
PCU: Perturbation-Calibrated Uncertainty for Unsupervised Anomaly Detection

Mebarka Allaoui^{*1}

Rachid Hedjam¹

Mohand Saïd Allili²

Guoqiang Zhong^{†3}

¹Computer Science Dept., Bishop’s University, Sherbrooke, Quebec, Canada

²Computer Science and Engineering Dept., Université du Québec en Outaouais, Gatineau, Quebec, Canada

³College of Computer Science and Technology, Ocean University of China, Qingdao, Shandong, China

^{1*}Corresponding author: mallaoui@ubishops.ca. [†]Corresponding author for inquiries from China: gqzhong@ouc.edu.cn.

Abstract

Unsupervised anomaly detection requires deciding whether a test observation belongs to the nominal data distribution when only normal data are available for training. In this setting, failures arise primarily from epistemic uncertainty: the model must determine whether a sample lies within the learned support of the data or in an unseen region of the input space. Existing methods rely on density surrogates, reconstruction error, or distance-based scores, which often become unreliable near the nominal manifold and do not explicitly encode uncertainty. We propose *Perturbation-Calibrated Uncertainty (PCU)*, a representation-learning framework that estimates epistemic uncertainty through controlled input perturbations. An encoder is trained on a ladder of known corruption magnitudes so that latent displacement varies predictably with perturbation strength, enforced by a ranked-sensitivity constraint, a scale-aware prediction head, and variance-covariance regularization. An exponential moving average of clean embeddings provides a reference, yielding an anomaly score that combines global displacement, learned perturbation sensitivity, and local stability. By making perturbation response an intrinsic property of the representation rather than a post hoc test, PCU provides a compact, label-free uncertainty signal for unsupervised anomaly detection. Experiments on diverse tabular benchmarks demonstrate stable anomaly scores and competitive detection performance, particularly in near-manifold regimes, while mixed-feature perturbation design remains a limitation of the current Gaussian instantiation. PCU source code is available on GitHub.

Keywords: Uncertainty Quantification, Representation Learning, Anomaly Detection, One-Class Learning, Calibration

1 INTRODUCTION

Unsupervised anomaly detection in *one-class* settings requires reliable uncertainty estimates in the absence of labels and explicit class boundaries. This problem arises in high-stakes domains such as fraud monitoring, security screening, and operational diagnostics, where models are trained only on nominal data and must decide whether future observations deviate from normal behavior (Huang et al. (2025)). In such settings, uncertainty is predominantly *epistemic*: it reflects incomplete knowledge of the support of the normal data manifold and how far a test point departs from it, rather than probabilistic class membership (Kendall and Gal (2017); Gal and Ghahramani (2016)). Although measurement noise introduces some *aleatoric* variability, the main challenge is determining whether an observation lies within the learned normal region or in an unseen part of the input space (Ovadia et al. (2019)).

This view links anomaly detection directly to uncertainty estimation. Because anomalies are never observed during training, abnormality cannot be defined as membership in a known class but instead corresponds to insufficient evidence that a sample belongs to the learned data-generating process. Previous work shows that reliable decisions require calibrated uncertainty signals, particularly in OOD and safety-critical settings (Amodei et al. (2016); Gal and Ghahramani (2016); Lakshminarayanan et al. (2017); Ovadia et al. (2019)). From this perspective, anomaly detection amounts to identifying regions of epistemic uncertainty in representation space, making confidence estimation as important as modeling the nominal distribution.

Most unsupervised detectors rely on surrogate uncer-

tainty signals derived from density, proximity, or reconstruction. Neighborhood methods such as LOF and k NN and isolation-based approaches such as Isolation Forest remain competitive due to minimal assumptions but are sensitive to feature scaling and sampling density (Breunig et al. (2000); Liu et al. (2008)). Deep models attempt to learn task-specific representations using reconstruction, denoising, or energy-based objectives (Vincent et al. (2008); Ruff et al. (2018); Zhai et al. (2016)), yet in tabular domains these signals often become ambiguous for samples close to the nominal manifold. Large-scale studies confirm substantial performance variability and hyperparameter sensitivity, with learned models not consistently outperforming classical baselines (Han et al. (2022)). Even calibrated density-based methods such as copula- and CDF-based scoring improve robustness but remain affected by feature scaling and local density distortions (Li et al. (2020, 2022)). These observations suggest that the central difficulty is not representation capacity but the absence of a principled mechanism for encoding epistemic uncertainty within learned features.

We propose *Perturbation-Calibrated Uncertainty (PCU)*, a representation-learning framework for unsupervised anomaly detection that models uncertainty via controlled perturbations rather than reconstruction or likelihood estimation. The key idea is that a reliable representation should respond to perturbations in a calibrated and stable manner. During training, an encoder is exposed to a ladder of Gaussian corruptions and optimized so that latent displacement grows consistently with perturbation magnitude. A lightweight head predicts a proxy of corruption strength directly from the embedding, variance-covariance regularization prevents degenerate solutions, and an exponential moving average of clean embeddings provides a nominal reference. At inference, the anomaly score combines predicted perturbation sensitivity, deviation from the reference, and local stability under infinitesimal noise to produce a label-free decision rule.

Recent work has strengthened tabular anomaly detection along several directions, including rejection-aware anomaly decisions (Perini and Davis (2023)), robust autoencoder formulations for transductive and inductive outlier detection (Lindenbaum et al. (2024)), internal contrastive learning for tabular representations (Shenkar and Wolf (2022)), and deep extensions of isolation-based detection (Xu et al. (2023)). Other methods incorporate calibration or perturbations by adjusting decision boundaries with uncertainty-aware penalties (Xu et al. (2024)), probing trained detectors using input perturbations (Dai et al. (2025)), or learning auxiliary representation objectives while relying on external scoring rules (Ye et al. (2025)). PCU differs

by making the perturbation response itself the training signal: the embedding is explicitly optimized so that its variation reflects known perturbation magnitude. Consequently, perturbation sensitivity becomes an intrinsic property of the learned representation and acts as an estimator of epistemic uncertainty rather than a post hoc diagnostic.

Experiments on diverse tabular benchmarks show that perturbation-calibrated representations produce stable anomaly scores and competitive unsupervised detection performance, particularly in near-manifold regimes where reconstruction- and density-based surrogates are ambiguous. More broadly, PCU suggests that reliable unsupervised decisions benefit from representations that encode epistemic uncertainty rather than only reconstruction fidelity or likelihood.

The remainder of this paper is organized as follows. Section 2 introduces the PCU framework. Section 3 presents empirical evaluation and ablations. Section 4 concludes the paper. A broader discussion of related work is provided in App. A.

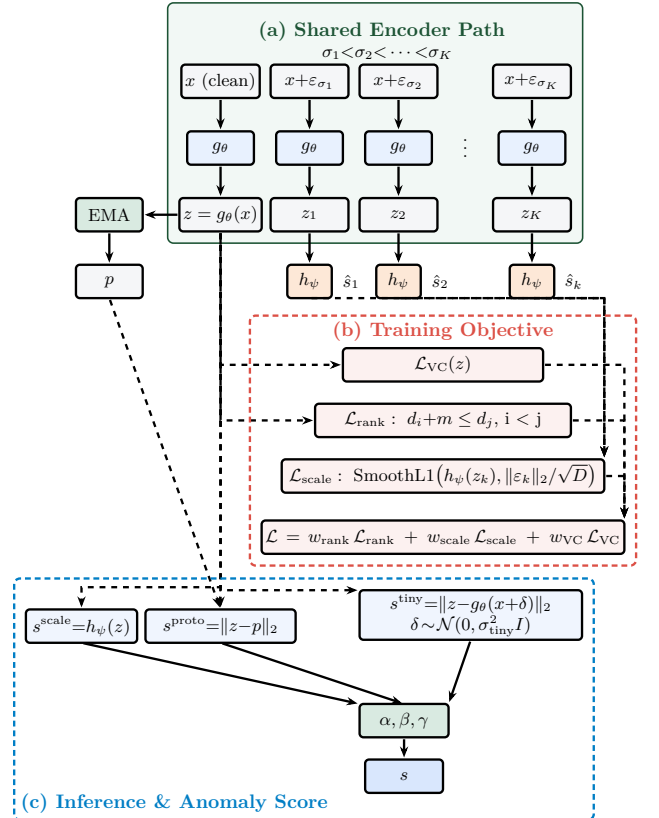


Figure 1: PCU training and inference. (a) Clean sample x and noised copies share encoder g_θ to produce z and $\{z_k\}$. (b) The training objective aggregates VC, ranking over d_k , and scale regression into the *joint loss* $\mathcal{L}$. (c) the Inference & Anomaly Score mixes s^{scale} , s^{proto} , s^{tiny} into the final anomaly score s .

2 PCU METHOD

PCU models uncertainty in unsupervised anomaly detection by learning how representations respond to controlled input perturbations. The central hypothesis is that epistemic uncertainty manifests as instability of the learned representation: samples well supported by the training distribution exhibit predictable responses to perturbations, whereas unsupported samples produce irregular responses. Accordingly, a shared MLP encoder is exposed during training to a multi-scale ladder of perturbations and is optimized to (i) produce scale-ordered latent displacements (ranked sensitivity), (ii) encode perturbation magnitude in the embedding (scale awareness), and (iii) maintain a well-conditioned latent geometry (variance-covariance regularization). A lightweight exponential-moving-average (EMA) prototype summarizes the nominal embedding structure. At test time, PCU does not reconstruct or estimate density; instead, it evaluates the learned perturbation response to obtain an uncertainty score used for anomaly detection.

2.1 MODEL

Let $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^D$ be normal training N samples of dimension D . We learn a d_z -dimensional representation $z = g_\theta(x)$ with a compact MLP g_θ (ReLU nonlinearities; final LayerNorm to stabilize latent statistics), and attach a lightweight scalar head $h_\psi : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^+$ that is used in both training and inference. Each clean input x is augmented into a strictly increasing *perturbation ladder* of K corrupted copies whose perturbation magnitude is controlled by a known scale parameter. PCU is not tied to a particular corruption distribution; its objective requires an ordered perturbation family whose strength is known or can be estimated. In this work, we instantiate the ladder with additive Gaussian perturbations because this choice is simple, analytically convenient, and appropriate for standardized continuous or approximately continuous features. Specifically, we use standard deviations $\sigma_1 < \dots < \sigma_K$ sampled linearly in $[\sigma_{\min}, \sigma_{\max}]$:

$$\begin{aligned} x_k &= x + \varepsilon_k, \quad \varepsilon_k \sim \mathcal{N}(0, \sigma_k^2 I_D), \\ z &= g_\theta(x), \quad z_k = g_\theta(x_k), \quad \forall k = 1..K. \end{aligned} \quad (1)$$

where I_D is the $D \times D$ identity matrix and ε_k denotes the injected Gaussian noise. The Gaussian ladder should be understood as the continuous-feature instantiation of PCU. For binary, one-hot, ordinal, or categorical attributes, additive Gaussian noise may generate values outside the valid feature semantics. In such settings, the same PCU objective can be used with a feature-type-aware ordered perturbation family, for

example, Gaussian noise on continuous variables and masking, empirical resampling, or category swapping on discrete or categorical variables. Thus, the ranked-sensitivity and scale-regression objectives require controllable perturbation strength, but not necessarily Gaussian perturbations.

Latent displacements $d_k = \|z - z_k\|_2$ quantify how the representation reacts to perturbations of increasing scale. Under mild local Lipschitz assumptions on the encoder, the expected displacement induced by Gaussian perturbations is controlled by the perturbation magnitude, admitting an upper bound proportional to the noise scale; a formal bound is given in Lemma B.1 (Appendix B). The ranking objective then enforces an ordered response across noise levels during training. For each noised copy x_k in the ladder, we pass it through the same encoder g_θ to obtain $z_k = g_\theta(x_k) \in \mathbb{R}^{d_z}$. We then feed z_k into a lightweight scalar prediction head h_ψ to obtain a single nonnegative value

$$\hat{s}_k = h_\psi(z_k). \quad (2)$$

Concretely, h_ψ is a small multilayer perceptron with one hidden layer and elementwise ReLU activations, followed by a final linear projection to a scalar. Its parameters ψ are learned jointly with θ during training. Thus, h_ψ implements a learned mapping $h_\psi : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^+$, which attempts to read out from the latent code z_k an estimate of the magnitude of the injected perturbation. Under the Gaussian instantiation, the training target for this head is a normalized measure of the actual perturbation injected into the input. For level k , the Gaussian corruption vector is $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2 I_D)$, and we define the target scalar

$$s_k^{\text{tgt}} = \frac{\|\varepsilon_k\|_2}{\sqrt{D}}. \quad (3)$$

Writing $\varepsilon_k = \sigma_k \xi$ with $\xi \sim \mathcal{N}(0, I_D)$ makes clear that the normalization removes the trivial growth of Gaussian norms with the ambient dimension, i.e., $\|\varepsilon_k\|_2 / \sqrt{D} = \sigma_k \|\xi\|_2 / \sqrt{D}$. Here $\|\xi\|_2^2$ follows a $\chi^2(D)$ (chi-square) distribution with D degrees of freedom. Moreover, the standard concentration for Gaussian norms implies $\|\varepsilon_k\|_2 / \sqrt{D}$ is tightly concentrated around σ_k with fluctuations of order $O(D^{-1/2})$.¹ Thus, the target in Eq. (3) is (i) dimensionally comparable across datasets with different D , (ii) centered near the true corruption scale σ_k , and (iii) low-variance for moderate-to-large D , providing a well-conditioned supervision signal for the scale head. Thus, s_k^{tgt} is a well-conditioned and dimension-robust proxy for the

¹See, e.g., concentration of Lipschitz functions of Gaussians or the χ_D tail bounds; a convenient reference is Vershynin (Vershynin, 2018), Secs. 1.1–1.2.

injected corruption scale. The scale head is then trained by regression to match $\hat{s}_k$ (Eq. 2) to s_k^{tgt} (Eq. 3). Intuitively, h_ψ learns to answer the question: “Given only the latent representation z_k of a (possibly corrupted) sample, how much corruption does this sample appear to contain?” During inference, we will apply the same head to clean test embeddings $z = g_\theta(x)$ to obtain $s^{\text{scale}} = h_\psi(z)$, which serves as one of the calibrated uncertainty cues used for one-class decision making.

In parallel, we maintain an exponential moving average (EMA) prototype $p_t \in \mathbb{R}^{d_z}$ that tracks the center of clean embeddings over training steps $t = 1, 2, \dots$:

$$p_t = \mu p_{t-1} + (1 - \mu) \bar{z}_t, \\ \bar{z}_t = \frac{1}{B_t} \sum_{b=1}^{B_t} g_\theta(x_t^{(b)}), \quad \mu \in (0, 1). \quad (4)$$

where B_t is the minibatch size at step t , and $\bar{z}_t$ is computed on *clean* inputs. Unrolling the recursion shows that p_t is a geometrically weighted average of all past batch means:

$$p_t = (1 - \mu) \sum_{i=1}^t \mu^{t-i} \bar{z}_i + \mu^t p_0,$$

so the contribution of batch i is $(1 - \mu)\mu^{t-i}$: recent batches receive larger weight while older information decays geometrically. The weights approximately span an effective window of $1/(1 - \mu)^2$ steps (e.g., $\mu=0.99$ averages ~ 100 steps), which lowers variance by aggregating many batches yet maintains short lag through recency emphasis. In practice, we update p_t once per iteration *after* the optimizer step using only clean embeddings; optionally, a bias-corrected estimate $\hat{p}_t = p_t/(1 - \mu^t)$ may be used in early iterations, though we find it unnecessary for our scoring. The EMA prototype concentrates around the nominal embedding center with bounded variance, providing a low-noise reference for decision scoring; a formal tracking bound is given in Lemma B.2. Thus, p serves as an epistemic reference point representing the typical embedding of nominal data rather than a decision boundary.

EMA-style running statistics are a standard stabilization device in deep learning, e.g., BatchNorm keeps running means/variances for test-time normalization (Ioffe and Szegedy (2015)), optimizers such as Adam maintain EMA moments of gradients (Kingma and Ba (2015)), and modern self-supervised systems use EMA “targets” to provide stable references during representation learning (Tarvainen and Valpola (2017); Grill et al. (2020)). In PCU, p serves the same purpose at the

embedding level: it provides a noise-free, low-variance epistemic reference for global position. The prototype deviation $s^{\text{proto}} = \|z - p\|_2$ thus measures displacement from the center of the nominal embedding distribution in latent space, complementing (i) the scale response learned from corrupted views and (ii) the tiny-noise sensitivity probe. We update p only from clean embeddings to ensure that the anchor reflects the unperturbed nominal data manifold.

Although training uses corrupted copies, these perturbations are never treated as negatives nor used as reference points for decision making. They only calibrate the encoder’s perturbation–response and the scale head, while inference depends solely on a sample’s predicted scale, its distance to the clean EMA prototype, and its tiny-noise stability, preventing near-manifold normal neighbors from being penalized even if they resemble synthetic corruptions of other points.

2.2 OBJECTIVE

Training learns an uncertainty-aware representation rather than a classifier: the objective calibrates how the embedding should react to perturbations, not how to separate classes, where it minimizes a single objective that (i) aligns latent sensitivity with the noise ladder, (ii) learns a calibrated proxy of corruption strength, and (iii) preserves well-conditioned batch statistics in the latent space.

Ranking loss. To couple latent sensitivity to perturbation scale, we encourage an ordered response of the latent displacements $d_k = |z - z_k|_2$ along a strictly increasing noise ladder $\sigma_1 < \dots < \sigma_K$. For any two levels with $\sigma_i < \sigma_j$, the response at the higher noise level is encouraged to exceed that at the lower level by a margin $m > 0$, i.e., $d_i + m \leq d_j$. This is implemented via a hinge-based pairwise ranking objective,

$$\mathcal{L}_{\text{rank}} = \frac{1}{K(K-1)} \sum_{1 \leq i < j \leq K} \max(0, d_i + m - d_j). \quad (5)$$

Here $d_k = |z - z_k|_2$ measures the encoder’s response to injecting Gaussian noise $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2 I_D)$ at level k , with the ℓ_2 norm matched to the isotropic corruption model. The margin $m > 0$ controls the desired separation between responses at different scales. When minimized, the objective enforces *scale-ordered responses*, suppressing inversions in which smaller perturbations induce larger latent changes and aligning the latent geometry with predictable sensitivity to perturbation magnitude. Importantly, this constraint regularizes the encoder g_θ rather than imposing smoothness on the data manifold: neural networks are piecewise Lipschitz mappings, and the loss encourages locally consistent behavior of the learned representation under controlled

²EMA inherits classical variance-reduction properties of exponentially weighted averages and iterate averaging Polyak and Juditsky (1992).

perturbations. Because deviations from the nominal manifold produce disproportionately large latent responses, the ordered sensitivity learned here provides a foundation for the uncertainty cues used at inference.

Proposition 2.1 (Scale-ordered responses). *Assume that for a given training sample, the ranking constraints with margin $m > 0$ satisfy $\max(0, d_i + m - d_j) = 0$ for all $1 \leq i < j \leq K$, where $d_k = \|z - z_k\|_2$ and $\sigma_i < \sigma_j$. Then the latent responses obey*

$$d_1 \leq d_k - (k-1)m, \quad k = 1, \dots, K.$$

Proof. From $\max(0, d_i + m - d_j) = 0$ we obtain $d_j \geq d_i + m$ for all $i < j$. In particular, $d_2 \geq d_1 + m$, $d_3 \geq d_2 + m \geq d_1 + 2m$, and so on. By induction, $d_k \geq d_1 + (k-1)m$, which is equivalent to $d_1 \leq d_k - (k-1)m$. $\square$

Margin-based pairwise objectives are a standard and effective mechanism for imposing relative-order constraints in representation and metric learning, offering stable gradients when constraints are violated and becoming inactive once the margin is satisfied (Burges et al. (2005); Weinberger and Saul (2009); Hadsell et al. (2006)). Compared to absolute-regression surrogates for scale, the hinge formulation directly encodes the ordinal structure ($\sigma_i < \sigma_j \Rightarrow d_i < d_j$), is temperature-free, and naturally aggregates information across all ordered pairs in the noise ladder. In PCU, this term complements the scale-regression head: the ranking loss calibrates *relative* sensitivity across perturbation levels, while the regression objective learns an *absolute* scalar proxy of corruption strength.

Scale-regression loss. To make the representation explicitly *scale-aware*, we regress from the noised embedding $z_k = g_\theta(x_k)$ to a dimensionless proxy of the injected perturbation magnitude $s_k^{\text{tgt}} = \|\varepsilon_k\|_2 / \sqrt{D}$ (Eq. (3)) using a robust Smooth- ℓ_1 penalty. The scalar head $h_\psi : \mathbb{R}^{d_z} \rightarrow \mathbb{R}_{\geq 0}$ outputs $s_k^{\text{pred}} = h_\psi(z_k)$ (Eq. (2)); the regression error is $e_k = s_k^{\text{pred}} - s_k^{\text{tgt}}$. With transition parameter $\beta > 0$ (default $\beta=1$), the per-sample loss is

$$\text{Smooth-}\ell_1(e_k; \beta) = \begin{cases} \frac{1}{2} \frac{e_k^2}{\beta}, & |e_k| < \beta, \\ |e_k| - \frac{1}{2}\beta, & \text{otherwise,} \end{cases} \quad (6)$$

and averaging across the K ladder levels yields

$$\mathcal{L}_{\text{scale}} = \frac{1}{K} \sum_{k=1}^K \text{Smooth-}\ell_1(e_k; \beta). \quad (7)$$

Notation and rationale. z_k is the latent of $x_k = x + \varepsilon_k$ with $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2 I_D)$, D is the input dimension, and $\sqrt{D}$ normalizes $\|\varepsilon_k\|_2$ to remove trivial growth with

dimensionality, producing targets that are comparable across datasets. The Smooth- ℓ_1 penalty behaves quadratically for small errors (stable and precise gradients near correct calibration) and linearly for large errors (bounded influence of outliers or occasional large $\|\varepsilon_k\|_2$), combining the bias-variance benefits of ℓ_2 near zero with the robustness of ℓ_1 away from zero. This choice is a standard robust alternative to pure ℓ_2 in regression, and it has been widely adopted in deep objectives for stable training under heavy-tailed noise and label noise (Girshick (2015); Barron (2019)). By directly regressing s_k^{tgt} under a robust loss, PCU couples latent features to perturbation strength, providing a calibrated signal that later contributes to the final uncertainty-driven decision score.

VC regularizer. To stabilize the latent geometry without relying on negative pairs or contrastive sampling, we adopt the variance-covariance (VC) components of *Variance-Invariance-Covariance Regularization* (VICReg) on *clean* embeddings $z = g_\theta(x)$ (Bardes et al. (2022)). VICReg was introduced as a simple and scalable self-supervised objective that prevents representational collapse by (i) enforcing a per-dimension variance floor and (ii) decorrelating latent coordinates. These two components alone have been shown to yield strong, well-conditioned representations in practice.

In PCU, this choice is particularly appropriate. The variance term blocks degenerate solutions in which the encoder shrinks latent directions to trivially satisfy the ranking or scale-regression objectives, thereby preserving identifiability of the uncertainty calibration signal. The covariance term suppresses cross-dimensional redundancy, reducing interference between latent coordinates and aligning the ℓ_2 geometry used by the ranking loss with an approximately isotropic latent space. We intentionally omit VICReg’s invariance term because PCU already enforces structured consistency across perturbation levels through the ranked-sensitivity and scale-prediction objectives; an additional invariance penalty would be redundant and could counteract the desired monotone response to increasing noise. Importantly, the VC regularizer is applied only to clean embeddings, not to corrupted copies: it stabilizes the baseline latent geometry while the ranking and scale objectives govern how perturbed embeddings move relative to that baseline. Consequently, it does not suppress noise-induced deviations but prevents trivial sensitivity arising from collapsed or ill-conditioned representations.

Proposition 2.2 (Variance-covariance regularization prevents collapse). *Let $\hat{\Sigma}_z \in \mathbb{R}^{d_z \times d_z}$ denote the empirical covariance matrix of clean latent embeddings within*

a minibatch, and define the VC penalty as

$$\mathcal{L}_{\text{VC}} = \lambda_{\text{var}} \frac{1}{d_z} \sum_{u=1}^{d_z} \max(0, 1 - \hat{\sigma}_u) + \lambda_{\text{cov}} \frac{1}{d_z(d_z - 1)} \sum_{u \neq v} \hat{\Sigma}_{z,uv}^2, \quad \hat{\sigma}_u^2 = \hat{\Sigma}_{z,uu}.$$

If $\mathcal{L}_{\text{VC}} = 0$, then $\hat{\sigma}_u \geq 1$ for all u and $\hat{\Sigma}_{z,uv} = 0$ for all $u \neq v$. In particular, no latent dimension collapses to a constant, and all cross-dimensional correlations vanish.

Proof. Given that $\sigma_u(z) = \sqrt{\text{Cov}(z)_{uu}}$ is the empirical standard deviation of the u -th latent coordinate computed over the current minibatch and $\text{Cov}(z) \in \mathbb{R}^{d_z \times d_z}$ is the empirical covariance matrix of batch embeddings, and both components of $\mathcal{L}_{\text{VC}}$ are averages of nonnegative terms. Since $\lambda_{\text{var}}, \lambda_{\text{cov}} > 0$, the condition $\mathcal{L}_{\text{VC}} = 0$ implies that each summand must vanish. For the variance term, $\max(0, 1 - \hat{\sigma}_u) = 0$ implies $1 - \hat{\sigma}_u \leq 0$, hence $\hat{\sigma}_u \geq 1$ for all u . For the covariance term, $\hat{\Sigma}_{z,uv}^2 = 0$ implies $\hat{\Sigma}_{z,uv} = 0$ for all $u \neq v$. $\square$

Joint objective. The overall loss is a nonnegative weighted sum,

$$\mathcal{L} = w_{\text{rank}} \mathcal{L}_{\text{rank}} + w_{\text{scale}} \mathcal{L}_{\text{scale}} + w_{\text{VC}} \mathcal{L}_{\text{VC}}, \quad (8)$$

with $w_{\text{rank}}, w_{\text{scale}}, w_{\text{VC}} \geq 0$ balancing monotone sensitivity, scale calibration, and statistical regularity. In practice, we optimize Eq. (8) end-to-end with AdamW, gradient-norm clipping, and cosine-annealed learning rate; the optimizer is stepped before advancing the scheduler each epoch, and the EMA prototype p (Eq. (4)) is updated once per iteration from clean embeddings after the weight update to provide a stable epistemic anchor for inference.

2.3 INFERENCE

At test time, a sample $x \in \mathbb{R}^D$ is mapped to $z = g_\theta(x) \in \mathbb{R}^{d_z}$ and scored by a convex combination of three complementary signals that probe scale, position, and local robustness. (i) *Scale response* $s^{\text{scale}} = h_\psi(z)$ asks how strongly the point responds to perturbations under the scale-aware head trained to estimate perturbation magnitude. (ii) *Prototype deviation* $s^{\text{proto}} = \|z - p\|_2$ measures displacement from the EMA prototype p , a low-variance estimate of the nominal center. (iii) *Tiny-noise sensitivity* perturbs the input by a very small isotropic noise $\delta \sim \mathcal{N}(0, \sigma_{\text{tiny}}^2 I_D)$ and re-encodes to quantify local fragility:

$$s^{\text{tiny}} = \|z - g_\theta(x + \delta)\|_2, \quad \sigma_{\text{tiny}} \ll \sigma_{\text{min}}. \quad (9)$$

The final uncertainty-driven decision score aggregates three complementary cues after standardization to ensure commensurate scales. For each component $s^c \in \{s^{\text{scale}}, s^{\text{proto}}, s^{\text{tiny}}\}$, we compute its mean μ_k and standard deviation τ_c on the normal training data and form a normalized score $\tilde{s}^c = (s^c - \mu_c)/(\tau_c + \varepsilon)$, with a small $\varepsilon > 0$ for numerical stability. The overall decision score is then defined as the convex combination

$$s = \alpha \tilde{s}^{\text{scale}} + \beta \tilde{s}^{\text{proto}} + \gamma \tilde{s}^{\text{tiny}}, \quad \alpha, \beta, \gamma \geq 0, \quad \alpha + \beta + \gamma = 1, \quad (10)$$

with $(\alpha, \beta, \gamma) = (0.6, 0.3, 0.1)$ in all experiments (see App. C.1 and C.2)). Although the score aggregates three quantities, they are not independent heuristics. The training objective learns a representation with a calibrated perturbation response: the ranking loss encourages ordered sensitivity to perturbation magnitude, the scale head reads out this sensitivity from the embedding, and the VC regularizer stabilizes the reference geometry. The prototype deviation and tiny-noise sensitivity, therefore, act as diagnostic probes of the same learned structure, measuring the representation’s global position and local stability. Inference thus evaluates complementary aspects of a single uncertainty field learned during training rather than optimizing a separate criterion.

This normalization renders the combination scale-invariant across datasets and stabilizes the relative contribution of each uncertainty cue. The three components capture complementary failure modes: the scale response $\tilde{s}^{\text{scale}}$ identifies samples that appear intrinsically over-perturbed, the prototype deviation $\tilde{s}^{\text{proto}}$ highlights points displaced from the nominal embedding center, and the tiny-noise sensitivity $\tilde{s}^{\text{tiny}}$ elevates cases whose local neighborhoods are unstable or weakly supported. By jointly aggregating perceived corruption level, global displacement, and local robustness, PCU provides a label-free uncertainty score that is effective across many tabular datasets, particularly when the perturbation family is aligned with the feature semantics, without relying on reconstruction or density modeling. Appendix B includes a short remark showing that the convex aggregation preserves a positive mean gap when such a gap is already present in the standardized components.

3 EXPERIMENTS: EVALUATION AND VALIDATION

We evaluate PCU on 10 standard tabular *unsupervised anomaly detection* benchmarks drawn from *ADBench* (Han et al. (2022)) following the standard *one-class* protocol (train on nominal data only; deviations appear at test time). Unless noted, we reuse the official

preprocessing and splits from ADBench to ensure comparability, and report mean $\pm$ std over multiple random seeds. Throughout this section, labels are used only for post hoc evaluation and never during training, calibration, or model selection. Implementation details (optimizer, schedules, and hyperparameters) are provided in App. C.1. Additional experiments are reported in the appendix: sensitivity to the noise-ladder length and final decision-score weights, together with component-level ablations, are provided in App. C.2; the feature-type-aware perturbation ablation is discussed in App. C.3; and robustness to contaminated training data is evaluated in App. C.4. The experimental section is organized as follows. We first validate that PCU learns a calibrated perturbation–response in representation space (Sec. 3.1). Then, we analyze the stability of the resulting uncertainty scores under infinitesimal input perturbations (Sec. 3.2). Finally, we evaluate anomaly detection performance across datasets spanning diverse dimensionalities and anomaly characteristics (Sec. 3.3). Complete results on all 47 ADBench datasets are reported in App. C.5.

3.1 PERTURBATION–RESPONSE CALIBRATION

To examine whether PCU learns a representation calibrated to perturbation magnitude, we evaluate how embeddings react to controlled Gaussian corruptions of nominal samples. For a clean input x from the training distribution, we generate perturbed copies $x + \varepsilon_\sigma$ with $\varepsilon_\sigma \sim \mathcal{N}(0, \sigma^2 I_D)$ over a range of noise levels σ . Each corrupted input is encoded as $z_\sigma = g_\theta(x + \varepsilon_\sigma)$ and compared with the clean embedding $z = g_\theta(x)$. We measure (i) the latent displacement $\|z - z_\sigma\|_2$, (ii) the training corruption proxy, and (iii) the scale-head prediction $h_\psi(z_\sigma)$. If the learned geometry is calibrated, increasing noise should induce larger latent responses on average, and the scale head should consistently recover perturbation magnitude from the embedding, indicating that uncertainty-relevant information is encoded directly in the representation and can be read out at test time.

The results in Fig. 2 show consistent behavior across all datasets: as σ increases, latent displacement increases on average, and the scale head tracks the corruption proxy using only the latent representation. The latent displacement increases monotonically with the perturbation magnitude, indicating that the encoder organizes the representation according to corruption strength rather than learning strict noise invariance. Thus, progressively perturbed inputs are mapped farther from the clean embedding, which aligns with the interpretation of anomalies as unknown deviations from nominal data. The rate of increase varies across datasets:

higher-complexity datasets (e.g., Cover and Optdigits) produce steeper responses, while lower-dimensional datasets (e.g., Ionosphere and Thyroid) exhibit slower growth, suggesting that the encoder adapts its sensitivity to the intrinsic data geometry. Meanwhile, the scale head remains stable while closely tracking the corruption proxy using only the latent representation. Its consistency across noise levels shows that perturbation magnitude is recoverable directly from the embedding. Together, these observations confirm the PCU hypothesis: the learned representation forms a calibrated geometric space where distance from the nominal embedding correlates with deviation from the data manifold, providing the signal later used by the anomaly score.

3.2 DECISION STABILITY UNDER SMALL PERTURBATIONS

Calibration of the representation does not by itself guarantee reliable decisions. A detector may still be unstable if small input variations change the anomaly prediction. We therefore evaluate the stability of the PCU decision function under infinitesimal perturbations. For each test sample x , we generate a perturbed copy $x' = x + \delta$, $\delta \sim \mathcal{N}(0, \sigma_{\text{tiny}}^2 I_D)$, with $\sigma_{\text{tiny}} \ll \sigma_{\text{min}}$, and recompute the anomaly score $s(x')$. We then compare $s(x')$ with the original score $s(x)$ to quantify how sensitive the final decision is to negligible input changes.

We report three complementary stability metrics. (i) **Relative Score Variation (RSV)** (Hendrycks and Dietterich (2019)) measures score sensitivity to infinitesimal perturbations, $\text{RSV} = \mathbb{E}_x \left[\frac{|s(x') - s(x)|}{|s(x)| + \varepsilon} \right]$, where ε is a small constant; lower values indicate stable scores. (ii) **Flip Rate** (Zhang et al. (2019)) measures the fraction of samples whose predicted label changes when thresholded with the same decision boundary. (iii) **Rank Correlation** computes the Spearman correlation between $\{s(x_i)\}$ and $\{s(x'_i)\}$, reflecting preservation of anomaly ranking. Table 1 summarizes the results across 10 tabular datasets.

Table 1: Decision stability under small perturbations. Lower RSV and Flip Rate are better, while higher Rank Correlation is better. PCU produces stable decisions across datasets.

Dataset	RSV ↓	Flip Rate ↓	Rank Corr ↑
WDBC	0.17	0.24	0.97
Pendigits	0.37	0.01	0.91
HTTP	0.04	0.20	0.96
Letter	0.10	0.00	0.98
Ionosphere	0.35	0.13	0.98
Wilt	0.01	0.00	1.00
Cover	0.01	0.00	0.96
Optdigits	0.36	0.26	0.86
Thyroid	0.35	0.28	0.93
Anthyroid	0.39	0.29	0.89

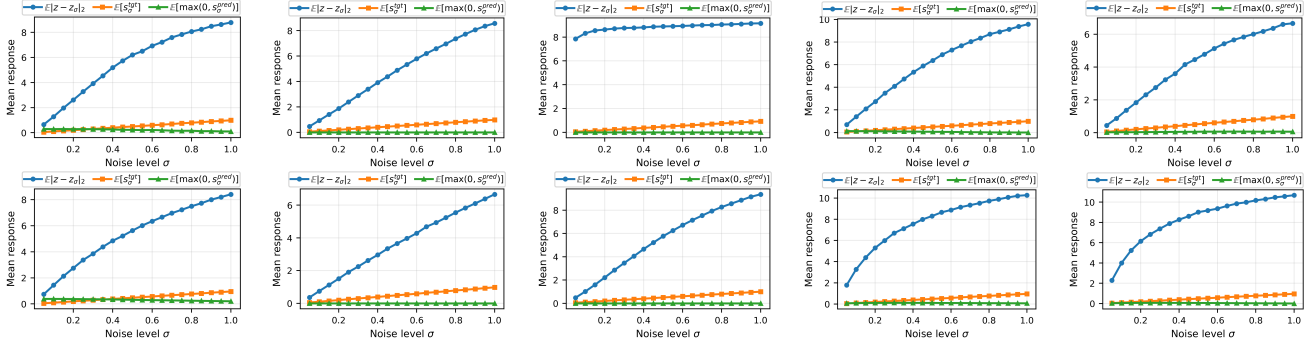


Figure 2: Perturbation-response calibration across 10 representative datasets. Increasing σ produces progressively larger latent displacements across datasets from left to right and up to down: WDBC, Pendigits, Htt, Letter, Ionosphere, Wilt, Cover, OptDigits, Thyroid, and Annthyroid.

Across datasets, the three stability metrics reveal a coherent and interpretable behavior of PCU. First, the rank correlation is consistently high on all datasets (0.86–1.00), indicating that small perturbations rarely alter the global ordering of anomaly scores. This confirms that PCU learns a stable ranking of anomaly severity rather than relying on fragile local fluctuations. Datasets such as Wilt and Cover exhibit near-perfect rank preservation together with extremely small RSV and zero flip rate, showing that both the score magnitude and the final decision remain unchanged under infinitesimal perturbations. On moderately complex datasets such as Letter and HTTP, the score varies slightly, but the flip rate remains low, and the ordering is preserved, meaning that perturbations affect calibration but not the relative anomaly structure.

A different but informative regime appears on datasets such as Ionosphere, Optdigits, Thyroid, and Annthyroid, where higher RSV and noticeable flip rates are observed while the rank correlation remains high (≥ 0.86). This indicates boundary sensitivity rather than instability: only samples located near the decision threshold change labels, whereas the global ranking of anomaly scores remains essentially unchanged. In other words, PCU maintains a consistent ordering of anomaly severity while expressing uncertainty for ambiguous near-manifold samples. Such behavior is desirable in one-class decision making, since clear normal and clear anomalous samples remain stable while only borderline cases are sensitive to small perturbations. Overall, the results demonstrate that PCU achieves both local score stability and meaningful uncertainty awareness: the scoring function is stable within well-supported nominal regions and responsive primarily near ambiguous near-manifold cases, which is precisely where uncertainty should occur.

3.3 BENCHMARK EVALUATION ON ANOMALY DETECTION TASK

We evaluate PCU under the standard unsupervised anomaly detection protocol, where models are trained using only nominal data and must identify unseen abnormal samples at test time. The goal is to assess whether perturbation-calibrated uncertainty in representation space improves practical detection quality on real datasets. Tab. 2 reports AUROC on 10 representative benchmarks spanning low- and high-dimensional settings and diverse anomaly rates. The comparison covers representative families of unsupervised tabular anomaly detectors: Local Outlier Factor (LOF) (Breunig et al. (2000)) and k NN distance (Angiulli and Pizzuti (2002)) represent proximity-based detection; Isolation Forest (IForest) (Liu et al. (2008)) and Deep Isolation Forest (DIF) (Xu et al. (2023)) represent isolation-based detection; COPOD (Li et al. (2020)) and ECOD (Li et al. (2022)) represent distributional and empirical-CDF scoring; Probabilistic Robust AutoEncoder (PRAE) (Lindenbaum et al. (2024)) represents robust reconstruction-based deep outlier detection; and MLP-NE (Dai et al. (2025)) represents recent perturbation-based neural scoring. Across these datasets, PCU consistently achieves the best or tied-best performance, outperforming distance-, density-, reconstruction-, and isolation-based detectors while matching or exceeding recent learned baselines.

PCU performs strongly on large or heterogeneous datasets such as COVER, PENDIGITS, and LETTER, where neighborhood-based methods are often brittle. In these settings, PCU remains competitive with MLP-NE while exhibiting substantially lower variance across random seeds, reflecting stable training and uncertainty scores. This stability is consistent with the use of variance-covariance regularization and the EMA prototype, which together prevent representational collapse and provide a low-noise reference for scoring. Additional

Table 2: AUROC score of the proposed method compared with baselines on 10 benchmark datasets. Each experiment is repeated 10 times with a random seed from 0 to 9, and the mean value and standard deviation are reported.

Dataset	IForest	DIF	COPOD	KNN	LOF	ECOD	PRAE	MLP-NE	PCU
WDBC	0.987±0.00	1.000±0.00	0.994±0.00	0.975±0.01	0.980±0.00	0.971±0.01	0.967±0.01	1.000±0.00	1.000±0.00
Pendigits	0.981±0.02	0.968±0.00	0.499±0.25	0.996±0.01	0.991±0.01	0.499±0.23	0.864±0.02	0.998±0.00	0.999±0.00
Httt	0.993±0.00	0.119±0.00	0.991±0.00	0.171±0.12	0.358±0.24	0.979±0.00	0.996±0.01	0.996±0.00	0.998±0.00
Letter	0.912±0.03	0.374±0.06	0.500±0.16	0.979±0.02	0.956±0.03	0.501±0.15	0.732±0.03	0.994±0.00	0.997±0.00
Ionosphere	0.895±0.03	0.206±0.12	0.801±0.02	0.975±0.01	0.954±0.01	0.743±0.02	0.859±0.02	0.975±0.00	0.994±0.01
Wilt	0.432±0.05	0.770±0.05	0.345±0.13	0.605±0.07	0.698±0.04	0.394±0.15	0.343±0.01	0.987±0.00	0.993±0.00
Cover	0.893±0.03	0.906±0.01	0.884±0.01	0.789±0.01	0.521±0.01	0.920±0.01	0.953±0.01	0.979±0.01	0.987±0.00
OptDigits	0.976±0.02	0.968±0.00	0.717±0.09	0.973±0.02	0.972±0.02	0.687±0.10	0.606±0.03	0.980±0.02	0.984±0.01
Thyroid	0.901±0.01	0.952±0.03	0.785±0.00	0.917±0.00	0.880±0.00	0.792±0.00	0.923±0.02	0.969±0.00	0.981±0.00
Annnthyroid	0.806±0.04	0.922±0.02	0.776±0.04	0.808±0.03	0.726±0.05	0.789±0.06	0.667±0.03	0.973±0.00	0.980±0.00

results on all 47 benchmarks, along with extended per-dataset analysis, are provided in the Appendix C.5.

4 CONCLUSION AND FUTURE WORK DIRECTIONS

We propose *PCU*, a perturbation-calibrated uncertainty framework for *unsupervised anomaly detection* that learns uncertainty from controlled perturbations rather than reconstruction or explicit density estimation. By exposing the encoder to a ladder of controlled perturbations, instantiated here with Gaussian corruptions for standardized continuous features, PCU aligns latent geometry with perturbation magnitude through ranked sensitivity, scale-aware prediction, and variance-covariance regularization. The resulting decision score combines perceived corruption level, displacement from the nominal embedding, and local stability under infinitesimal noise. Experiments on diverse tabular benchmarks demonstrate three consistent properties. First, the learned representation is *calibrated*: latent displacement and predicted scale grow coherently with perturbation strength. Second, PCU yields *stable decisions*: anomaly scores, rankings, and predicted labels remain robust under small input perturbations. Third, this calibrated stability translates into competitive unsupervised anomaly detection performance, particularly in near-manifold regimes where reconstruction and density surrogates become ambiguous.

The accompanying analysis supports the main design choices rather than proving universal anomaly separability: encoder smoothness motivates scale-ordered latent responses, and the EMA update provides a variance-reduced nominal reference under the one-class protocol. The results should also be interpreted within the scope of the current Gaussian instantiation: Gaussian perturbations are most appropriate for standardized continuous or approximately continuous features, while mixed, binary, ordinal, and categorical data require more carefully designed feature-type-aware per-

turbation ladders. Our mask-aware ablation indicates that simple masking is not sufficient to resolve this issue, confirming that mixed-feature perturbation design is a substantive modeling challenge rather than a minor implementation detail.

These observations open several directions for future work. First, more expressive non-Gaussian perturbation families could be developed for heterogeneous tabular data, including empirical resampling, category swapping, ordinal-preserving perturbations, learned masking policies, and generative corruptions that better respect feature semantics. Second, PCU could be extended beyond purely tabular settings to multimodal anomaly detection, where perturbation ladders may need to combine modality-specific corruptions for numerical, categorical, textual, visual, or temporal inputs. Third, although the present work follows the standard one-class protocol, contaminated-training regimes motivate robust prototype estimation, such as score-filtered, trimmed, or median-based reference updates. Finally, a stronger theoretical analysis is needed to characterize when perturbation-calibrated representations yield anomaly separation under distribution shift and how the choice of perturbation family affects this separation. Overall, PCU provides a simple and practical step toward learning representations that encode epistemic uncertainty directly from nominal data, while leaving rich opportunities for extending perturbation calibration to broader data types and deployment conditions.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant RGPIN-2020-05653. G. Zhong acknowledges support from the National Natural Science Foundation of China (NSFC) under Grant Nos. 42476194 and U24A20242, the Natural Science Foundation of Shandong Province under Grant No. ZR2024MF097, and the Natural Science Foundation of Hainan Province under Grant No. 425MS106.

References

- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.
- Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002.
- Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=xm6YD62D1Ub>.
- Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4331–4339, 2019.
- Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.
- Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.
- Wei Dai, Kai Hwang, and Jicong Fan. Unsupervised anomaly detection for tabular data using deep noise evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11553–11562, 2025.
- Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
- Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR’06)*, volume 2, pages 1735–1742. IEEE, 2006.
- 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.
- Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=HJz6tiCqYm>.
- Haoqi Huang, Ping Wang, Jianhua Pei, Jiacheng Wang, Shahan Alexanian, and Dusit Niyato. Deep learning advancements in anomaly detection: A comprehensive survey. *IEEE Internet of Things Journal*, 12(21): 44318–44342, 2025. doi: 10.1109/JIOT.2025.3585884.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30:5574–5584, 2017.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. URL <https://arxiv.org/abs/1412.6980>.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.
- 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)*, pages 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.
- Ofir Lindenbaum, Yariv Aizenbud, and Yuval Kluger. Transductive and inductive outlier detection with robust autoencoders. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*,

- volume 244 of *Proceedings of Machine Learning Research*, pages 2271–2293, 2024.
- Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.
- Andrei Margeloiu, Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. Tabebm: A tabular data augmentation method with distinct class-specific energy-based models. *Advances in Neural Information Processing Systems*, 37:72094–72144, 2024.
- Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.
- Lorenzo Perini and Jesse Davis. Unsupervised anomaly detection with rejection. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dc48c738d3ef8c81b6e968453a84a819-Abstract-Conference.html.
- Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.
- 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*, pages 4393–4402. PMLR, 2018.
- Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2022.
- Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30:1195–1204, 2017.
- Hugo Thimonier, Fabrice Popineau, Arpad Rimmel, and Bich-Liên Doan. Retrieval augmented deep anomaly detection for tabular data. In *Proceedings of the 33rd ACM international conference on information and knowledge management*, pages 2250–2259, 2024.
- Che-Ping Tsai, Ganyu Teng, Phillip Wallis, and Wei Ding. Anollm: Large language models for tabular anomaly detection. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.
- Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.
- 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, 2023. doi: 10.1109/TKDE.2023.3270293.
- 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*, 36(11):5723–5736, 2024.
- Hangting Ye, He Zhao, Wei Fan, Mingyuan Zhou, Dan Guo, and Yi Chang. Drl: Decomposed representation learning for tabular anomaly detection. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International conference on machine learning*, pages 1100–1109. PMLR, 2016.
- Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

Supplementary Material

Mebarka Allaoui^{*1}

Rachid Hedjam¹

Mohand Saïd Allili²

Guoqiang Zhong^{†3}

¹Computer Science Dept., Bishop’s University, Sherbrooke, Quebec, Canada

²Computer Science and Engineering Dept., Université du Québec en Outaouais, Gatineau, Quebec, Canada

³College of Computer Science and Technology, Ocean University of China, Qingdao, Shandong, China

^{1*}Corresponding author: mallaoui@ubishops.ca. [†]Corresponding author for inquiries from China: gqzhong@ouc.edu.cn.

A RELATED WORK

PCU targets *unsupervised anomaly detection* on tabular data under the standard *one-class* protocol, where only nominal samples are observed during training and anomalies are treated as OOD deviations from the nominal support. In this setting, uncertainty is predominantly epistemic. Our contributions intersect four lines of work: (i) tabular anomaly detection and one-class learning, (ii) uncertainty and OOD estimation and calibration without labels, (iii) perturbation-based self-supervision and sensitivity-driven scoring, and (iv) non-contrastive representation regularization and collapse prevention. We summarize closely related studies and clarify how PCU differs.

Tabular anomaly detection and one-class learning. Classical unsupervised detectors remain strong on tabular benchmarks due to their minimal inductive bias, including neighborhood and density surrogates (e.g., LOF (Breunig et al. (2000)), k NN distances (Angiulli and Pizzuti (2002))) and isolation-based methods such as IForest (Liu et al. (2008)). However, their reliability can degrade under heterogeneous feature scales, non-uniform sampling densities, and near-manifold anomalies, motivating the use of learned representations and more explicit calibration mechanisms. Deep tabular anomaly detection has historically relied on reconstruction- or energy-based criteria (e.g., denoising/reconstruction objectives and one-class embedding constraints), but large benchmark studies highlight substantial variance across datasets and sensitivity to hyperparameters, especially in fully unsupervised protocols. ADBench systematizes this evaluation landscape and has become a common reference point for tabular anomaly detection comparisons (Han et al. (2022)).

Recent deep tabular anomaly detectors address these limitations through more structured representation-learning, isolation, and reconstruction objectives. Internal Contrastive Learning (ICL) learns tabular representations through contrastive objectives constructed within the nominal class, providing an alternative route to one-class representation learning (Shenkar and Wolf (2022)). Deep Isolation Forest (DIF) extends the isolation principle by using learned representations to improve isolation-based detection in complex feature spaces (Xu et al. (2023)). Probabilistic Robust AutoEncoder (PRAE), proposed for transductive and inductive outlier detection, revisits reconstruction-based detection by using robust probabilistic modeling to attenuate the influence of outliers during autoencoder training (Lindenbaum et al. (2024)). Rejection-based anomaly detection further augments unsupervised decisions with a reject option when the detector is uncertain (Perini and Davis (2023)). PCU is complementary to these advances: it does not rely on contrastive pair construction, reconstruction fidelity, isolation scores, or rejection rules, but instead learns a perturbation-calibrated representation whose sensitivity to controlled corruption is used directly as an uncertainty signal.

More recently, tabular anomaly detection has begun to incorporate *retrieval* and *foundation-model* ingredients. For example, retrieval-augmented deep anomaly detection for tabular data investigates how external memory banks and nearest-neighbor evidence can improve detection and robustness beyond purely parametric scoring (Thimonier et al. (2024)). In parallel, recent LLM-based approaches for tabular anomaly detection have emerged, for example, by formulating anomalies as reasoning problems handled through structured prompting or language-model

inference. These works reflect a broader shift toward foundation-model methodologies for anomaly detection and OOD identification (Tsai et al. (2025)). PCU is complementary to these directions: rather than adding external memory or language priors, it focuses on making the *representation itself* encode a calibrated notion of deviation through controlled perturbations.

Uncertainty, OOD detection, and calibration without labels. A substantial literature connects anomaly detection to OOD detection by interpreting anomalies as samples drawn from distributions not seen during training. In this view, reliable detection requires uncertainty signals that reflect *epistemic* mismatch with the nominal support rather than confidence in class membership. Practical OOD scoring for deep models often uses proxies such as distances in embedding space, density surrogates, or energy scores, but these can be poorly calibrated when anomalous points lie near the nominal support. Recent work in tabular settings reinforces that *calibration*, not merely model capacity, is a central bottleneck. In particular, a recent study on class-conditional tabular modeling and augmentation highlights that one-class modeling can be sensitive to how uncertainty and class-conditional structure are captured, and that robustness can depend on avoiding noise from other classes (Margeloiu et al. (2024)). PCU aligns with this perspective by explicitly supervising how representations should respond as a function of *known* perturbation magnitude, thereby turning calibration into a training signal that does not require anomaly labels.

Perturbation-based learning and perturbation-based scoring. PCU is motivated by the idea that controlled perturbations can act as self-supervision for representation learning. Corruption-based learning has long been used to obtain robust features, most commonly through denoising or reconstruction objectives in which the model learns to remove noise or recover the clean input. Such approaches encourage invariance: the representation is trained to ignore perturbations. PCU adopts a different perspective. Instead of suppressing perturbations, it uses them to structure the latent geometry. The encoder is trained to exhibit a scale-ordered response to increasing corruption, and a dedicated head predicts a proxy of perturbation magnitude directly from the embedding. The objective is therefore not robustness through invariance but *calibrated sensitivity*: perturbation magnitude becomes observable in the representation and can be used as an uncertainty cue for anomaly detection.

This perspective relates to recent work in tabular anomaly detection that moves beyond reconstruction-based objectives. Decomposed Representation Learning (DRL) (Ye et al. (2025)) shows that anomaly detection improves when representations are trained using auxiliary structural objectives rather than reconstruction error alone, suggesting that reconstruction fidelity is a weak surrogate for abnormality in tabular data. PCU shares this motivation but differs in mechanism: DRL modifies the representation before applying a separate scoring rule, whereas PCU trains the representation itself to encode perturbation strength and directly uses this response as an uncertainty signal for unsupervised anomaly detection.

A closely related line of work estimates anomalies by measuring prediction sensitivity to input perturbations. The Noise Evaluation method (Dai et al. (2025)) perturbs each test sample with small random noise and evaluates how the detector’s prediction changes. The underlying hypothesis is that nominal samples lie in stable regions of the data manifold, while anomalous samples occupy unstable regions where small perturbations significantly alter the prediction. The magnitude of this prediction change is then used as an anomaly indicator.

PCU differs fundamentally from such approaches. Noise Evaluation measures the instability of an already trained detector at inference time and treats perturbation sensitivity as a diagnostic signal. In contrast, PCU incorporates perturbations directly into the training objective: the representation is explicitly optimized to produce an ordered and calibrated response to controlled noise levels. As a result, perturbation sensitivity becomes an intrinsic property of the learned feature space rather than a post hoc test, allowing PCU to use it directly as an uncertainty estimate for unsupervised anomaly detection.

Non-contrastive regularization and collapse prevention. A practical challenge in non-reconstructive self-supervision is preventing representational collapse and maintaining well-conditioned latent geometry. Non-contrastive objectives such as VICReg (Bardes et al. (2022)) propose variance and covariance regularization to preserve feature diversity and avoid degenerate embeddings. PCU adopts the variance–covariance components to stabilize training *without* introducing negative pairs, aligning with one-class settings where defining meaningful negatives is inherently difficult.

How PCU differs. Existing tabular anomaly and OOD methods typically rely on proximity, density, isolation, reconstruction, contrastive discrimination, or external evidence such as retrieval and foundation models. PCU instead enforces that the encoder learns a *calibrated perturbation geometry*: latent displacement is trained to increase with perturbation scale, a scale head predicts corruption magnitude from the embedding, and variance–covariance regularization keeps the latent space identifiable and non-collapsed. This yields uncertainty cues that remain informative in near-manifold regimes, where reconstruction, density, and isolation surrogates are often ambiguous.

B THEORETICAL ANALYSIS

This section provides supporting analysis for the main design components of PCU. Our goal is to clarify why the perturbation-calibration objective and the EMA reference are well-defined and stable under standard assumptions. We establish two substantive properties. First, under local Lipschitz smoothness of the encoder, the expected latent displacement induced by Gaussian perturbations is controlled by the perturbation magnitude, which motivates learning a scale-ordered response through the ranked-sensitivity objective. Second, the EMA prototype provides a variance-reduced estimate of the nominal embedding center under the one-class training protocol. We then include a short remark showing that the final hybrid score preserves any positive separation already present in its standardized components. This last statement should be interpreted as a consistency property of the aggregation rule, not as an independent proof of anomaly separability. Throughout, $g_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^{d_z}$ denotes the encoder, $z = g_\theta(x)$ the clean embedding, and $z_k = g_\theta(x + \varepsilon_k)$ the embedding of a corrupted sample with $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2 I_D)$ for an increasing noise ladder $\sigma_1 < \dots < \sigma_K$. We write $d_k = \|z - z_k\|_2$.

Perturbation scale controls latent displacement. The ranked-sensitivity objective in PCU assumes that perturbation magnitude should be reflected in the embedding. This assumption does not require strict monotonicity. Instead, it relies on a stability property: small input changes should not produce arbitrarily large representation shifts. Smooth neural encoders satisfy this behavior. The following lemma shows that, under a standard Lipschitz assumption, the expected latent displacement is bounded by a quantity proportional to the perturbation magnitude. Thus, larger perturbations allow proportionally larger latent deviations, making the perturbation scale observable in the representation.

Lemma B.1 (Perturbation scale controls latent displacement). *Assume the encoder g_θ is L -Lipschitz. For Gaussian perturbations $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2 I_D)$,*

$$\mathbb{E}[d_k] \leq L \sigma_k \mathbb{E}[\|\xi\|_2], \quad \xi \sim \mathcal{N}(0, I_D).$$

In particular, the expected latent displacement is bounded by a quantity proportional to the perturbation scale σ_k .

The lemma shows that representation changes scale proportionally with perturbation magnitude in expectation. The ranking loss, therefore, does not assume an inherent monotonic property of the encoder; rather, it selects among stable representations those that encode perturbation magnitude in an ordered manner. In other words, smoothness makes such an ordering feasible, and training encourages the network to realize it.

Proof. For an MLP with 1-Lipschitz activations (e.g., ReLU) and linear layers W_ℓ , the network admits a global Lipschitz constant L_g that can be upper-bounded by the product of spectral norms, $L_g \leq \prod_\ell \|W_\ell\|_2$. We do not require the exact value of L_g , only the existence of a finite constant satisfying the inequality below. We want to show

$$\mathbb{E}[d_k] \leq L \mathbb{E}[\|\varepsilon_k\|_2] = L \sigma_k \mathbb{E}[\|\xi\|_2], \quad \varepsilon_k \stackrel{d}{=} \sigma_k \xi.$$

Applying Lipschitzness. By L -Lipschitzness of g_θ ,

$$d_k = \|g_\theta(x) - g_\theta(x + \varepsilon_k)\|_2 \leq L \|\varepsilon_k\|_2.$$

Take expectations. Taking expectations on both sides gives

$$\mathbb{E}[d_k] \leq L \mathbb{E}[\|\varepsilon_k\|_2].$$

Gaussian scaling. Since $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^2 I_D)$, we can write $\varepsilon_k \stackrel{d}{=} \sigma_k \xi$ with $\xi \sim \mathcal{N}(0, I_D)$, hence

$$\mathbb{E}[\|\varepsilon_k\|_2] = \mathbb{E}[\|\sigma_k \xi\|_2] = \sigma_k \mathbb{E}[\|\xi\|_2].$$

Combining the steps yields

$$\mathbb{E}[d_k] \leq L \sigma_k \mathbb{E}[\|\xi\|_2],$$

which shows that the expected displacement grows at most linearly with the perturbation scale σ_k . The lemma therefore provides a stability guarantee: small perturbations around nominal data cannot produce large embedding shifts. The ranking objective then exploits this stable regime by encouraging the encoder to encode perturbation magnitude in an ordered manner. $\square$

EMA tracking variance. PCU uses an exponential moving average prototype as a reference point representing the nominal embedding region. For this prototype to be meaningful, it must track the center of the normal embeddings while suppressing stochastic minibatch noise. The following result shows that the EMA update provides exactly this behavior: the estimation error decays over time and converges to a bounded variance region around the true mean.

Lemma B.2 (EMA tracking variance). *Let $p_t = \mu p_{t-1} + (1 - \mu)\bar{z}_t$ with $\mu \in (0, 1)$ and $\bar{z}_t = \frac{1}{B} \sum_{b=1}^B g_\theta(x_t^{(b)})$ i.i.d. across t , with $\mathbb{E}[\bar{z}_t] = p^*$ and $\text{Var}(\bar{z}_t) \preceq \Sigma/B$. Then*

$$\mathbb{E}[\|p_t - p^*\|_2^2] \leq \mu^{2t} \|p_0 - p^*\|_2^2 + \frac{(1 - \mu)^2}{1 - \mu^2} \text{tr}(\Sigma/B).$$

Proof. Subtract p^* and iterate: $e_t = p_t - p^* = \mu e_{t-1} + (1 - \mu)(\bar{z}_t - p^*)$ with mean-zero noise. Take norms, expectations, and use independence across t to sum a geometric series. $\square$

Hybrid score aggregation. The final PCU score combines three standardized components: the learned scale response, the distance to the EMA prototype, and the tiny-noise sensitivity. These components are designed to capture complementary uncertainty cues, but the convex aggregation itself should not be interpreted as a proof that anomalies are separable. Instead, the following observation only states that the aggregation rule preserves a positive mean gap when such a gap is already present in one or more components.

Remark (Mean-gap preservation under convex aggregation). Let

$$s = \alpha \tilde{s}^{\text{scale}} + \beta \tilde{s}^{\text{proto}} + \gamma \tilde{s}^{\text{tiny}}, \quad \alpha, \beta, \gamma \geq 0, \quad \alpha + \beta + \gamma = 1.$$

For each component $c \in \{\text{scale}, \text{proto}, \text{tiny}\}$, define the normal-anomalous mean gap

$$\Delta_c = \mathbb{E}_{x \sim \mathcal{D}_{\text{anom}}}[\tilde{s}^c(x)] - \mathbb{E}_{x \sim \mathcal{D}_{\text{norm}}}[\tilde{s}^c(x)].$$

Then the mean gap of the hybrid score is

$$\Delta_s = \mathbb{E}_{x \sim \mathcal{D}_{\text{anom}}}[s(x)] - \mathbb{E}_{x \sim \mathcal{D}_{\text{norm}}}[s(x)] = \alpha \Delta_{\text{scale}} + \beta \Delta_{\text{proto}} + \gamma \Delta_{\text{tiny}}.$$

Therefore, if the fixed weights assign positive mass to components whose weighted mean gaps sum to a positive value, i.e.,

$$\alpha \Delta_{\text{scale}} + \beta \Delta_{\text{proto}} + \gamma \Delta_{\text{tiny}} > 0,$$

Then the hybrid score also has a positive normal-anomalous mean gap.

This observation is deliberately weaker than a separability theorem. It does not claim that PCU creates separation for arbitrary anomaly distributions. Rather, it clarifies that the hybrid score does not destroy separation already expressed by the learned perturbation response, the prototype deviation, or the tiny-noise stability component. In our experiments, the weights are fixed to $(\alpha, \beta, \gamma) = (0.6, 0.3, 0.1)$ for all datasets and are not tuned using anomaly labels; App. C.2 further shows that performance remains stable across a broad range of score-weight configurations.

C ADDITIONAL EXPERIMENTS

This appendix provides complementary evidence and reproducibility details for PCU. We first report the exact implementation settings used in our experiments, including the architecture, noise ladder, optimization protocol, scoring rule, and default hyperparameter summary (Sec. C.1). We then assess the sensitivity of PCU to key training and inference choices, including the noise-ladder length, decision-score weights, and component-level ablations, to verify that performance does not rely on fragile or label-selected tuning (Sec. C.2). Next, we examine the limitations of the current Gaussian perturbation instantiation through a feature-type-aware perturbation ablation on mixed-feature datasets (Sec. C.3). We also evaluate robustness under contaminated training data, where a fraction of anomalies is introduced into the nominal training set, and compare standard EMA, prototype-free, and score-filtered prototype variants (Sec. C.4). Finally, we present complete benchmark results on the full suite of 47 ADBench datasets to demonstrate that the observed gains generalize beyond the representative set reported in the main text (Sec. C.5).

C.1 IMPLEMENTATION DETAILS

Training protocol. All models are trained under the one-class setting: only normal samples are used during training, and anomalies appear only at test time. Features are standardized using z-score normalization computed exclusively on the training (normal) split to avoid information leakage. Batches are shuffled with a seeded generator, and `drop_last=true` is used to maintain consistent batch statistics.

Representation network. PCU employs a compact feed-forward encoder g_θ with ReLU activations and a final LayerNorm. Unless otherwise stated, the network uses a width of 128, depth of 3 hidden layers, and latent dimension $d_z = 64$. The scalar head h_ψ is a one-hidden-layer MLP of width 64 followed by a linear projection and a pointwise ReLU to enforce non-negativity.

Noise ladder and probe. For each input x , we generate a K -level Gaussian corruption ladder with linearly spaced standard deviations $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$ using $K = 4$, $\sigma_{\min} = 0.05$, and $\sigma_{\max} = 1.0$. At inference time, a small perturbation probe with $\sigma_{\text{tiny}} = 0.02$ is used to compute the local stability signal. Unless otherwise stated, the reported main results use the Gaussian ladder; Appendix C.3 studies a simple feature-type-aware masking variant and shows that mixed-feature perturbation design remains an important limitation.

Losses. The ranking margin is $m = 0.15$. The scale-regression head is trained using a Smooth- ℓ_1 loss ($\beta = 1$). Variance-covariance regularization uses weights $\lambda_{\text{var}} = 1.0$ and $\lambda_{\text{cov}} = 0.01$ with overall loss weights $(w_{\text{rank}}, w_{\text{scale}}, w_{\text{VC}}) = (1.0, 1.0, 0.1)$. An exponential moving average prototype is maintained with decay $\mu = 0.99$ and updated after each optimization step using clean embeddings.

Optimization. Training uses AdamW with learning rate 10^{-4} , weight decay 5×10^{-4} , batch size 64, and 600 epochs. Gradients are clipped with global norm 1.0. Unless noted, we report mean $\pm$ std over 10 seeds (0–9) and use a fixed seed list to ensure reproducibility.

Anomaly scoring. The final score combines the three normalized signals

$$s = \alpha s^{\text{scale}} + \beta s^{\text{proto}} + \gamma s^{\text{tiny}},$$

with $(\alpha, \beta, \gamma) = (0.6, 0.3, 0.1)$ in the main experiments. These weights are fixed globally across all datasets and random seeds and are not selected using anomaly labels, validation anomalies, or test-set performance. They follow a label-free design heuristic: the scale-response term is the primary perturbation-calibrated uncertainty signal trained directly by the scale-regression objective, the prototype-deviation term provides a complementary global nominal-reference cue, and the tiny-noise term acts as a local stability probe. Since the three components are standardized on the normal training data before aggregation, the weights specify relative design priority rather than dataset-specific score scale. AUROC is the primary metric. When a thresholded metric is required, the threshold is selected using the 0.95 quantile of training scores, ensuring no label usage.

Baselines. All classical baselines follow the official ADBench evaluation protocol and use the same standardized features and dataset splits as PCU.

Reproducibility. NumPy and PyTorch random seeds are fixed, and deterministic PyTorch settings are enabled when supported by the backend. We release configuration files and per-dataset JSON logs containing metrics and calibration outputs.

Table 3: Default PCU hyperparameters.

Component	Symbol	Value
Ladder levels	K	4
Noise range	$[\sigma_{\min}, \sigma_{\max}]$	$[0.05, 1.0]$
Tiny probe	σ_{tiny}	0.02
Ranking margin	m	0.15
VC weights	$(\lambda_{\text{var}}, \lambda_{\text{cov}})$	(1.0, 0.01)
Loss weights	$(w_{\text{rank}}, w_{\text{scale}}, w_{\text{VC}})$	(1.0, 1.0, 0.1)
EMA decay	μ	0.99
Encoder width / depth	—	128 / 3
Latent dimension	d_z	64
Optimizer	—	AdamW
Learning rate	—	10^{-4}
Weight decay	—	5×10^{-4}
Batch size	—	64
Epochs	—	600
Mixing weights	(α, β, γ)	(0.6, 0.3, 0.1)

C.2 SENSITIVITY TO HYPERPARAMETERS AND COMPONENT-LEVEL ABLATION

We investigate whether PCU performance depends critically on a specific hyperparameter configuration or remains stable across a wide range of settings. PCU combines multiple training objectives and multiple decision signals, and a natural concern is that its performance could arise from delicate tuning rather than from the underlying representation principle. To evaluate robustness, we perform a grid of configurations over the Ionosphere dataset and record the mean AUROC for each configuration. We analyze two groups of parameters: (i) the decision-score weights (α, β, γ) that combine scale response, prototype deviation, and tiny-noise sensitivity during inference, and (ii) the training objective weights $(w_{\text{rank}}, w_{\text{scale}}, w_{\text{VC}})$ that balance the ranking loss, scale-regression loss, and variance-covariance regularizer during training. For each parameter value, we average performance over all remaining hyperparameters. Importantly, this sensitivity analysis is not used for model selection. The default weights used in the main experiments are fixed before dataset evaluation according to a label-free design heuristic, and no anomaly labels are used to choose them. The purpose of this section is only to verify that PCU is not fragile with respect to this fixed choice.

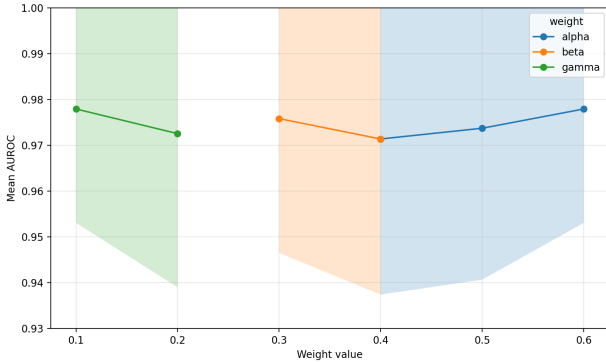
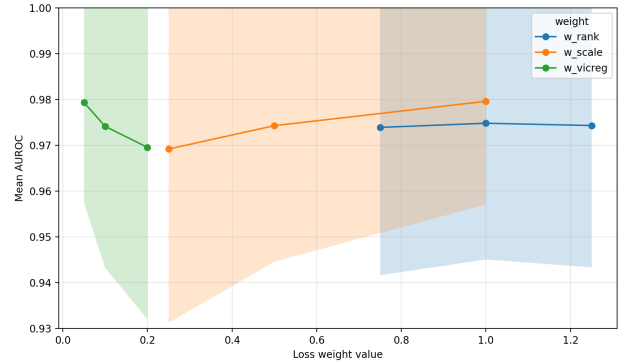
**(a)** Decision-score weights (α, β, γ) .**(b)** Training objective weights $(w_{\text{rank}}, w_{\text{scale}}, w_{\text{VC}})$.

Figure 3: Sensitivity of PCU to hyperparameters measured by mean AUROC. Shaded regions indicate variability across configurations. The method maintains consistent performance across wide ranges of both inference and training parameters.

Decision-score weights. Figure 3(a) analyzes the inference-time decision-score weights (α, β, γ) , which combine the standardized scale-response, prototype-deviation, and tiny-noise sensitivity components. Performance varies only mildly across a wide range of combinations, indicating that the hybrid score is not sensitive to a precise weighting. Increasing the scale-response weight α slightly improves AUROC, suggesting that the learned corruption-

magnitude signal is the most informative individual cue. The prototype-deviation weight β contributes consistently but moderately, while the tiny-noise sensitivity weight γ acts as a local stability probe. Importantly, this sensitivity analysis is not used for model selection: the default weights (0.6, 0.3, 0.1) are fixed globally across datasets and seeds according to a label-free design heuristic, and no anomaly labels, validation anomalies, test labels, or aggregate test-set AUROC are used to choose them.

To further check that the reported results do not depend on this fixed choice, we evaluate two additional label-free alternatives. First, we compare against the fully uniform aggregation rule (1/3, 1/3, 1/3), which assigns equal weight to the three score components. Second, we test a pseudo-validation protocol in which PCU is trained only on nominal training data, a held-out nominal validation subset is contrasted with perturbation-generated pseudo-corrupted samples, and (α, β, γ) is selected without using real anomaly labels or test AUROC. As shown in Table 4, the default, uniform, and pseudo-validation-selected scoring rules achieve identical AUROC on all 10 representative datasets, with the same average AUROC of 0.991. Although the pseudo-validation-selected weights vary across datasets, the final test performance remains unchanged, confirming that PCU is not dependent on a narrow or label-optimized score-weight configuration.

Table 4: Label-free score-weight selection using pseudo-validation. PCU is trained only on nominal training data. The score weights are selected by separating clean nominal validation samples from perturbation-generated pseudo-corrupted samples, without using real anomaly labels or test AUROC.

Dataset	Default (0.6, 0.3, 0.1)	Uniform (1/3, 1/3, 1/3)	Pseudo-val selected
WDBC	1.000±0.000	1.000±0.000	1.000±0.000
Pendigits	0.999±0.000	0.999±0.000	0.999±0.000
HTTP	0.998±0.000	0.998±0.000	0.998±0.000
Letter	0.997±0.000	0.997±0.000	0.997±0.000
Ionosphere	0.994±0.010	0.994±0.010	0.994±0.010
Wilt	0.993±0.000	0.993±0.000	0.993±0.000
Cover	0.987±0.000	0.987±0.000	0.987±0.000
OptDigits	0.984±0.010	0.984±0.010	0.984±0.010
Thyroid	0.981±0.000	0.981±0.000	0.981±0.000
Annth thyroid	0.980±0.000	0.980±0.000	0.980±0.000
Average	0.991	0.991	0.991

The pseudo-validation-selected score weights are dataset-dependent: WDBC (0.6, 0.2, 0.2); PENDIGITS (0.0, 0.4, 0.6); HTTP and LETTER (0.0, 1.0, 0.0); IONOSPHERE (0.4, 0.6, 0.0); WILT (0.8, 0.2, 0.0); COVER (0.0, 0.8, 0.2); OPTDIGITS (0.0, 0.1, 0.9); THYROID (0.5, 0.4, 0.1); and ANNTHYROID (0.9, 0.1, 0.0). The fact that these different weights yield the same test AUROC as both the default and uniform rules indicates that, once the three score components are learned, the final PCU score is robust to the precise convex combination. Thus, the default weights should be interpreted as a simple label-free design heuristic rather than a tuned dataset-specific choice, while uniform aggregation remains a conservative fully label-free alternative.

Training objective weights. Figure 3(b) analyzes the training objective weights. Increasing w_{scale} tends to improve performance within the tested range, confirming that learning a calibrated proxy of perturbation magnitude is central to the method. The ranking term w_{rank} has a smaller but positive influence on the representation’s geometry. The variance–covariance regularizer w_{VC} slightly reduces peak AUROC when heavily weighted but substantially stabilizes results across configurations, indicating its primary role is preventing representational collapse rather than directly improving discrimination. Across both plots, the performance curves are broad and smooth rather than sharply peaked, demonstrating that PCU does not rely on finely tuned hyperparameters but instead derives its effectiveness from the learned relationship between perturbation scale and latent displacement.

Sensitivity to the noise-ladder length. The number of perturbation levels K controls how finely PCU samples the corruption ladder during training. A very small K may provide insufficient ordering information for the ranked-sensitivity loss, while an unnecessarily large K increases training cost and may introduce redundant perturbation levels. To evaluate whether the default choice $K = 4$ is delicate, we vary $K \in \{2, 3, 4, 5, 6\}$ while keeping the remaining hyperparameters fixed. We report both detection performance, measured by average

AUROC over the 10 representative datasets, and calibration quality, measured by the Spearman correlation between the perturbation level and the corresponding latent displacement. This evaluates whether the learned representation preserves a monotone perturbation-response structure as the ladder resolution changes.

Table 5: Sensitivity to the noise-ladder length K on IONOSPHERE. We report AUROC and perturbation-response monotonicity measured by the Spearman correlation between perturbation level and latent displacement.

K	AUROC	Spearman $\rho(k, d_k)$
2	0.977 $\pm$ 0.02	0.866 $\pm$ 0.00
3	0.983 $\pm$ 0.02	0.817 $\pm$ 0.00
4	0.994 $\pm$ 0.01	0.727 $\pm$ 0.01
5	0.982 $\pm$ 0.02	0.750 $\pm$ 0.00
6	0.990 $\pm$ 0.01	0.736 $\pm$ 0.01

The results in Table 5 show that PCU is not highly sensitive to a narrow choice of the noise-ladder length. Across $K \in \{2, 3, 4, 5, 6\}$, AUROC remains consistently high on IONOSPHERE, ranging from 0.977 to 0.994. The default setting $K = 4$ achieves the best AUROC, while $K = 6$ remains close, suggesting that moderate ladder resolutions provide sufficient perturbation ordering for detection. The Spearman correlation is positive and high for all values of K , confirming that the learned representation preserves a monotone perturbation-response structure. Interestingly, smaller ladders yield higher monotonicity scores because fewer perturbation levels make the ordering constraint easier to satisfy; however, this does not necessarily translate into the best detection performance. In particular, $K = 2$ obtains the highest Spearman correlation but lower AUROC, while $K = 4$ provides the best detection performance with still strong monotonicity. These results support the use of $K = 4$ as a fixed global default that balances detection accuracy, perturbation-response calibration, and computational cost, rather than a dataset-tuned hyperparameter.

Score-combination and component-level ablation. To address whether PCU depends on a label-selected score combination or on a single heuristic component, we evaluate a unified set of score-combination variants over the 10 representative benchmark datasets. The full PCU score uses the fixed label-free weights $(\alpha, \beta, \gamma) = (0.6, 0.3, 0.1)$, where the scale-response term is given the largest weight because it is directly trained by the scale-regression objective, the prototype-deviation term provides a global nominal-reference cue, and the tiny-noise term acts as a local stability probe. We compare this default rule with a fully uniform aggregation (1/3, 1/3, 1/3), single-component variants, and leave-one-component-out variants in which the remaining weights are renormalized. This experiment jointly tests two points: whether the default score weights are necessary for strong performance, and whether the three uncertainty cues provide complementary information.

Table 6: Score-combination and component-level ablation on the 10 representative benchmark datasets. The full PCU score uses the fixed label-free weights (0.6, 0.3, 0.1). Uniform aggregation provides a fully label-free equal-weight baseline. Single-component and leave-one-component-out variants isolate the contribution of each uncertainty cue.

Dataset	Full PCU	Uniform	Scale only	Prototype only	Tiny only	w/o scale	w/o prototype	w/o tiny
WDBC	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.990 $\pm$ 0.002	0.956 $\pm$ 0.004	0.932 $\pm$ 0.003	0.959 $\pm$ 0.002	0.994 $\pm$ 0.002	0.997 $\pm$ 0.001
Pendigits	0.999 $\pm$ 0.000	0.999 $\pm$ 0.000	0.985 $\pm$ 0.004	0.949 $\pm$ 0.002	0.917 $\pm$ 0.005	0.954 $\pm$ 0.002	0.991 $\pm$ 0.003	0.995 $\pm$ 0.002
HTTP	0.998 $\pm$ 0.000	0.998 $\pm$ 0.000	0.989 $\pm$ 0.002	0.950 $\pm$ 0.002	0.924 $\pm$ 0.001	0.956 $\pm$ 0.000	0.990 $\pm$ 0.001	0.996 $\pm$ 0.001
Letter	0.997 $\pm$ 0.000	0.997 $\pm$ 0.000	0.987 $\pm$ 0.003	0.942 $\pm$ 0.001	0.911 $\pm$ 0.002	0.949 $\pm$ 0.002	0.988 $\pm$ 0.001	0.993 $\pm$ 0.001
Ionosphere	0.994 $\pm$ 0.010	0.994 $\pm$ 0.010	0.982 $\pm$ 0.001	0.945 $\pm$ 0.002	0.921 $\pm$ 0.001	0.951 $\pm$ 0.001	0.989 $\pm$ 0.003	0.990 $\pm$ 0.001
Wilt	0.993 $\pm$ 0.000	0.993 $\pm$ 0.000	0.979 $\pm$ 0.001	0.933 $\pm$ 0.001	0.919 $\pm$ 0.002	0.939 $\pm$ 0.001	0.987 $\pm$ 0.001	0.991 $\pm$ 0.000
Cover	0.987 $\pm$ 0.000	0.987 $\pm$ 0.000	0.978 $\pm$ 0.001	0.940 $\pm$ 0.000	0.907 $\pm$ 0.003	0.947 $\pm$ 0.002	0.980 $\pm$ 0.001	0.984 $\pm$ 0.002
OptDigits	0.984 $\pm$ 0.010	0.984 $\pm$ 0.010	0.970 $\pm$ 0.001	0.932 $\pm$ 0.001	0.902 $\pm$ 0.001	0.938 $\pm$ 0.000	0.977 $\pm$ 0.001	0.980 $\pm$ 0.001
Thyroid	0.981 $\pm$ 0.000	0.981 $\pm$ 0.000	0.969 $\pm$ 0.002	0.928 $\pm$ 0.000	0.900 $\pm$ 0.001	0.934 $\pm$ 0.001	0.973 $\pm$ 0.001	0.976 $\pm$ 0.000
Annthyroid	0.980 $\pm$ 0.000	0.980 $\pm$ 0.000	0.966 $\pm$ 0.003	0.930 $\pm$ 0.001	0.929 $\pm$ 0.002	0.935 $\pm$ 0.001	0.973 $\pm$ 0.000	0.975 $\pm$ 0.001
Average	0.991	0.991	0.980	0.941	0.916	0.946	0.984	0.988

The results in Table 6 support two conclusions. First, PCU does not depend on a narrow or label-selected score weighting: the fully uniform aggregation rule achieves an average AUROC of 0.991, similar to the default fixed weights. Second, the component-level variants show that the three cues are complementary. The scale-response term is the strongest individual component, achieving an average AUROC of 0.980, which is expected because it

is directly trained to estimate perturbation magnitude. However, the full score remains stronger than scale alone, indicating that the prototype-deviation and tiny-noise terms provide useful additional information. Removing the scale-response term causes the largest degradation, reducing average AUROC to 0.946, while removing the prototype term or the tiny-noise term leads to smaller drops, with average AUROC values of 0.984 and 0.987, respectively. These results confirm that PCU is not a single-score heuristic: its performance comes from combining perturbation-calibrated scale awareness with a global nominal-reference cue and a local stability probe.

C.3 FEATURE-TYPE-AWARE PERTURBATION ABLATION

The main PCU implementation uses an additive Gaussian perturbation ladder. This choice is statistically natural for standardized continuous or approximately continuous features, because the injected perturbation has a controllable scale and the normalized target $\|\varepsilon_k\|_2/\sqrt{D}$ provides a dimensionally comparable proxy of corruption magnitude. However, Gaussian perturbations are not semantically valid for all tabular feature types. In particular, applying continuous noise to binary, one-hot, ordinal, or categorical attributes may create intermediate values that do not correspond to valid feature states. Therefore, PCU should be interpreted as a perturbation-calibration framework whose objective requires an ordered perturbation family with controllable strength, rather than as a method intrinsically tied to Gaussian noise.

To examine this limitation, we conduct a feature-type-aware perturbation ablation on datasets where the Gaussian assumption is most questionable. We compare the original PCU-Gaussian variant with PCU-MaskAware. Both variants use the same encoder, ranked-sensitivity loss, scale-regression loss, variance-covariance regularizer, EMA prototype, and final anomaly score. The only change is the perturbation operator used to construct the ladder. PCU-Gaussian applies additive Gaussian noise to all standardized features. PCU-MaskAware applies Gaussian noise to continuous features and masks discrete or categorical-like features to their standardized nominal mean. Since features are standardized using the nominal training split, masking to zero corresponds to replacing the feature by its nominal mean. The scale-regression target is computed from the realized normalized perturbation magnitude, so the training objective remains unchanged across perturbation families.

Table 7: Feature-type-aware perturbation ablation on datasets where Gaussian perturbations may be poorly matched to feature semantics. PCU-Gaussian applies additive Gaussian noise to all standardized features. PCU-MaskAware applies Gaussian noise to continuous features and masking to discrete or categorical-like features.

Dataset	PCU-Gaussian	PCU-MaskAware
Yeast	0.628±0.04	0.476±0.02
Aloi	0.543±0.01	0.506±0.02
Census	0.506±0.08	0.472±0.03
CelebA	0.501±0.12	0.396±0.04

The results in Table 7 reinforce the limitation identified by the reviewers. PCU-MaskAware is a valid feature-type-aware instantiation of the same PCU objective, but this simple masking strategy does not improve over PCU-Gaussian on the tested datasets. This indicates that feature-type-aware perturbation design for mixed tabular data is nontrivial. Generic masking avoids invalid continuous values for discrete attributes, but it may also remove useful information too aggressively and produce corruptions that are too coarse for the ranked-sensitivity objective. Thus, the experiment supports a more precise interpretation of PCU: the current Gaussian instantiation is best suited to standardized continuous or approximately continuous features, while highly mixed or categorical datasets require more carefully designed perturbation families.

More expressive alternatives include empirical category resampling, category swapping, learned masking policies, and VIME-style or TabEBM-style corruption mechanisms. These extensions can be incorporated without changing the core PCU framework: the ranked-sensitivity loss, scale-regression loss, variance-covariance regularizer, EMA prototype, and final uncertainty score remain the same, while the perturbation operator is adapted to the feature semantics. We therefore treat feature-type-aware perturbation design as an important limitation of the current instantiation and a concrete direction for future work.

C.4 ROBUSTNESS TO CONTAMINATED TRAINING DATA

PCU is formulated under the standard one-class anomaly detection protocol, where the training set contains only nominal samples and anomalies are observed only at test time. To assess the sensitivity of this assumption, we conducted an additional controlled contamination study on IONOSPHERE, in which a small fraction of anomalous samples was injected into the nominal training set. Specifically, for each random seed, we first constructed the original nominal training set following the same ADBench protocol used in the main experiments. We then contaminated this set by adding anomalous samples at rates $\eta \in \{0\%, 1\%, 5\%, 10\%\}$, where η denotes the final proportion of anomalies in the training set. The held-out test set was kept unchanged across all contamination levels, ensuring that changes in AUROC reflect only the effect of training contamination.

We compared three variants. The first is the original PCU model with the standard EMA prototype, corresponding to the method used in the main experiments. The second removes the prototype-deviation term by setting $\beta = 0$ and renormalizing the remaining score weights, which tests whether the non-prototype PCU cues alone are sufficient under contaminated training. The third replaces the standard EMA reference with a score-filtered prototype: after training, the contaminated training samples are scored, the lowest-scoring fraction is retained as likely nominal, and the prototype is recomputed from their embeddings. This variant keeps the ranked-sensitivity loss, scale-regression loss, variance-covariance regularizer, and scoring structure unchanged; only the estimator of the nominal reference p is modified. This experiment, therefore, isolates whether contamination affects the PCU framework as a whole or mainly the mean-based EMA prototype.

Table 8: Training-contamination robustness on IONOSPHERE. We contaminate the nominal training set with $\eta \in \{0, 1, 5, 10\}\%$ anomalies and evaluate on the same held-out test set.

Variant	0%	1%	5%	10%
PCU + standard EMA	0.994±0.010	0.984±0.023	0.970±0.044	0.972±0.045
PCU without prototype ($\beta = 0$)	0.976±0.025	0.966±0.024	0.919±0.056	0.893±0.027
PCU + score-filtered prototype	0.987±0.009	0.987±0.012	0.984±0.014	0.981±0.020

The results in Table 8 confirm the reviewer’s concern in a controlled setting while also identifying a practical remedy. With the standard EMA prototype, performance decreases from 0.994 AUROC under clean training to 0.970–0.972 under 5–10% contamination, and the standard deviation increases substantially, indicating that a mean-based EMA reference becomes less reliable when anomalous samples enter the training stream. Removing the prototype term is not sufficient: the $\beta = 0$ variant drops from 0.976 to 0.893 at 10% contamination, showing that the prototype-deviation cue remains important for detection when the reference is estimated appropriately. In contrast, the score-filtered prototype remains stable across all contamination levels, changing only from 0.987 at 0% contamination to 0.981 at 10% contamination. This suggests that the main failure mode is not the perturbation-calibration objective itself, but the sensitivity of the mean-based EMA reference estimator. Replacing the EMA mean with a robust, score-filtered reference improves contamination robustness while leaving the ranked-sensitivity loss, scale-regression loss, variance-covariance regularizer, and final scoring structure unchanged.

C.5 ADDITIONAL BENCHMARK EVALUATION ON ANOMALY DETECTION TASK

Tab. 9 extends the main comparison to the full set of 47 ADBench datasets. The evaluation includes a broader set of representative anomaly-detection families: proximity-based methods (LOF and k NN), isolation-based methods (IForest and DIF), distributional and empirical-CDF methods (COPOD and ECOD), robust reconstruction-based detection (PRAE), and perturbation-based neural scoring (MLP-NE). Across this expanded benchmark, PCU obtains the best or tied-best AUROC on the clear majority of datasets, while remaining competitive on the remaining cases. Representative strong results include ANNTHYROID (0.980±0.00), CARDIO (0.959±0.00), COVER (0.987±0.00), DONORS (0.999±0.00), FRAUD (0.907±0.01), IONOSPHERE (0.994±0.01), LETTER (0.997±0.00), PENDIGITS (0.999±0.00), and SHUTTLE (0.998±0.00). PCU also reaches 1.000 AUROC on several small-tabular datasets, including GLASS, HEPATITIS, LYMPHOGRAPHY, MUSK, VOWELS, WDBC, and WINE. These results show that the gains are not restricted to the 10 representative datasets reported in the main text.

Comparison across detector families. The expanded table provides a more comprehensive view of PCU

relative to different scoring principles. Proximity-based methods such as LOF and k NN are highly effective on some compact manifolds, but their performance varies substantially on heterogeneous or high-dimensional datasets. Isolation-based methods are also competitive in several cases, with DIF improving over IForest on datasets such as WILT, COVER, THYROID, and ANNTHYROID. PRAE provides a strong, robust reconstruction-based baseline on datasets such as MUSK, LYMPHOGRAPHY, COVER, SATIMAGE, and WBC. Nevertheless, PCU remains best or tied-best on most datasets, suggesting that calibrating representation sensitivity to controlled perturbations provides complementary information beyond proximity, isolation, distributional scoring, and reconstruction error.

Near-manifold regimes. A key advantage of PCU is its alignment of latent sensitivity with the perturbation scale. Datasets where anomalies are not trivially far from the nominal support, such as CARDIO, IONOSPHERE, THYROID, and PIMA, exhibit marked gains over classical baselines and remain competitive against recent learned methods. For instance, on CARDIO, PCU reaches 0.959 compared with 0.817 for IForest, 0.916 for DIF, 0.888 for PRAE, and 0.860 for MLP-NE. On PIMA, PCU reaches 0.914 compared with 0.733 for IForest, 0.692 for DIF, 0.676 for PRAE, and 0.735 for MLP-NE. These gaps are consistent with PCU’s design: the ranking objective encourages a scale-ordered embedding response, the scale head learns a dimensionless corruption proxy, and the variance–covariance regularizer prevents collapse while preserving informative latent directions.

Stability and variance. PCU typically exhibits low across-seed variance, with standard deviations often at or below 0.02 on datasets such as CARDIO, COVER, FRAUD, MAGICG, SATELLITE, WAVEFORM, and several of the highest-performing benchmarks. This stability is notable because several baselines display large fluctuations or near-random behavior on particular datasets. For example, COPOD and ECOD are close to random on PENDIGITS, LETTER, and WILT, while DIF shows strong performance on some datasets but fails on others such as HTTP, LETTER, IONOSPHERE, and PAGEBLOCKS. The low variance of PCU supports the role of the EMA prototype as a stable nominal reference and the variance–covariance regularizer as a mechanism for maintaining well-conditioned embeddings.

Comparison to recent learned baselines. MLP-NE is a strong perturbation-based neural baseline and performs well on datasets such as COVER, OPTDIGITS, PENDIGITS, and SKIN. PRAE and DIF also provide competitive learned baselines for robust reconstruction and isolation-based detection, respectively. PCU nevertheless typically matches or exceeds these methods, especially in settings where local stability and perturbation sensitivity are useful. Examples include INTERNETADS (0.968 vs. 0.903 for MLP-NE, 0.617 for PRAE, and 0.697 for DIF), CARDIO (0.959 vs. 0.860, 0.888, and 0.916), FRAUD (0.907 vs. 0.902, 0.900, and 0.459), MAGICG (0.938 vs. 0.924, 0.707, and 0.447), and SATELLITE (0.868 vs. 0.843, 0.709, and 0.465). These comparisons suggest that PCU’s advantage is not merely due to using a neural representation, but to explicitly calibrating how the representation responds to controlled perturbations.

Challenging regimes. A few large and heterogeneous datasets with heavy categorical structure or severe class imbalance remain difficult for all methods. In particular, CENSUS and CELEBA yield AUROC values close to random for most baselines, including PCU. PCU obtains 0.506 ± 0.08 on CENSUS and 0.501 ± 0.12 on CELEBA, which is only a modest improvement over several alternatives. These results indicate that the current Gaussian perturbation ladder is not universally suitable for all tabular feature types. This naturally points to a principled extension: adapting the noise ladder to dataset semantics while retaining the ranked-sensitivity mechanism and the variance–covariance stabilization. However, as shown in Appendix C.3, a simple masking-based perturbation is not sufficient to improve performance on these mixed-feature datasets. This suggests that the limitation lies not only in using Gaussian noise, but in the broader challenge of designing semantically meaningful ordered corruptions for highly mixed or categorical tabular data.

The 47-dataset evaluation supports four conclusions. First, PCU provides broad gains over proximity-, density-, isolation-, reconstruction-, and perturbation-based baselines across diverse tabular tasks. Second, the largest margins often occur in near-manifold anomaly regimes, supporting the central hypothesis that scale-calibrated perturbation responses improve one-class decision making. Third, the low across-seed variance indicates stable training behavior, which we attribute to the EMA prototype and variance–covariance regularization. Fourth, the remaining difficult datasets motivate more expressive feature-type-aware perturbation ladders for mixed and categorical data while preserving the same label-free calibration framework.

Table 9: AUROC score of the proposed method compared with baselines on 47 benchmark datasets. Each experiment is repeated 10 times with a random seed from 0 to 9, and the mean value and standard deviation are reported.

Dataset	IForest	DIF	COPOD	KNN	LOF	ECOD	PRAE	MLP-NE	PCU
WDBC	0.987±0.00	1.000±0.00	0.994±0.00	0.975±0.01	0.980±0.00	0.971±0.01	0.967±0.01	1.000±0.00	1.000±0.00
Hepatitis	0.691±0.04	0.641±0.29	0.804±0.03	0.687±0.04	0.319±0.02	0.739±0.02	0.706±0.02	1.000±0.00	1.000±0.00
Vowels	0.782±0.02	0.672±0.03	0.497±0.01	0.972±0.00	0.955±0.01	0.594±0.01	0.879±0.03	0.994±0.00	1.000±0.00
Wine	0.963±0.03	1.000±0.00	0.494±0.08	0.978±0.03	0.979±0.02	0.494±0.12	0.909±0.02	0.988±0.02	1.000±0.00
Musk	0.467±0.04	0.999±0.00	0.320±0.00	0.819±0.00	0.813±0.01	0.302±0.00	0.998±0.01	0.851±0.01	1.000±0.00
Glass	0.769±0.11	0.777±0.08	0.446±0.26	0.767±0.09	0.711±0.10	0.447±0.24	0.630±0.10	0.844±0.10	1.000±0.00
Lymphography	0.751±0.05	0.805±0.17	0.534±0.04	0.779±0.03	0.765±0.03	0.535±0.04	0.997±0.01	0.824±0.05	1.000±0.00
Pendigits	0.981±0.02	0.968±0.00	0.499±0.25	0.996±0.01	0.991±0.01	0.499±0.23	0.864±0.02	0.998±0.00	0.999±0.00
Donors	0.774±0.02	0.950±0.01	0.815±0.01	0.644±0.03	0.574±0.02	0.888±0.01	0.887±0.03	0.932±0.00	0.999±0.00
Http	0.993±0.00	0.119±0.00	0.991±0.00	0.171±0.12	0.358±0.24	0.979±0.00	0.996±0.01	0.996±0.00	0.998±0.00
Skin	0.664±0.12	0.872±0.05	0.471±0.14	0.643±0.06	0.569±0.23	0.489±0.36	0.756±0.03	0.996±0.00	0.998±0.00
Shuttle	0.916±0.07	0.961±0.00	0.311±0.31	0.987±0.01	0.956±0.06	0.355±0.31	0.986±0.00	0.994±0.01	0.998±0.00
Letter	0.912±0.03	0.374±0.06	0.500±0.16	0.979±0.02	0.956±0.03	0.501±0.15	0.732±0.03	0.994±0.00	0.997±0.00
Stamps	0.895±0.01	0.849±0.03	0.930±0.01	0.846±0.02	0.559±0.06	0.876±0.01	0.816±0.02	0.990±0.00	0.997±0.00
Ionosphere	0.895±0.03	0.206±0.12	0.801±0.02	0.975±0.01	0.954±0.01	0.743±0.02	0.859±0.02	0.975±0.00	0.994±0.01
Wilt	0.432±0.05	0.770±0.05	0.345±0.13	0.605±0.07	0.698±0.04	0.394±0.15	0.343±0.01	0.987±0.00	0.993±0.00
Breastw	0.986±0.00	0.991±0.00	0.990±0.00	0.978±0.00	0.447±0.13	0.991±0.00	0.978±0.00	0.976±0.00	0.993±0.00
Cover	0.893±0.03	0.906±0.01	0.884±0.01	0.789±0.01	0.521±0.01	0.920±0.01	0.953±0.01	0.979±0.01	0.987±0.00
OptDigits	0.976±0.02	0.968±0.00	0.717±0.09	0.973±0.02	0.972±0.02	0.687±0.10	0.606±0.03	0.980±0.02	0.984±0.01
Thyroid	0.901±0.01	0.952±0.03	0.785±0.00	0.917±0.00	0.880±0.00	0.792±0.00	0.923±0.02	0.969±0.00	0.981±0.00
Amnthyroid	0.806±0.04	0.922±0.02	0.776±0.04	0.808±0.03	0.726±0.05	0.789±0.06	0.667±0.03	0.973±0.00	0.980±0.00
Satimage	0.953±0.04	0.783±0.02	0.692±0.27	0.949±0.04	0.897±0.07	0.628±0.23	0.976±0.02	0.945±0.04	0.978±0.02
WBC	0.957±0.01	0.888±0.15	0.865±0.01	0.947±0.01	0.951±0.01	0.625±0.01	0.975±0.01	0.969±0.01	0.977±0.00
WPBC	0.507±0.08	0.564±0.13	0.523±0.06	0.511±0.03	0.522±0.05	0.481±0.09	0.481±0.01	0.956±0.01	0.972±0.02
InternetAds	0.718±0.05	0.697±0.03	0.676±0.06	0.648±0.04	0.610±0.03	0.677±0.03	0.617±0.02	0.903±0.01	0.968±0.01
Cardio	0.817±0.02	0.916±0.01	0.660±0.01	0.739±0.01	0.774±0.02	0.780±0.01	0.888±0.02	0.860±0.03	0.959±0.00
MagicG	0.726±0.02	0.447±0.00	0.681±0.03	0.810±0.01	0.694±0.03	0.638±0.04	0.707±0.01	0.924±0.01	0.938±0.01
PageBlocks	0.898±0.02	0.206±0.06	0.875±0.02	0.864±0.01	0.725±0.05	0.914±0.01	0.914±0.01	0.907±0.01	0.919±0.04
Vertebral	0.861±0.02	0.863±0.04	0.786±0.02	0.872±0.01	0.882±0.02	0.600±0.03	0.308±0.02	0.909±0.01	0.915±0.01
Pima	0.733±0.01	0.692±0.05	0.656±0.01	0.746±0.01	0.711±0.02	0.595±0.01	0.676±0.01	0.735±0.03	0.914±0.01
Fraud	0.850±0.02	0.459±0.11	0.847±0.03	0.850±0.01	0.506±0.06	0.849±0.04	0.900±0.01	0.902±0.00	0.907±0.01
Mammo	0.879±0.01	0.753±0.02	0.905±0.00	0.876±0.00	0.841±0.01	0.906±0.00	0.852±0.01	0.901±0.01	0.907±0.02
Speech	0.469±0.02	0.524±0.04	0.492±0.01	0.487±0.01	0.498±0.01	0.471±0.01	0.468±0.02	0.584±0.02	0.907±0.01
Waveform	0.716±0.04	0.851±0.00	0.734±0.03	0.735±0.04	0.708±0.03	0.604±0.07	0.773±0.01	0.856±0.01	0.879±0.02
SpamBase	0.610±0.06	0.782±0.02	0.688±0.05	0.501±0.11	0.457±0.14	0.656±0.07	0.550±0.01	0.803±0.02	0.871±0.01
Satellite	0.712±0.03	0.465±0.02	0.633±0.06	0.674±0.03	0.546±0.08	0.583±0.05	0.709±0.02	0.843±0.02	0.868±0.02
SMTP	0.806±0.02	0.447±0.00	0.812±0.01	0.810±0.01	0.731±0.04	0.780±0.03	0.856±0.01	0.828±0.04	0.861±0.03
MNIST	0.807±0.03	0.779±0.03	0.774±0.04	0.841±0.02	0.679±0.02	0.746±0.03	0.851±0.01	0.849±0.04	0.859±0.03
Backdoor	0.738±0.01	0.788±0.01	0.789±0.02	0.729±0.02	0.744±0.01	0.786±0.00	0.788±0.02	0.516±0.05	0.797±0.02
CardioTG	0.673±0.05	0.754±0.02	0.663±0.06	0.489±0.08	0.524±0.07	0.785±0.05	0.571±0.02	0.783±0.02	0.796±0.03
Campaign	0.614±0.04	0.675±0.02	0.683±0.03	0.644±0.04	0.621±0.03	0.670±0.02	0.673±0.03	0.676±0.05	0.698±0.06
Fault	0.577±0.05	0.345±0.08	0.455±0.03	0.526±0.02	0.595±0.02	0.469±0.04	0.633±0.01	0.638±0.03	0.698±0.04
Landsat	0.493±0.06	0.575±0.07	0.422±0.08	0.582±0.05	0.548±0.04	0.368±0.11	0.425±0.03	0.621±0.02	0.645±0.07
Yeast	0.390±0.06	0.596±0.02	0.381±0.08	0.396±0.04	0.457±0.05	0.444±0.03	0.391±0.01	0.613±0.03	0.628±0.04
Aloi	0.514±0.01	0.460±0.00	0.515±0.03	0.492±0.01	0.473±0.02	0.530±0.01	0.522±0.01	0.517±0.02	0.543±0.01
Census	0.403±0.02	0.501±0.00	0.474±0.01	0.476±0.02	0.435±0.04	0.446±0.03	0.503±0.00	0.504±0.00	0.506±0.08
Celeba	0.489±0.01	0.437±0.01	0.451±0.01	0.409±0.03	0.476±0.01	0.457±0.02	0.423±0.10	0.500±0.00	0.501±0.12