintain strong performance and smooth runtime growth with n. As d becomes very large, accuracy decreases due to dilution of sparse anomaly signals by nuisance dimensions.

We further evaluate how RGLD scales with the number of samples n and the ambient dimension d using controlled synthetic data. We use synthetic data to isolate scaling behavior. Normal samples are drawn from a Gaussian mixture, while anomalies follow a local-shift mechanism where they are sampled from the same cluster structure and then shifted along a small randomly chosen subset of informative dimensions. We vary either the number of samples n or the ambient dimension d, keeping the anomaly ratio and shift magnitude fixed. In the dimension-scaling experiment, the anomaly signal

remains confined to a small number of informative features as d grows, making detection increasingly challenging. The goal of this experiment is to complement the main benchmark by isolating runtime and accuracy trends under increasing dataset size and dimensionality.

Figure 6 shows the results. When scaling the number of samples n, RGLD and RGLD-Fast maintain high AUROC and AUPRC across the full range while runtime grows smoothly. This behavior is consistent with the complexity analysis in Section 5: RGLD avoids explicit pairwise KDE and full-reference kNN computations, replacing them with randomized density and sampled-reference local-support estimates. RGLD-Fast further reduces runtime while preserving similar accuracy, illustrating the controllable accuracy–efficiency tradeoff of the randomized-view design.

When scaling the ambient dimension d, runtime increases for all methods, but RGLD remains efficient over most of the range. Accuracy decreases at very large d, especially in AUPRC, reflecting the increasing difficulty of detecting sparse anomaly signals as they are diluted by many nuisance dimensions. This stress test highlights both the benefit and limitation of randomized feature-bagged views: they improve efficiency and can expose subspace anomalies, but extremely high ambient dimension remains challenging when the informative anomaly signal occupies only a small fraction of features.

Interestingly, several deep methods (DIF, DeepSVDD) exhibit flatter runtime growth than statistical methods as the ambient dimension d increases. This likely reflects their use of fixed neural architectures, whose runtime is less sensitive to large input dimension. By contrast, statistical detectors often operate directly on distances, projections, or neighborhoods in the input space. Therefore, RGLD's main scaling advantage is most evident in sample size n, while very high-dimensional inputs suggest a direction for further optimization through sparser or more adaptive projections.

M Detailed Comparison with ADERH

ADERH is the strongest statistical baseline in our main benchmark, achieving the highest mean AUROC and AUPRC among the evaluated methods. Since both ADERH and RGLD are randomized statistical detectors, we provide a more detailed comparison between the two methods. Table 7 summarizes their aggregate performance over the 47 ADBench datasets. ADERH obtains slightly higher mean AUROC and AUPRC. In contrast, RGLD achieves more first-place wins among all evaluated methods in both AUROC and AUPRC, while being substantially faster. In particular, RGLD runs in 0.22 seconds on average, compared with 1.06 seconds for ADERH, corresponding to an average runtime advantage of approximately 4.86x.

Table 7: Head-to-head comparison between RGLD and ADERH over 47 ADBench datasets.

Metric	RGLD	ADERH
Mean AUROC	77.50	78.18
Mean AUPRC	38.59	39.72
Average rank	9.72	8.22
AUROC wins among all methods	8	4
AUPRC wins among all methods	5	3
Mean runtime (s)	0.22	1.06
Relative runtime	1.00x	4.86x slower

To assess whether the observed differences are statistically meaningful, we perform paired Wilcoxon signed-rank tests across the 47 datasets. For AUROC and AUPRC, paired differences are computed as RGLD minus ADERH, so positive values indicate that RGLD has higher detection accuracy. For runtime, paired differences are computed as ADERH runtime minus RGLD runtime, so positive values indicate that RGLD is faster. The results are shown in Table 8.

The paired tests show that ADERH has slightly higher mean AUROC and AUPRC, but do not detect a statistically significant accuracy difference at the 0.05 level; this should not be interpreted as evidence of equivalence. By contrast, the runtime difference is highly significant, with RGLD being faster across the benchmark. These results support the main accuracy–efficiency claim of the paper:

Table 8: Paired statistical comparison between RGLD and ADERH across 47 datasets. Positive differences favor RGLD.

Metric	Median Diff.	Mean Diff.	p-value	Interpretation
AUROC	-1.45	-0.68	0.162	Not significant
AUPRC	-1.42	-1.13	0.228	Not significant
Runtime (s)	0.77	0.84	2.39×10^{-9}	RGLD faster

RGLD remains competitive with the strongest statistical baseline in detection accuracy while offering substantially lower runtime.

The two methods also rely on different randomized inductive biases. ADERH builds an ensemble from randomized hypersphere-pair comparisons, whereas RGLD explicitly combines global random-feature density support with local projected-neighbor isolation across feature-bagged randomized views. Thus, RGLD provides a complementary randomized statistical approach: rather than relying on a single randomized scoring principle, it aggregates global and local density evidence across multiple views and scales.

N Full Pareto Frontier Graph

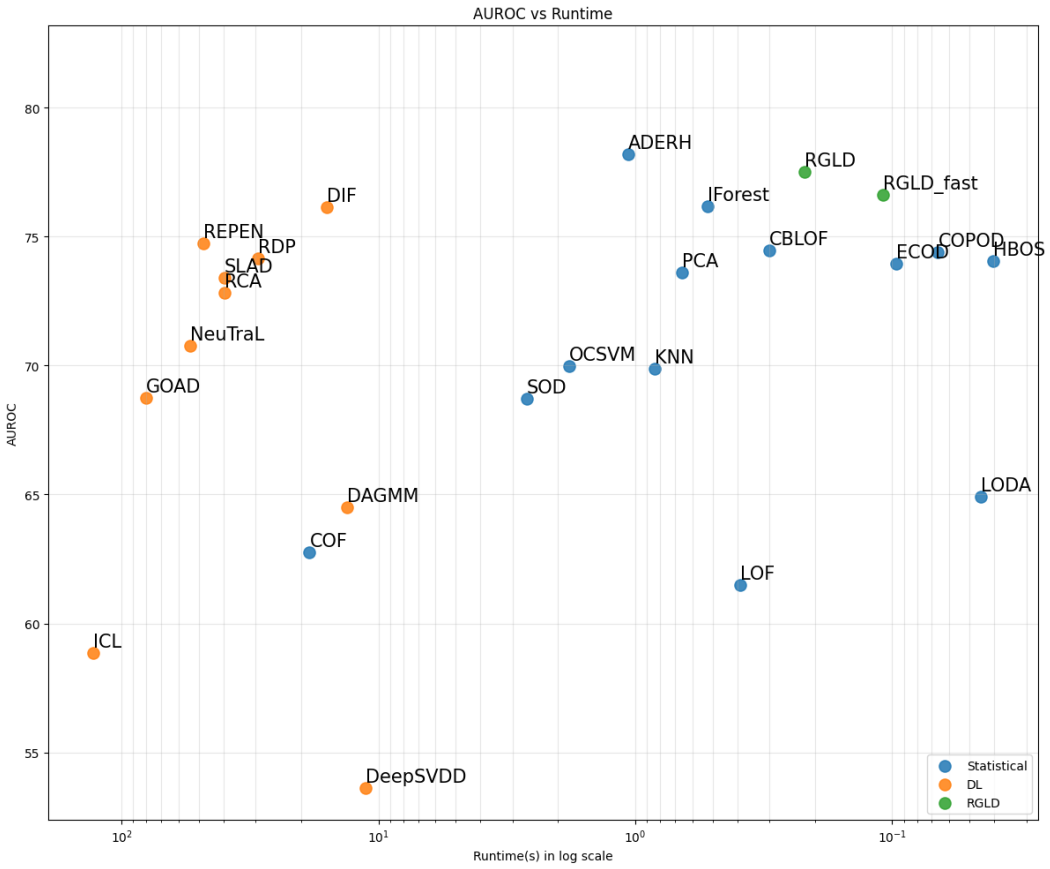


Figure 7: Full Pareto frontier graph of RGLD and 23 unsupervised anomaly detection methods using the average AUROC and time over 47 ADBench Datasets.

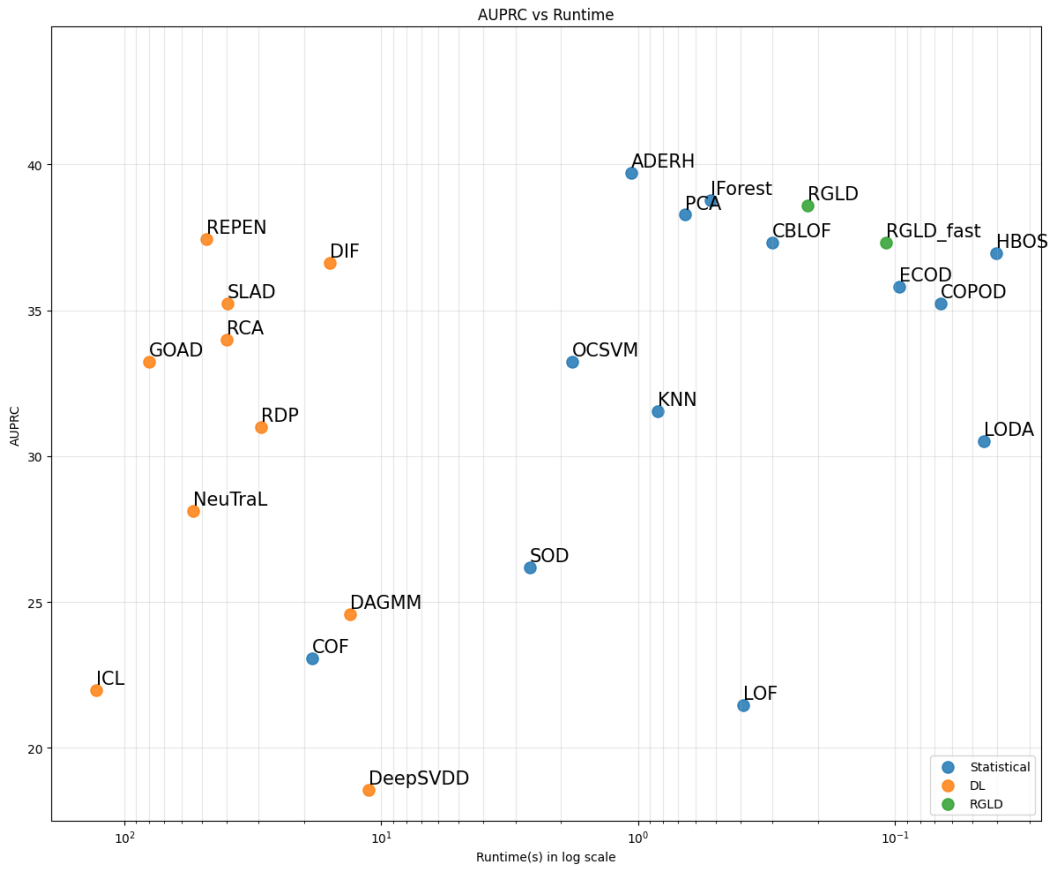


Figure 8: Full Pareto frontier graph of RGLD and 23 unsupervised anomaly detection methods using the average AUPRC and time over 47 ADBench Datasets.

O Full Benchmark Results

This appendix reports the full per-dataset benchmark results across RGLD and 23 deep learning and statistical baselines. We report AUROC, AUPRC, and total runtime.

Table 9: Full benchmark AUROC results over 47 ADBench tabular datasets.

Datasets	PCA	OCSVM	LOF	CBLOF	COF	HBOS	*k*NN	SOD	COPOD	ECOD	DAGMM	LODA	IForest	ADERH	DeepSVDD	DIF	RCA	GOAD	NeuTraL	ICL	SLAD	REPEN	RDP	RGLD
ALOI	56.65	55.85	66.63	55.22	64.68	52.63	61.47	61.09	53.75	56.6	51.96	51.33	56.66	54.39	50.29	56.48	54.38	57.08	56.71	52.07	53.67	57.39	56.99	51.79
annthyroid	66.25	57.23	70.2	62.26	65.92	60.15	71.69	77.38	76.8	78.03	56.53	41.02	82.01	63.96	76.62	66.34	60.55	57.91	62.98	37.7	75.88	65.23	62.2	93.53
backdoor	80.13	86.2	85.68	81.16	73.03	71.43	80.82	69.54	80.97	86.33	56.26	69.22	72.15	89.68	55.16	92.24	86.18	78.8	86.47	63.23	87.47	85.07	85.78	75.99
breastw	95.13	80.3	40.61	96.81	38.84	98.94	97.01	93.97	99.68	99.17	N/A	98.49	98.32	99.14	65.66	96.95	98.88	11.44	57.66	75.26	49.07	98.89	97.58	98.79
campaign	72.78	65.52	58.85	66.61	57.26	78.61	72.1	69.04	77.69	76.78	56.08	51.43	71.71	68.3	48.7	65.86	64.76	71.78	73.05	70.82	71.36	63.72	68.51	80.12
cardio	95.55	93.91	66.33	89.93	71.41	84.67	76.64	73.25	92.35	94.44	75.01	90.34	93.19	94.95	58.96	93.27	90.01	93.14	82.24	74.22	82.07	86.42	87.72	91.01
Cardiotocography	74.67	77.86	59.51	64.54	53.77	60.86	56.23	51.69	67.02	68.92	62.01	73.65	67.57	66.56	53.53	69	72.63	69.45	55.86	50.49	59.73	60.46	61.15	70.89
celeba	79.38	70.7	38.55	73.99	38.58	76.18	59.63	47.85	75.68	72.82	44.74	60.11	70.41	73.11	50.36	72.47	66.25	63.33	71.95	58.07	76.98	76.14	66.47	78.75
census	68.74	54.58	47.19	59.41	41.35	64.94	66.75	62.31	69.07	68.44	59.29	36.86	59.52	64.7	51.07	62.45	62.35	60.21	66.49	68.35	62.73	54.56	67.36	63.32
cover	93.73	92.62	84.58	89.3	76.91	80.24	85.97	74.46	88.64	93.42	89.89	92.34	86.74	87.23	46.2	92.63	81.47	95.17	92.89	65.98	91.19	90.32	94.98	83.22
donors	83.15	71.93	55.49	60.44	70.54	78.23	81.09	55.21	81.76	74.45	70.57	24.86	77.68	76.1	50.27	72.11	82.19	46.47	54.66	60.02	50.48	74.23	76.28	85.95
fault	46.02	47.69	58.93	64.06	62.1	51.28	72.98	68.11	43.88	43.41	45.86	41.71	57.02	70.25	51.67	67.11	61.6	52.94	69.69	68.62	66.63	65.69	70.53	49.93
fraud	90.35	90.62	94.92	91.7	93.05	90.29	93.56	94.97	88.32	89.85	89.53	88.99	90.38	92.08	64.98	91.3	86.54	99.69	98.48	99.71	99.71	98.98	99.31	90.06
glass	66.29	35.36	69.2	82.94	72.24	77.23	82.29	73.36	72.43	75.7	76.09	73.13	77.13	84.72	47.49	83.95	62.33	79.84	82.41	61.25	87.67	78.06	87.35	80.85
Hepatitis	75.95	67.75	38.02	66.4	41.45	79.85	52.76	68.17	82.05	79.67	54.8	64.87	69.75	74.31	50.96	67.13	59.48	72.31	59.14	26.74	67.38	45.91	60.66	77.82
http	99.72	99.59	27.46	99.6	88.78	99.53	3.37	78.04	99.29	98.1	N/A	12.48	99.96	99.52	69.05	99.59	99.49	98.99	98.7	49.98	99.73	99.06	99.1	99.99
InternetAds	61.67	68.28	65.83	70.58	63.79	68.03	69.99	61.85	67.05	67.1	N/A	55.38	69.01	66.85	60.2	68.67	70.45	46.46	56.04	53.91	59.09	67.75	63.84	70.76
Ionosphere	79.19	75.92	90.59	90.72	86.76	62.49	88.26	86.41	79.34	75.59	73.41	78.42	84.5	89.94	50.89	90.58	77.8	85.27	93.73	33.78	95.2	80.4	92.6	88.35
landsat	35.76	36.15	53.9	63.55	53.5	55.14	57.95	59.54	41.55	56.61	43.92	38.17	47.64	59.56	63.61	50.5	29.63	45.84	57.44	65.76	64.25	55.09	47.63	50.35
letter	50.29	46.18	84.49	75.62	80.03	59.74	86.19	84.09	54.32	50.76	50.42	50.24	61.07	72.72	56.64	68.38	58.74	52.39	83.86	72.01	73.39	67.87	76.88	65.65
Lymphography	99.82	99.54	89.86	99.83	90.85	99.49	55.91	72.49	99.48	99.52	72.11	85.55	99.81	99.89	32.29	99.78	99.85	92.86	75.94	64.74	73.95	100	81.88	99
magic.gamma	67.22	60.65	68.51	75.13	66.64	70.86	82.38	75.4	68.33	64.36	58.58	68.02	73.25	80.94	60.26	76.82	74.51	65.63	65.8	64.26	68.5	67.82	75.23	69.04
mammography	88.72	84.95	74.39	83.74	77.53	86.27	84.53	81.51	90.69	90.75	N/A	83.91	86.39	82.82	56.98	82.53	85.54	29.24	36.63	40.31	26.41	63.93	79.65	71.45
mnist	85.29	82.95	67.13	79.45	70.78	60.42	80.58	60.1	77.74	84.6	67.23	72.27	80.98	86.99	53.4	79.24	72.77	88.44	74.61	83.85	93.13	79.6	79.23	65.8
musk	100	80.58	41.18	100	38.69	100	69.89	74.09	94.2	95.11	76.85	95.11	99.99	100	43.52	99.96	99.15	99.36	97.74	100	99.99	100	91.53	97.82
optdigits	51.65	54	56.1	87.51	49.15	81.63	41.73	58.92	68.71	61.04	62.57	61.74	70.92	76.74	38.89	60.15	78.06	48.58	89.38	62.3	61.12	43.88	63.3	66.31
PageBlocks	90.64	88.76	75.9	85.04	72.65	80.58	81.94	77.75	88.05	90.92	89.61	83.34	89.57	89.37	57.77	90.34	71.31	88.46	89.41	73.24	93.4	88.23	84.02	91.23
pendigits	93.73	93.75	47.99	90.4	45.07	93.04	72.95	66.29	90.68	91.22	64.22	89.1	94.76	96.5	39.92	95.32	88.53	93.77	85.99	50.6	91.55	93.33	82.07	97.61
Pima	70.77	66.92	65.71	71.42	61.05	71.07	73.43	61.25	69.1	51.54	55.92	65.93	72.87	78.36	51.03	75.2	77.63	59.81	57.49	47.19	56.98	60.6	65.31	73.58
satellite	59.62	59.02	55.88	71.32	54.74	74.8	65.18	63.96	63.2	75.06	62.33	61.98	70.43	78.85	55.3	70.31	53.36	68.74	72.45	53.12	70.1	71.97	63.17	71.9
satimage-2	97.62	97.35	47.36	99.84	56.7	97.65	92.6	83.08	97.21	97.11	96.29	97.56	99.16	99.87	53.14	99.84	98.36	98.23	95.92	66.8	94.27	99.26	98.19	99.37
shuttle	98.62	97.4	57.11	83.48	51.72	98.63	69.64	69.51	99.35	99.4	97.92	60.95	99.56	98.27	52.05	96.91	97.95	99.17	85.64	36.07	99.37	99.46	97.49	98.22
skin	45.26	49.45	46.47	69.49	41.66	60.15	71.46	60.35	47.55	39.09	N/A	45.75	68.21	78.89	44.05	77.98	69.83	56.4	43.8	42.75	81.03	63.85	70.21	75.55
smtp	88.41	80.7	71.84	79.68	79.6	70.52	89.62	59.85	79.09	71.86	71.32	67.43	89.73	78.89	78.24	80.74	63.14	100	100	50	100	99.93	100	94.31
SpamBase	54.66	52.47	43.33	54.97	40.96	64.74	53.35	52.35	70.09	66.89	N/A	41.99	64.76	58.7	53.55	56.04	55.83	19.86	35.64	42.83	29.76	45.24	58.39	73.52
speech	50.79	50.19	52.48	50.58	55.97	50.59	51.03	55.86	52.89	51.58	52.75	49.84	50.74	53.03	53.43	51.35	49.96	53.43	55.34	41.14	55.6	55.15	53.45	51.26
Stamps	91.47	83.86	51.26	68.18	53.81	90.73	68.61	73.26	93.4	91.41	88.88	87.18	91.21	91.18	55.84	86.7	75.13	87.52	75.23	46.97	91.5	86.84	72.49	89.73
thyroid	96.34	87.92	86.86	94.73	90.87	95.62	95.93	92.81	94.3	97.78	79.75	74.3	98.3	94.9	49.64	96.01	86.4	94.56	98.34	73.63	98.89	95.2	94.26	98.99
vertebral	37.06	37.99	49.29	41.41	48.71	28.56	33.79	40.32	25.64	37.51	53.2	30.57	36.66	35.48	36.67	35.34	41.39	50.57	56.62	45.19	52.37	45.38	52.51	32.22
vowels	65.29	61.59	93.12	89.92	94.04	72.21	97.26	92.65	53.15	45.81	60.58	70.36	73.94	93.37	52.49	88.53	80.45	53.78	85.5	75.89	90.16	91.41	89.51	80.5
Waveform	65.48	56.29	73.32	72.42	72.56	68.77	73.78	68.57	75.03	73.25	49.35	60.13	71.47	80.93	54.47	73.59	73.43	63.84	58.06	44.72	70.84	80.99	52.51	76.66
WBC	98.2	99.03	54.17	99.46	60.9	98.72	90.56	94.6	99.11	99.11	N/A	96.91	99.01	99.24	55.5	95.33	97.97	93.16	37.09	55.32	63.54	99.6	95.13	98.49
WDBC	99.05	98.86	89	99.32	96.26	99.5	91.72	91.9	99.42	97.2	76.67	98.26	98.95	98.34	65.69	96.57	96.72	97.71	78.16	85.32	87.27	99.37	93.32	99.06
Wilt	20.39	31.28	50.65	32.54	49.66	32.49	48.42	53.25	33.4	39.43	37.29	26.42	41.94	30.44	46.08	30.03	38.27	35.9	81.08	49.74	56.07	38.22	30.47	59.94
wine	84.37	73.07	37.74	25.86	44.44	91.36	44.98	46.11	88.65	71.34	61.7	90.12	80.37	78.55	59.52	70.11	82.72	68	42.07	58.06	75.44	84.6	48.54	79.01
WPBC	46.01	45.35	41.41	44.77	45.88	51.24	46.59	51.14	49.34	46.83	47.8	49.31	46.63	49.46	49.79	47.3	47.38	44.46	48.77	57.41	55.22	44.43	48.38	47.47
yeast	41.15	41	45.31	44.85	44.48	39.64	39.06	42.46	36.99	39.61	41.11	44.58	37.76	36.17	47.92	38.85	41.17	40.5	42.4	46.33	39.16	42.33	42.1	37.13

Table 10: Full benchmark AUPRC results over 47 ADBench tabular datasets.

Datasets	PCA	OCSVM	LOF	CBLOF	COF	HBOS	*k*NN	SOD	COPOD	ECOD	DAGMM	LODA	IForest	ADERH	DeepSVDD	DIF	RCA	GOAD	NeuTraL	ICL	SLAD	REPEN	RDP	RGLD
ALOI	4.17	5.02	8.08	4.46	6.85	3.69	6.02	5.97	3.62	3.9	4.33	4.53	3.9	4.75	4.01	4.47	4.67	6.44	7.66	3.54	5.55	5.88	6.7	3.33
annthyroid	16.12	10.37	15.71	13.69	14.39	16.99	16.74	18.84	16.58	24.65	9.64	7.06	30.47	13.59	21.95	16.44	13.21	11.46	16.34	10.79	29.53	13.97	14.62	48.96
backdoor	31.29	9.69	26.14	6.96	24.68	4.91	45.22	39.41	7.69	11.25	6.5	14.51	4.75	20.78	12.85	25.5	9.49	10.52	48.79	32.69	44.28	10.57	19.88	5.17
breastw	95.11	82.7	28.55	91.54	27.6	97.71	92.19	84.88	99.4	98.54	N/A	97.04	96.04	98.07	50.92	88.97	97.5	20.21	34.8	59.54	30.65	97.14	92.24	97.25
campaign	27.9	29.22	14.51	23.99	13.01	37.99	27.18	18.88	38.58	37.4	14.62	13.47	32.26	27.76	11.6	22.85	22.35	32.77	30.35	26.86	30.82	26.71	26.52	35.45
cardio	66.06	62.89	23.79	61.95	28.67	52.1	40.72	28.54	60.42	68.59	28.92	53.41	59.95	64.25	22.5	55.18	53.4	55.86	39.41	22.21	39.89	47.28	41.17	45.45
Cardiotocography	47.95	52.61	30.66	45.44	28.21	38.28	34.79	27.99	40.46	43.57	30.61	48	41.47	44.56	34.03	42.79	44.52	46.61	37.63	31.8	43.25	40.28	38.7	40.02
celeba	15.89	10.73	1.71	11.33	1.77	13.82	3.14	2.66	13.69	12.37	1.95	4.04	8.96	7.14	2.34	9.08	5.05	3.67	4.37	2.61	5.07	5.71	3.88	13.07
census	10.02	6.76	5.45	7.44	4.88	8.69	9	8.52	9.92	9.72	8.71	5.01	7.78	8.32	6.87	8.04	8.17	7.46	10.09	10.8	8.23	6.93	10.33	8.31
cover	9.8	11.41	8.12	5.83	4	6.83	6.16	3.88	11.37	15.63	27.59	13.06	8.85	6.15	8.12	10	3.68	12.62	12.49	1.88	12.46	5.89	20.73	7.34
donors	17.9	9.86	7.88	6.89	8.8	23.36	14.75	9.69	21.58	14.17	10.53	3.78	12.74	11.48	6.38	10.09	17.06	5.2	6.07	6.77	5.63	10.14	11.43	25.2
fault	32.76	38.44	38.38	43.98	41.56	36.47	54.45	48.01	30.54	30.82	33.48	31.03	41.09	54.21	39.15	51.61	45.49	46.74	54.42	50.77	54.91	52.22	54.07	33.09
fraud	22.91	47.58	47.4	47.52	22.86	25.89	47.3	31.37	42.82	42.99	21.32	46.37	21.67	26.29	8.97	38.55	32.38	60.49	23.17	41.52	57.74	54.75	42.3	26.53
glass	10.05	8.02	20.11	13.84	11.81	11.82	20.26	18.73	9.78	18.43	24.58	13.37	10.99	15.05	8.72	15.1	6.86	26.55	14.23	10.6	20.65	15.57	22.29	11.48
Hepatitis	36.65	29.44	13.69	31.54	14.39	37.73	21.95	24.89	41.5	37.82	22.96	30.9	26.25	28.45	22.17	24.87	18.02	28.72	26.01	12.16	24.03	15.98	31.75	31.68
http	56.43	46.86	3.82	47.53	9.57	44.79	0.7	8.32	35.19	16.61	N/A	0.67	90.83	43.52	29.3	48.87	26.83	31.33	26.12	0.5	61.61	33.44	21.65	95.69
InternetAds	32.55	54.68	40.49	58.13	38.67	53.97	43.23	27.69	50.97	51.07	N/A	23.89	48.6	49.94	27.91	52.12	57.87	16.13	20.55	20.77	22.93	47.15	34.97	55.38
Ionosphere	73.92	74.54	88.06	90.27	82.91	41.78	90.41	85.87	69.89	65.99	64.98	73.04	80.41	88.2	41.79	88.76	75.01	76.13	92.31	35.6	92.42	69.43	90.26	85.49
landsat	16.18	16.21	24.69	30.97	24.95	22.03	24.65	26.38	17.48	25.17	24.48	18.86	19.81	22.99	38.83	20.58	13.98	27.02	28.09	38.58	52.35	22.94	20.03	21
letter	6.86	6.1	34.02	14.8	21.43	8.38	30	28.63	6.77	6.94	11.68	6.87	8.49	12.76	9.29	12.03	8.27	12.87	32.23	19.36	20.97	12	15.87	10.72
Lymphography	97.02	93.59	23.08	97.62	36.68	91.83	38.69	22.65	88.68	90.87	19.52	44.54	97.31	98.08	4.58	95.24	97.19	54.72	11.02	11.9	10.14	100	19.25	83.24
magic.gamma	59.27	51.43	54.76	68.85	54.12	62.41	75.63	67.89	59.18	54.38	46.92	58.49	64.72	73.51	49.17	69.5	64.8	59.17	56.15	52.79	65.01	63.78	68.15	61.97
mammography	19.25	12.94	9.8	11.14	11.14	21.31	15.91	13.41	40.67	41.28	N/A	14.75	20.67	10.38	6.26	11.76	18.8	1.52	1.67	3.46	1.45	3.41	10.99	5.75
mnist	39.93	33.2	20.9	28.82	25.51	12.51	35.53	19.15	21.35	31.93	23.75	25.86	27.71	37.91	19.72	24.59	27.13	46.2	33.04	34.42	55.84	37.68	31.54	14.37
musk	99.89	10.61	2.82	100	2.61	100	9.65	7.59	34.79	34.95	32.75	47.6	99.61	100	5.39	98.21	85.36	88.36	61.07	100	99.64	100	37.68	65.16
optdigits	2.76	2.92	6.06	10.08	4.42	10.03	3.06	4.39	4.36	3.43	5.59	3.95	5.09	5.88	2.5	3.65	7.33	2.61	12.05	4.44	3.67	2.42	4.08	4.46
PageBlocks	51.71	49.14	39.64	49.65	41.02	33.32	45.39	37.83	37.65	49.3	53.25	51.29	46.04	53.37	31.45	52.47	47.12	49.78	52.38	29.02	58.44	53.89	47.98	51.29
pendigits	23.65	23.52	3.78	17.27	2.89	29.27	6.5	4.46	21.22	23.07	4.67	18.71	26.05	30.59	2.45	23.91	10.4	33.76	8.69	2.87	12.81	36.69	7.69	49.47
Pima	54.03	50	47.18	53.19	44.7	56.61	55.14	48.24	55.19	37.3	41.55	44.09	55.82	59.35	35.87	56.77	59.54	47.9	45.89	37.57	46.28	52.38	50.86	57.54
satellite	59.64	57.61	37.68	61.48	39.7	67.25	50.01	47.23	56.58	65.94	58.33	61.94	65.92	72.26	40.11	59.94	43.59	69.61	60.96	45.59	63.46	68.37	50.76	68.61
satimage-2	85.69	82.71	4.29	97.09	8.8	78.04	39.14	26.11	76.55	63.25	22.07	80.52	93.45	97.47	3.08	94.61	95.25	74.65	19.49	4.36	10.09	91.28	41.82	92.93
shuttle	92.35	85.29	13.76	60.98	12.17	96.4	20.38	20.27	96.56	95.76	93.2	48.75	97.62	89.31	15.86	65.68	95.11	95.93	43.82	8.41	97.95	97.95	84.38	89.06
skin	17.4	19.03	18.25	29.82	16.38	23.7	28.72	24.61	17.99	15.96	N/A	18.44	26.08	34.65	18.48	33.95	30.02	21.31	20.28	20.71	36.84	23.83	27.55	31.55
smtp	66.7	18.9	20.69	26.68	61.13	35.2	66.7	33.36	1.08	50.01	50.03	35.77	1.24	50.02	50.02	50.02	18.52	100	100	0.03	100	33.33	100	4.16
SpamBase	41.57	40.12	35.16	41.18	34.73	50.03	41.42	40.03	56.68	53.95	N/A	35.88	51.75	43.15	42.23	42.07	41.97	26.8	32.26	34.76	30.05	37.03	43.39	65.1
speech	1.97	1.96	2.52	1.99	2.25	2.09	2.02	2.13	1.94	1.77	2.03	1.79	2.31	2.02	5.12	2.33	3.05	2.63	2.13	1.39	1.9	2.2	3.33	1.99
Stamps	41.09	31.39	21.29	23.66	16.5	35.24	23.53	20.28	43.1	38.17	43.72	34.6	39.49	39.56	11.4	33.23	20.44	26.24	16.61	12.04	33.57	24.48	15.34	32.87
thyroid	44.34	21.23	20.81	29.95	28.5	50.98	34.98	23.56	19.64	54.05	16.06	14.68	63.11	31.12	2.5	41.26	26.72	36.09	57.2	46.1	82.81	31.63	38.54	67.46
vertebral	10.49	10.94	14.24	11.58	13.85	9.23	10.57	11.79	8.89	11.24	15.24	9.68	10.46	10.28	10.49	10.29	10.54	11.44	14.62	11.99	13.7	9.6	12.72	9.34
vowels	8.92	8.24	34.42	22.12	55.96	13.41	63.41	38.88	4.14	3.92	12.22	13.82	15.12	43.18	4.99	33.72	20.37	11.09	14.72	16.98	28.95	22.6	20.61	13.41
Waveform	5.79	4.37	11.33	18.98	14.11	5.86	13.04	9.66	6.9	6.86	3.11	4.71	6.24	14.35	4.83	7.83	8.44	4.03	4.22	2.9	38.45	22.03	3.27	10.29
WBC	82.29	89.87	5.57	92.27	9.73	73.56	66.55	54	86.19	86.19	N/A	78.67	90.49	90.54	6.38	47.61	77.4	40.24	4.09	7.74	6.67	94.04	32.6	80.44
WDBC	75.46	71.88	14.93	79.62	50.52	88.98	43.72	35.6	84.78	57.91	18.48	66.11	78.53	59.26	6.57	47.97	44.04	48.58	5.96	21.61	12.64	76.26	18.51	71.47
Wilt	3.13	3.62	5.05	3.64	4.98	3.84	4.73	5.53	3.69	4.14	4	3.36	4.23	3.53	4.67	3.51	3.99	3.95	12.84	5.33	5.66	3.99	3.52	6.04
wine	30.87	21.56	7.77	5.83	8.45	43.08	8.43	7.95	45.71	18.37	17.51	48.82	25.96	17.94	21.14	14.17	23.73	13.7	6.78	10.84	14.27	24.67	7.23	22.41
WPBC	23.01	22.93	20.29	21.32	21.3	23.04	21.49	25.37	22.81	21.38	22.49	25.58	22.42	22.88	26.24	22.4	23.77	20.79	25.64	34.09	32.75	20.83	22.2	21.64
yeast	29.9	29.84	31.64	30.93	31.27	32.75	29.33	29.96	30.71	31.36	29.92	33.29	29.8	27.81	33.03	28.5	29.63	32.73	32.97	32.02	29.94	31.32	33.34	30.88

Table 11: Full benchmark Runtime (s) results over 47 ADBench tabular datasets.

Datasets	PCA	OCSVM	LOF	CBLOF	COF	HBOS	*k*NN	SOD	COPOD	ECOD	DAGMM	LODA	IForest	ADERH	DeepSVDD	DIF	RCA	GOAD	NeuTraL	ICL	SLAD	REPEN	RDP	RGLD
ALOI	0.23	3.86	1.65	0.83	27.69	0.94	1.62	6.05	0.1	0.13	31.17	0.06	0.39	1.56	20.55	34.42	95.16	159.27	137.83	158.46	37.88	100.33	43.15	0.29
annthyroid	0.01	1.18	0.17	0.39	5.13	0	0.19	2.25	0.02	0.03	17.89	0.05	0.27	1.23	12.82	20.38	63.47	110.87	84.46	268.3	25.93	67.19	36.41	0.21
backdoor	0.73	6.43	1.85	0.14	121.26	0.06	3.51	5.15	N/A	N/A	24.8	0.08	2.02	2.15	19.25	33.8	63.06	162.97	118.98	60.04	40.91	100.84	51.66	0.5
breastw	0	0.03	0.01	0.12	0.15	0.01	0.02	0.13	0.01	0	N/A	0.02	0.12	0.2	1.79	2.42	8.93	15.78	11.05	38.06	48.75	7.63	7.08	0.06
campaign	0.15	4.38	1.71	0.29	45.75	0.02	2.3	4.85	0.13	0.19	18.41	0.06	0.79	1.38	24.39	33.6	82.43	155.97	118.96	118.81	52.13	101.46	78.8	0.37
cardio	0.02	0.1	0.07	0.23	0.38	0	0.07	0.28	0.02	0.02	3.4	0.04	0.16	0.4	4.98	6.46	17.53	28.82	25.31	39.18	22.68	14.72	9.63	0.09
Cardiotocography	0.03	0.15	0.08	0.24	0.51	0	0.1	0.38	0.02	0.02	3.42	0.04	0.17	0.42	4.49	9.1	17.24	32.9	28.43	45.55	17.88	16.16	11.19	0.09
celeba	0.1	3.65	1.52	0.29	27.96	0.02	1.99	4.8	0.07	0.1	19.79	0.07	0.65	1.45	23.77	35.23	77.89	160.95	127.82	162.91	35.39	97.78	69.31	0.32
census	4.52	18.94	1.99	0.65	316.96	0.17	15.79	5.65	N/A	N/A	33.12	0.1	5.7	3.22	25.35	35.22	72.15	167.27	127.84	172.04	46.93	104.73	86.05	1.15
cover	0.04	3.21	0.37	0.34	12.37	0.01	0.38	4.04	0.05	0.07	32.81	0.06	0.38	1.58	22.42	31.74	89.71	154.97	140.79	332.33	43.72	97.32	67.16	0.28
donors	0.04	2.96	0.29	0.14	11.03	0	0.33	3.74	0.02	0.04	34.5	0.06	0.4	1.49	21.07	28.95	88.05	157.72	126.86	302.71	37.32	99.51	72.77	0.24
fault	0.06	0.17	0.06	0.26	0.59	0.01	0.08	0.33	0.02	0.04	6.49	0.04	0.18	0.4	4.57	6.97	12.59	30.13	28.15	34.38	21.32	16.45	17.83	0.09
fraud	0.22	4.34	1.75	0.42	30.76	0.01	1.69	4.81	0.15	0.25	34.85	0.06	0.46	1.53	24.06	33.54	82.52	155.65	129.69	173.66	53.94	99.55	82.73	0.32
glass	0	0.04	0.01	0.2	0.17	0	0.01	0.12	0.02	0	2.47	0.01	0.13	0.2	2.45	2.63	8.54	15.38	11.36	40.52	30.88	7.03	8.35	0.06
Hepatitis	0.03	0.06	0.03	0.2	0.18	0.01	0.04	0.13	0	0.01	2.63	0.02	0.14	0.25	2.31	2.66	9.18	15.69	13.79	20.85	30.08	7.49	8.39	0.06
http	0	2.93	0.11	0.19	11.14	0	0.16	3.93	0.02	0.02	N/A	0.05	0.35	1.56	22.28	28.06	91.43	157.66	100.9	453.11	35.85	100.82	66.36	0.26
InternetAds	15.68	1.86	0.14	0.46	40.37	0.2	1.34	0.75	0.56	0.79	N/A	0.07	2.91	1.41	4.66	6.83	13.49	36.62	19.31	44.88	27.59	17.66	14.24	0.61
Ionosphere	0.08	0.04	0.03	0.19	0.23	0	0.04	0.14	0.02	0.02	2.6	0.03	0.15	0.31	1.64	2.54	6.21	16.4	9.95	19.09	39.06	6.89	6.56	0.07
landsat	0.28	1.57	0.69	0.4	15.3	0.01	0.73	2.2	0.1	0.16	21.95	0.05	0.38	1.13	11.72	19.31	40.87	101.35	60.07	111.3	56.6	59.29	49.12	0.25
letter	0.13	0.08	0.04	0.2	0.49	0	0.07	0.25	0.02	0.03	5.21	0.05	0.16	0.37	3.21	6.66	15.51	25.22	17.78	28.14	18.55	12.48	13.57	0.1
Lymphography	0.03	0.04	0.01	0.17	0.19	0	0.03	0.13	0.01	0.01	2.61	0.02	0.14	0.26	3.01	3.05	6.59	16.28	11.54	26.35	35.15	7.07	7.15	0.07
magic.gamma	0.04	3.25	0.23	0.39	12	0	0.35	6.42	0.04	0.09	32.91	0.07	0.37	1.58	22.45	28.68	72.77	158.06	79.75	313.71	35.67	98.46	71.75	0.3
mammography	0.01	2.96	0.19	0.25	10.54	0.01	0.3	9.15	0.01	0.04	N/A	0.06	0.31	1.89	24.58	22.05	75.61	156.66	82.74	356.13	36.61	101.65	42.04	0.26
mnist	0.8	2.76	1.08	0.6	45.1	0.04	1.53	5.11	N/A	N/A	26.33	0.06	0.86	1.88	18.45	23.18	49.98	119.6	86.85	42.63	41.35	70.54	42.47	0.34
musk	2	0.59	0.18	0.28	11.97	0.05	0.4	1.29	0.24	0.38	12.16	0.05	0.55	0.81	7.42	11.06	24.35	48.8	29.44	18.96	33.18	26.78	13.22	0.25
optdigits	0.56	1.12	0.48	0.34	12.85	0.04	0.76	3.09	N/A	N/A	17.09	0.05	0.41	1	13.01	20.67	45.81	82.42	49.19	56.27	63.18	48	24.04	0.25
PageBlocks	0.01	0.86	0.1	0.32	3.61	0.01	0.14	3	0.02	0.04	18.06	0.05	0.22	1.14	12.55	17.38	43.56	84.95	47.57	170.22	45.68	48.41	21.55	0.18
pendigits	0.09	1.87	0.76	0.36	6.22	0.01	0.86	4.86	0.04	0.07	22.57	0.06	0.33	1.5	11.58	23.89	55.66	107.62	57.65	180.92	65.4	65.48	28.44	0.21
Pima	0	0.04	0.01	0.17	0.18	0.01	0.02	0.2	0	0.01	2.54	0.02	0.13	0.25	1.82	2.83	8.54	15.73	10.12	35.39	48.23	7.54	4.05	0.08
satellite	0.19	1.57	0.73	0.47	14.12	0.02	0.79	4.69	0.09	0.17	14.59	0.07	0.38	1.59	14.85	19.47	41.83	101.04	60.93	101.16	73.17	55.4	26.49	0.25
satimage-2	0.22	1.37	0.53	0.38	10.45	0.01	0.63	3.76	0.09	0.15	10.13	0.05	0.35	1.56	12.73	15.79	38.08	90.22	58.17	109.39	49.93	52.77	23.59	0.23
shuttle	0.03	3.16	0.17	0.22	11.07	0.01	0.25	8.75	0.03	0.05	18	0.06	0.36	2.19	21.56	31.35	68.32	157.14	86.32	319.05	40.19	100.2	43.96	0.28
skin	0	2.52	0.05	0.2	9.74	0	0.17	5.95	0.02	0.02	N/A	0.05	0.36	2.13	22.15	30.12	77.24	157.23	97.78	447.26	54.79	97.65	48.23	0.25
smtp	0	2.75	0.09	0.22	10.09	0	0.16	6.19	0.01	0.02	17.43	0.04	0.32	1.98	23.1	26.45	72.19	157.84	94.37	452.46	53.7	100.14	47.7	0.26
SpamBase	0.34	0.76	0.31	0.31	6.33	0.02	0.46	2.36	0.06	0.09	N/A	0.05	0.29	0.98	8.2	13.41	35.16	66.97	33.38	43.95	49.75	37.78	19.38	0.21
speech	3.98	1.43	0.26	0.56	30.86	0.15	1.39	2.19	0.71	0.91	8.3	0.06	1.5	1.1	7.38	13.35	33.69	60.37	38.33	26.18	36.01	29.67	19.96	0.42
Stamps	0	0.04	0.02	0.21	0.17	0	0.03	0.22	0	0	1.32	0.02	0.13	0.35	2.16	3.22	9.14	16.08	11.81	36	31.08	6.64	4.47	0.06
thyroid	0.01	0.42	0.07	0.35	1.58	0	0.1	1.31	0.02	0.02	6.5	0.04	0.19	0.81	8.36	12.36	33.78	59.73	35.09	161.78	33.33	29.38	15.38	0.14
vertebral	0	0.04	0.01	0.21	0.15	0.01	0.02	0.21	0	0.01	1.34	0.01	0.13	0.33	2.13	2.44	8.08	15.42	8.57	44.49	48.32	6.76	4.05	0.06
vowels	0.04	0.07	0.04	0.27	0.31	0.01	0.05	0.38	0.01	0.02	2	0.02	0.15	0.47	3.57	5.04	11.3	23.11	13.17	54.84	18.23	11.6	6.42	0.08
Waveform	0.16	0.42	0.21	0.39	1.57	0	0.24	1.59	0.03	0.06	5.98	0.05	0.22	0.94	7.18	12.31	29.66	54.34	32.54	80.1	34.76	30.02	14.91	0.16
WBC	0	0.02	0.02	0.16	0.17	0.01	0.02	0.24	0.01	0	N/A	0.02	0.13	0.3	2.14	2.54	8.2	15.17	8.61	35.47	41.99	7.66	4.59	0.06
WDBC	0.04	0.04	0.03	0.17	0.19	0.01	0.04	0.26	0.02	0.02	1.33	0.03	0.14	0.44	1.65	2.58	7.85	15.62	8.27	15.86	30.51	7.65	4.5	0.07
Wilt	0	0.56	0.06	0.36	2.59	0	0.09	2.01	0.01	0.02	8.21	0.03	0.21	0.94	10.22	14.01	34.23	76.74	38.37	201.36	43.17	40.04	22.43	0.17
wine	0.01	0.04	0.03	0.15	0.18	0.01	0.03	0.25	0.01	0.01	1.32	0.01	0.13	0.36	2.37	2.1	6.66	16.78	7.85	34.21	36.71	7.95	5.87	0.06
WPBC	0.06	0.04	0.03	0.22	0.18	0	0.05	0.26	0.01	0.03	1.34	0.03	0.14	0.45	2.47	2.26	6.23	15.73	7.99	15.8	48.7	7.8	5.99	0.08
yeast	0	0.06	0.03	0.22	0.27	0	0.04	0.42	0.01	0	2.58	0.02	0.14	0.45	2.73	4.38	12.32	23.11	12.16	56.32	17.26	12.13	9.16	0.07