An Evidence-Based Post-Hoc Adjustment Framework for Anomaly Detection Under Data Contamination

Sukanya Patra [1] **Souhaib Ben Taieb** [1 2]

Abstract

Unsupervised anomaly detection (AD) methods typically assume clean training data, yet real-world datasets often contain undetected or mislabeled anomalies, leading to significant performance degradation. Existing solutions require access to the training data, model pipeline or model parameters, limiting real-world applicability. To address this challenge, we propose `EPHAD`, a simple yet effective test-time adaptation framework that updates the outputs of AD models trained on contaminated datasets using evidence gathered at inference. Our approach integrates the prior captured by the AD model trained on the contaminated dataset with the output of an auxiliary evidence function at test-time using exponential tilting. This evidence can be derived from foundation models like CLIP, classical methods such as the Latent Outlier Factor or domain-specific knowledge. We validate its effectiveness through extensive experiments across eight image-based AD datasets, twenty-seven tabular datasets, and a real-world industrial dataset. Our code is publicly available[1].

1. Introduction

Anomaly detection (AD) is the basis of many critical applications, including cybersecurity (Xiao et al., 2024; Li et al., 2023a), and industrial maintenance (Schwarz et al., 2025; Patra et al., 2024). By enabling the identification of abnormalities, potential threats, or critical system failures, AD contributes to the robustness and safety of real-world systems. Despite its significance, AD remains a challenging task due to the inherent difficulty in characterising anomalous behaviours and the lack of prior knowledge about anomalous samples (Ruff et al., 2021). Thus, AD is commonly approached as an unsupervised representation learning problem without access to labelled anomalies (Batzner et al., 2024; You et al., 2022).

A standard approach in unsupervised AD involves training a model to learn a "compact" representation of the normal samples from a training dataset under the assumption that the training data is "clean", i.e. contains only normal samples (Ruff et al., 2021). Then, anomalies are identified as deviations from this learned normality. One-class (OC) classification methods (Ruff et al., 2018; Tax and Duin, 2004) learn a decision boundary that encompasses all the normal samples. In contrast, density-based methods (Gudovskiy et al., 2022; Yu et al., 2021) learn the probability distribution of normal samples. Furthermore, memory bank-based approaches (Roth et al., 2022) store the features corresponding to normal samples in a memory bank.

However, real-world datasets are often contaminated with undetected anomalies (Hien et al., 2023; Qiu et al., 2022), which leads to biased AD models that struggle to reliably distinguish between normal and anomalous instances. Thus, we consider a more realistic setting where the training data may be contaminated with anomalies. Existing approaches to handle contamination in the *unsupervised* setting primarily follow two strategies. The first employs an auxiliary OC classifier to filter out suspected anomalies (Yoon et al., 2022; Jiang et al., 2022), while the second modifies the training pipeline to enhance robustness against contamination (Qiu et al., 2022; Eduardo et al., 2020). Although effective, these methods rely on prior knowledge of the proportion of anomalies in the training data, i.e. the contamination ratio, which is typically unknown. Also, such methods are often computationally expensive. In the *semi-supervised* setting, methods leverage additional labelled datasets containing normal and anomalous samples (Hien et al., 2024; Ruff et al., 2020). However, their impact diminishes when the anomalies encountered during training do not resemble real anomalies (Perini et al., 2025).

In this work, we aim to mitigate the possible adverse effects of data contamination on the performance of *unsupervised* AD models (Bouman et al., 2024). Specifically,

[1]Faculty of Science, University of Mons, Belgium [2]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates. Correspondence to: Sukanya Patra <sukanya.patra@umons.ac.be>.

ICML 2025 Workshop on Test-Time Adaptation: Putting Updates to the Test! (PUT), Vancouver, Canada, 2025.

[1]https://github.com/sukanyapatra1997/EPAF

we address the challenging setting where the training data, model pipeline, and model parameters cannot be accessed or modified. This scenario reflects the growing trend of deploying proprietary AD models in real-world applications, where access to internal model components is often restricted. Even when fine-tuning is permitted, it is not only computationally intensive but also unreliable due to the absence of guaranteed clean training data, as anomalies are inherently unknown a priori. This setup aligns with *preparation-agnostic, inference-time* adaptation strategies (Karmanov et al., 2024; Zhang et al., 2023) within the broader class of *test-time adaptation* (TTA) methods (Xiao and Snoek, 2024), which remain largely unexplored in the context of AD. To address this gap, we make three key contributions: (i) We introduce **E**vidence-based **P**ost-**H**oc Adjustment Framework for **A**nomaly **D**etection (EPHAD), a *simple yet effective test-time adaptation* framework for unsupervised AD models trained on contaminated datasets; (ii) EPHAD combines the prior captured by the AD model trained on the contaminated dataset with the output of an auxiliary evidence function at test-time using exponential tilting; (iii) Extensive experiments across eight image-based AD, twenty-seven tabular datasets, and a real-world industrial dataset demonstrate the effectiveness of EPHAD.

2. Background

Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ denote a pair of random variables following a joint probability distribution $\mathrm{P}_{X,Y}$ over the space $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} := \{-1, +1\}$. Here, the label $Y = +1$ and $Y = -1$ correspond to the normal and anomalous classes, respectively. The conditional distribution of normal samples is P_X^+ with PDF $f_X^+(x)$ at $X = x$. Similarly, the conditional distribution of anomalous samples is P_X^-, with PDF $f_X^-(x)$. The training dataset without contamination is $\mathcal{D}_{\text{train}}^+ := \{x_i\}_{i=1}^m$, where $x_i \overset{\text{iid}}{\sim} \mathrm{P}_X^+$. The test dataset is $\mathcal{D}_{\text{test}} := \{(x_i, y_i)\}_{i=1}^n$, where $(x_i, y_i) \overset{\text{iid}}{\sim} \mathrm{P}_{X,Y}$.

Density-based Anomaly Detection. An anomaly can be defined as "an observation that deviates significantly from some concept of normality" (Ruff et al., 2021). This definition comprises two key aspects: the *concept of normality* and the *significant deviation* from it, which can be formalised using a probabilistic framework. The *concept of normality* is defined as the probability distribution of normal samples P_X^+. To formalise this further, we adopt the *concentration assumption* (Steinwart et al., 2005), which posits that although the data space $\mathcal{X}$ is unbounded, the high-density regions of P_X^+ are bounded and concentrated. In contrast, P_X^- is assumed to be non-concentrated (Schölkopf and Smola, 2002), and is often approximated by a uniform distribution over $\mathcal{X}$ (Tax, 2001). Given the PDF $f_X^+(x)$ associated with P_X^+, which we refer to as *inlier density*, a data point $x \in \mathcal{X}$ is identified as an anomaly if it *deviates substantially* from this concept of normality, i.e., if it resides in a low-probability region under P_X^+. However, since $f_X^+(x)$ is typically unknown in practice, we approximate it using a density estimator.

Score-based Anomaly Detection. Density estimation poses significant challenges, particularly in high-dimensional spaces or when data availability is limited, and often incurs substantial computational cost. Fortunately, in the context of anomaly detection, the goal is typically not to recover the exact data likelihood but rather to establish a ranking of data points based on their degree of normality. This motivates an alternative strategy: learning an *anomaly score function* $s^-(x) : \mathcal{X} \to \mathbb{R}$, which directly assigns an anomaly score to a data point $x \in \mathcal{X}$, thereby quantifying its *degree of anomalousness* (Ruff et al., 2021). Consequently, the *inlier score function* is defined as $s^+(x) = -s^-(x)$, capturing the *degree of normality*, where higher values indicate that x is normal. For AD, first, we train a model to learn the anomaly score function $s^-(x)$ using $\mathcal{D}_{\text{train}}^+$. Then, we define the anomaly detector as

$$g_{\lambda_s}(x) = \begin{cases} +1, & \text{if } s^-(x) \leq \lambda_s \\ -1, & \text{if } s^-(x) > \lambda_s, \end{cases} \tag{1}$$

where $\lambda_s \geq 0$ is the threshold (Perini et al., 2023; 2022). The density-based AD method can also be interpreted as a specific case of the score-based AD methods where the anomaly score $s^-(x) = -\phi(f_X^+(x))$ and the inlier score $s^+(x) = \phi(f_X^+(x))$. Here, $\phi(\cdot)$ is an order-preserving transformation typically chosen to be the logarithm.

Data Contamination. For training the AD model, a common assumption is that the training dataset $\mathcal{D}_{\text{train}}^+$ consists solely of i.i.d. samples from the normal data distribution P_X^+, without anomalies. However, this assumption is rarely satisfied in practice, since anomalies are typically unknown *a priori*. As a result, the training dataset is often contaminated with undetected anomalies. A more realistic assumption is that the dataset $\mathcal{D}_{\text{train}}^u := \{x_i\}_{i=1}^m$ comprises of both normal and anomalous samples drawn from a mixture distribution $\mathrm{P}_X^{\mathrm{u}}$ with density $f_X^u(x)$ (Huber and Ronchetti, 2011; Huber, 1992). Let $\epsilon = \mathbb{P}(Y = -1)$ denote the *contamination factor*. The data distribution can be written as

$$\mathrm{P}_X^{\mathrm{u}} = \epsilon\, \mathrm{P}_X^- + (1 - \epsilon)\, \mathrm{P}_X^+. \tag{2}$$

As ϵ increases, the model trained on $\mathcal{D}_{\text{train}}^u$ becomes increasingly biased (Qiu et al., 2022; Yoon et al., 2022), and tends to misclassify anomalous samples as normal.

3. EPHAD: An Evidence-based Post-Hoc Adjustment Framework

We consider the realistic scenario where an AD model has already been trained on a contaminated dataset $\mathcal{D}_{\text{train}}^u$. The

goal is to adapt the model's predictions at inference time to reduce the impact of contamination. To this end, we introduce our **E**vidence-based **P**ost-**H**oc Adjustment Framework for **A**nomaly **D**etection (EPHAD), a simple yet effective method to mitigate the adverse effects of training data contamination using an auxiliary evidence function at test time. Here, the *auxiliary evidence function* $T(x) : \mathcal{X} \to \mathbb{R}$ quantifies how likely a sample x is to be normal, based on a domain-specific *concept of normality*. The function $T(x)$ should assign higher values to samples deemed more likely to be normal and can incorporate domain-specific knowledge. As such, EPHAD aligns with *preparation-agnostic* TTA methods (Xiao and Snoek, 2024).

Given an AD model trained on the contaminated dataset $\mathcal{D}^u_{\text{train}}$, we denote the anomaly score for a data point x as $s^-_u(x)$. Then, we can compute the inlier score as $s^+_u(x) = -s^-_u(x)$. Considering $s^+_u(x)$ as prior and an auxiliary evidence function $T(x)$, EPHAD computes the revised score $s^+_c(x)$ using exponential tilting. It is a technique used to adjust a PDF by "tilting" it toward a specific outcome. Recall that the inlier score is an order-preserving transformation of the inlier PDF, i.e., $s^+(x) = \phi(f^+_X(x))$ where ϕ is a transformation function. Thus, given a score-based AD model that learns $s^-_u(x)$, tilting increases the relative scores of the normal samples over the anomalous samples, steering the model toward an outcome supported by the evidence function. We first exponentiate the inlier score $s^+_u(x) = -s^-_u(x)$ to ensure non-negativity. Then, we rescale $T(x)$ with a temperature parameter β and exponentiate it. Finally, the revised inlier score is computed as:

$$s^+_c(x) = \exp(s^+_u(x)) \exp(T(x)/\beta). \tag{3}$$

Consequently, we can rewrite (1) using (3) as:

$$g_{\lambda_s}(x) = \begin{cases} +1, & \text{if } s^+_c(x) \geq \lambda_s, \\ -1, & \text{otherwise.} \end{cases} \tag{4}$$

In doing so, EPHAD enables post-hoc adjustment of AD models trained on contaminated datasets without requiring access to the training procedure. Thus, EPHAD offers a simple, yet effective solution for real-world AD systems.

Application to Density-based Anomaly Detection. Contrary to score-based AD models, density-based models directly learn to approximate the inlier PDF $f^+_X(x)$. Considering the model is trained on the contaminated dataset $\mathcal{D}^u_{\text{train}}$, we denote the learned inlier PDF as $f^u_X(x)$ and compute the revised PDF $f^c_n(x)$ using exponential tilting:

$$f^c_X(x) = \frac{f^u_X(x) \exp(T(x)/\beta)}{Z^\beta_X}. \tag{5}$$

Proposition 3.1 provides a condition under which the revised PDF $f^c_X(x)$ is closely aligned with the true PDF $f^+_X(x)$ than the contaminated PDF $f^u_X(x)$, in terms of Kullback–Leibler (KL) divergence.

Proposition 3.1. *Let f^+_X, f^u_X, and f^c_X be PDFs over same domain $\mathcal{X}$. Then the KL divergence between f^+_X and f^c_X is strictly less than the divergence between f^+_X and f^u_X iff*

$$\mathbb{E}_{x\sim \mathrm{P}^+_X}\left[\log \frac{\exp(T(x)/\beta)}{Z^\beta_X}\right] > 0. \tag{6}$$

The proof is provided in Appendix A. Thus, we expect $f^c_X(x)$ will result in an improved AD performance compared to using $f^u_X(x)$, assuming a well-chosen threshold.

Connection with TTA of generative models using Reinforcement Learning (RL) with KL penalties. We highlight the conceptual connection between the application of EPHAD to density-based AD methods and a well-established TTA approach used in generative models (Mudgal et al., 2024; Li et al., 2024). A generative model π_θ is treated as an RL policy and is refined using a reward function r that encodes the desired evidence or alignment criteria. The model is initially set to a prior π_0 and fine-tuned using a KL-regularised RL objective (Korbak et al., 2022):

$$J_{\text{KL-RL}}(\pi_\theta) = \mathbb{E}_{x\sim\pi_0}\left[r(x)\right] - \beta\, \mathrm{KL}(\pi_\theta \| \pi_0), \tag{7}$$

where β is a temperature hyperparameter. It can be shown that the optimal solution to this objective is given by:

$$\pi^*_\theta(x) = \frac{\pi_0(x) \exp\left(r(x)/\beta\right)}{Z}, \tag{8}$$

where Z is a normalization constant ensuring that π^*_θ is a valid PDF. Interestingly, this optimal solution is equivalent to (5) when we set $\pi_0(x) = f^u_X(x)$, $r(x) = T(x)$, and $\pi^*_\theta(x) = f^c_X(x)$. This equivalence offers a valuable interpretation of EPHAD as a form of inference-time alignment: it shifts the prior density $f^u_X(x)$ toward regions favoured by the evidence function $T(x)$ while maintaining consistency with the prior through KL regularisation.

4. Experiments

We evaluate the effectiveness of EPHAD for unsupervised AD across a range of datasets, including image datasets (Section 4.1), tabular datasets (Appendix C.2), and an industrial use case (Appendix C.3). The evidence functions used are computed in an unsupervised manner without utilising ground-truth labels in the test set $\mathcal{D}_{\text{test}}$, mitigating the risk of overfitting. Unless stated otherwise, we use $\epsilon = 0.1$ and $\beta = 0.5$. An ablation study on different values of ϵ and β is presented in Appendix C.5 and on varying number of test samples n in Appendix C.6. Following prior work (Roth et al., 2022; Gudovskiy et al., 2022), we provide AUROC averaged across all categories for each dataset.

Table 1. Performance on AD benchmarking datasets with 10% contamination ratio. Style: AUROC % (± SE). Best in **bold**.

Method	Non-overlap					Overlap		
	MNIST	FMNIST	CIFAR10	SVHN	RealIAD	MVTec	MPDD	ViSA
CLIP	71.15	95.63	98.63	58.46	65.74	86.34	60.02	74.47
CFLOW	77.24 (± 1.01)	72.87 (± 0.48)	65.47 (± 0.02)	55.09 (± 0.09)	**76.42** (± 0.47)	87.58 (± 0.77)	**66.69** (± 2.06)	75.71 (± 1.28)
+ EPHAD	**78.40** (± 0.81)	**92.97** (± 0.19)	**97.38** (± 0.01)	**55.82** (± 0.06)	71.58 (± 0.17)	**87.98** (± 0.12)	65.22 (± 0.93)	**78.53** (± 0.27)
DRÆM	71.44 (± 0.29)	76.53 (± 0.18)	63.41 (± 0.26)	51.55 (± 0.07)	67.46 (± 0.21)	70.55 (± 1.97)	62.32 (± 1.96)	69.61 (± 1.57)
+ EPHAD	**73.51** (± 0.39)	**92.46** (± 0.25)	**97.17** (± 0.02)	**54.18** (± 0.07)	**69.89** (± 0.23)	**87.13** (± 0.39)	**67.02** (± 0.29)	**76.89** (± 0.99)
FastFlow	82.65 (± 0.43)	83.66 (± 0.06)	62.94 (± 0.37)	54.02 (± 0.11)	**82.03** (± 0.08)	84.24 (± 1.07)	**71.94** (± 0.87)	77.83 (± 0.22)
+ EPHAD	**83.20** (± 0.43)	**93.49** (± 0.07)	**97.34** (± 0.02)	**55.07** (± 0.07)	77.22 (± 0.08)	**87.68** (± 0.5)	66.84 (± 0.34)	**80.29**(± 0.07)
PaDiM	**87.50** (± 0.23)	86.84 (± 0.06)	62.53 (± 0.4)	55.49 (± 0.28)	**80.39** (± 0.35)	77.85 (± 0.43)	36.58 (± 2.58)	73.07 (± 0.27)
+ EPHAD	87.45 (± 0.22)	**94.66** (± 0.03)	**97.10** (± 0.03)	**56.94** (± 0.22)	75.94 (± 0.25)	**86.58** (± 0.38)	**55.48** (± 0.72)	**77.73** (± 0.27)
PatchCore	86.33 (± 0.09)	78.97 (± 0.06)	75.69 (± 0.09)	**69.64** (± 0.04)	**70.08** (± 0.07)	70.51 (± 0.7)	53.58 (± 0.54)	27.2 (± 0.31)
+ EPHAD	**86.36** (± 0.1)	**94.73** (± 0.01)	**97.74** (± 0.01)	61.31 (± 0.0)	69.76 (± 0.2)	**86.45** (± 0.14)	**60.58** (± 1.12)	**62.94** (± 0.41)
RD	77.33 (± 0.09)	84.11 (± 0.72)	66.29 (± 0.31)	55.54 (± 0.58)	**89.13** (± 0.18)	80.08 (± 1.32)	**75.08** (± 1.75)	**86.33** (± 0.46)
+ EPHAD	**78.19** (± 0.28)	**95.77** (± 0.03)	**98.40** (± 0.0)	**57.38** (± 0.14)	69.35 (± 0.26)	**85.82** (± 0.31)	62.62 (± 0.27)	77.76 (± 0.19)
ULSAD	**90.83** (± 0.08)	88.64 (± 0.13)	72.45 (± 0.18)	**64.27** (± 0.22)	**89.06** (± 0.01)	**91.93** (± 0.15)	77.67 (± 0.42)	**86.58** (± 0.13)
+ EPHAD	90.41 (± 0.06)	**95.03** (± 0.07)	**97.90** (± 0.02)	58.17 (± 0.18)	80.58 (± 0.06)	91.31 (± 0.06)	**72.79** (± 1.05)	85.82 (± 0.1)

4.1. Experiments on Image Datasets

Benchmark Datasets. We evaluate on sensory datasets and semantic anomaly detection (AD) using eight well-established benchmarks: MVTecAD (Bergmann et al., 2019), MPDD (Jezek et al., 2021), ViSA (Zou et al., 2022), RealIAD (Wang et al., 2024), CIFAR-10, Fashion-MNIST, MNIST, and SVHN. For MVTecAD, ViSA, and MPDD, we adopt the "overlap" setting, introducing $\epsilon\%$ contamination into the training set by randomly selecting anomalous samples from the test set while retaining them in the test set (Jiang et al., 2022). For the remaining datasets, we follow the "non-overlapping" setting, excluding anomalous samples used for contamination simulation from the test set. Additional details are provided in Appendix B.1.

Baseline AD Methods. We evaluate the performance of several state-of-the-art unsupervised anomaly detection (AD) methods, including PatchCore (Roth et al., 2022), PaDim (Defard et al., 2021), CFLOW (Gudovskiy et al., 2022), FastFLOW (Yu et al., 2021), DRÆM (Zavrtanik et al., 2021), Reverse Distillation (RD) (Deng and Li, 2022), and ULSAD (Patra and Ben Taieb, 2024), both with and without the integration of EPHAD.

Evidence Function. For the experiments, we use Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) as the evidence function following (Jeong et al., 2023). Additional details are in Appendix B.2.1.

Results. In Table 1, we observe that while zero-shot AD using CLIP performs well on real-world image datasets such as CIFAR10 and FMNIST, its effectiveness declines on domain-specific datasets like MVTec, MPDD, and ViSA, where existing AD methods, such as ULSAD, achieve superior performance. However, when these AD methods are used within the EPHAD framework with CLIP as an evidence function in a post-hoc manner, their performance improves in most cases. Notably, even when CLIP-based AD alone does not achieve the best results, as seen in SVHN, incorporating it within EPHAD still leads to significant improvements. For instance, CFLOW, PaDiM, and RD exhibit enhanced performance after using EPHAD, surpassing both CLIP and the standalone AD methods. This highlights the effectiveness of EPHAD in refining anomaly scores for better AD performance. In some cases, such as ULSAD on MNIST, we observe a decline in performance when integrating EPHAD compared to the standalone AD method. This typically occurs when the AD method substantially outperforms the evidence function. In such scenarios, overly relying on the evidence can diminish overall performance. To mitigate this effect, careful tuning of β enables the framework to adapt effectively to different datasets, AD methods, and evidence functions. A detailed analysis of the impact of varying β values is presented in Appendix C.5. Additionally, an unsupervised approach for selecting the optimal β value is presented in Appendix C.7.

5. Conclusion

Unsupervised AD methods typically assume anomaly-free training data, yet real-world datasets often contain undetected or mislabeled anomalies, leading to performance degradation. Existing approaches to address contamination often require access to model parameters, training data, or the training pipeline, limiting their practicality in real-world deployments. In this work, we introduce EPHAD, a simple, post-hoc adjustment framework that refines the outputs of any AD method trained on contaminated data by incorporating evidence collected at inference time. Experiments demonstrate the effectiveness of EPHAD across diverse scenarios. Further exploring the interplay between datasets, AD methods, and evidence functions remains an open direction for future work. Additionally, ablation studies analyse the impact of hyperparameters and varying contamination levels, highlighting the robustness of EPHAD.

References

Batzner, K., Heckler, L., and König, R. (2024). Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE.

Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. (2019). Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592.

Bouman, R., Bukhsh, Z., and Heskes, T. (2024). Unsupervised anomaly detection algorithms on real-world data: how many do we need? *Journal of Machine Learning Research*, 25(105):1–34.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, page 93–104, New York, NY, USA. Association for Computing Machinery.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Defard, T., Setkov, A., Loesch, A., and Audigier, R. (2021). *PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization*, page 475–489. Springer International Publishing.

Deng, H. and Li, X. (2022). Anomaly detection via reverse distillation from one-class embedding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9727–9736.

Eduardo, S., Nazábal, A., Williams, C. K., and Sutton, C. (2020). Robust variational autoencoders for outlier detection and repair of mixed-type data. In *International Conference on Artificial Intelligence and Statistics*, pages 4056–4066. PMLR.

Gudovskiy, D., Ishizaka, S., and Kozuka, K. (2022). Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE.

Han, S., Hu, X., Huang, H., Jiang, M., and Zhao, Y. (2022). Adbench: Anomaly detection benchmark. In *Neural Information Processing Systems (NeurIPS)*.

Hendrycks, D., Mazeika, M., and Dietterich, T. (2019). Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*.

Hien, L. T. K., Patra, S., and Ben Taieb, S. (2024). Anomaly detection with semi-supervised classification based on risk estimators. *Transactions on Machine Learning Research*.

Hien, L. T. K., Patra, S., and Taieb, S. B. (2023). Anomaly detection with semi-supervised classification based on risk estimators. In *Advances in Neural Information Processing Systems*. Submitted.

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer.

Huber, P. J. and Ronchetti, E. M. (2011). *Robust statistics*. John Wiley & Sons.

Jeong, J., Zou, Y., Kim, T., Zhang, D., Ravichandran, A., and Dabeer, O. (2023). Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19606–19616.

Jezek, S., Jonak, M., Burget, R., Dvorak, P., and Skotak, M. (2021). Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In *2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 66–71.

Jiang, X., Liu, J., Wang, J., Nie, Q., Wu, K., Liu, Y., Wang, C., and Zheng, F. (2022). Softpatch: Unsupervised anomaly detection with noisy data. *Advances in Neural Information Processing Systems*, 35:15433–15445.

Karmanov, A., Guan, D., Lu, S., El Saddik, A., and Xing, E. (2024). Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14162–14171.

Kim, J. and Scott, C. D. (2012). Robust kernel density estimation. *The Journal of Machine Learning Research*, 13(1):2529–2565.

Korbak, T., Perez, E., and Buckley, C. (2022). RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1083–1091, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lee, S., Lee, S., and Song, B. C. (2022). Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access*, 10:78446–78454.

Li, R., Li, Q., Zhang, Y., Zhao, D., Jiang, Y., and Yang, Y. (2023a). Interpreting unsupervised anomaly detection in security via rule extraction. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Li, X., Zhao, Y., Wang, C., Scalia, G., Eraslan, G., Nair, S., Biancalani, T., Ji, S., Regev, A., Levine, S., and Uehara, M. (2024). Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding.

Li, Z., Zhao, Y., Botta, N., Ionescu, C., and Hu, X. (2020). COPOD: copula-based outlier detection. In *IEEE International Conference on Data Mining (ICDM)*. IEEE.

Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., and Chen, G. H. (2023b). Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, 6(1).

Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F., and Jin, Y. (2024). Deep industrial image anomaly detection: A survey. *Machine Intelligence Research*, 21(1):104–135.

Mudgal, S., Lee, J., Ganapathy, H., Li, Y., Wang, T., Huang, Y., Chen, Z., Cheng, H.-T., Collins, M., Strohman, T., Chen, J., Beutel, A., and Beirami, A. (2024). Controlled decoding from language models. In *Forty-first International Conference on Machine Learning*.

Patra, S. and Ben Taieb, S. (2024). Revisiting deep feature reconstruction for logical and structural industrial anomaly detection. *Transactions on Machine Learning Research*.

Patra, S., Sournac, N., and Ben Taieb, S. (2024). Detecting abnormal operations in concentrated solar power plants from irregular sequences of thermal images. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5578–5589, New York, NY, USA. Association for Computing Machinery.

Perini, L., Bürkner, P.-C., and Klami, A. (2023). Estimating the contamination factor's distribution in unsupervised anomaly detection. In *International Conference on Machine Learning*, pages 27668–27679. PMLR.

Perini, L., Rudolph, M., Schmedding, S., and Qiu, C. (2025). Uncertainty-aware evaluation of auxiliary anomalies with the expected anomaly posterior. *Transactions on Machine Learning Research*.

Perini, L., Vercruyssen, V., and Davis, J. (2020). Quantifying the confidence of anomaly detectors in their example-wise predictions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 227–243. Springer.

Perini, L., Vercruyssen, V., and Davis, J. (2022). Transferring the contamination factor between anomaly detection domains by shape similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4128–4136.

Press, O., Shwartz-Ziv, R., Lecun, Y., and Bethge, M. (2024). The entropy enigma: Success and failure of entropy minimization. In *International Conference on Machine Learning*, pages 41064–41085. PMLR.

Qiu, C., Li, A., Kloft, M., Rudolph, M., and Mandt, S. (2022). Latent outlier exposure for anomaly detection with contaminated data. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18153–18167. PMLR.

Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., and Rudolph, M. (2021). Neural transformation learning for deep anomaly detection beyond images. In *International conference on machine learning*, pages 8703–8714. PMLR.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Roth, K., Pemula, L., Zepeda, J., Scholkopf, B., Brox, T., and Gehler, P. (2022). Towards total recall in industrial anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 14298–14308. IEEE.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018). Deep one-class classification. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR.

Ruff, L., Vandermeulen, R. A., Görnitz, N., Binder, A., Müller, E., Müller, K.-R., and Kloft, M. (2020). Deep

semi-supervised anomaly detection. In *International Conference on Learning Representations*.

Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schwarz, A., Rahal, J. R., Sahelices, B., Barroso-García, V., Weis, R., and Duque Antón, S. (2025). Data augmentation in predictive maintenance applicable to hydrogen combustion engines: a review. *Artificial Intelligence Review*, 58(1):1–24.

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

Steinwart, I., Hush, D., and Scovel, C. (2005). A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6(2).

Tax, D. M. (2001). *One-class classification*. PhD thesis, Technische Universiteit Delft.

Tax, D. M. and Duin, R. P. (1999). Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199.

Tax, D. M. and Duin, R. P. (2004). Support Vector Data Description. *Machine Learning*, 54(1):45–66.

Wang, C., Zhu, W., Gao, B.-B., Gan, Z., Zhang, J., Gu, Z., Qian, S., Chen, M., and Ma, L. (2024). Real-iad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22883–22892.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2021). Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.

Wang, S., Zeng, Y., Liu, X., Zhu, E., Yin, J., Xu, C., and Kloft, M. (2019). Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. *Advances in neural information processing systems*, 32.

Xiao, J., Xu, Z., Zou, Q., Li, Q., Zhao, D., Fang, D., Li, R., Tang, W., Li, K., Zuo, X., et al. (2024). Make your home safe: Time-aware unsupervised user behavior anomaly detection in smart homes via loss-guided mask. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3551–3562.

Xiao, Z. and Snoek, C. G. (2024). Beyond model adaptation at test time: A survey. *arXiv preprint arXiv:2411.03687*.

Yoon, J., Sohn, K., Li, C.-L., Arik, S. O., Lee, C.-Y., and Pfister, T. (2022). Self-supervise, refine, repeat: Improving unsupervised anomaly detection. *Transactions on Machine Learning Research*.

You, Z., Cui, L., Shen, Y., Yang, K., Lu, X., Zheng, Y., and Le, X. (2022). A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems*, 35:4571–4584.

Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R., and Wu, L. (2021). Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows.

Zavrtanik, V., Kristan, M., and Skočaj, D. (2021). DrÆm – a discriminatively trained reconstruction embedding for surface anomaly detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8310–8319.

Zhang, J., Suganuma, M., and Okatani, T. (2024). Contextual affinity distillation for image anomaly detection. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, page 148–157. IEEE.

Zhang, Y., Wang, X., Jin, K., Yuan, K., Zhang, Z., Wang, L., Jin, R., and Tan, T. (2023). Adanpc: Exploring nonparametric classifier for test-time adaptation. In *International conference on machine learning*, pages 41647–41676. PMLR.

Zhou, Q., Pang, G., Tian, Y., He, S., and Chen, J. (2024). AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. In *The Twelfth International Conference on Learning Representations*.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*.

Zou, Y., Jeong, J., Pemula, L., Zhang, D., and Dabeer, O. (2022). Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer.

A. Proof of Proposition 3.1

Proof. From (2), we have

$$f_X^u(x) = \epsilon f_X^-(x) + (1-\epsilon) f_X^+(x).$$

Additionally, from (5), we have

$$f_X^c(x) = \frac{f_X^u(x)\exp(T(x)/\beta)}{Z_X^\beta}$$

Then,

$$\begin{aligned}
\mathrm{KL}(f_X^+ \| f_X^c) &= \mathbb{E}_{x\sim \mathrm{P}_X^+}\left[\log \frac{f_X^+(x)}{f_X^c(x)}\right] && (9)\\
&= \mathbb{E}_{x\sim \mathrm{P}_X^+}\left[\log f_X^+(x) - \log f_X^c(x)\right] && (10)\\
&= \mathbb{E}_{x\sim \mathrm{P}_X^+}\left[\log f_X^+(x) - \log \frac{f_X^u(x)\exp(T(x)/\beta)}{Z_X^\beta}\right] && (11)\\
&= \mathbb{E}_{x\sim \mathrm{P}_X^+}\left[\log f_X^+(x) - \log f_X^u(x)\exp(T(x)/\beta) + \log Z_X^\beta\right] && (12)\\
&= \mathbb{E}_{x\sim \mathrm{P}_X^+}\left[\log f_X^+(x) - \log f_X^u(x) - \log\exp(T(x)/\beta) + \log Z_X^\beta\right] && (13)\\
&= \mathrm{KL}(f_X^+ \| f_X^u) - \mathbb{E}_{x\sim \mathrm{P}_X^+}\left[\log\exp(T(x)/\beta) - \log Z_X^\beta\right] && (14)\\
&= \mathrm{KL}(f_X^+ \| f_X^u) - \mathbb{E}_{x\sim \mathrm{P}_X^+}\left[\log \frac{\exp(T(x)/\beta)}{Z_X^\beta}\right] && (15)
\end{aligned}$$

We are interested in increasing the alignment between f_X^+ and f_X^c. As KL-divergence is always non-negative if the expectation term is positive, it results in $\mathrm{KL}(f_X^+ \| f_X^c) \leq \mathrm{KL}(f_X^+ \| f_X^u)$. Thus, we want the following condition to hold:

$$\mathbb{E}_{x\sim \mathrm{P}_X^+}\left[\log \frac{\exp(T(x)/\beta)}{Z_X^\beta}\right] \geq 0. \tag{16}$$

□

B. Additional Implementation Details

B.1. Benchmark Datasets

For sensory AD in industrial settings, we use three widely recognised benchmark datasets. MVTecAD (Bergmann et al., 2019) comprises images from 15 categories (10 objects and 5 textures) with 3629 normal training images and 1258 anomalous and 467 normal test images, each containing pixel-level annotations of defects. MPDD (Jezek et al., 2021) targets metal part defects under varying conditions, offering 888 training images and test datasets consisting of 176 normal and 282 anomalous images across 6 metal part categories. ViSA (Zou et al., 2022) provides 10821 high-resolution images (9621 normal and 1200 anomalous) spanning 12 categories, capturing a range of anomalies such as scratches, cracks, missing parts, and misplacements. Each defect type is represented by 15–20 images, and some images feature multiple defects. RealIAD (Wang et al., 2024) is a large-scale industrial AD dataset comprising $\sim 150k$ images across 30 categories and having various types of defects such as scratches, dirt and missing parts. For experiments with RealIAD, we use the training split with 10% contamination and the test split provided by the authors. For the semantic datasets, using the one-vs-rest protocol, we create k AD tasks for each dataset, where k is the number of classes. In each task, one class is designated as normal, while the remaining classes are treated as anomalous. Across both sensory and semantic AD, the training datasets consist of a mixture of normal samples and a fraction ϵ of anomalous samples, reflecting realistic contamination scenarios.

B.2. Computing Evidence Functions

EPHAD relies on an evidence function $T(x)$, computed during inference, to refine anomaly scores by assigning higher values to samples from P_X^+ than those from P_X^-. In this section, we introduce domain-agnostic evidence functions applicable to image (Section B.2.1) and tabular datasets (Section B.2.2). While these functions are commonly used as standalone methods for anomaly detection, their role as evidence functions is novel and complementary to our framework. By operating in a transductive setting, they refine the outputs of an AD model initially trained in an inductive setting. Moreover, as shown in Section 4, using these evidence functions solely as anomaly scores does not always yield strong AD performance. However, when integrated into EPHAD, they significantly enhance the performance of a pre-trained model. Finally, the choice of an $T(x)$ is not restricted to AD methods and can be adapted to incorporate domain-specific knowledge for improved effectiveness.

B.2.1. Evidence for Image Datasets

For the evidence function in image-based AD, we propose using Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021), a robust large-scale framework that learns joint vision-language representations from web-collected image-text pairs. While CLIP has been explored in prior work as a zero-shot AD method (Jeong et al., 2023; Zhou et al., 2024), its performance varies across different datasets. Although CLIP excels in detecting anomalies in real-world image datasets such as CIFAR10, it faces significant challenges when applied to domain-specific datasets, particularly those used for industrial inspection, like MVTec. This limitation stems from the lack of domain-specific knowledge in CLIP's pre-training. In this section, we describe how CLIP is integrated into EPHAD as an evidence function $T(x)$, leveraging its strengths while mitigating its limitations in specialized domains.

Given a dataset $\mathcal{D} := \{(x_j, t_j)\}_{j=1}^n$, CLIP trains an image encoder e_i and a text encoder e_t using contrastive learning (Chen et al., 2020), maximizing the cosine similarity between $e_i(x_j)$ and $e_t(t_j)$ for all $(x_j, t_j) \in \mathcal{D}$. For an input image x, CLIP performs zero-shot classification (Radford et al., 2021) by computing a k-way categorical distribution over a set of candidate class texts $\mathcal{C} = \{c_1, \ldots, c_k\}$

$$\mathbb{P}(C = c_j \mid x; c \in \mathcal{C}) := \frac{\exp\left(\langle e_i(x), e_t(c_j)\rangle/\gamma\right)}{\sum_{s\in\mathcal{C}} \exp\left(\langle e_i(x), e_t(s)\rangle/\gamma\right)},$$

where $C \in \mathcal{C}$ is a random variable, $\langle\cdot,\cdot\rangle$ denotes the cosine similarity, and γ is a temperature parameter that controls the sharpness of the distribution. Pairing class labels $c \in \mathcal{C}$ with prompt templates (e.g., `a photo of a` $[c]$) improves classification accuracy, and aggregating embeddings from multiple prompt variations (e.g., `a cropped photo of a` $[c]$) further enhances performance.

Building on Jeong et al. (2023), we use CLIP as evidence function $T(x)$ in EPHAD. We start by defining two lists of textual prompt templates, $\mathcal{T}_N = \{n_1, \cdots, n_k\}$ and $\mathcal{T}_A = \{a_1, \cdots, a_k\}$, corresponding to normal and anomalous classes, respectively. The list of prompts is provided in Table 2. These templates are dataset-dependent, reflecting subjectivity (e.g., "missing wire" as anomalous for cables). For each label, we generate two lists of prompts for normal and anomalous cases using $\mathcal{T}_N$ and $\mathcal{T}_A$ and compute the mean of text embeddings t_N and t_A. Finally, given an input image x, the evidence $T(x)$ during inference is computed as:

$$T(x) := \frac{\exp\left(\langle e_i(x), t_A\rangle/\gamma\right)}{\exp\left(\langle e_i(x), t_N\rangle/\gamma\right) + \exp\left(\langle e_i(x), t_A\rangle/\gamma\right)}.$$

On the use of CLIP for computing $T(x)$. CLIP, being a pre-trained model, raises the possibility of overlap between its pre-training data and the samples encountered during testing. Such overlap could challenge the assumption that test-time statistics are computed solely on test data during inference, independent of the training datasets. However, Radford et al. (2021) systematically analyzed the data overlap in CLIP's pre-training process and demonstrated that removing all overlapping data results in only a negligible drop in performance. This finding underscores that CLIP's performance primarily reflects its ability to generalize rather than leveraging specific training data. Consequently, our experiments with CLIP focus on evaluating its generalization capabilities, making it applicable to our proposed framework without access to the training dataset.

Table 2. Prompts for CLIP where "c" denotes the category.

Semantic AD		Sensory AD	
Normal	Anomalous	Normal	Anomalous
"c"	damaged "c"	a photo of the number "c"	a photo of something
flawless "c"	"c" with flaw		
perfect "c"	"c" with defect		
unblemished "c"	"c" with damage		
"c" without flaw			
"c" without defect			
"c" without damage			

B.2.2. Evidence for Tabular Datasets

For tabular datasets, we use the output of two classical unsupervised AD methods as evidence functions $T(x)$, namely, Local Outlier Factor (LOF) (Breunig et al., 2000) and Isolation Forest (IForest) (Liu et al., 2012).

Local Outlier Factor. To detect anomalies, the local density of a point is compared to that of its k-nearest neighbours. Specifically, given a dataset $\mathcal{D} := \{x_j\}_{j=1}^n$, the k-distance of a point x, denoted as k-distance(x), is defined as the distance from x to its k-th nearest neighbor.

Based on this, the k-distance neighborhood of x, denoted as $\mathcal{N}_k(x)$, consists of all points whose distance from x is at most k-distance(x). Additionally, the reachability distance of x from a neighbor x_i is computed as reach-dist$_k(x, x_i) = \max\{k\text{-distance}(x), d(x, x_i)\}$, where $d(x, x_i)$ represents the distance between x and x_i.

Then, local reachability density (LRD) of x is computed as

$$\text{LRD}_k(x) = \left[\frac{\sum_{x_i \in N_k(x)} \text{reach-dist}_k(x, x_i)}{|N_k(x)|}\right]^{-1}.$$

Finally, the LOF-based evidence is computed as

$$T(x) = \frac{\sum_{x_i \in N_k(x)} \frac{\text{LRD}_k(x_i)}{\text{LRD}_k(x)}}{|N_k(x)|}.$$

Isolation Forest. Anomalies are identified by recursively partitioning the data using a tree-based method, where features and split values are selected randomly. IForest operates under the assumption that anomalies are more susceptible to isolation due to their sparsity and distinctiveness in the feature space. Given $\mathcal{D}$, IForest constructs multiple isolation trees (ITrees), where each data point x is assigned a depth representing the number of splits required to isolate it, referred to as the *path length*. Specifically, the evidence function $T(x)$ is computed as:

$$T(x) = 2^{-\frac{E(h(x))}{c(n)}},$$

where $h(x)$ is the path length of x, i.e., the number of edges traversed from the root node to the leaf node where x is isolated in an ITree. $\mathbb{E}(h(x))$ is the expected path length, i.e., the average path length across multiple ITrees, and $c(n)$ is the average path length of an unsuccessful search.

B.3. Experimental Setup

For training the base AD methods, we use open-source Anomalib and ADBench libraries for experiments with image and tabular datasets, respectively. Our decision to rely on these public libraries was intentional, ensuring transparency and facilitating unbiased comparisons. For the training of each base AD model, we used a single NVIDIA A100 GPU. Then, we run inference using `EPHAD` on CPU.

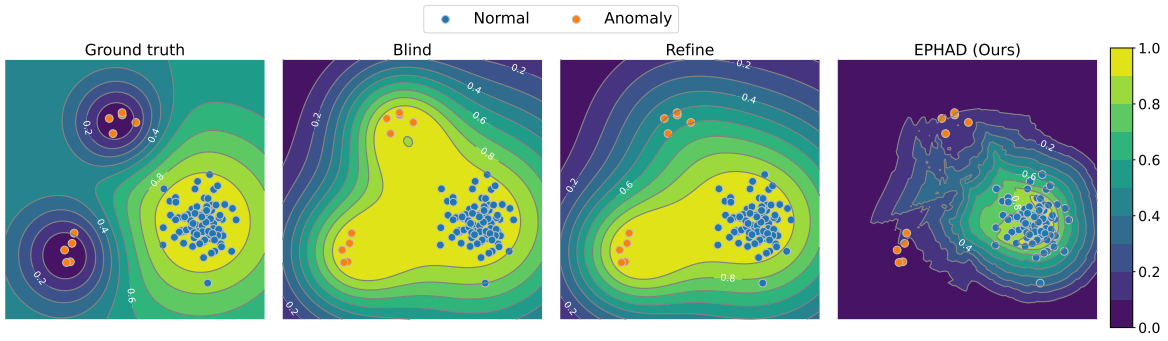

Figure 1. DeepSVDD trained on 2D synthetic contaminated training data with different configurations: (I) Supervised AD with ground truth labels for reference, (ii)"Blind" considering all samples as normal, (iii) "Refine" filtering out a fraction of the anomalies, and (iv) EPHAD updating the "Blind" anomaly detector using evidence computed on test samples during inference.

C. Extended Results

C.1. Experiments on Synthetic Example

We evaluate EPHAD with a toy dataset inspired by Qiu et al. (2022). The dataset is generated using a two-dimensional mixture model comprising three Gaussian components: $c_1 := \mathcal{N}(\mu_1, \Sigma_1), c_2 := \mathcal{N}(\mu_2, \Sigma_2), c_3 := \mathcal{N}(\mu_3, \Sigma_3)$. Here, each component follows a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ with mean μ and covariance Σ. Normal samples are drawn from $f_X^+(x) = c_1$, with $\mu_1 = [1, 1]^T$ and $\Sigma_1 = 0.07\ \mathbf{I}_2$. Anomalous samples are drawn from a mixture distribution $p_X^-(x) := 0.5c_2 + 0.5c_3$ where $\mu_2 = [-0.25, 2.5]^T, \mu_3 = [-1, 0.5]^T$ and $\Sigma_2 = \Sigma_3 = 0.03\ \mathbf{I}_2$. Using this setting, we create a contaminated dataset consisting 100 data points. We compare the baseline DeepSVDD (Ruff et al., 2018) across three configurations as illustrated in Figure 1: (i) "Blind", (ii) "Refine", and (iii) with EPHAD. "Blind" treats all samples as normal while "Refine" iteratively filters out suspected anomalies during training.

For the experiments, we use DeepSVDD with a one-layer radial basis function (RBF) network. The hidden layer comprises three neurons, with their centres fixed at the mean of each Gaussian component, while the scales are optimised during training. The RBF network outputs a 1D scalar obtained as a linear combination of the outputs from the hidden layer. The centre is initialised randomly and made trainable, with an added bias term in the final layer. Although these modifications are not recommended by Ruff et al. (2018) to avoid collapse to a trivial solution, Qiu et al. (2022) observed that these changes enhance model flexibility and convergence. Following this, we train DeepSVDD using the Adam optimiser with a learning rate of 0.01, 200 epochs, and a mini-batch size of 25. As an evidence function in EPHAD, LOF (Breunig et al., 2000) is computed on test samples during inference. The results in Figure 1 demonstrate that the "Blind" configuration mistakenly considers all anomalies as normal. The "Refine" configuration improves performance by filtering out a subset of anomalies. Finally, EPHAD establishes a clearer boundary around normal samples.

C.2. Experiments on Tabular Datasets

Benchmark Datasets. We evaluate our proposed framework on 27 classical benchmark datasets from ADBench (Han et al., 2022). The classical datasets include datasets from different domains such as healthcare (e.g., antithyroid, cardio), astronautics (e.g., Landsat, satellite), and finance (fraud). Following Qiu et al. (2022), we preprocess, split the dataset in the train and test set and simulate contamination using synthetic anomalies created by adding zero-mean Gaussian noise with a large variance to the anomalous sample from the test set.

Baseline AD Methods. We compare EPHAD against IFOREST (Liu et al., 2012), LOF (Breunig et al., 2000), DeepSVDD (Ruff et al., 2018), ECOD (Li et al., 2023b) and COPOD (Li et al., 2020) using ADBench (Han et al., 2022).

Evidence Function. We use the output of Local Outlier Factor (LOF) (Breunig et al., 2000) and Isolation Forest (IForest) (Liu et al., 2012). Additional details provided in the Appendix B.2.2.

Results. The experimental results for the 27 benchmarking datasets are presented in Table 3 and 4. We observe that most AD methods benefit from our post-hoc adjustment framework EPHAD, often achieving performance improvements that surpass both the evidence function and the AD method in isolation. For example, COPOD, when updated with LOF as the

Table 3. Performance on tabular datasets with 10% contamination ratio and LOF as evidence function. Style: AUROC % (± SE). Best in **bold**. † represents transductive inference.

Dataset	LOF†	COPOD		DeepSVDD		ECOD		IForest		LOF	
		Blind	+ EPHAD	Blind	+ EPHAD	Blind	+ EPHAD	Blind	+ EPHAD	Blind	+ EPHAD
aloi	72.64 (± 0.1)	51.46 (± 0.05)	**52.55** (± 0.06)	54.06 (± 0.54)	**64.36** (± 0.21)	53.14 (± 0.03)	**54.33** (± 0.05)	54.05 (± 0.21)	**71.75** (± 0.08)	73.57 (± 0.1)	**73.62** (± 0.07)
annthyroid	68.53 (± 0.12)	73.45 (± 0.08)	**73.82** (± 0.08)	62.69 (± 3.33)	**67.00** (± 2.15)	76.05 (± 0.11)	**76.31** (± 0.11)	**71.39** (± 0.34)	70.41 (± 0.13)	**72.12** (± 0.57)	71.06 (± 0.24)
backdoor	70.43 (± 0.08)	75.06 (± 0.07)	**78.88** (± 0.08)	**78.34** (± 1.21)	76.48 (± 0.57)	83.00 (± 0.09)	**85.48** (± 0.08)	51.29 (± 1.29)	**70.13** (± 0.12)	46.65 (± 0.26)	**69.11** (± 0.1)
breastw	46.31 (± 0.92)	**99.46** (± 0.06)	98.52 (± 0.14)	**98.65** (± 0.05)	95.13 (± 0.97)	**99.01** (± 0.04)	97.44 (± 0.03)	**99.46** (± 0.04)	64.05 (± 1.17)	**73.39** (± 1.35)	62.4 (± 1.25)
celeba	41.45 (± 0.32)	**72.09** (± 0.01)	61.86 (± 0.1)	**67.51** (± 3.07)	55.60 (± 2.13)	**73.99** (± 0.01)	63.2 (± 0.09)	40.09 (± 0.83)	**40.32** (± 0.23)	**42.97** (± 0.23)	40.52 (± 0.38)
cover	52.12 (± 0.1)	78.70 (± 0.03)	**79.01** (± 0.02)	75.11 (± 11.37)	**75.74** (± 11.06)	85.34 (± 0.02)	**85.45** (± 0.02)	**72.59** (± 1.59)	63.64 (± 0.92)	22.44 (± 0.1)	**44.20** (± 0.07)
fault	55.00 (± 0.53)	**45.69** (± 0.58)	45.66 (± 0.57)	47.34 (± 0.99)	**48.59** (± 0.99)	**47.00** (± 0.4)	46.87 (± 0.4)	**58.08** (± 0.94)	55.92 (± 0.68)	**64.41** (± 1.35)	59.93 (± 0.37)
fraud	45.75 (± 0.13)	**94.39** (± 0.0)	94.24 (± 0.0)	**89.98** (± 0.97)	85.1 (± 0.66)	**93.86** (± 0.0)	93.62 (± 0.01)	**92.95** (± 0.29)	61.88 (± 0.49)	33.92 (± 0.34)	**45.26** (± 0.16)
glass	77.52 (± 0.93)	76.11 (± 0.77)	**79.45** (± 0.95)	64.52 (± 6.87)	**80.94** (± 3.31)	67.65 (± 0.44)	**72.59** (± 0.61)	78.50 (± 1.47)	**79.12** (± 1.01)	71.79 (± 1.08)	**76.40** (± 0.68)
http	37.65 (± 0.09)	**94.91** (± 0.01)	90.26 (± 0.04)	**99.17** (± 0.08)	94.97 (± 0.2)	**92.35** (± 0.02)	87.88 (± 0.04)	**96.82** (± 0.37)	69.51 (± 0.62)	17.85 (± 2.03)	**24.61** (± 0.89)
ionosphere	82.43 (± 0.16)	79.42 (± 1.03)	**81.67** (± 0.95)	83.09 (± 0.57)	**84.90** (± 0.17)	73.04 (± 0.84)	**74.34** (± 0.85)	**89.58** (± 1.57)	83.50 (± 0.16)	**94.64** (± 0.52)	89.74 (± 0.55)
letter	83.15 (± 0.73)	56.71 (± 0.12)	**57.62** (± 0.09)	50.51 (± 2.54)	**61.26** (± 2.42)	56.41 (± 0.29)	**57.17** (± 0.29)	59.84 (± 0.64)	**81.53** (± 0.59)	**85.74** (± 0.54)	84.84 (± 0.39)
lymphography	99.44 (± 0.26)	99.52 (± 0.22)	**99.76** (± 0.19)	98.57 (± 0.74)	**99.53** (± 0.19)	99.60 (± 0.23)	**99.76** (± 0.19)	**99.76** (± 0.19)	99.52 (± 0.19)	98.57 (± 0.59)	**99.36** (± 0.32)
mammography	67.29 (± 0.19)	**89.29** (± 0.05)	89.28 (± 0.05)	87.23 (± 0.95)	**87.29** (± 1.22)	**89.38** (± 0.06)	89.26 (± 0.05)	**80.44** (± 0.29)	73.93 (± 0.04)	69.70 (± 0.36)	**72.29** (± 0.18)
mnist	59.63 (± 0.19)	75.87 (± 0.03)	**75.89** (± 0.03)	**74.26** (± 4.38)	73.93 (± 4.24)	72.62 (± 0.05)	**72.64** (± 0.05)	**71.27** (± 0.7)	62.75 (± 0.16)	**94.55** (± 0.36)	83.26 (± 0.45)
musk	39.44 (± 0.57)	**91.95** (± 0.32)	91.91 (± 0.33)	**88.57** (± 5.4)	87.17 (± 5.87)	**71.84** (± 0.34)	71.78 (± 0.34)	**89.39** (± 1.88)	57.06 (± 2.03)	20.17 (± 0.48)	**32.93** (± 0.04)
optdigits	59.58 (± 0.26)	62.26 (± 0.24)	**62.49** (± 0.23)	40.01 (± 10.2)	**46.77** (± 8.53)	54.04 (± 0.21)	**54.36** (± 0.21)	40.87 (± 4.5)	**56.80** (± 0.68)	18.45 (± 0.59)	**50.59** (± 0.07)
pendigits	47.21 (± 0.12)	**88.44** (± 0.2)	88.38 (± 0.2)	**74.87** (± 9.91)	72.68 (± 8.72)	90.63 (± 0.17)	**90.65** (± 0.17)	**81.86** (± 1.48)	55.56 (± 0.98)	14.87 (± 0.18)	**37.64** (± 0.13)
satellite	52.90 (± 0.31)	64.33 (± 0.25)	**64.40** (± 0.25)	**60.59** (± 1.77)	62.63 (± 1.38)	57.57 (± 0.16)	**57.61** (± 0.16)	**76.31** (± 0.7)	63.85 (± 0.4)	61.01 (± 0.29)	**66.72** (± 0.28)
satimage-2	52.80 (± 0.15)	97.03 (± 0.06)	**97.20**(± 0.06)	92.65 (± 0.46)	**96.16** (± 0.31)	94.21 (± 0.03)	**94.39** (± 0.02)	**98.91** (± 0.09)	70.75 (± 0.44)	24.52 (± 0.87)	**47.14** (± 0.17)
shuttle	55.54 (± 0.11)	**99.26** (± 0.0)	99.19 (± 0.0)	**97.83** (± 0.91)	97.78 (± 0.79)	**98.82** (± 0.01)	98.64 (± 0.01)	**99.57** (± 0.02)	81.72 (± 0.27)	99.21 (± 0.01)	**99.69** (± 0.02)
smtp	89.77 (± 0.55)	79.64 (± 0.01)	**80.56** (± 0.12)	84.05 (± 0.57)	**86.10** (± 0.5)	87.98 (± 0.02)	**88.28** (± 0.09)	89.27 (± 0.88)	**89.80** (± 0.5)	43.01 (± 1.57)	**89.82** (± 0.27)
thyroid	75.91 (± 0.79)	88.45 (± 0.35)	**88.71** (± 0.31)	86.73 (± 3.72)	**88.33** (± 3.15)	**94.91** (± 0.14)	94.85 (± 0.14)	**93.67** (± 0.27)	83.42 (± 0.29)	73.59 (± 1.69)	**77.10** (± 0.53)
vowels	89.10 (± 0.67)	56.10 (± 0.32)	**58.87** (± 0.34)	64.47 (± 2.55)	**76.61** (± 1.24)	54.29 (± 0.06)	**56.82** (± 0.14)	66.01 (± 0.57)	**88.59** (± 0.65)	**93.04** (± 0.54)	91.30 (± 0.1)
wilt	64.63 (± 0.72)	33.45 (± 0.11)	**35.55** (± 0.1)	35.79 (± 1.97)	**46.44** (± 1.4)	38.06 (± 0.13)	**39.80** (± 0.15)	42.92 (± 1.11)	**61.30** (± 0.81)	**81.09** (± 0.41)	73.37 (± 0.3)
wine	97.57 (± 1.46)	80.51 (± 1.36)	**86.78** (± 1.96)	82.26 (± 2.29)	**92.94** (± 1.74)	67.12 (± 2.04)	**74.97** (± 2.88)	80.40 (± 3.42)	**97.51** (± 1.51)	**99.94** (± 0.05)	**99.94** (± 0.05)

evidence function, shows this behaviour. Additionally, as seen in the image-based experiments, performance degradation in certain cases arises when the framework places excessive emphasis on an evidence function that is substantially weaker than the AD method. However, as previously discussed, this limitation can be mitigated by appropriately tuning β.

C.3. Experiments on Industrial Use Case

Concentrated Solar Power (CSP) Plant Dataset. For the industrial setting, we utilise the simulated dataset introduced by Patra et al. (2024), which is generated by training a variational autoencoder on real-world data collected from an operational CSP plant. The dataset consists of thermal images of solar panels captured using infrared (IR) cameras, distinguishing it from the semantic and sensory anomaly datasets, as the images lack semantic structure and do not depict specific objects.

Baseline AD Method. We evaluate the performance of the forecasting-based anomaly detection method `ForecastAD`, as proposed by the original authors, both with and without the integration of `EPHAD`. All experiments are conducted using the original implementation provided by the authors.

Rule-based Evidence. To compute evidence, we utilise two of the four rules proposed by Patra et al. (2024) that indicate normal operational behaviour of the CSP plant. The first rule (**R1**) is based on the *difference between consecutive images*. Under normal conditions, the plant's temperature is expected to remain relatively stable; therefore, substantial deviations from one image to the next suggest potential anomalies. To quantify this, pixel-wise squared differences are computed between every pair of consecutive images, and the 95th percentile of these differences is extracted as the representative evidence for each pair. The second rule (**R2**) involves the *difference from the average daily temperature*. Here, samples with average temperatures significantly diverging from the typical daily average could indicate anomalous behaviour. For this, the mean temperature of each day is first determined, and then the absolute difference between each image's average temperature and that day's mean is computed to serve as the evidence.

Table 5. Performance on CSP plant dataset.

Setting	Method	AUROC (± SE)
Evidence	Rule-based (R1, R2)	69.46 (± 0.0)
Clean	`ForecastAD`	94.91 (±0.09)
Contaminated	`ForecastAD`	90.45 (± 0.8)
($\epsilon = 0.1$)	+ `EPHAD`	**93.51 (± 0.45)**

Results. The results presented in Table 5 underscore the effectiveness and adaptability of our approach. Under a 10% contamination setting, the baseline method `ForecastAD` experiences a performance drop of approximately 5%. However, by incorporating domain-specific rules R1 and R2 as sources of evidence using `EPHAD`, the performance nearly matches

Table 4. Performance on tabular datasets with 10% contamination ratio and IForest as evidence function. Style: AUROC % (± SE). Best in **bold**. † represents transductive inference.

Dataset	IForest†	COPOD		DeepSVDD		ECOD		IForest		LOF	
		Blind	+ EPHAD	Blind	+ EPHAD	Blind	+ EPHAD	Blind	+ EPHAD	Blind	+ EPHAD
aloi	54.18 (± 0.31)	51.46 (± 0.05)	**51.48** (± 0.04)	54.06 (± 0.54)	**54.43** (± 0.51)	53.14 (± 0.03)	**53.16** (± 0.03)	54.05 (± 0.21)	**54.26** (± 0.22)	**73.57** (± 0.1)	69.30 (± 0.18)
annthyroid	78.62 (± 1.01)	73.45 (± 0.08)	**73.85** (± 0.05)	62.69 (± 3.33)	**66.63** (± 2.19)	76.05 (± 0.11)	**76.20** (± 0.09)	71.39 (± 0.34)	**76.91** (± 0.88)	72.12 (± 0.57)	**76.67** (± 0.39)
backdoor	67.83 (± 1.69)	75.06 (± 0.07)	**75.06** (± 0.06)	78.34 (± 1.21)	**81.43** (± 0.72)	**83.00** (± 0.09)	82.95 (± 0.09)	51.29 (± 1.29)	**66.48** (± 1.43)	46.65 (± 0.26)	**66.23** (± 1.04)
breastw	97.97 (± 0.14)	99.46 (± 0.06)	**99.46** (± 0.05)	98.65 (± 0.05)	**98.96** (± 0.04)	99.01 (± 0.04)	**99.07** (± 0.04)	**99.46** (± 0.04)	98.98 (± 0.09)	73.39 (± 1.35)	**81.16** (± 1.08)
celeba	66.62 (± 1.04)	**72.09** (± 0.01)	72.00 (± 0.01)	67.51 (± 3.07)	**68.20** (± 2.59)	**73.99** (± 0.01)	73.87 (± 0.01)	40.09 (± 0.83)	**60.55** (± 1.07)	42.97 (± 0.23)	**49.73** (± 0.63)
cover	86.11 (± 1.6)	78.70 (± 0.03)	**79.01** (± 0.09)	75.11 (± 11.37)	**77.54** (± 9.82)	85.34 (± 0.02)	**85.44** (± 0.06)	72.59 (± 1.59)	**82.94** (± 1.71)	22.44 (± 0.1)	**76.71** (± 2.42)
fault	52.02 (± 0.18)	45.69 (± 0.58)	**45.73** (± 0.58)	47.34 (± 0.99)	**47.89** (± 0.94)	47.00 (± 0.4)	**47.04** (± 0.39)	**58.08** (± 0.94)	53.76 (± 0.41)	**64.41** (± 1.35)	58.97 (± 0.96)
fraud	94.87 (± 0.11)	94.39 (± 0.0)	**94.40** (± 0.0)	89.98 (± 0.97)	**92.26** (± 0.5)	93.86 (± 0.0)	**93.87** (± 0.01)	92.95 (± 0.29)	**94.60** (± 0.08)	33.92 (± 0.34)	**85.94** (± 0.34)
glass	77.60 (± 1.77)	76.11 (± 0.77)	**76.29** (± 0.8)	64.52 (± 6.87)	**69.28** (± 5.85)	67.65 (± 0.44)	**68.26** (± 0.52)	**78.50** (± 1.47)	77.85 (± 1.64)	71.79 (± 1.08)	**81.23** (± 0.95)
http	99.99 (± 0.0)	94.91 (± 0.01)	**96.84** (± 0.05)	99.17 (± 0.08)	**99.24** (± 0.05)	92.35 (± 0.02)	**94.49** (± 0.07)	96.82 (± 0.37)	**99.63** (± 0.02)	17.85 (± 2.03)	**94.04** (± 0.05)
ionosphere	81.80 (± 0.28)	79.42 (± 1.03)	**79.49** (± 1.0)	83.09 (± 0.57)	**83.57** (± 0.62)	73.04 (± 0.84)	**73.21** (± 0.85)	**89.58** (± 1.57)	85.24 (± 0.63)	**94.64** (± 0.52)	94.23 (± 0.68)
letter	61.76 (± 0.26)	56.71 (± 0.12)	**56.76** (± 0.12)	50.51 (± 2.54)	**52.37** (± 2.32)	56.41 (± 0.29)	**56.47** (± 0.29)	59.84 (± 0.64)	**61.35** (± 0.32)	**85.74** (± 0.54)	80.36 (± 0.32)
lymphography	99.92 (± 0.07)	**99.52** (± 0.22)	**99.52** (± 0.22)	98.57 (± 0.74)	**99.28** (± 0.41)	99.60 (± 0.23)	**99.68** (± 0.17)	99.76 (± 0.19)	**99.84** (± 0.13)	98.57 (± 0.59)	**99.68** (± 0.26)
mammography	83.98 (± 0.32)	**89.29** (± 0.05)	89.22 (± 0.04)	87.23 (± 0.95)	**87.76** (± 0.85)	**89.38** (± 0.06)	89.24 (± 0.04)	80.44 (± 0.29)	**83.14** (± 0.17)	69.70 (± 0.36)	**83.30** (± 0.15)
mnist	75.50 (± 0.08)	75.87 (± 0.03)	**75.88** (± 0.03)	74.26 (± 4.38)	**76.20** (± 3.66)	72.62 (± 0.05)	**72.65** (± 0.05)	71.27 (± 0.7)	**74.86** (± 0.18)	**94.55** (± 0.36)	91.46 (± 0.39)
musk	99.29 (± 0.33)	91.95 (± 0.32)	**92.00** (± 0.32)	88.57 (± 5.4)	**91.39** (± 4.15)	71.84 (± 0.34)	**71.92** (± 0.35)	89.39 (± 1.88)	**98.74** (± 0.21)	20.17 (± 0.48)	**89.22** (± 2.5)
optdigits	58.65 (± 3.55)	**62.26** (± 0.24)	62.25 (± 0.26)	40.01 (± 10.2)	**42.56** (± 9.28)	54.04 (± 0.21)	**54.09** (± 0.24)	40.87 (± 4.5)	**53.81** (± 1.83)	18.45 (± 0.59)	**38.72** (± 2.67)
pendigits	92.04 (± 0.23)	88.44 (± 0.2)	**88.58** (± 0.21)	74.87 (± 9.91)	**79.77** (± 8.09)	90.63 (± 0.17)	**90.73** (± 0.18)	81.86 (± 1.48)	**90.40** (± 0.11)	14.87 (± 0.18)	**68.81** (± 1.12)
satellite	64.44 (± 0.57)	**64.33** (± 0.25)	**64.33** (± 0.25)	60.59 (± 1.77)	**60.84** (± 1.49)	57.57 (± 0.16)	**57.60** (± 0.16)	**76.31** (± 0.7)	68.34 (± 0.51)	61.01 (± 0.29)	**72.19** (± 0.45)
satimage-2	99.43 (± 0.07)	97.03 (± 0.06)	**97.06** (± 0.06)	92.65 (± 0.46)	**95.23** (± 0.06)	94.21 (± 0.03)	**94.27** (± 0.03)	98.91 (± 0.09)	**99.41** (± 0.06)	24.52 (± 0.87)	**92.79** (± 0.16)
shuttle	98.97 (± 0.08)	99.26 (± 0.0)	**99.28** (± 0.01)	97.83 (± 0.91)	**98.30** (± 0.78)	98.82 (± 0.01)	**98.85** (± 0.0)	**99.57** (± 0.02)	99.46 (± 0.04)	99.21 (± 0.01)	**99.89** (± 0.01)
smtp	90.95 (± 0.28)	79.64 (± 0.01)	**81.14** (± 0.06)	84.05 (± 0.57)	**87.46** (± 0.73)	87.98 (± 0.02)	**88.41** (± 0.04)	89.27 (± 0.88)	**90.78** (± 0.3)	43.01 (± 1.57)	**88.96** (± 0.35)
thyroid	96.65 (± 0.26)	88.45 (± 0.35)	**89.21** (± 0.32)	86.73 (± 3.72)	**89.21** (± 2.86)	94.91 (± 0.14)	**95.06** (± 0.15)	93.67 (± 0.27)	**96.02** (± 0.18)	73.59 (± 1.69)	**93.41** (± 0.25)
vowels	72.73 (± 0.8)	56.10 (± 0.32)	**56.50** (± 0.31)	64.47 (± 2.55)	**66.27** (± 2.37)	54.29 (± 0.06)	**54.65** (± 0.06)	66.01 (± 0.57)	**71.08** (± 0.84)	**93.04** (± 0.54)	91.68 (± 0.34)
wilt	42.57 (± 1.63)	33.45 (± 0.11)	**33.70** (± 0.17)	35.79 (± 1.97)	**36.43** (± 1.88)	38.06 (± 0.13)	**38.14** (± 0.17)	**42.92** (± 1.11)	42.66 (± 1.4)	**81.09** (± 0.41)	71.40 (± 0.64)
wine	58.98 (± 0.68)	**80.51** (± 1.36)	80.34 (± 1.39)	**82.26** (± 2.29)	81.07 (± 2.51)	**67.12** (± 2.04)	67.06 (± 2.08)	**80.40** (± 3.42)	68.47 (± 2.3)	**99.94** (± 0.05)	99.72 (± 0.12)

that on the clean dataset. It emphasises the value of leveraging structured, context-aware evidence to enhance the detection of anomalies. Importantly, foundation models like CLIP are unsuitable in this context due to the lack of semantic content in thermal imagery, rendering zero-shot approaches such as WinCLIP (Jeong et al., 2023) and AnoCLIP (Zhou et al., 2024) ineffective. EPHAD addresses this limitation by providing a flexible framework that integrates both powerful foundation models, where applicable, and domain-specific knowledge when necessary. This versatility enables EPHAD to deliver robust performance across diverse real-world anomaly detection tasks while maintaining efficiency and ease of deployment.

C.4. Comparison against LOE

To ensure a comprehensive evaluation, we compare the performance of our proposed post-hoc framework against both variants of LOE (Qiu et al., 2022). However, it is important to note that, unlike our approach, LOE modify the training process to account for contamination, making it inapplicable to pre-trained networks without access to the training dataset and pipeline, which is our main focus.

Table 6. Comparison with LOE (AUROC %)

	Method	Semantic AD				Sensory AD		
		MNIST	FMNIST	CIFAR10	SVHN	MVTec	MPDD	ViSA
	CLIP	71.15	95.63	98.63	58.46	86.34	60.02	74.47
NTL	Blind	90.15	89.01	90.79	**61.82**	78.13	80.41	61.95
NTL	Refine	91.35	91.37	92.79	61.78	82.54	87.32	65.63
NTL	LOE-Hard	86.89	90.53	93.10	53.86	79.28	83.34	**78.82**
NTL	LOE-Soft	**91.56**	92.89	94.71	61.69	85.46	**92.31**	74.5
	EPHAD	78.96	**95.99**	**98.65**	57.64	**86.20**	59.88	74.22

For comparison with LOE, we conduct experiments using the Neural Transformation Learning-based (NTL) AD method (Qiu et al., 2021) and evaluate it under four configurations: "Blind", "Refine", LOE-Hard and LOE-Soft. Additionally, we follow the same setup as LOE by extracting image features using pre-trained ResNet152 and WideResNet50 for semantic and sensory datasets, respectively, which are then used to train NTL. The results, summarised in Table 6, show that given a good evidence function, i.e. the performance of the evidence is better than the "Blind" configuration, our simple inference-time framework outperforms LOE.

Results on MVTec, CIFAR10, FMIST, and SVHN are examples of this behaviour. Also, on the ViSA dataset, the performance improves over the "Blind" and "Refine" configurations. In the converse situations where the performance of the

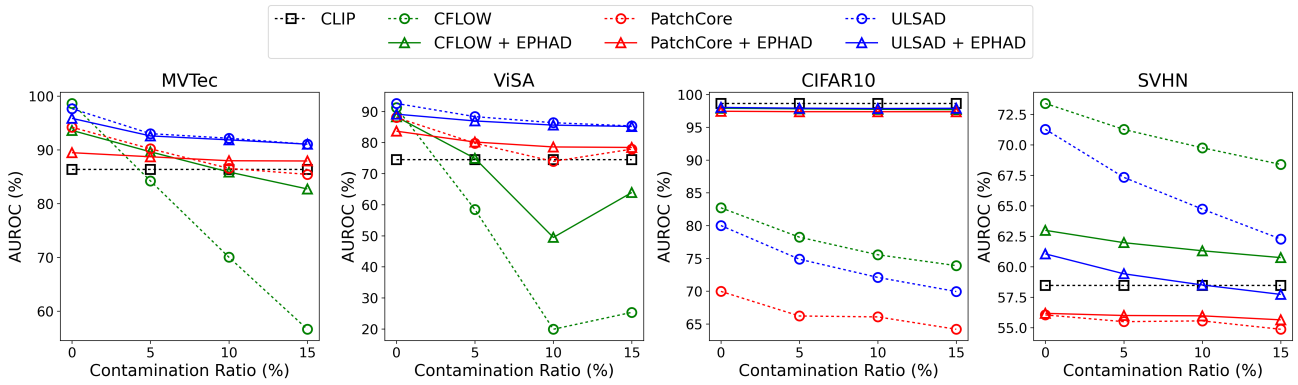

Figure 2. Ablation on ϵ.

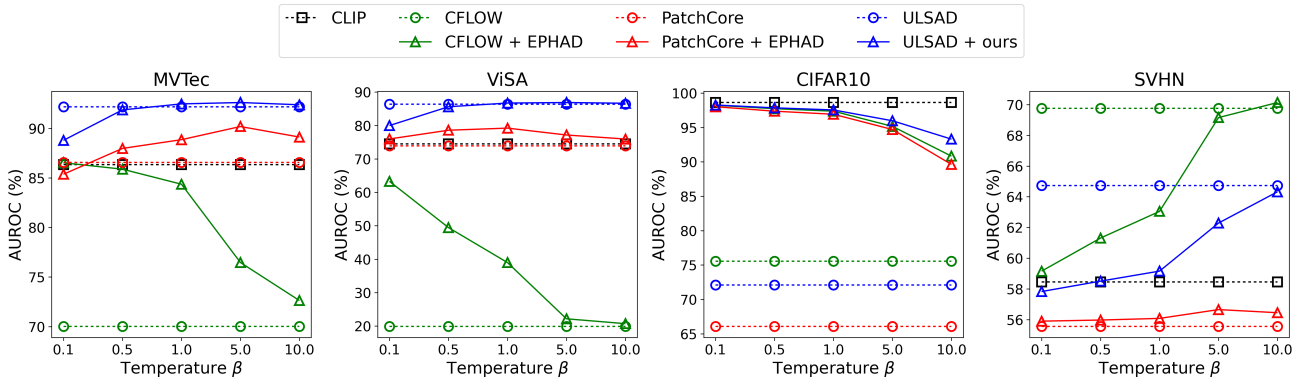

Figure 3. Ablation on β.

evidence is lower than the "Blind" configuration, we observe a reduction in performance which can be accounted for by putting more emphasis on the AD model by adjusting β.

C.5. Ablation on ϵ and β

In this section, we first analyse the sensitivity of EPHAD to various contamination ratios. Then, we investigate the effect of the temperature β on AD performance.

Effect of varying contamination ratio. Here, we evaluate the sensitivity of our proposed framework by varying the contamination ratio $\{0\%, 5\%, 10\%, 15\%\}$. The results are summarised in the Figure 2. Applying EPHAD results in improvements across all contamination ratios for most of the AD methods. Furthermore, in the presence of a strong evidence function, such as CLIP, we can observe that the performance becomes almost constant even as the contamination ratio increases from 5% to 15%.

Effect of temperature parameter β. We also analyse the performance of the EPHAD by varying the temperature parameter β. In Figure 3, we can see how β allows for controlling the trade-off between the prior AD method and the evidence. As discussed earlier, we observe that setting $\beta \approx 0$ results in full reliance on $T(x)$, while with increasing β, $T(x)$ is disregarded and it defaults to the prior.

C.6. Effect of Test Set Size n

The performance of our proposed framework, EPHAD, is influenced by both the pre-trained AD method and the evidence function. While the pre-trained AD method is affected only by the training data, for the evidence function, we evaluated two scenarios: (1) When using foundation models such as CLIP, the evidence function remains independent of the test sample distribution. (2) When employing traditional AD methods like Isolation Forest or Local Outlier Factor, the evidence

function relies on the local density of test samples, meaning that an insufficient number of test samples could lead to less informative evidence which can be accounted for in EPHAD by adjusting the temperature parameter β. In Figure 4, we analyse the impact of varying the proportion of anomalies in the test set, which exhibits consistent improvements across all tested settings.

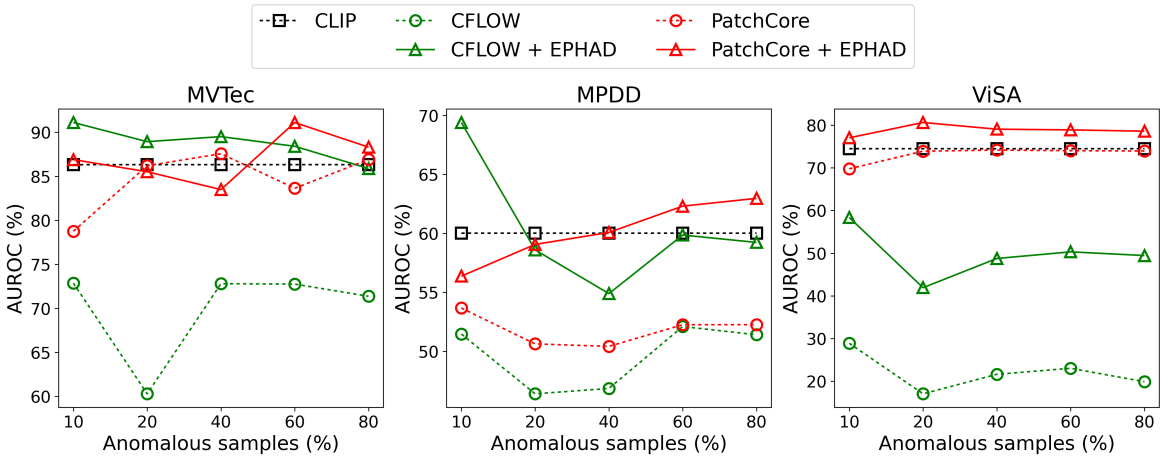

Figure 4. Ablation on varying proportion of anomalies in the test set.

C.7. Determining the Temperature Parameter β

As previously discussed, EPHAD has only a single hyperparameter, β, which controls the trade-off between reliance on the prior AD model and the evidence function $T(x)$. A straightforward approach to selecting β would involve evaluating the AD performance of the prior and $T(x)$ individually on a validation set and choosing β accordingly. However, this strategy introduces additional computational overhead during inference and requires access to a labelled validation set of sufficient size to ensure reliable performance estimation – conditions often impractical in real-world deployments. To address this limitation, we propose an adaptive extension of our approach, termed EPHAD-Ada, which determines the optimal β in an unsupervised manner using only the test data during inference. This adaptation is inspired by the principle of Entropy Minimization (EM) (Press et al., 2024), a widely-used technique in test-time adaptation (Xiao and Snoek, 2024). Motivated by the observation from Wang et al. (2021) that models tend to be more accurate when predictions are made with high confidence, we apply it to compute the hyperparameter β. We begin by calculating the inlier probability from the output scores to derive the entropy of the predictions from either the evidence function or the prior model.

Computing inlier probability from anomaly scores. Let $S = s_\theta(X)$ and $s = s_\theta(x)$, then the inlier probability is given by

$$p_{Y=+1}(x) := \mathbb{P}(Y = +1 \mid s_\theta(X) = s_\theta(x)) = \mathbb{P}(Y = +1 \mid S = s) = \mathbb{P}(S > s) = 1 - p_s, \tag{17}$$

where $p_s := \mathbb{P}(S \leq s)$. Since p_s is unknown in practice, we follow the approach of Perini et al. (2020) and treat it as a random variable P_s with a prior distribution $\text{Beta}(1, 1)$, corresponding to a uniform prior over $[0, 1]$. Given that the label $Y \in \{+1, -1\}$, we model the conditional distribution $Y \mid S = s$ as a Bernoulli random variable. To estimate p_s, we draw samples $a \sim S$ by first sampling $x' \sim \mathcal{X}$ and computing the corresponding anomaly score $a = s_\theta(x')$. We record a success ($b = 1$) if $a \leq s$, and a failure ($b = 0$) otherwise. Repeating this procedure n times yields t successes and $n - t$ failures. Then, according to Theorem 2 in Perini et al. (2020), the posterior distribution of P_s given the observed binary outcomes $b_1, \ldots, b_n$ is $\text{Beta}(1 + t, 1 + n - t)$. We estimate p_s using the posterior mean of P_s as

$$p_s := \mathbb{E}[P_s] = \frac{1 + t}{2 + n}. \tag{18}$$

In practice, the posterior is inferred from test samples, so the sample size n is constrained by the number of available test points. Finally, combining Equations (17) and (18), we obtain the estimated inlier probability for a data point x as

$$p_{Y=+1}(x) = 1 - p_s = 1 - \frac{1 + t}{2 + n}. \tag{19}$$

Table 7. Performance on both sensory and semantic AD benchmarking datasets with 10% contamination ratio. Style: AUROC % (± SE). Best in **bold**.

Method	Non-overlap					Overlap		
	MNIST	FMNIST	CIFAR10	SVHN	RealIAD	MVTec	MPDD	ViSA
CLIP	71.15	95.63	98.63	58.46	65.74	86.34	60.02	74.47
CFLOW	77.24 (± 1.01)	72.87 (± 0.48)	65.47 (± 0.02)	55.09 (± 0.09)	**76.42 (± 0.47)**	87.58 (± 0.77)	66.69 (± 2.06)	75.71 (± 1.28)
+ EPHAD-Ada	**78.08 (± 0.91)**	**91.63 (± 0.29)**	**96.43 (± 0.0)**	**55.78 (± 0.04)**	73.86 (± 0.24)	**89.84 (± 0.3)**	**67.81 (± 1.63)**	**79.64 (± 0.63)**
DRÆM	71.44 (± 0.29)	76.53 (± 0.18)	63.41 (± 0.26)	51.55 (± 0.07)	67.46 (± 0.21)	70.55 (± 1.97)	62.32 (± 1.96)	69.61 (± 1.57)
+ EPHAD-Ada	**72.88 (± 0.33)**	**84.96 (± 0.97)**	**87.73 (± 1.52)**	**53.79 (± 0.36)**	**70.15 (± 0.05)**	**87.24 (± 0.39)**	**69.55 (± 0.42)**	**74.95 (± 1.15)**
FastFlow	82.65 (± 0.43)	83.66 (± 0.06)	62.94 (± 0.37)	54.02 (± 0.11)	**82.03 (± 0.08)**	84.24 (± 1.07)	**71.94 (± 0.87)**	77.83 (± 0.22)
+ EPHAD-Ada	**82.83 (± 0.44)**	**92.1 (± 0.14)**	**96.24 (± 0.05)**	**55.26 (± 0.17)**	81.1 (± 0.06)	**88.07 (± 0.8)**	70.08 (± 0.41)	**80.71 (± 0.08)**
PaDiM	87.5 (± 0.23)	86.84 (± 0.06)	62.53 (± 0.4)	55.49 (± 0.28)	**80.39 (± 0.35)**	77.85 (± 0.43)	36.58 (± 2.58)	73.07 (± 0.27)
+ EPHAD-Ada	**87.56 (± 0.23)**	**92.87 (± 0.02)**	**90.23 (± 0.67)**	**57.09 (± 1.05)**	79.56 (± 0.28)	**86.1 (± 0.52)**	**49.06 (± 1.52)**	**76.62 (± 0.38)**
PatchCore	86.33 (± 0.09)	78.97 (± 0.06)	75.69 (± 0.09)	**69.64 (± 0.04)**	70.08 (± 0.07)	70.51 (± 0.7)	53.58 (± 0.54)	27.2 (± 0.31)
+ EPHAD-Ada	**86.38 (± 0.1)**	**89.99 (± 0.2)**	**96.63 (± 0.09)**	68.4 (± 0.52)	**77.18 (± 0.09)**	**83.53 (± 0.18)**	**56.97 (± 1.23)**	**48.6 (± 0.51)**
RD	77.33 (± 0.09)	84.11 (± 0.72)	66.29 (± 0.31)	55.54 (± 0.58)	**89.13 (± 0.18)**	80.08 (± 1.32)	**75.08 (± 1.75)**	**86.33 (± 0.46)**
+ EPHAD-Ada	**78.91 (± 0.21)**	**95.64 (± 0.04)**	**98.0 (± 0.17)**	**57.78 (± 0.5)**	72.78 (± 0.43)	**86.69 (± 0.38)**	63.97 (± 0.88)	79.42 (± 0.34)
ULSAD	**90.83 (± 0.08)**	88.64 (± 0.13)	72.45 (± 0.18)	**64.27 (± 0.22)**	**89.06 (± 0.01)**	91.93 (± 0.15)	77.67 (± 0.42)	86.58 (± 0.13)
+ EPHAD-Ada	90.8 (± 0.07)	**94.55 (± 0.08)**	**97.29 (± 0.02)**	59.68 (± 0.16)	85.84 (± 0.04)	**92.25 (± 0.07)**	**76.31 (± 1.04)**	**87.23 (± 0.05)**

Computing the value of hyperparameter β. We begin by converting the anomaly scores from the prior AD model and $T(x)$ into inlier probabilities $p^p_{Y=+1}(x)$ and $p^t_{Y=+1}(x)$, respectively, using (18). We then compute the entropy of the prior model's prediction H_{prior} as:

$$H_{\text{prior}} = \sum_{x \in \mathcal{D}_{\text{test}}} \left[-(p^p_{Y=+1}(x) \ \log p^p_{Y=+1} + p^p_{Y=-1}(x) \ \log p^p_{Y=-1}) \right], \tag{20}$$

where $p^p_{Y=-1}(x) = 1 - p^p_{Y=+1}(x)$. Similarly, we compute the entropy of the evidence model's prediction H_{evi} as

$$H_{\text{evi}} = \sum_{x \in \mathcal{D}_{\text{test}}} \left[-(p^e_{Y=+1}(x) \ \log p^e_{Y=+1} + p^e_{Y=-1}(x) \ \log p^e_{Y=-1}) \right], \tag{21}$$

with $p^e_{Y=-1}(x) = 1 - p^e_{Y=+1}(x)$. A low H_{prior} indicates that the prior model is confident in its predictions, suggesting that a higher value of β is appropriate to place greater trust in the prior. Conversely, a lower H_{evi} implies greater reliability in the evidence function, advocating for a smaller β. Based on this intuition, we define the adaptive temperature parameter as:

$$\beta_{\text{adaptive}} = \frac{H_{\text{evi}}}{H_{\text{prior}} + \delta}, \tag{22}$$

where δ is a small positive constant introduced to ensure numerical stability. Through this formulation, EPHAD-Ada effectively enables unsupervised, test-time selection of β, thereby enhancing practicality and reducing reliance on labelled validation data.

Results. The effect of applying EPHAD-Ada is summarised in Table 7. We can observe that in most of the scenarios, applying EPHAD-Ada improves the performance of the base AD method. Moreover, in some cases, such as applying EPHAD-Ada to CFLOW on CIFAR10, the performance improvement is significant. Interestingly, even when the evidence function in isolation does not achieve good performance as for RealIAD, its use as a part of EPHAD-Ada significantly improves the performance of PatchCore, showing the framework's effectiveness. While EPHAD-Ada provides a way to determine the hyperparameter in an unsupervised manner, we observe a performance drop in certain scenarios, specifically when the performance of the base AD method is better than the evidence function. We hypothesise that this occurs as the inlier probability computed from the anomaly scores is not calibrated. We leave the further investigation and the development of a better approach for determining the value of β as a promising future research.

D. Related Work

Unsupervised AD. Over the years, numerous approaches have been developed for unsupervised AD, which can be broadly categorized into four main families: *one-class classifiers* (OCC), *feature embedding-based*, *density-based*, and

reconstruction-based methods. *One-class classifiers* aim to learn a decision boundary that encapsulates all normal samples. Classical OCC approaches employ shallow models such as support vector-based methods that learn a maximum-margin hyperplane (Schölkopf et al., 2001) or a hypersphere (Tax and Duin, 1999). To mitigate the limitations of manual feature engineering and extend to high-dimensional data, deep learning-based variants like DeepSVDD (Ruff et al., 2018) have been introduced.

Feature embedding-based methods, on the other hand, leverage pre-trained deep models to extract representations of input data. These representations are then either stored in a memory bank (Roth et al., 2022; Lee et al., 2022) or used to train a student-teacher network (Zhang et al., 2024; Batzner et al., 2024; Patra and Ben Taieb, 2024). *Density-based* methods detect anomalies by estimating the probability distribution of normal samples, assuming that anomalies reside in low-density regions. While early methods include KDE (Kim and Scott, 2012), more recent deep-learning-based variants include DAGMM (Zong et al., 2018), CFLOW (Gudovskiy et al., 2022), and FastFlow (Yu et al., 2021). Lastly, *reconstruction-based* approaches learn to map normal samples into a lower-dimensional bottleneck and reconstruct them. The inability to accurately reconstruct samples during inference serves as a detection criterion. For a more comprehensive survey, we refer readers to Liu et al. (2024) and Ruff et al. (2021).

Data Contamination. Handling dataset contamination in AD typically assumes a low proportion of anomalies, allowing methods to prioritise normal instances (inlier priority) (Wang et al., 2019). However, in practice, this assumption is difficult to ensure since anomalies are often unknown. To mitigate contamination, Yoon et al. (2022) proposed a data refinement approach using an ensemble of one-class classifiers (OCCs) to filter suspected anomalies and create a cleaner dataset. While effective, this method incurs high computational costs and discards anomalies rather than leveraging them for improved generalisation via Outlier Exposure (Hendrycks et al., 2019).

To address this, Qiu et al. (2022) introduced Latent Outlier Exposure (LOE), which iteratively assigns anomaly scores and infers labels using block coordinate descent while incorporating the contamination ratio to prevent degenerate solutions. However, estimating the contamination ratio remains a challenge. Perini et al. (2022) tackled this by leveraging an auxiliary dataset with a known contamination ratio, assuming domain similarity. Alternatively, Perini et al. (2023) fits a Dirichlet Process Gaussian Mixture Model to anomaly scores, though this approach lacks a closed-form solution. Despite these advancements, existing methods introduce computational overhead and are often impractical for modern pre-trained proprietary models, limiting their real-world applicability.