Université de Mons
Faculté des Sciences

Deep Visual Anomaly Detection under Data Contamination and Anomaly Heterogeneity

Sukanya Patra

A dissertation submitted in fulfilment of the requirements of the degree of *Docteur en Sciences*

Advisor

Prof. Souhaib Ben Taieb
Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates
University of Mons, Belgium

Co-Advisor

Prof. Stéphane Dupont
University of Mons, Belgium

Members of the Jury

Prof. Hadrien Melot
University of Mons, Belgium
Prof. Olivier Caelen
Université Catholique de Louvain, Belgium
Prof. Karthik Nandakumar
Michigan State University, United States

Acknowledgements

Throughout my PhD journey, I have had the privilege of working and interacting with many remarkable individuals to whom I owe deep gratitude for making this experience meaningful and memorable.

First and foremost, I would like to express my heartfelt thanks to my supervisor, Prof. Souhaib Ben Taieb, for believing in me and providing the opportunity to embark on this doctoral journey. The path has not always been easy, and there were times when I struggled to keep pace. However, your constant support, constructive feedback, and patient guidance helped me recognise my shortcomings and work towards continuous improvement. During our meetings, I often worried about the questions you might pose, but in retrospect, those moments were invaluable. I have grown considerably both as a researcher and as an individual under your mentorship. Beyond academic matters, your assistance with personal and logistical challenges, from settling in Mons to finding accommodation in Abu Dhabi, was deeply appreciated. I am truly grateful to have had such a supportive supervisor.

I would also like to extend my sincere gratitude to Prof. Sidi Ahmed Mahmoudi, Prof. Stéphane Dupont and Prof. Gilles Louppe for their valuable guidance as members of my PhD committee. Your insightful suggestions greatly contributed to shaping my research. I am equally grateful for your continued support beyond my doctoral studies, particularly for your willingness to act as a reference during my job applications. I would also like to thank Prof. Hadrien Melot, Prof. Olivier Caelen, and Prof. Karthik Nandakumar for graciously agreeing to serve on my thesis defense committee. This research was supported by the Federated Learning and Augmented Reality for Advanced Control Centres project, whose funding is gratefully acknowledged. Within the

scope of this project, I had the pleasure of interacting with Adrien Farinelle, Thibault Georges, and Geoffroy Herbin. I am sincerely thankful for their guidance and for the valuable discussions that deepened my understanding of the project's use cases.

My warm thanks go to my lab mates, Victor and Tanguy, for making this journey truly enjoyable. Your assistance was invaluable on many occasions, especially when language barriers posed challenges. I will always cherish the summer we spent together in Oxford and our memorable outing in Abu Dhabi. I am also grateful to my collaborators Hien, Yorick, David, and Nicolas. Hien, I learned a great deal from working with you, and I am especially thankful for your last-minute feedback on my thesis. Yorick, I will never forget the tight deadlines we faced during our PKDD project and the numerous calls that helped us meet them. I truly admire your clarity of thought and strong work ethic. Thank you also for your valuable feedback on my thesis, and I wish you continued success in your PhD journey. David, it was a pleasure working with you, and I wish you all the best in your doctoral research. Nicolas, collaborating with you was a wonderful experience. Your enthusiasm and readiness to take on challenges made it possible for us to conclude your thesis work with a publication at KDD. Our time in Barcelona remains a highlight, and I wish you continued success in your professional career.

I would also like to thank Naomi for encouraging an introvert like me to engage socially. I greatly enjoyed our time together in Abu Dhabi. I would also like to acknowledge Elnura for contributing to the memorable moments I had there. Natarajan, thank you for welcoming me on my first day in Mons and introducing me to the city. As fellow internationals navigating a new environment and language, your company in handling administrative processes, and house-hunting was immensely helpful. Beyond Mons, I would also like to extend my gratitude to my friends in Eindhoven. During challenging times, a short conversation with them could always brighten my day. To all others who made my stay in Eindhoven and Mons a little easier and more enjoyable, even if I have not mentioned you by name, please accept my heartfelt thanks for being part of this journey.

Living far from home throughout this period was challenging, particularly being away from my family. I am deeply grateful to my parents, sisters, and brother-in-law for their unwavering love and support through every phase of this journey. The most difficult part for me was leaving behind my little niece just when she was learning to recognise and bond with family. I was afraid that she might not remember me. Fortunately, we got used to the distance and now

we share a very special bond. I cannot wait to witness all her adventures and accomplishments as she grows up. Finally, to Sayak, thank you for being my constant source of encouragement, for believing in me when I doubted myself, and for standing by me through every challenge. This accomplishment would not have been possible without your support.

Abstract

Anomaly detection (AD) is the task of identifying rare and unusual events that deviate from expected behaviour. It plays a crucial role in various high-stakes domains, including industrial quality inspection, healthcare, fraud detection, and predictive maintenance. A standard approach involves learning a "compact" representation of the normal samples. Once this notion of normality is established, instances that significantly deviate from it are identified as anomalies. Traditional shallow AD methods often struggle in high-dimensional data settings due to the *curse of dimensionality*, where the performance deteriorates as the number of input features grows. Consequently, *deep learning*-based methods have gained attention due to their ability to learn effective representations directly from the data.

Despite remarkable progress in deep learning, the practical deployment of deep AD models remains hindered by several fundamental challenges. This thesis advances the field through four key contributions, each addressing specific practical limitations of existing approaches. The work is conducted as a part of the Federated Learning and Augmented Reality for Advanced Control Centres (FLARACC) project, which aims to develop solutions for real-world industrial problems. FLARACC is a collaboration among the University of Mons, the University of Namur, John Cockerill, IBA and CETIC.

First, in real-world applications, different types of anomalies often occur simultaneously, rendering existing methods ineffective s they typically focus on a single anomaly type. As our first contribution, we develop a unified method for the detection of both structural and logical anomalies. The proposed method achieves competitive performance across multiple benchmark datasets, demonstrating the ability to detect co-occurring anomaly types.

Second, contamination in the training dataset undermines the common assumption that training datasets are "clean", i.e. free of anomalous samples. To address this, our second contribution introduces two complementary strategies. In semi-supervised AD, we propose two risk-based estimators: a shallow method with a regularised unbiased risk estimator and a deep method employing a non-negative risk estimator, both supported by theoretical guarantees. In the fully unsupervised setting, we develop a test-time adaptation framework that dynamically adjusts model predictions using exponential tilting, improving robustness against contamination without requiring labelled data.

Third, motivated by a real-world use case of AD in solar power plants from John Cockerill, we address the challenge of learning effective representations for AD given a thermal image dataset with complex temporal features such as non-stationarity, strong daily seasonal patterns, irregular sampling intervals, and temporal dependencies. Our third contribution proposes a forecasting-based AD framework, where a deep sequence model predicts the next thermal image under normal operating conditions. Then anomalies are identified as deviations between predicted and observed thermal images. This approach enables the detection of anomalous behaviours by capturing temporal dynamics and extracting meaningful representations from thermal data.

Finally, while deep learning-based methods can learn expressive representations, they often produce unreliable and overly optimistic predictions, which is harmful for safety-sensitive applications. To address this, our fourth contribution proposes a risk-controlling thresholding strategy for anomaly scores that ensures finite-sample performance guarantees for any user-defined risk function, including false positive rates and F1-scores. This contribution builds upon the distribution-free Learn then Test framework and introduces two adaptive thresholds accounting for overlap between normal and anomalous score distributions. In addition, we develop a density-forecasting-based AD model using conditional normalising flows to support likelihood-based anomaly scoring.

Overall, these contributions advance the methodological foundations of deep AD and strengthen its applicability to safety-critical domains, paving the way for more reliable deployment in real-world systems.

Contents

List of Symbols **xii**

1 Introduction **1**
1.1 Why anomaly detection? . . . 2
1.2 Challenges in anomaly detection . . . 4
1.3 Thesis objectives and contributions . . . 6
1.4 List of publications . . . 10

2 Background **12**
2.1 Anomaly detection . . . 13
2.1.1 Formal definition of anomaly . . . 13
2.1.2 Anomaly threshold . . . 13
2.1.3 Anomaly score . . . 14
2.1.4 Anomaly, novelty or outlier? . . . 15
2.1.5 Different types of anomalies . . . 16
2.1.6 Evaluation metrics . . . 18
2.2 Data settings . . . 21
2.2.1 Supervised anomaly detection . . . 22
2.2.2 Semi-supervised anomaly detection . . . 22
2.2.3 Unsupervised anomaly detection . . . 23
2.3 Unsupervised anomaly detection approaches . . . 24
2.3.1 Classification-based methods . . . 24
2.3.2 Density-based methods . . . 26
2.3.3 Reconstruction-based methods . . . 27
2.3.4 Deep feature-based methods . . . 28
2.4 Related fields . . . 29

2.4.1 Hypothesis testing 29
2.4.2 Conformal prediction 32

3 Detecting Logical and Structural Anomalies 38
3.1 Introduction 39
3.2 The ULSAD framework for anomaly detection 41
3.2.1 Feature extractor 42
3.2.2 Detecting structural anomalies 43
3.2.3 Detecting logical anomalies 44
3.2.4 ULSAD algorithm overview 47
3.2.5 Anomaly detection and localisation 47
3.3 Experimental evaluation 50
3.3.1 Setup 50
3.3.2 Evaluation results 52
3.3.3 Ablation study 53
3.4 Memory and computational complexity 55
3.5 Limitations 56
3.6 Conclusion 56

4 Risk Estimator-based Semi-supervised AD 58
4.1 Introduction 59
4.2 Related work 60
4.3 Background on risk estimators 61
4.4 The proposed semi-supervised AD methods 63
4.5 Risk bounds 67
4.6 Experiments 69
4.6.1 Experiments with shallow rAD 69
4.6.2 Experiments with deep rAD 71
4.7 Discussion 75

5 Evidence-Based Test-time Adaptation Framework 76
5.1 Introduction 77
5.2 Related work 78
5.3 Background 79
5.4 EPHAD: An evidence-based post-hoc adjustment framework 81
5.4.1 Extension to score-based anomaly detection 82
5.4.2 An illustrative example 84
5.4.3 Determining the temperature parameter β 84
5.5 Experiments 86

5.5.1 Experiments on visual AD datasets . . . 86
5.5.2 Experiments on tabular AD datasets . . . 89
5.5.3 Experiments on industrial use case . . . 91
5.5.4 Ablation study . . . 92
5.6 Conclusion . . . 93

6 Detecting Anomalies in Irregular Image Sequences 95
6.1 Introduction . . . 96
6.2 A case study on detecting anomalous behaviours in CSP plants 98
6.2.1 Data description . . . 99
6.2.2 Data labelling . . . 101
6.2.3 General problem formulation . . . 104
6.3 Forecasting-based AD model . . . 104
6.3.1 Image encoder . . . 105
6.3.2 Context encoder . . . 105
6.3.3 Image decoder . . . 106
6.4 Experimental setup . . . 106
6.5 Results and discussion . . . 108
6.6 Ablation study . . . 109
6.7 Interpretability of `ForecastAD` . . . 111
6.8 Simulated dataset . . . 112
6.9 Deployment . . . 112
6.10 Conclusion . . . 114

7 Risk-Based Thresholding for Reliable Anomaly Detection 115
7.1 Introduction . . . 116
7.2 Background . . . 118
7.3 Reliable decision thresholds for AD . . . 120
7.4 Density-based AD model . . . 122
7.5 Experiments . . . 124
7.5.1 Experimental setup . . . 124
7.5.2 Results and discussion . . . 125
7.6 Deployment . . . 127
7.7 Simulated dataset . . . 129
7.8 Related work . . . 130
7.9 Conclusion . . . 130

8 Conclusion 132
8.1 Summary of contributions . . . 133
8.2 Future research directions . . . 134

8.2.1 Zero-shot anomaly detection using foundation models . 135
8.2.2 Synthetic anomaly generation 136
8.2.3 Conformal risk control for anomaly detection 137
8.2.4 Federated anomaly detection 138

Appendices **163**

A Detecting Logical and Structural Anomalies **164**
A.1 Implementation details 165
A.2 Extended results 167
A.2.1 Performance on MVTecLOCO: logical and structural AD 167
A.3 Extended ablations 175

B Risk estimator-based Semi-supervised Anomaly Detection **177**
B.1 Some definitions 178
B.2 Additional experiments 178
B.2.1 Additional experiments for shallow rAD 178
B.2.2 Additional experiments for deep rAD 178

C Evidence-Based Test-time Adaptation Framework **184**
C.1 Proofs 185
C.1.1 Proof of Proposition 5.4.1 185
C.2 Additional implementation details 186
C.2.1 Benchmark datasets 186
C.2.2 Details of the experiment using synthetic data 186
C.2.3 Computing evidence functions 187
C.2.4 Experimental setup 189
C.3 Extended results 190
C.3.1 Additional experiments on tabular datasets 190
C.3.2 Comparison against LOE and SoftPatch 190
C.3.3 Ablation on ϵ and β 193
C.3.4 Effect of test set size n 193

D Detecting Anomalies in Irregular Image Sequences **195**
D.1 Network architecture 196
D.2 Data generation 196
D.3 Sensitivity to data labelling 197
D.4 Additional results 198

E Risk-Based Thresholding for Reliable Anomaly Detection **200**

E.1 Ablation study 201
E.2 Details on dataset simulation 203
E.2.1 Description and performances. 203

List of Figures **204**

List of Tables **207**

List of Symbols

$\mathbb{R}_+$ — Positive real number.

$\mathbb{R}^d$ — d-dimensional space of real numbers.

$\mathcal{A}$ — Set of anomalous samples.

$|\mathcal{B}|$ — Cardinality of a set $\mathcal{B}$.

$\mathcal{X}$, $\mathcal{Y}$ — Input and output spaces.

$2^{\mathcal{Y}}$ — Power set of $\mathcal{Y}$.

X, Y — Input and output random variables.

$\mathbb{P}(\mathcal{A})$ — Probability of event $\mathcal{A}$.

P_{XY} — Joint distribution of random variables X and Y.

P_X — Marginal distribution of random variable X.

$\mathrm{P}_X^+ := \mathrm{P}_{X|Y=+1}$ — Conditional distribution of normal samples.

$\mathrm{P}_X^- := \mathrm{P}_{X|Y=-1}$ — Conditional distribution of anomalous samples.

f_X — Probability density function (PDF) of X.

p_Y — Probability mass function (PMF) of Y.

$\mathbb{E}_X[f(X)]$ — Expectation of $f(X)$ with respect to X.

$\mathcal{D}$ — Full dataset.

$\mathcal{D}_{\text{train}} \subset \mathcal{D}$ — Training split of the dataset $\mathcal{D}$.

$\mathcal{D}_{\text{val}} \subset \mathcal{D}$ — Validation split of the dataset $\mathcal{D}$.

$\mathcal{D}_{\text{test}} \subset \mathcal{D}$ — Test split of the dataset $\mathcal{D}$.

ϵ — The contamination factor, i.e., the proportion of anomalous samples.

$\boldsymbol{w}$ — Weight vector.

$\mathbf{W}$ — Weight matrix.

$\boldsymbol{\theta} \in \Theta$ — Trainable parameters belonging to some parameter space $\Theta \subseteq \mathbb{R}^d$.

$\mathcal{L}(\cdot)$ — Objective function.

$\nabla_{\boldsymbol{\theta}}$ — Gradient with respect to $\boldsymbol{\theta}$.

η — Learning rate.

$||\cdot||_2$ — The ℓ_2 norm.

CHAPTER 1

Introduction

1.1. Why anomaly detection?

> The thing that doesn't fit is the thing that's the most interesting: the part that doesn't go according to what you expected.
>
> Richard P. Feynman

An anomaly is "*an observation that deviates significantly from some concept of normality*" (Chandola et al., 2009; Pang et al., 2021; Ruff et al., 2021). Such observations have also been referred to as irregular, atypical, inconsistent, unexpected, and rare. Depending on the application context, such deviations may indicate errors, faults, fraudulent behaviour, or critical system failures. Anomaly detection (AD) is the task of identifying such anomalous observations using data-driven models and algorithms. AD is the basis of many critical applications, including cybersecurity (Ahmed et al., 2016; Hilal et al., 2022), healthcare (Tibshirani and Hastie, 2007; Fernando et al., 2021), finance, and industrial maintenance (Choi et al., 2022; 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.

This thesis is conducted as part of the Federated Learning and Augmented Reality for Advanced Control Centres (FLARACC) project, a collaboration between the University of Mons, the University of Namur, John Cockerill, IBA, and CETIC. To illustrate a representative application of anomaly detection (AD), we consider a use case from John Cockerill focused on predictive maintenance in a Concentrated Solar Power (CSP) plant. A CSP plant, shown in Figure 1.1, is designed to harness solar energy for large-scale electricity generation. Central to the plant's operation is the Thermal Solar Receiver, which serves as a high-temperature furnace that absorbs concentrated sunlight. Operating under extreme temperatures, these receivers are susceptible to a range of failures, including metal fatigue, corrosion, and tube blockages. Such faults can, in turn, compromise system efficiency, reduce component lifespan, and lead to costly downtime. Consequently, thermal images are captured multiple times per day to detect abnormal behaviours or anomalies that may indicate potential failures. However, manual inspection of thermal images is infeasible at scale due to the high volume and complexity of the data. In such scenarios, data-driven AD techniques provide a viable option for effectively processing and interpreting thermal imagery, thereby enabling continuous monitoring and real-time fault detection.

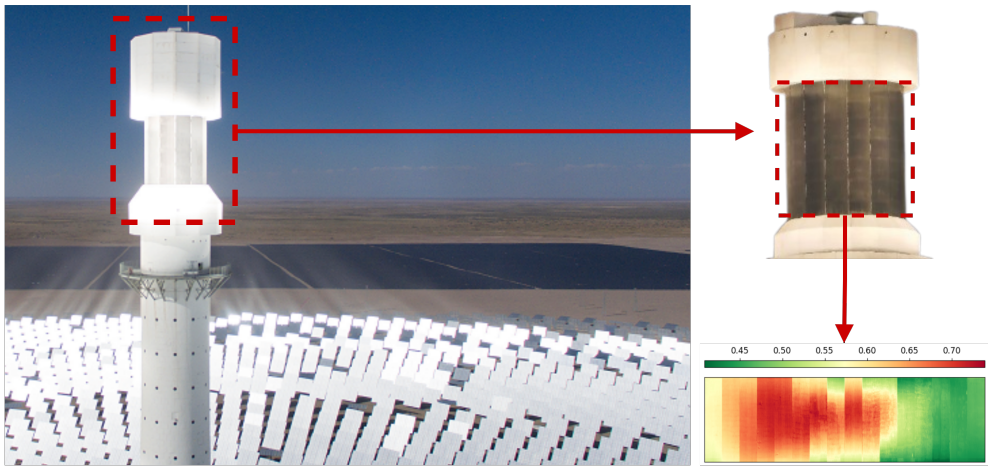


Figure 1.1: (Left) Illustration of a Concentrated Solar Power plant. (Right) Zoomed-in image of a thermal solar receiver with the thermal image captured using infrared cameras on its surface[1]. Our goal is to detect anomalous behaviours using such thermal images.

The core challenge of AD lies in learning a data-driven representation of what constitutes a "normal" given the application context. This inherently stems from the lack of prior knowledge about anomalous behaviours a priori (Ruff et al., 2021). This makes supervised methods, which require labelled data, largely unsuitable. Consequently, 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, probabilistic methods such as 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 representative features corresponding to normal samples for comparison.

In addition to technical challenges, anomaly detection in real-world applica-

[1]Source: https://www.johncockerill.com accessed on December 16, 2025

tions carries significant economic implications. For instance, in the case of CSP plants, undetected failures can lead to prolonged operation of faulty components, resulting in increased wear, reduced efficiency, and ultimately costly repairs or replacements. Moreover, in financial applications, failure to detect fraudulent transactions has direct monetary consequences. Therefore, timely and accurate detection of anomalies is not only critical for maintaining system reliability but also essential for minimising operational and economic losses.

This thesis explores the development of data-driven approaches for AD, with a particular focus on unsupervised and semi-supervised methods that generalise effectively from limited or unlabeled image data (Cao et al., 2024). The overarching goal is to design AD systems that are both accurate and reliable, with practical relevance to industrial applications such as those in CSP plants.

1.2. Challenges in anomaly detection

Binary classification (Hastie et al., 2009) is a fundamental problem in machine learning, where the goal is to predict a binary output given a set of input features. To achieve this, a model is trained to distinguish between two well-defined classes using a labelled dataset composed of representative, independently and identically distributed (i.i.d.) samples from the positive and negative classes. Once trained, the classifier aims to generalise its decision boundary to accurately classify previously unseen data. At first glance, AD appears similar to binary classification, as it also involves separating instances into two categories, namely, normal (negative class) and anomalous (positive class). However, despite this superficial resemblance, AD presents additional challenges (Pang et al., 2021; Ruff et al., 2021; Perini, 2024). Summarised below are some of the significant challenges that make AD inherently more difficult than conventional binary classification, and remain active areas of research.

Unsupervised or weakly supervised. Anomalies are inherently unknown until they are observed, and different classes of anomalies can exhibit distinct characteristics or distributions. This heterogeneity, combined with the absence of a universally accepted definition of what constitutes an anomaly, renders the task of data labelling highly non-trivial. Compounding this, anomalies are inherently rare in real-world datasets and simulating realistic anomalous behaviour is often infeasible or inappropriate, particularly in high-stakes domains (e.g., healthcare, autonomous systems), due to practical, ethical, or safety constraints. Thus, the labelled samples can be *partial/incomplete* (i.e.,

they do not span the entire set of anomaly class), *inexact* (i.e., coarse-grained labels), or *inaccurate* (i.e., some given labels can be incorrect) (Pang et al., 2021). Consequently, a central challenge in AD is to develop methods capable of learning robust and discriminative representations of both normal and anomalous patterns in the absence of reliable labels, i.e., within *unsupervised or weakly supervised* frameworks.

Extreme data imbalance. A defining characteristic of anomaly detection problems is the pronounced class imbalance as anomalies occur infrequently relative to normal instances (Chalapathy and Chawla, 2019; Ruff et al., 2021; Perini, 2024). This skew in class distribution leads to several complications. First, models trained under such an imbalance tend to be biased toward the majority class, often failing to recognise subtle yet critical deviations. Second, the cost of false negatives (i.e., missing anomalies) is typically much higher than that of false positives, especially in high-stakes domains like fault diagnosis or security monitoring. Finally, the rarity of anomalies restricts the size and diversity of training datasets, making it difficult to learn generalisable patterns. As collecting additional anomalous samples is often impractical, researchers must rely on domain assumptions, introducing further bias into the learning process.

Absence of representative samples. Anomalies are inherently diverse and can manifest in multiple, heterogeneous forms, each exhibiting distinct characteristics and statistical signatures. Importantly, the complete set of possible anomaly types is typically *unknown a priori*. For example, in a CSP plant, anomalous thermal patterns may result from a variety of failure modes, such as metal fatigue, corrosion, or tube blockages, each inducing different signatures in the sensor or image data. Since such failure types may occur infrequently or may not have yet been observed, it is highly unlikely that a dataset can capture all relevant forms of anomalous behaviour. This absence of comprehensive and representative anomalous samples severely limits the effectiveness of supervised learning approaches, which rely on well-defined class boundaries. It also complicates the development of generalizable models, as methods trained on a limited subset of anomaly types may fail to detect novel or rare events in practice. As a result, robust anomaly detection requires models that can generalise beyond the specific types of anomalies seen during training. Instead, AD methods predominantly learn expressive representations of normal behaviour, enabling the identification of diverse and previously unseen deviations.

Representation learning from high-dimensional data. With the growing volume of complex data, such as images, multivariate time series, and graphs, the problem complexity of AD is further increased. Traditional shallow AD methods struggle in such settings due to the *curse of dimensionality*, where the performance deteriorates as the number of input features grows (Erfani et al., 2016). A common strategy to address this involves reducing the data dimensionality by selecting a subset of features. However, identifying feature subsets that preserve the high-order, non-linear dependencies necessary for reliable anomaly detection remains a major challenge. Specifically, it encompasses identifying co-occurrences and long-range relationships that cannot be identified by simple local or pairwise comparisons. For instance, in images, anomalies may arise not from individual pixel intensities (low-order) but from unusual combinations of textures, shapes, or regions (high-order). Consequently, *deep learning*-based methods have gained attention due to their ability to learn effective representations directly from the data (Bishop and Nasrabadi, 2006). Deep neural networks can learn hierarchical, non-linear representations, making them particularly suitable for capturing the latent structure of complex data. Recent advancements in hardware acceleration, stochastic optimisation, and automatic differentiation have further facilitated the deployment of deep models on large-scale, heterogeneous datasets (Faust et al., 2018; Hoogeboom et al., 2022). Despite their representational power, deep learning-based AD methods introduce new challenges. These include defining appropriate learning objectives tailored for anomaly detection, ensuring computational efficiency for real-time or large-scale applications, and improving the interpretability of deep models. The *black-box* nature of deep learning hinders its adoption in high-stakes domains where transparency and trust are essential. Thus, while deep learning offers a promising direction for AD, several open challenges must be addressed for real-world applications.

Given the challenges above, it is evident that AD remains a fundamentally complex problem, particularly in real-world scenarios. These challenges underscore the critical need for continued investigation, particularly in the area of *deep AD*, to design models capable of learning expressive representations, handling uncertainty, and generalising to unseen and evolving anomaly patterns.

1.3. Thesis objectives and contributions

Motivated by the critical role of AD across a wide range of practical applications, this thesis aims to advance the field by making four key contributions

in deep AD that hinder the effectiveness of existing approaches, as outlined below. Each contribution is accompanied by an underlying research question and a summary of the contribution.

Contribution 1: Identifying different types of anomalies

Anomaly detection in real-world settings is a complex task due to the diverse nature and co-occurrence of different anomaly types. Traditionally, anomalies have been classified as point, contextual (or conditional), and group anomalies (Pang et al., 2021). Point anomalies refer to individual outliers, contextual anomalies are dependent on external conditions, and group anomalies involve sets of instances that collectively deviate from expected patterns but can be deemed as normal individually. More recent advances in deep learning have introduced two additional categories: low-level structural anomalies, which are subtle, localised irregularities in image features such as texture or edges, and high-level logical anomalies, which involve violations of semantic or geometrical constraints, such as missing, misplaced, or surplus components in visual data (Ruff et al., 2021; Bergmann et al., 2019, 2022). In practical industrial applications, such as automated inspection manufacturing plants, structural and logical anomalies frequently co-occur, making detection especially challenging.

In Chapter 3, we develop a unified method for the detection of both structural and logical anomalies, building on the Deep Feature Reconstruction (DFR) approach. To detect structural anomalies, we consider both magnitude and angular differences between the representations extracted using a pre-trained deep neural network and reconstructed by our method. To detect logical anomalies, we propose a novel attention-based loss for learning the logical constraints. Extensive empirical comparison with eight baseline methods across five widely adopted benchmark datasets demonstrates the effectiveness of our proposed method.

Contribution 2: Addressing training data contamination

Supervised learning is the least considered paradigm for AD, as acquiring large amounts of normal and anomalous samples is expensive and difficult. Consequently, the majority of research efforts have focused on *unsupervised* learning approaches. A common strategy in unsupervised AD involves training models to learn a compact representation of normal samples under the assumption that the training dataset is *clean*, i.e., free from anomalous instances (Ruff et al., 2021). However, in real-world applications, this assumption rarely holds (Das

et al., 2025; Hien et al., 2024; Qiu et al., 2022). For example, a dataset collected for industrial maintenance may already include unnoticed defects. These contaminations can significantly bias the learned representations, resulting in performance deterioration and reduced model reliability in distinguishing between normal and anomalous data.

We propose two novel approaches tailored to these questions. First, in Chapter 4, we frame anomaly detection as a semi-supervised binary classification problem. Here, the training dataset comprises a larger unlabelled set potentially containing anomalies and a smaller labelled set containing both normal and anomalous samples. Within this framework, we develop two risk-based AD methods: (i) a shallow method based on an unbiased risk estimator, and (ii) a deep learning-based method that employs a nonnegative risk estimator. To ensure robustness and avoid overfitting, we introduce a regularisation strategy for the shallow model that guarantees the nonnegativity of the empirical risk. Furthermore, we derive estimation error bounds and excess risk bounds for both risk minimisers, building upon results from Kiryo et al. (2017) and Niu et al. (2016).

Second, to address contamination at inference time, in Chapter 5 we introduce a test-time adaptation framework applicable to unsupervised AD models trained on contaminated data. This approach does not require access to additional labelled data. Instead, it leverages the prior captured by the AD model trained on the contaminated dataset. At test time, this is combined with the output of an auxiliary evidence function through exponential tilting. This mechanism enables the model to adapt its decision boundary dynamically in the presence of contamination. Comprehensive experimental evaluations across several standard AD benchmarks confirm the effectiveness of both proposed methods in improving AD performance under data contamination.

Contribution 3: Handling temporal features of the normal data for a real-world industrial application

In the context of the John Cockerill use case for CSP plants, as described in Section 1.1, the data exhibit several complex temporal features that pose significant challenges for AD. These include non-stationarity, temporal dependencies across consecutive observations, and strong daily seasonal patterns. Non-stationarity arises from variations in the underlying statistical properties of the data over time. Moreover, the data exhibits strong temporal correlations between consecutive thermal images, combined with recurring seasonal patterns driven largely by weather conditions. Accurately capturing these tem-

poral features is essential for learning the concept of normality. However, as the thermal images are captured over irregular intervals, it is further difficult to model the temporal features required for AD.

In Chapter 6, to address these issues, we develop a model that is capable of extracting meaningful representations for anomaly detection from thermal images while modelling the temporal features of the data. Specifically, we propose a forecasting-based approach, in which a deep sequence model is trained to predict the next thermal image conditioned on past observations, assuming normal behaviours. Anomalies are then detected based on the deviation between predicted and observed images. We further empirically demonstrate that our approach is well-suited for real-world AD scenarios.

Contribution 4: Determining risk-controlling anomaly thresholds with finite-sample performance guarantees

Deep learning-based AD methods are scalable and capable of learning rich representations from high-dimensional data without manual feature engineering. Nevertheless, a critical limitation of deep AD models is that they frequently yield unreliable and overly optimistic predictions (Nalisnick et al., 2019a). This issue is particularly problematic in safety-critical applications such as healthcare, fraud detection, and predictive maintenance, where model reliability is paramount. Consequently, there is growing scepticism regarding the deployment of deep AD systems in practice. A natural response to these concerns has been the development of methods for quantifying predictive uncertainty (Perini et al., 2021). While uncertainty estimates provide useful information, they do not directly inform practitioners about how to act upon individual predictions, which often leads to hesitation in adopting deep AD models even in cases where uncertainty is low (Perini, 2024). To address this issue, Perini and Davis (2023) introduced a learning-with-reject framework for unsupervised AD, based on an estimated measure of uncertainty called ExCeeD (Perini et al., 2021). It aims to find a constant rejection threshold to defer uncertain predictions to domain experts. This framework offers theoretical guarantees on the number of rejections, false positives and false negatives. However, it lacks statistical guarantees with respect to arbitrary user-defined risk functions.

In Chapter 7, we address this challenge in the context of predictive maintenance for CSP plants. We present a novel AD method based on density forecasting with conditional normalising flows, which models the likelihood of normal images given past thermal images and timestamps. Based on this deep AD model, we introduce adaptive thresholds that adjust for the overlap

between normal and anomalous score distributions by extending the machine-learning-with-abstention framework of Perini and Davis (2023).

Specifically, we propose a risk-controlling thresholding strategy for anomaly scores that provides finite-sample performance guarantees for any chosen risk function, such as the false positive rate or the F1-score. Our approach is built upon Learn then Test, a framework for distribution-free control of general risk (Angelopoulos et al., 2025).

1.4. List of publications

1. **Sukanya Patra** & Souhaib Ben Taieb (2025a). An Evidence-Based Post-Hoc Adjustment Framework for Anomaly Detection Under Data Contamination. In the *Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

    - **Code:** `https://github.com/sukanyapatra1997/EPHAD`.

2. Yorick Estievenart, **Sukanya Patra** & Souhaib Ben Taieb (2025b). Risk-Based Thresholding for Reliable Anomaly Detection in Concentrated Solar Power Plants. In the *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.

    - **Code:** `https://github.com/yoest/reliable-ad-csp`.

3. **Sukanya Patra** & Souhaib Ben Taieb (2024a). Revisiting Deep Feature Reconstruction for Logical and Structural Industrial Anomaly Detection. In the *Transactions of Machine Learning Research (TMLR)*.

    - **Code:** `https://github.com/sukanyapatra1997/ULSAD-2024`.

4. Le Thi Khanh Hien, **Sukanya Patra**, & Souhaib Ben Taieb. Anomaly detection with semi-supervised classification based on risk estimators (2024b). In the *Transactions of Machine Learning Research (TMLR)*.

    - **Code:** `https://github.com/LeThiKhanhHien/rAD`.

5. **Sukanya Patra**, Nicolas Sournac, & Souhaib Ben Taieb. Detecting Abnormal Operations in Concentrated Solar Power Plants from Irregular Sequences of Thermal Images (2024c). In the *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

- **Code:** `https://github.com/sukanyapatra1997/ForecastAD`.

6. **Sukanya Patra**, Le Thi Khanh Hien, & Souhaib Ben Taieb (2023). Anomaly detection in irregular image sequences for concentrated solar power plant. In the *European Symposium on Artificial Neural Networks (ESANN)*.

   - **Code:** `https://github.com/sukanyapatra1997/ForecastAD`.

[2] 7. David Vallmanya Poch, Yorick Estievenart, Elnura Zhalieva, **Sukanya Patra**, Mohammad Yaqub & Souhaib Ben Taieb (2025c). Segmentation-Guided CT Synthesis with Pixel-Wise Conformal Uncertainty Bounds. *arXiv Preprint arXiv:2503.08515, 2025.*

   - **Code:** `https://github.com/fabibombo/cbct2ct_translation`.

[2]Not included in this thesis.

CHAPTER 2

Background

2.1. Anomaly detection

In this section, anomaly detection is formally defined, and key concepts essential for the remainder of this thesis are introduced. The discussion encompasses the role of thresholding in AD, the various categories of anomalies described in the literature and commonly employed evaluation metrics.

2.1.1 Formal definition of anomaly

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 $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} := \{-1, +1\}$. Here, $Y = +1$ corresponds to the normal class, while $Y = -1$ represents the anomalous class. The conditional distribution of normal samples is $\mathrm{P}_{X|Y=+1}$ denoted as P_X^+ with PDF f_X^+. Likewise, the conditional distribution of anomalous samples is $\mathrm{P}_{X|Y=-1}$ denoted as P_X^-, with PDF f_X^-.

An anomaly is 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^+. Thus, we can define the *set of anomalies* $\mathcal{A}$ as

$$\mathcal{A} = \{x \in \mathcal{X} \mid f_X^+(x) \leq \lambda\}, \quad \lambda \geq 0, \tag{2.1}$$

where λ is a chosen threshold such that $\mathrm{P}_X^+(\mathcal{A})$ is *sufficiently small*.

2.1.2 Anomaly threshold

In practice, precisely specifying λ is challenging as the law of normality P_X^+ is rarely known. Consequently, the objective of anomaly detection to estimate the low-density regions in the data space $\mathcal{X}$ can be formally expressed as the problem of *density level set estimation* (Steinwart, 2011; Chen et al., 2017).

Definition 2.1.1 (α-density level set)**.** The set $C_\alpha := \{x \in \mathcal{X} \mid f_X^+(x) > \lambda_\alpha\}$ is defined as the α-density level set of the distribution P_X^+ if it satisfies the condition $\mathrm{P}_X^+(C_\alpha) \geq 1 - \alpha$. Here, $\alpha \in [0, 1]$ is the density level and λ_α is the corresponding threshold.

Considering the concentration assumption holds, there will always exist a level α and threshold λ_α, such that C_α exists and is bounded. Using C_α, we can define the corresponding anomaly detector $g_\alpha : \mathcal{X} \to \{+1, -1\}$:

$$g_\alpha(x) = \begin{cases} +1, & \text{if } x \in C_\alpha \\ -1, & \text{if } x \notin C_\alpha. \end{cases}$$

Each anomaly detection problem requires making various modelling choices and assumptions. As the value of α increases, the detector focuses only on the most likely regions under P_X^+. This is desirable when the cost of missing anomalies is high, such as in fraudulent transaction detection. However, it will create many false alarms, which are costly for online applications. On the contrary, as $\alpha \to 0$, the number of false alarms would reduce at the cost of missing some anomalies. Hence, there exists an inherent application-specific trade-off. In Chapter 7, we will discuss an approach for computing reliable decision thresholds for any chosen risk function which quantifies the expected performance of a model given the costs associated with various types of errors.

2.1.3 Anomaly score

A key challenge with the approach discussed above is its reliance on access to the true PDF $f_X^+(x)$ of the normal data distribution P_X^+. However, in practice, $f_X^+(x)$ is unknown. Consequently, it is approximated using a density estimator. Density estimation poses significant challenges, particularly in high-dimensional spaces or when data is scarce, and often incurs substantial computational cost. Fortunately, in the context of anomaly detection, the goal is 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* $s_a(x) : \mathcal{X} \to \mathbb{R}$, which directly assigns an anomaly score to a data point $x \in \mathcal{X}$, thereby quantifying its *degree of anomalousness* (Ruff et al., 2021). To complement this, the *inlier score function* is defined as $s_n(x) = -s_a(x)$, capturing the *degree of normality*, where higher values indicate that x is normal. Given the PDF $f_X^+(x)$, we can express the anomaly score as $s_a(x) = -\phi(f_X^+(x))$ where $\phi(\cdot)$ is a transformation commonly chosen to be the logarithm. Lastly, using the anomaly score

and a corresponding threshold $\lambda_s \in \mathbb{R}$, we can define the score-based anomaly detector $g_s : \mathcal{X} \to \{+1, -1\}$ as

$$g_s(x) = \begin{cases} +1, & \text{if } s_a(x) < \lambda_s, \\ -1, & \text{if } s_a(x) \geq \lambda_s. \end{cases}$$

2.1.4 Anomaly, novelty or outlier?

While the terms anomaly, novelty and outlier all refer to data samples that reside in a low-probability region under P_X^+, the literature often draws distinctions among them (Ruff et al., 2021). An instance is referred to as an anomaly if it lies in a low probability region under P_X^+ and is sampled from a distribution P_X^- that is substantially different from P_X^+. In contrast, an outlier is a rare or low-probability instance that still belongs to the same normal data distribution P_X^+. A novelty, on the other hand, corresponds to a low-probability instance that is sampled from a newly evolving region or mode when the distribution P_X^+ is non-stationary. To illustrate this distinction, if the distribution of dogs is considered normal, then a cat would be classified as an anomaly, a rare breed of dog would be classified as an outlier, and a newly emerging breed of dog would be regarded as a novelty.

In the context of a CSP plant, previously discussed, a failure mode such as tube blockage or corrosion would be regarded as an anomaly. The timely detection of such instances is of particular interest, as they are critical for ensuring the continued operation and maintenance of the plant. In contrast, an artefact appearing on a thermal image of a solar panel due to dust or dirt would be classified as an outlier. Such instances are not of interest for predictive maintenance, as they are best understood as noise and are usually removed during data pre-processing, a step commonly referred to as outlier removal. Furthermore, if a change in the thermal camera alters the appearance of the images, this would be considered a novelty. In this case, the inspection model would need to be adapted to account for the new normal operating condition.

Despite these distinctions, the approaches for detecting anomalies, outliers, and novelties share both technical and conceptual similarities, as they all involve the identification of instances residing in low-probability regions. Therefore, throughout the remainder of this thesis, and without loss of generality, all instances $x \in \mathcal{A}$ (2.1) will be referred to as anomalies.

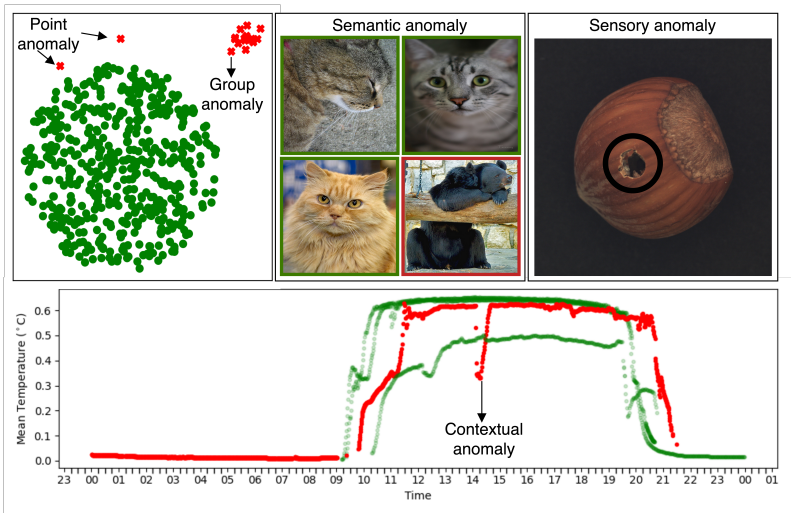


Figure 2.1: Illustration of different types of anomalies.

2.1.5 Different types of anomalies

Several types of anomalies have been reported in the literature (Chandola et al., 2009; Ruff et al., 2021). We highlight below the most commonly referred types of anomalies as shown in Figure 2.1.

- **Point or individual anomalies**: These refer to single anomalous instances that deviate significantly from a predefined notion of normal behaviour. Their identification does not require additional contextual information, and they can be detected by examining the instance in isolation. For example, in a CSP plant, the presence of a hot spot in a thermal image caused by a blocked tube can be considered a point anomaly. A considerable portion of the anomaly detection literature concentrates on this type of anomaly, with applications spanning healthcare, finance, industrial inspection and scientific discovery.

- **Group or collective anomalies**: These occur when a collection of instances $\{x_j \in \mathcal{X} \mid j \in J, J \subseteq \mathbb{N}\}$ collectively exhibit anomalous behaviour, even though each individual instance x_j may appear normal when considered in isolation. Here, J, is a set of indices that captures some relation or dependency among the points. Detection of such anomalies requires the analysis of interactions or dependencies within the group and the study of their joint distribution. For instance, in a CSP plant,

if a vertical heat exchanger tube becomes blocked with molten salt, its temperature remains elevated over an extended period until maintenance is performed. Thus, a single instance from that day, when viewed in isolation, would appear normal, as multiple similar instances exist in the data. However, the anomaly becomes evident when the particular day's images are considered as a group and compared across multiple days.

- **Contextual or conditional anomalies**: These are instances that are anomalous only under specific contextual conditions, such as time, space, or environmental factors. Identification requires the joint consideration of both the instance's features and its associated context. For example, in a CSP plant, a thermal image indicating a low surface temperature cannot be unambiguously classified as normal or anomalous without reference to its temporal context. If such an image is recorded at the beginning of plant operation, the observation is consistent with expected behaviour and thus considered normal. In contrast, if the same low-temperature image is obtained during the middle of the operational cycle, it indicates abnormal behaviour and is therefore classified as anomalous. In this case, the time of image acquisition serves as the critical contextual factor.

- **Sensory anomalies**: Also referred to as low-level or structural anomalies, these represent localised defects or imperfections. The term low-level refers to features at the lower end of the hierarchical structure of data, such as edges or textures in images, or words and characters in text. Examples include broken objects in manufacturing inspection or localised hot spots visible in thermal images of CSP plants. Detecting sensory anomalies generally requires modelling local dependencies or structural patterns in high-dimensional data.

- **Semantic anomalies**: In contrast to sensory anomalies, semantic anomalies correspond to high-level deviations. Here, high-level refers to more abstract representations in the hierarchical structure of data. In the case of images, this involves semantic concepts such as object class identity, while in text, it involves topics or themes. Within industrial anomaly detection, such anomalies are often termed logical anomalies, which arise when elements are missing, misplaced, present in excess, or in violation of geometrical or structural constraints (Bergmann et al., 2022). Detection of semantic anomalies, therefore, requires an understanding of long-range dependencies and the global structure within high-dimensional data.

2.1.6 Evaluation metrics

Given an anomaly detector, it is essential for any real-world application to measure its performance quantitatively. We discuss below some of the commonly used evaluation metrics in the literature.

Confusion matrix. Any anomaly detection model is subject to two primary types of classification errors. The first occurs when normal samples are incorrectly identified as anomalous. Such misclassifications are referred to as *false positives (FP)* or *Type I errors*. The second type arises when anomalous samples are mistakenly classified as normal, known as *false negatives (FN)* or *Type II errors*.

The relative importance of these errors is application-dependent, as there is no universal guideline to balance the risk associated with these errors. For example, in a medical healthcare scenario, false positives are generally less critical, since a healthy patient would merely undergo unnecessary treatment. Conversely, false negatives can have severe, potentially life-threatening consequences if a sick patient remains untreated. In contrast, within the context of a CSP plant monitoring application, false positives are typically more costly, as they lead to unnecessary manual inspections. False negatives, although initially less expensive, may result in significant long-term damage to equipment and reduced system longevity.

		Truth class	
		$Y = 1$	$Y = 0$
Predicted class	$\hat{Y} = 1$	TP	FP
	$\hat{Y} = 0$	FN	TN

Table 2.1: Confusion matrix

Therefore, the evaluation of an anomaly detection model must always be considered in relation to the specific application domain. The *confusion matrix*, shown in Table 2.1, provides a structured overview of classification outcomes. It is divided into four categories: false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN). Here, TP and TN denote correctly classified anomalous and normal instances, respectively. This framework is crucial for assessing model performance and for determining the optimal threshold λ (defined in Section 2.1.2) to convert continuous anomaly scores into class labels.

Based on the confusion matrix, several evaluation metrics can be defined.

These metrics quantify different aspects of performance by relating the predicted outcomes to the ground truth labels.

1. **True Positive Rate (TPR) / Recall / Sensitivity**: Proportion of actual positive samples that are correctly identified as positive, i.e.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}.$$

2. **False Positive Rate (FPR)**: Proportion of actual negative samples that are incorrectly classified as positive, i.e.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}.$$

3. **False Negative Rate (FNR) / Type II Error**: Proportion of actual positive samples that are incorrectly classified as negative, i.e.

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{P}} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{TPR}.$$

4. **True Negative Rate (TNR) / Specificity**: Proportion of actual negative samples that are correctly identified as negative, i.e.

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{FPR}.$$

5. **False Discovery Rate (FDR)**: Proportion of predicted positive samples that are actually negative, i.e.

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{Precision}.$$

6. **Precision**: Proportion of predicted positive samples that are truly positive, i.e.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{FDR}.$$

7. **Accuracy**: Proportion of correctly classified samples (both positive and negative) among all samples, i.e.

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}.$$

8. **F1 Score**: Harmonic mean of Precision and Recall, providing a balanced measure between the two, i.e.

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{TPR}}{\text{Precision} + \text{TPR}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}.$$

Area Under the Curve. In contrast to the threshold-dependent metrics introduced previously, often the costs or constraints of a specific application are not explicitly defined a priori. In such cases, it is necessary to evaluate the performance of an anomaly detection method across multiple thresholds λ, thereby capturing its general discriminative ability. To this end, the most widely employed metrics are the *Area Under the Receiver Operating Characteristic curve* (AUROC) and the *Area Under the Precision-Recall curve* (AUPR), with the latter also commonly referred to as *Average Precision (AP)*.

The Receiver Operating Characteristic (ROC) curve is constructed by plotting the *recall* (True Positive Rate) against the *False Positive Rate* while varying the decision threshold λ_s over the anomaly scores. The AUROC can be interpreted as the probability that the model ranks a randomly chosen anomalous instance higher than a randomly chosen normal instance. An important advantage of AUROC is its ease of comparison across different applications, since a random baseline always yields a score of 0.5, independent of class imbalance in the test set. However, AUROC is known to provide overly optimistic performance estimates in the presence of highly imbalanced datasets (Davis and Goadrich, 2006; Ahmed and Courville, 2020), which are characteristic of AD problems.

To address this limitation, practitioners frequently complement AUROC with AUPR. The Precision-Recall (PR) curve illustrates the trade-off between *precision* and *recall* at different thresholds, and AUPR summarises this trade-off into a single scalar value. Unlike AUROC, AUPR provides more meaningful and discriminative performance estimates in imbalanced settings. Nevertheless, Flach and Kull (2015) highlighted certain limitations of PR curves, such as the absence of a universal baseline, the lack of a convex Pareto front, and challenges in interpreting the area under the curve. Despite these shortcomings, the majority of the anomaly detection literature reports both AUROC and AUPR, as their combination provides a more comprehensive and robust evaluation of model performance.

Beyond image-level detection, anomaly localisation and segmentation tasks require evaluation at the pixel level. Several specialised metrics have therefore been proposed. The *Per-Region Overlap* (PRO) measures the degree of

overlap between predicted anomalous regions and the corresponding ground-truth regions. Its integral across thresholds, referred to as *AUPRO*, provides a threshold-independent summary analogous to AUROC and AUPR. Furthermore, to provide a fairer evaluation of performance on datasets with logical anomalies, *Saturated Per-Region Overlap* (sPRO) is introduced by Bergmann et al. (2022) as a generalisation of PRO, which prevents very large or small anomalies from overly influencing the score. While PRO weights ground-truth regions of different sizes equally, sPRO adds the saturation component to further refine this. Specifically, given a saturation threshold for each defect region, sPRO saturates the contribution of a region once the overlap exceeds the threshold, thus deeming the anomaly localisation "solved" if a minimal sufficient area is detected. Additionally, *pixel-level AUROC* is widely reported as a complementary measure, as it quantifies the anomaly detector's ability to distinguish between normal and anomalous pixels over all possible thresholds. Together, these metrics provide a fine-grained and application-relevant assessment of model performance in anomaly localisation and segmentation tasks.

2.2. Data settings

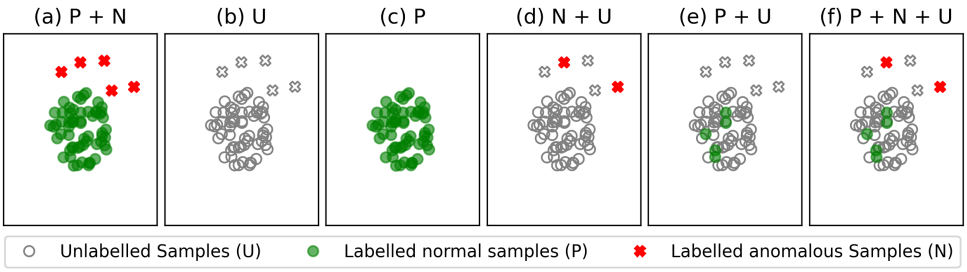


Figure 2.2: Illustration of common data settings in anomaly detection.

One of the fundamental considerations in designing anomaly detection algorithms is the level of supervision (Perini, 2024). Depending on the available supervision, researchers make specific assumptions about the anomalous data distribution P_X^-. In the absence of supervision, unsupervised methods rely on the concentration assumption discussed earlier in Section 2.1.1. This assumption states that normal samples are concentrated in high-density regions, whereas anomalous samples are distributed more uniformly across the data space (Tax, 2001). Such a formulation represents an uninformative prior over

the anomalous distribution. In contrast, when supervision is available, as in semi-supervised or fully supervised approaches, these restrictive assumptions can be relaxed, allowing for more informative priors to be employed. Although these assumptions play a critical role in addressing weak supervision or the complete lack of labels, they are rarely stated explicitly in the literature. Nonetheless, it is essential to note that each paradigm entails a trade-off between practicality, robustness, and detection accuracy. A summary of the main data settings is provided below and illustrated in Figure 2.2. For extensive discussions, we refer to (Ruff et al., 2021; Chandola et al., 2009).

2.2.1 Supervised anomaly detection

In fully supervised anomaly detection (Figure 2.2-(a)), both labelled normal and anomalous samples are assumed to be available. This setting is typically formulated as a supervised binary classification problem, where the objective is to learn a model that separates the two classes (Chandola et al., 2009). Despite its apparent simplicity, several challenges arise in this setting. First, anomalous samples are rare, resulting in highly imbalanced datasets. Acquiring additional anomalous samples is often infeasible, as replicating failure scenarios is costly or impractical in most real-world applications. Moreover, synthetically generated anomalies typically do not reflect the complexity of anomalies encountered in practice (Perini et al., 2025). Second, anomalies are heterogeneous in nature, and their distribution evolves over time. This is particularly critical in adversarial domains such as fraud detection (Dastidar et al., 2024; Abdallah et al., 2016), where malicious actors continuously develop novel strategies to evade detection. Due to these challenges, specialised supervised AD methods are scarce, and standard classifiers such as Random Forests or deep neural networks are often employed. However, these models are not tailored for AD and generally perform poorly in this setting (Görnitz et al., 2013).

2.2.2 Semi-supervised anomaly detection

Semi-supervised approaches provide a compromise between supervised and unsupervised methods. They assume access to a limited number of labelled instances in combination with a larger pool of unlabelled data. In practice, the labelled dataset often contains more normal than anomalous samples, since anomalies are scarce and difficult to label. The labelling process itself is challenging due to the absence of a universal definition of anomaly and the heterogeneity of anomalous patterns. As a result, the labelled data may be *partial or incomplete* (covering only a subset of anomaly classes), *inexact* (coarse-grained

labels), or *inaccurate* (containing incorrect annotations).

Depending on the availability of labels, different subcategories of semi-supervised AD have been proposed. If only normal or only anomalous labelled samples are available, the problem is formulated as a PU (positive-unlabelled) (Figure 2.2-(d)) or NU (negative-unlabelled) (Figure 2.2-(e)) setup, respectively. The PU setup is more common in practice, as verifying normal samples is relatively straightforward, given their abundance. A special case of learning from PU samples, known as Learning from Positive and Unlabelled Examples (LPUE), has therefore been widely applied in AD (Bekker and Davis, 2020; Chandola et al., 2009). However, it is well established that even a small number of labelled anomalous samples can substantially improve performance (Görnitz et al., 2013). Consequently, semi-supervised methods that leverage both normal and anomalous labelled data, in addition to unlabelled data, have shown highly promising results (Han et al., 2022; Ruff et al., 2021, 2020; Hien et al., 2024). This is referred to as a PNU (positive-negative-unlabelled) setting (Figure 2.2-(f)).

2.2.3 Unsupervised anomaly detection

In unsupervised anomaly detection, no labelled data are assumed to be available (Figure 2.2-(b)). This setting is particularly prevalent in practice due to the difficulties associated with acquiring labelled anomalies. A common assumption in this case is that all training samples are i.i.d. drawn from the normal data distribution (Figure 2.2-(c)). In other words, it is assumed that the training set is clean, or contains only a negligible fraction of anomalies, which allows the model to prioritise the normal instances (Wang et al., 2019). In practice, however, this assumption is rarely satisfied, as the training data are typically affected by noise and contamination.

Noise refers to inherent randomness in the data, such as measurement noise, which may be irreducible. Moreover, too much noise has an adverse effect on the learning process. A common assumption, however, is that the noise is unbiased and spherically symmetric. In addition, contamination, i.e., the presence of undetected anomalous samples in the training set, violates the assumption of clean data and biases the learned model. In this case, a more realistic assumption is therefore that the data are sampled from a mixture distribution $\mathrm{P}_X^{\mathrm{u}}$ with density $f_X^u(x)$ (Huber and Ronchetti, 2011; Huber, 1992), containing both normal and anomalous components. Considering the contamination

factor ϵ, the mixture distribution can be expressed as

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

As the contamination factor ϵ increases, models trained on x_i are i.i.d. samples from $\mathrm{P}_X^{\mathrm{u}}$ become increasingly biased, leading to systematic misclassification of anomalous samples as normal (Qiu et al., 2022; Yoon et al., 2022). Both noise and contamination must therefore be carefully accounted for in the development of robust unsupervised anomaly detection methods.

2.3. Unsupervised anomaly detection approaches

In this thesis, the focus is on methods that can operate under weak supervision or unsupervised settings, while remaining robust to noise and contamination. These represent the most realistic conditions in applications such as thermal image-based anomaly detection for CSP plants. As discussed previously, to account for the lack of supervision, unsupervised methods leverage the concentration assumption. Consequently, 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. Four main categories of unsupervised anomaly detection are described below. We discuss both traditional shallow AD methods that use shallow feature maps with manual feature engineering and deep AD approaches that leverage deep neural networks to extract hierarchical feature maps without manual intervention.

2.3.1 Classification-based methods

Key idea. *The goal of one-class classification methods (Ruff et al., 2018; Tax and Duin, 2004) is to learn a decision boundary encompassing all the normal samples. Samples lying outside the boundary are considered anomalous.*

Traditional shallow AD approaches. Anomaly detection can be formulated as a one-class classification problem (Tax, 2001; Khan and Madden, 2014). Classification-based techniques such as the One-Class Support Vector Machine (OC-SVM) (Schölkopf et al., 2001) directly estimate a decision boundary that separates normal and anomalous samples. However, this task is challenging in practice, since normal samples vastly outnumber anomalous samples in most applications. Support Vector Data Descriptor (SVDD) (Tax

and Duin, 1999; Tax, 2001; Tax and Duin, 2004) addresses this by constructing a tight spherical boundary. However, a spherical model can only approximate limited distributions, such as an isotropic Gaussian. For improved generalisation, elliptical boundaries (Rousseeuw, 1985; Rousseeuw and Driessen, 1999) have been proposed.

Models like OC-SVM and SVDD can be further extended using kernels to define non-linear decision boundaries. Kernel functions implicitly map data into a high-dimensional feature space where linear separation is possible. Currently, many variants of the kernel-based one-class classifiers have been proposed in the literature, including Kernel Fisher Discriminants (Roth, 2004, 2006), Bayesian Data Descriptors (Ghasemi et al., 2012), Multi-sphere SVDD (Gornitz et al., 2018), and group anomaly detection with OC-SVM (Muandet and Schölkopf, 2013).

Deep AD methods. Although the kernel-based variants SVDD and OC-SVM significantly improve the model's expressivity, they still rely on manual kernel selection. Hence, the focus has shifted lately to learning feature maps using neural networks. Some of the prominent approaches include DeepSVDD (Ruff et al., 2018) and deep OC-SVM variants (Erfani et al., 2016). Despite the deep methods requiring a massive amount of data for generalisation, the capability to parallelise makes the deep variants capable of scaling to bigger and more complex datasets. However, the key concern with such deep models is ensuring that the feature maps do not collapse to a constant value. Several solutions have been proposed to address this problem in the literature, such as architectural constraints (Ruff et al., 2018), freezing of the embedding layers (Erfani et al., 2016; Oza and Patel, 2019; Ruff et al., 2019), and integrating some manifold assumption (Goyal et al., 2020). Extensions include transfer learning (Reiss et al., 2021) and adversarial learning (Sabokrou et al., 2018).

Anomalous labelled samples can be integrated into one-class classification. These may be artificial, auxiliary, or true negative samples (real anomalous samples), each having different degrees of informativeness. While artificial samples are drawn from an assumed distribution, auxiliary negative samples are collected from publicly available data sources to incorporate domain knowledge. Auxiliary samples are thus more likely to resemble real-world data samples and are therefore more suitable than artificial samples. Outlier Exposure (Hendrycks et al., 2019) leverages such auxiliary negative samples and has been successfully applied in specific domains. Nevertheless, true negative data points are the most informative and can significantly improve performance even in considerably smaller amounts (Tax, 2001; Görnitz et al., 2013). Semi-

supervised extensions, such as DeepSAD (Ruff et al., 2020), utilise a small labelled subset in conjunction with larger unlabeled data. The unlabelled data consists of both normal and anomalous samples. PU learning methods can extract informative anomalies from unlabelled data using clustering (Chaudhari and Shevade, 2012), distance-based prototypes (Zhang and Zuo, 2009), or density estimation (He et al., 2020). Other PU learning methods consider unlabeled data for learning from noisy anomalous samples using approaches such as sample reweighing (Menon et al., 2015) and label cleaning (Scott, 2015).

2.3.2 Density-based methods

Key idea. *The goal of density-based methods (Gudovskiy et al., 2022; Yu et al., 2021) is to model the probability distribution of normal samples. Samples in low-density regions are considered anomalous.*

Traditional shallow AD approaches. Classical parametric methods (Yang et al., 2024; Ruff et al., 2021) assume a specific distribution for the normal data, such as the multivariate Gaussian distribution. Then, the distribution parameters are estimated using the training data. During inference, anomaly scores are computed using distance measures, such as the Mahalanobis distance (Leys et al., 2018), between the sample and the estimated normal data distribution. Non-parametric methods relax distributional assumptions and can thus learn any arbitrary complex distribution. Prominent examples of non-parametric methods include histogram-based approaches (Van Ryzin, 1973; Kind et al., 2009; Xie et al., 2012), Kernel Density Estimation (KDE) (Parzen, 1962) and Gaussian Mixture Models (GMMs) (Ruff et al., 2021). Although these classical methods work well in low dimensions, they face scalability issues in high-dimensional spaces due to the curse of dimensionality (Ruff et al., 2021). To overcome this, deep density-based AD methods have been proposed in the literature.

Deep AD methods. Neural generative models, such as VAE (Kingma and Welling, 2014), Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Normalising Flows (Rezende and Mohamed, 2015a), are deep learning-based methods that learns to map a predetermined latent distribution to the input data distribution based on the training dataset. The most common assumption for the latent distribution is an isotropic Gaussian. As VAEs and GANs cannot estimate the exact likelihood of a data point, for anomaly detection, the reconstructed sample generated by the VAE or the discriminator network in GANs is used. On the contrary, normalising flows allow for the exact calculation of the likelihood using the change of variable formula. Re-

cent AD methods such as CFlow (Gudovskiy et al., 2022) and FastFlow (Yu et al., 2021) further build on normalising flows. However, recent studies have shown that normalising flow often fails to detect anomalies and assign a higher likelihood to them (Nalisnick et al., 2019a,b; Kirichenko et al., 2020).

Finally, Energy-Based Models (EBMs) are statistical models where the density of a point is expressed as an energy function (Ngiam et al., 2011). While exact density evaluation is intractable, energy scores correlate with likelihood and can be used for anomaly detection (Zhai et al., 2016). Moreover, the performance and robustness of deep classifiers have also been improved by considering an energy-based training framework (Grathwohl et al., 2020).

2.3.3 Reconstruction-based methods

Key idea. *The key assumption employed in reconstruction-based methods (Gong et al., 2019) is that encoder-decoder models trained on normal samples will perform poorly for anomalous samples. Hence, the model's performance can serve as an indicator in anomaly detection tasks.*

Deep encoder-decoder models, including AE or VAE-based approaches, are widely used. To prevent identity mapping, structural assumptions about the data-generating process are imposed. One such assumption is the *Manifold Assumption*, which assumes that the observed data reside in a lower-dimensional manifold within the data space (Bengio et al., 2013). Methods that build on the manifold assumption restrict the encoded space to a lower dimensionality than the actual data space. Such a setup is often employed in other unsupervised tasks, such as manifold learning, dimensionality reduction, and representation learning. A model trained with a reconstruction objective is expected to extract critical features or patterns from the data. Consequently, a model trained on normal data points can accurately reconstruct normal samples from their compressed representation while incurring considerable loss for anomalous points (Ruff et al., 2021). Advanced strategies include memory-based reconstruction (Gong et al., 2019), adaptive architectures (Lai et al., 2020), and partial reconstruction (Nguyen et al., 2019; Yan et al., 2021). Recent approach DRÆM (Zavrtanik et al., 2021) trains a discriminative network alongside the reconstruction network to localise anomalies without requiring any further post-processing steps. Generative models, such as GANs, are also widely employed for anomaly detection, as the discriminator inherently calculates reconstruction loss for samples Zenati et al. (2018). Another approach is sparse representation modelling (Bruckstein et al., 2009). Here, it is assumed that normal samples can be expressed sparsely, while anomalies require dense

representation (Adler et al., 2013; Li et al., 2017).

Another prominent assumption is the *Prototype Assumption*, which states that normal samples can be represented by a finite set of prototypes. Primarily used in clustering-based approaches, the reconstruction objective attempts to learn a function that encodes the data points and maps them to a discrete number of prototypes, unlike the continuous function learned under the manifold assumption. Then the reconstruction loss, i.e., the difference between the data point and the mapped prototype, is used as the anomaly score. Prominent methods include k-means (Dhillon et al., 2004), as well as GMM-based clustering with a reconstruction objective defined in terms of Mahalanobis distance to each of the GMM clusters (Amruthnath and Gupta, 2018).

2.3.4 Deep feature-based methods

Key idea. *The features of normal samples extracted using a pre-trained network are utilised in deep feature-based approaches (Roth et al., 2022). These approaches either learn a compact representation of the normal features or store them in a memory bank. During inference, anomalies are detected by comparing the features of the test sample against the learned or stored normal features.*

There are mainly three different types of methods which utilise feature embeddings from a pre-trained deep neural network: *memory bank* (Defard et al., 2021; Roth et al., 2022; Lee et al., 2022), *student-teacher* (Zhang et al., 2024; Batzner et al., 2024), and *density-based* (Gudovskiy et al., 2022; Yu et al., 2021). *Memory bank*-based methods extract features from normal images and store them during the training phase. During the testing phase, the feature of a test image is used as a query to match the stored normal features. There are two main constraints in these methods: *how to learn useful features* and *how to reduce the size of the memory bank*. While PatchCore (Roth et al., 2022) introduces a coreset selection algorithm, CFA (Lee et al., 2022) clusters the features in the memory bank to reduce the size of the memory bank. Nonetheless, the performance of the *memory bank* methods heavily depends on the completeness of the memory bank, requiring a large number of normal samples. Furthermore, the memory size, which is related to the number of training samples, makes these methods unsuitable for large datasets or very high-dimensional images.

In the *student-teacher* approach, the student network learns to extract features from normal images, similar to those of the teacher model. For anomalous

images, the features extracted by the student network differ from those of the teacher network. Batzner et al. (2024) propose to use an autoencoder model in addition to the student network to identify logical anomalies. To leverage the multiscale feature from the teacher network for detecting anomalies at various scales, Deng and Li (2022) propose Reverse Distillation. Zhang et al. (2024) extended it by proposing the use of two student networks to address structural and logical anomalies. Shi et al. (2021) proposes to learn a deep neural network for learning to reconstruct the features of the normal images extracted using the pre-trained backbone.

For *density-based* methods, a model is first trained to learn the distribution of features obtained from normal samples. Then, during inference, anomalies are detected based on the likelihood of features extracted from the test images. PaDiM (Defard et al., 2021) uses a multivariate Gaussian to estimate the density of the features corresponding to the samples from the normal class, while FastFlow (Yu et al., 2021) and CFLOW (Gudovskiy et al., 2022) utilise normalising flows.

2.4. Related fields

2.4.1 Hypothesis testing

Hypothesis testing constitutes one of the central pillars of statistical inference, providing a rigorous framework for addressing questions about data and assessing competing explanations (James et al., 2013). The process begins by formulating two mutually exclusive hypotheses: the *null hypothesis*, denoted H_0, and the *alternative hypothesis*, denoted H_a. The null hypothesis typically represents a baseline assumption, most often the absence of an effect or difference, whereas the alternative hypothesis asserts the presence of an effect or deviation from H_0. Importantly, these hypotheses are treated asymmetrically. Rejection of H_0 based on the observed data constitutes evidence in favour of H_a. However, failure to reject H_0 cannot be interpreted as definitive evidence for its truth, since such an outcome may reflect insufficient data rather than the validity of the null hypothesis itself.

To formally evaluate the plausibility of H_0, a *test statistic* T is computed to quantify the degree of agreement between the observed data and the null hypothesis. The form of the test statistic depends on both the data type and the specific hypothesis under consideration. Large values of T, in absolute terms, signal greater incompatibility with H_0, but a key challenge lies in de-

termining the threshold beyond which rejection is warranted. This difficulty is addressed through the *p-value*, defined as the probability of observing results as extreme as, or more extreme than, those observed, under the assumption that H_0 is true. The p-value provides a standardised measure on the unit interval $[0, 1]$, thereby converting the test statistic into an interpretable decision-making quantity. Small p-values indicate that the observed data would be highly unlikely under H_0, leading to its rejection in favour of H_a.

The calculation of a p-value relies on the null distribution of the test statistic, i.e., the distribution of T under the assumption that H_0 holds. Depending on the context, this distribution may follow a normal distribution, a t-distribution, a χ^2-distribution, or an F-distribution, among others. In practice, rejection thresholds are application-dependent. Nonetheless, in most cases, H_0 is rejected whenever the p-value is less than 0.05. This threshold implies that, if the null hypothesis were true, the probability of observing a test statistic at least as extreme as the one calculated is at most 5%.

Finally, hypothesis testing is subject to two forms of error. A *Type I error* occurs when H_0 is rejected despite being true, whereas a *Type II error* occurs when H_0 is not rejected despite being false. These errors represent fundamental trade-offs in hypothesis testing and are central to the design and interpretation of statistical procedures.

2.4.1.1 Family-wise error rate

	H_0 is True	H_0 is False	Total
Reject H_0	V	S	R
Do not Reject H_0	U	W	$m - R$
Total	m_0	$m - m_0$	m

Table 2.2: Summary of outcomes when testing m null hypotheses.

Hypothesis testing, as discussed above, provides a way to control Type I errors by rejecting the null hypothesis when the p-value is substantially small, e.g., 0.05. Specifically, in this case, there is no more than 5% chance of rejecting H_0 given it is true. However, in practice, we seldom perform a single hypothesis test. Modern research often involves testing many hypotheses simultaneously, for instance, in genomics or neuroimaging studies, where thousands of comparisons are made at once. This multiple testing creates a problem as the more tests are performed, the chances of incorrectly rejecting at least one hypothesis increase drastically. This notion can be formalised using the notion

of *family-wise error rate* (FWER). Consider Table 2.2 which summarises the result of m independent hypothesis tests where the null hypotheses are denoted as $H_{01}, \cdots, H_{0m}$. Here V, S, U and W refer to the number of Type I errors (False Positives), True Positives, True Negatives and Type II errors (False Negatives), respectively. Then the FWER is expressed as

$$\text{FWER} = \mathbb{P}(V \geq 1).$$

Now, if we consider the null hypothesis is rejected when the p-value is below α, the FWER is

$$\begin{aligned}
\text{FWER} &= 1 - \mathbb{P}(V = 0) \\
&= 1 - \mathbb{P}(\bigcap_{i=1}^{m} \{H_{0i} \text{ is not falsely rejected}\}) \\
&= 1 - \prod_{i=1}^{m} (1 - \alpha) = 1 - (1 - \alpha)^m.
\end{aligned}$$

Thus, performing $m = 1000$ tests with an $\alpha = 0.05$ results in an FWER of $1 - (1 - 0.05)^{1000} = 0.99$. In other words, at least one Type I error is virtually guaranteed to occur. This is highly undesirable in large-scale testing scenarios and motivates the development of multiple testing corrections.

2.4.1.2 The Bonferroni method

Building upon the discussion in the previous section, consider the setting where m hypothesis tests are to be performed simultaneously. Let A_j denote the event that a Type I error is committed for the jth null hypothesis. The FWER can then be expressed as

$$\begin{aligned}
\text{FWER} &= \mathbb{P}(\bigcup_{j=1}^{m} A_j) \\
&\leq \sum_{j=1}^{m} A_j, \quad (\text{as } \mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)).
\end{aligned}$$

The *Bonferroni method*, also known as the *Bonferroni correction*, addresses this issue by adjusting the significance level used for each individual test. Specifically, the rejection threshold is updated to α/m, ensuring that $\mathbb{P}(A_j) \leq \alpha/m$. Consequently, the FWER is bounded by

$$\text{FWER} \leq \sum_{j=1}^{m} A_j = m \times \frac{\alpha}{m} = \alpha, \tag{2.2}$$

thereby guaranteeing control of the family-wise error rate at the desired level α, regardless of the number of tests performed.

To illustrate, consider again the case of $m = 1000$ simultaneous hypothesis tests. With an overall significance level of $\alpha = 0.05$, the Bonferroni correction requires rejecting a null hypothesis only if its p-value falls below $0.05/1000 = 0.00005$. The simplicity of this adjustment, both in application and interpretation, explains its widespread use in practice.

However, the Bonferroni method is well known for its conservativeness. Because the inequality in (2.2) is often loose, the actual FWER tends to be substantially below the nominal level α. While this ensures strong error control, it also increases the likelihood of Type II errors. In other words, true effects may remain undetected because the rejection threshold is set too stringently. Less conservative methods, in contrast, can maintain control of the FWER while allowing more rejections of false null hypotheses. Alternative approaches to controlling the FWER include Holm's step-down procedure, Tukey's method, and Scheffé's method. For a detailed discussion on these approaches, we refer the readers to James et al. (2013, Chapter 13).

2.4.2 Conformal prediction

Conformal prediction (CP) (Vovk et al., 1999), also known as conformal inference, provides a general framework for constructing prediction sets using any underlying predictive model, with a finite-sample coverage guarantee under the *exchangability assumption* which is defined as follows:

Definition 2.4.1 (Exchangeability)**.** Given a finite sequence of random variables $(Z_1, \cdots, Z_n)$, it is said to be exchangeable if the joint probability distribution is invariant to any permutation of the indices, i.e.,

$$Z_1, \cdots, Z_n \overset{d.}{=} Z_{\pi(1)}, \cdots, Z_{\pi(n)},$$

where π denotes a permutation.

For a test point $(X_{\text{test}}, Y_{\text{test}})$, the objective is to construct a prediction set $\mathcal{C}(X_{\text{test}}) \subseteq \mathcal{Y}$ such that

$$\mathbb{P}(Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})) \geq 1 - \alpha, \tag{2.3}$$

for a user-specified error level $\alpha \in (0, 1)$. This guarantees that the true label lies in the prediction set with probability at least $1-\alpha$. The property is referred to as *marginal coverage guarantee*, since the probability is marginalised over

both the calibration set and the test instance. Among the different approaches within this framework, the most commonly applied method is *split conformal prediction* (SCP) due to its computational efficiency. Alternative variants include full conformal and cross-conformal prediction. For further details, we refer to Angelopoulos and Bates (2021) and Fontana et al. (2023).

General framework of SCP. SCP can be applied to any predictive model and any heuristic notion of uncertainty obtained from it to derive rigorous guarantees of uncertainty. The four steps of the general framework of SCP given an input x are summarised below:

1. Using the pre-trained model, determine a heuristic notion of uncertainty.
2. Formulate a conformity score function $s_c(x, y) \in \mathbb{R}$, which is larger when the model output and y do not align.
3. Compute $\hat{q}$ as the $\lceil (n+1)(1-\alpha) \rceil / n$ empirical quantile of $s_c^{(1)}, \cdots, s_c^{(n)}$, where $s_c^{(i)} = s_c(X_i, Y_i)$, i.e,
$$\hat{q} = \text{quantile}\left(s_c^{(1)}, \cdots, s_c^{(n)}; \frac{\lceil (n+1)(1-\alpha) \rceil}{n} \right).$$
4. Lastly, obtain the prediction set as:
$$\mathcal{C}(X_{\text{test}}) = \{ y \in \mathcal{Y} \ : \ s_c(X_{\text{test}}, y) \leq \hat{q} \},$$
which satisfies the property in (2.3).

Here, $\mathcal{C} : \mathcal{X} \to 2^{\mathcal{Y}}$ is a set-valued function, assigning each test data point a set of admissible class labels, which depends on the score function $s_c(x, y)$. Importantly, the size of $\mathcal{C}(X_{\text{test}})$ adapts to the level of uncertainty. It is small when the model is confident and larger when the task is difficult or the model is uncertain. Thus, the cardinality of the prediction set also quantifies the predictive uncertainty. Given the exchangeability assumption, the marginal coverage guarantee of SCP is formalised as in Theorem 2.4.2.

Theorem 2.4.2 (Marginal coverage guarantee (Vovk et al., 1999))**.** *Given* $\{(X_i, Y_i)\}_{i=1}^n$ *and* (X_{test}, Y_{test}) *are drawn exchangeably from some distribution,* $\hat{q}$ *and* $\mathcal{C}(X_{test})$ *are defined as above, the following holds:*

$$\mathbb{P}(Y_{test} \in \mathcal{C}(X_{test})) \geq 1 - \alpha.$$

The proof can be found in (Angelopoulos and Bates, 2021, Appendix D). It is important to note that while the marginal coverage guarantee in Theorem 2.4.2

is always satisfied, the choice of the score function $s_c(x, y)$ plays a critical role in determining the usefulness of the prediction set. Thus, designing an informative score function for various problem types, such as classification and regression, is an active area of research.

Example of SCP for classification. Let us consider a classification setting where a predictive model $\hat{f}$ has already been fitted. Using a separate set of *calibration data*, conformal prediction constructs a set of class labels, termed the *prediction set*, for a new test instance. To formalise this, let the input space $\mathcal{X}$ consist of images and the output space $\mathcal{Y}$ be a finite set of K class labels. The classifier $\hat{f} : \mathcal{X} \to [0,1]^K$ is assumed to be a neural network producing softmax values, which can be interpreted as estimated class probabilities. These softmax values provide a heuristic notion of uncertainty. However, in practice, the predictive model may be miscalibrated due to overfitting or underfitting. Thus, the calibration set plays a crucial role in SCP. Consider we have access to n calibration samples $(X_1, Y_1), \ldots, (X_n, Y_n)$, drawn independently from the same distribution as the training and test data. Then, for a new test point $(X_{\text{test}}, Y_{\text{test}})$, the objective is to construct a prediction set $\mathcal{C}(X_{\text{test}}) \subseteq \{1, \ldots, K\}$ such that it satisfies (2.3).

The construction of $\mathcal{C}(X_{\text{test}})$ proceeds by defining a *conformity score function* $s_c(X_i, Y_i) = 1 - \hat{f}(X_i)_{Y_i}$ as one minus the softmax score $\hat{f}(X_i)_{Y_i}$ assigned to the true class label Y_i that will be evaluated over the calibration set. Intuitively, the score $s_c^{(i)} = s_c(X_i, Y_i)$ is small when the model is confident and correct, and large when the prediction is uncertain or incorrect. Then, $\hat{q}$ is computed as the empirical quantile of $\{s_c^{(1)}, \ldots, s_c^{(n)}\}$ at level $\lceil (n+1)(1-\alpha) \rceil / n$. Finally, for a test instance X_{test}, the prediction set is created as $\mathcal{C}(X_{\text{test}}) = \{y \in \{1, \ldots, K\} : \hat{f}(X_{\text{test}})_y \geq 1 - \hat{q}\}$. This construction guarantees that (2.3) holds, independently of the predictive model and the data distribution.

2.4.2.1 Conformal anomaly detection

CP can be extended beyond supervised learning to unsupervised AD, where the aim is to construct an anomaly detector function $g : \mathcal{X} \to \{+1, -1\}$ with error control guarantee (Angelopoulos et al., 2024). In this setting, the calibration dataset is assumed to be *clean*, meaning it contains no anomalous instances. Furthermore, we assume access to an AD model that assigns each sample an anomaly score $s_a : \mathcal{X} \to \mathbb{R}$, with high values for anomalous instances and low values for normal instances. The procedure follows the general SCP framework described previously, with the distinction that, in the unsupervised

setting, the conformity score depends solely on the input instance. Specifically, we use the anomaly score directly as the conformity score. We compute the anomaly scores for all the calibration samples $\{s_a^{(1)}, \cdots, s_a^{(n)}\}$. Then, the conformal threshold $\hat{q}$ is computed as the $\lceil (n+1)(1-\alpha) \rceil / n$ empirical quantile of $s_a^{(1)}, \cdots, s_a^{(n)}$. Finally, X_{test} is flagged as anomalous if the score exceeds $\hat{q}$, i.e.,

$$g(X_{\text{test}}) = \begin{cases} +1, & \text{if } s_a(X_{\text{test}}) \leq \hat{q} \\ -1, & \text{if } s_a(X_{\text{test}}) > \hat{q}. \end{cases}$$

which allows for controlling the rate of falsely flagging normal samples as anomalies, i.e.,

$$\mathbb{P}(g(X_{\text{test}}) = -1) = \mathbb{P}(s_a(X_{\text{test}}) \ > \ \hat{q}) \leq \alpha. \tag{2.4}$$

The error control guarantee is further formalised in Proposition 2.4.3.

Proposition 2.4.3 (Error control guarantee for anomaly detection)**.** *Considering* $\{(X_i, Y_i)\}_{i=1}^n$ *and* (X_{test}, Y_{test}) *are drawn exchangeably from some distribution,* $\mathcal{C}(X_{test})$ *satisfies* $\mathbb{P}(\mathcal{C}(X_{test}) = -1) \leq \alpha$*.*

The anomaly score function s_a is therefore central to the effectiveness of conformal anomaly detection. As discussed in Section 2.3, there are several ways of computing the anomaly scores, depending on the underlying assumptions about the data.

2.4.2.2 Conformal risk control

In this section, we discuss the concept of *conformal risk control* (Angelopoulos et al., 2024), where beyond creating prediction sets $\mathcal{C}$ that bound the marginal coverage as

$$\mathbb{P}(Y_{\text{test}} \in \mathcal{C}(X_{\text{test}})) \geq 1 - \alpha,$$

or the miscoverage as

$$\mathbb{P}(Y_{\text{test}} \notin \mathcal{C}(X_{\text{test}})) \leq \alpha, \tag{2.5}$$

CP is used to provide guarantees that take the form

$$\mathbb{E}\left[l(\mathcal{C}(X_{\text{test}}), Y_{\text{test}})\right] \leq \alpha, \tag{2.6}$$

for any bounded loss function l that shrinks as $\mathcal{C}$ grows. While this allows us to extend conformal prediction for scenarios where other loss functions are more appropriate, we can also obtain (2.5) back from (2.6) when $l(\mathcal{C}(X_{\text{test}}), Y_{\text{test}}) = \mathbb{1}\{Y_{\text{test}} \notin \mathcal{C}(X_{\text{test}})\}$.

Formally, we are interested in creating a prediction set $\mathcal{C}_\lambda$ that processes the outputs y of a predictive model, i.e.,

$$\mathcal{C}_\lambda(X_{\text{test}}) = \{y \in \mathcal{Y} \ : \ s_c(X_{\text{test}}, y) \leq \lambda\}, \tag{2.7}$$

where the parameter $\lambda \in \Lambda$ controls the conservativeness of the function, resulting in larger prediction sets or lower anomaly decision threshold. Let $l(\mathcal{C}_\lambda(x), y) \in (-\infty, B]$ be a non-increasing loss function for some $B < \infty$, to measure the quality of the output from $\mathcal{C}_\lambda$. Further, consider an exchangeable collection of non-increasing functions $L_i : \Lambda \to (-\infty, B], i = 1, \cdots, n+1$. Specifically, we focus on the scenario where $L_i(\lambda) = l(\mathcal{C}_\lambda(X_i), Y_i)$. Let $\hat{R}_n(C_\lambda) = (L_1(\lambda) + \cdots + L_n(\lambda))/n$ be the empirical risk on the calibration set. Then, given any user-defined risk level upper bound $\alpha \in (-\infty, B)$, the threshold is defined as

$$\hat{\lambda} = \inf \left\{ \lambda : \hat{R}_n(C_\lambda) \leq \alpha - \frac{B - \alpha}{n} \right\}. \tag{2.8}$$

Then the prediction set $\mathcal{C}_{\hat{\lambda}}(X_{\text{test}})$ satisfies (2.6).

Theorem 2.4.4 (Conformal risk control (Angelopoulos et al., 2024))**.** *Consider $\{(X_i, Y_i)\}_{i=1}^n$ and (X_{test}, Y_{test}) are i.i.d samples from some distribution and l is a monotone function of λ, i.e.*

$$l(\mathcal{C}_{\lambda_1}(x), y) \geq l(\mathcal{C}_{\lambda_2}(x), y),$$

for all (x, y) and $\lambda_1 \leq \lambda_2$. Then

$$\mathbb{E}\left[l(\mathcal{C}_{\hat{\lambda}}(X_{test}), Y_{test})\right] \leq \alpha,$$

where $\hat{\lambda}$ is chosen such that it satisfies (2.8).

The proof can be found in (Angelopoulos et al., 2024, Theorem 1).

2.4.2.3 Distribution-free control of general risks

Following our discussion above, where a monotone risk function is controlled in expectation, we now focus on a generalised approach for controlling any risk, including the false-positive and false-negative rates, among others (refer to Section 2.1.6). Here, our goal is to ensure

$$\mathbb{P}(R_n(C_\lambda) < \alpha) \geq 1 - \delta, \tag{2.9}$$

where $R_n(C_\lambda) = \mathbb{E}\left[l(\mathcal{C}_\lambda(X_{\text{test}}), Y_{\text{test}})\right]$ is a user-chosen risk function. Moreover, $\alpha \in [0,1]$ is the risk tolerance and $\delta \in [0,1]$ is the error rate. The probability is taken over the calibration dataset used to obtain λ. For finding λ that satisfy 2.9, we focus on a distribution-free approach called Learn Then Test (LTT) (Angelopoulos et al., 2025). We start by defining a *risk-controlling prediction set* (RCPS) as:

Definition 2.4.5 (Risk-controlling prediction set (Bates et al., 2021))**.** Let $\lambda \in \Lambda$ be a random variable which takes values in a discrete set. The prediction set $\mathcal{C}_\lambda(X_{\text{test}})$ is defined as an (α, δ)-risk-controlling prediction set if it satisfies the condition $\mathbb{P}(R_n(C_\lambda) \leq \alpha) \geq 1 - \delta$, where $\alpha \in (0,1)$ is the risk tolerance and $\delta \in [0,1]$ is the error level.

For finding a prediction set $\mathcal{C}_\lambda(X_{\text{test}})$ whose risk is less than α, we search over a set of prediction sets $\{\mathcal{C}_\lambda(X_{\text{test}})\}_{\lambda\in\Lambda}$ and compute their risk on the calibration data $\{(X_i, Y_i)\}_{i=1}^n$. Here, our goal is to return a set of λ values $\hat{\Lambda} \subseteq \Lambda$ such that the risk is controlled as in (2.9). To do so, first, we associate a null hypothesis $H_{0\lambda} : R_n(C_\lambda) > \alpha$ with each $\lambda \in \Lambda$. Hence, rejecting $H_{0\lambda}$ would mean that the risk is controlled at λ. Then, we compute a p-value p_λ for each null hypothesis $H_{0\lambda}$ using a concentration inequality. For example, using Hoeffding's inequality, we get $p_\lambda = e^{-2n(\alpha - R_n(C_\lambda))_+^2}$. Finally, we can obtain $\hat{\Lambda} = \mathcal{F}(\{p_\lambda\}_{\lambda\in\Lambda})$, where $\mathcal{F}$ is an algorithm that controls FWER as discussed in Section 2.4.1.1. For instance, using the Bonferroni Correction (refer Section 2.4.1.2), results in $\hat{\Lambda} = \{\lambda : p_\lambda < \delta/|\Lambda|\}$. By splitting the conformal risk control into two subparts, namely computing p-values and combining them with multiple hypothesis testing, LTT provides the statistical guarantee in Theorem 2.4.6.

Theorem 2.4.6 (Learn Then Test Risk Control Guarantee)**.** *The $\hat{\Lambda}$ returned by the Learn Then Test procedure satisfies*

$$\mathbb{P}\left(\sup_{\lambda\in\hat{\Lambda}}\{R_n(C_\lambda)\} \leq \alpha\right) \geq 1 - \delta.$$

Thus, for any $\lambda \in \hat{\Lambda}$, $\mathcal{C}_\lambda$ is an (α, δ)-risk-controlling prediction set.

CHAPTER 3

Detecting Logical and Structural Anomalies

In many applications of AD, different types of anomalies can co-occur, making the task of detecting anomalies challenging. In this chapter, we take a closer look at two types of commonly occurring anomalies in industrial settings: (i) structural anomalies, where subtle localised structural defects can be observed in the images (Bergmann et al., 2019), and (ii) logical anomalies, where violations of logical constraints result in anomalies (Bergmann et al., 2022). We propose a unified framework to simultaneously detect both types of anomalies.

This chapter is based on the following publication.

- **Sukanya Patra** & Souhaib Ben Taieb (2024a). Revisiting Deep Feature Reconstruction for Logical and Structural Industrial Anomaly Detection. In the *Transactions of Machine Learning Research (TMLR)*.

3.1. Introduction

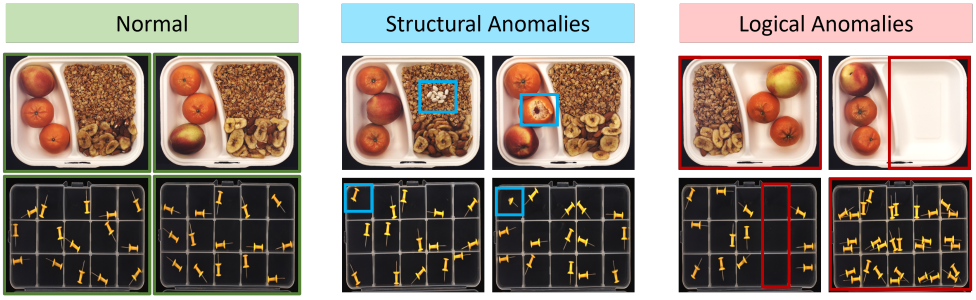


Figure 3.1: Types of anomalies for the categories "breakfast box" and "pushpin" in the MVTecLOCO dataset. Two normal samples (Left) along with structural (Middle) and logical anomalies (Right) where the anomalous regions are highlighted in blue and red, respectively.

In this chapter, we focus on the problem of detecting anomalies in the industrial setting, referred to in the literature as industrial anomaly detection (IAD) (Bergmann et al., 2019, 2022). Specifically, we aim to simultaneously detect both *structural* and *logical* anomalies, as shown in Figure 3.1, which are commonly observed in IAD settings.

To detect *structural anomalies*, state-of-the-art (SOTA) AD methods divide the image into smaller patches and leverage multi-scale features of the patches obtained using deep convolutional neural networks (Salehi et al., 2021). A widely used approach involves storing the extracted features in a memory bank during the training phase (Roth et al., 2022; Lee et al., 2022). During inference, the features of the test image are compared with their closest neighbours in the memory bank. However, such approaches require considerable storage to accumulate the extracted features, which can be challenging for large-scale datasets. As an alternative, (Bergmann et al., 2020, 2022) proposed to train a student network to mimic the teacher for normal samples. Discrepancy between the student and teacher outputs during inference would allow for the identification of anomalies. To prevent the student network from mimicking

the teacher on anomalous samples, several regularisation methods have been proposed (Batzner et al., 2024). However, they not only slow down the training but also increase the requirement for computing resources. Furthermore, excessive regularisation can prevent the student model from learning representations for normal images, thereby negatively impacting the AD performance. Other alternatives include estimating the distribution of features from normal images using a multivariate Gaussian distribution (Defard et al., 2021). Another approach is to train a deep feature reconstruction (DFR) network (Shi et al., 2021) to reconstruct the features of normal images.

Besides structural anomalies, *logical anomalies* occur when elements in the images are missing, misplaced, in surplus or violate geometrical constraints (Bergmann et al., 2022). Methods relying on multi-scale features of image patches would fail, as they would still be considered normal. It is the combination of objects in the image that makes the image anomalous. Thus, to detect such logical anomalies, it is necessary to look beyond image patches and develop a global understanding of the spatial relationships within normal images. Distillation-based methods, which are predominantly used for the detection of logical anomalies, rely on an additional network to learn the spatial relationships between items in the normal image (Batzner et al., 2024).

In this chapter, we focus on DFR, the benefits of which are four-fold. First, it does not need a large memory for storing the features, unlike PatchCore (Roth et al., 2022). Second, unlike PaDiM (Defard et al., 2021), it does not make any assumption about the distribution of features. Third, learning to reconstruct features in the latent space of a pre-trained network is less impacted by the curse of dimensionality than learning to reconstruct high-dimensional images. Fourth, deep networks trained to reconstruct normal images using the per-pixel distance suffer from the loss of sharp edges of the objects or textures in the background. As a result, AD performance deteriorates due to an increase in false positives. On the contrary, computing the distance features maps and their corresponding reconstructions during training is less likely to result in such errors (Assran et al., 2023).

We revisit DFR to develop a unified framework for the detection of both structural and logical anomalies. First, we modify the training objective by considering a combination of ℓ_2 and cosine distances between each feature and the corresponding reconstruction. The incorporation of the cosine distance addresses the curse of dimensionality, where high-dimensional features become orthogonal to each other in Euclidean space and the notion of distance disappears (Aggarwal et al., 2001). Second, to simultaneously allow for the detec-

tion of logical anomalies, we introduce an attention-based loss using a global autoencoder-like network. We empirically demonstrate that with our proposed changes, not only do the detection and localisation capabilities of DFR improve for structural anomalies, but also it delivers competitive results on the detection of logical anomalies. Our contributions can be summarised as follows.

- We propose a **U**nified framework for **L**ogical and **S**tructural **A**nomaly **D**etection referred as `ULSAD`, a framework for detection and localisation of both structural and logical anomalies, building on DFR.
- We consider both magnitude and angular differences between the extracted and reconstructed feature vectors to detect structural anomalies.
- We propose a novel attention-based loss for learning the logical constraints to detect logical anomalies.
- We demonstrate the effectiveness of `ULSAD` by comparing it with 8 SOTA methods across 5 widely adopted IAD benchmark datasets.
- We show the effect of each component of `ULSAD` on the overall performance of the end-to-end architecture through an extensive ablation study.

3.2. The `ULSAD` framework for anomaly detection

We propose `ULSAD`, a framework for the simultaneous detection and localisation of anomalies. Figure 3.2 shows the architectural components of our framework.. Firstly, we utilise a feature extractor network for extracting low-dimensional features from high-dimensional images, which we discuss in Section 3.2.1. Then, for the detection of both structural and logical anomalies, we rely on a dual-branch architecture. The local branch detects structural anomalies with the help of a feature reconstruction network applied to the features corresponding to patches in the image. We elaborate on this in Section 3.2.2. Conversely, the global branch, as discussed in Section 3.2.3, detects logical anomalies using an autoencoder-like network, which takes as input the image. Lastly, we provide an overview of the `ULSAD` algorithm in Section 3.2.4, followed by a discussion on the inference process in Section 3.2.5.

In this chapter, we follow the notations from the paper (Patra and Ben Taieb, 2024). We consider a dataset $\mathcal{D} = \{(\mathbf{X}_i, y_i)\}_{i=1}^{n}$ with n samples. Each $\mathbf{X}_i \in \mathcal{X}$ is an image and $y_i \in \mathcal{Y}$ is the corresponding label where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{0, 1\}$. We refer to the normal class with the label 0 and the anomalous class with the label 1. The samples belonging to the anomalous class can contain either logical

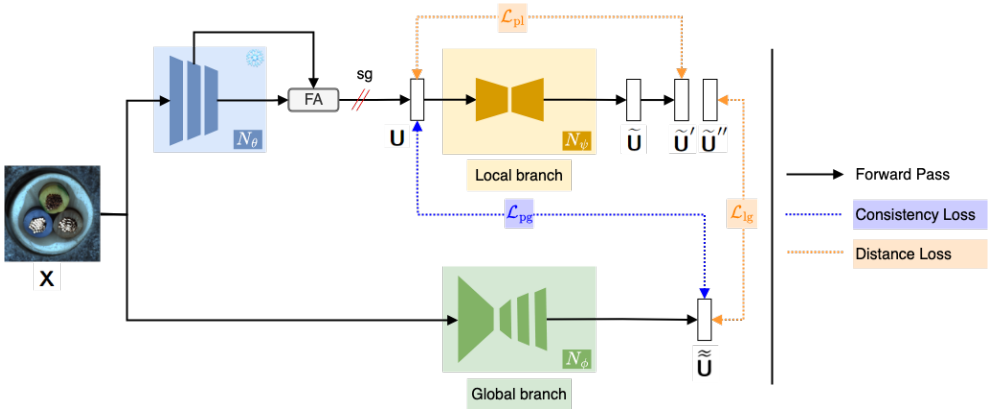


Figure 3.2: Overview of the end-to-end architecture of ULSAD.

or structural anomalies. We denote the train, validation and test partitions of $\mathcal{D}$ as $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{test}}$, respectively. The training and validation sets contain only normal samples, i.e., $y = 0$. The test set $\mathcal{D}_{\text{test}}$ includes both normal and anomalous samples.

3.2.1 Feature extractor

High-dimensional images pose a significant challenge for AD (Reiss et al., 2022). Recent studies have shown that deep convolutional neural networks (CNNs) trained on ImageNet (Russakovsky et al., 2015) capture discriminative features for several downstream tasks. Typically, AD methods (Salehi et al., 2021; Defard et al., 2021; Yoon et al., 2023) leverage such pre-trained networks to extract feature maps corresponding to partially overlapping regions or patches in the images. Learning to detect anomalies using the lower-dimensional features is beneficial as it results in reduced computational complexity. A key factor determining the efficiency of such methods is the size of the image patches being used, as anomalies can occur at any scale. To overcome this challenge, feature maps are extracted from multiple layers of the CNNs and fused together (Salehi et al., 2021; Roth et al., 2022; Shi et al., 2021). Each element in a feature map obtained from different layers of a convolutional network corresponds to a patch of a different size in the image, depending on its receptive field. Thus, combining feature maps from multiple layers results in a multi-scale representation of the image patches, which we refer to as **patch features**.

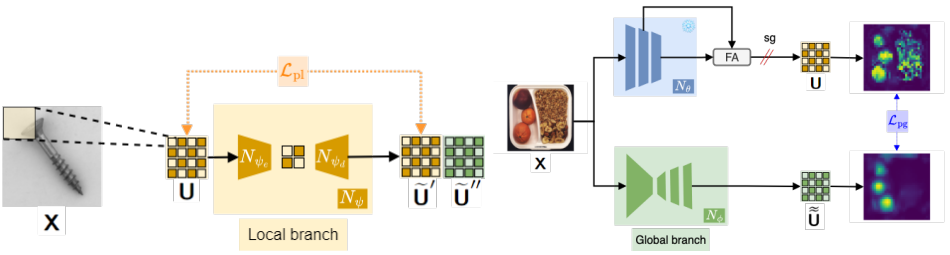
Similar to DFR, we extract low-dimensional feature maps by combining fea-

tures from multiple layers of a feature extractor, which is a pre-trained CNN N_θ parameterised by θ. In this paper, we consider ResNet-like architectures for N_θ. With the increasing number of layers, the computation becomes increasingly expensive as the resulting tensor becomes high-dimensional. In order to overcome this, we consider two intermediate or mid-level features. Our choice is guided by the understanding that the initial layers of such deep networks capture generic image features, while the latter layers are often biased towards the pre-training classification task (Roth et al., 2022). We denote the features extracted at a layer j for an image $\mathbf{X}$ as $N_\theta^j(\mathbf{X})$. Following this convention, we express the feature map $\mathbf{U} \in \mathcal{U} = \mathbb{R}^{c^* \times h^* \times w^*}$ produced by the *Feature Aggregator* (FA) as a concatenation of $N_\theta^j(\mathbf{X})$ and $N_\theta^{j+1}(\mathbf{X})$ obtained from layers j and $j+1$ of N_θ. Furthermore, to facilitate the concatenation of features extracted from multiple layers of the extractor N_θ, the features at the lower resolution layer $j+1$ are linearly rescaled by FA to match the dimension of the features at layer j. We define an invertible transformation $f : \mathbb{R}^{c^* \times h^* \times w^*} \to \mathbb{R}^{c^* \times k^*}$ where $k^* = h^* \times w^*$ to convert tensor to matrix and vice versa using f^{-1}. The function f can be computed in practice by reshaping the tensor to obtain a 2D matrix. Now, using f, we compute $\boldsymbol{Z} = f(\mathbf{U})$. We denote each patch feature within the feature map $\boldsymbol{Z}$ by $\boldsymbol{z}_k = \boldsymbol{Z}[:, k] = \mathbf{U}[:, h, w]$, where $k = (h-1) \times w^* + w$, $h \in \{1, 2, \ldots, h^*\}$, $w \in \{1, 2, \ldots, w^*\}$. Here, $\boldsymbol{z}_k = \boldsymbol{Z}[:, k] \in \mathbb{R}^{c^*}$ refers to the k-th column of $\boldsymbol{Z}$.

3.2.2 Detecting structural anomalies

Having defined $\boldsymbol{Z}$ in the previous section, we elaborate on the local branch of `ULSAD` for the detection of subtle localised defects in the images, i.e. structural anomalies. Specifically, our goal is to learn the reconstruction of the patch features using the dataset $\mathcal{D}_N$ composed of only normal images. Therefore, we can identify the structural anomalies when the network fails to reconstruct a patch feature during inference.

Feature reconstruction network (FRN). As shown in Figure 3.3a, `ULSAD` utilises a convolutional encoder-decoder architecture with a lower-dimensional bottleneck for learning to reconstruct the feature map $\mathbf{U}$ using the training dataset $\mathcal{D}_N$. First, the encoder network N_{ψ_e} compresses the feature $\mathbf{U}$ to a lower-dimensional space, which induces the information bottleneck. It acts as an implicit regulariser, preventing generalisation to features corresponding to anomalous images. The encoded representation is then mapped back to the latent space using a decoder network N_{ψ_d}. The output of FRN is $\widetilde{\mathbf{U}} = N_\psi(\mathbf{U}) \in \mathbb{R}^{2c^* \times h^* \times w^*}$ where $N_\psi = N_{\psi_e} \circ N_{\psi_d}$. Besides using the FRN to learn the reconstruction of the patch features for the detection of structural



(a) Feature Reconstruction Network (b) Global Branch of `ULSAD`

Figure 3.3: Overview of the local and global branch of `ULSAD`.

anomalies, we also utilise it to reduce errors during the detection of logical anomalies, as discussed in Section 3.2.3. To minimise computation costs and avoid the use of two separate FRNs, we adopt a shared FRN. This is achieved by doubling the number of output channels in the decoder to simultaneously produce two feature maps $\widetilde{\mathbf{U}}'$ and $\widetilde{\mathbf{U}}''$ for the detection of structural and logical anomalies, respectively, with both having dimension $c^* \times h^* \times w^*$.

Although the feature maps $\mathbf{U}$ have significantly lower dimensionality compared to the input images $\mathbf{X}$, they can still be considered high-dimensional tensors. In high-dimensional spaces, the ℓ_2 distance is not effective at distinguishing between the nearest and furthest points (Aggarwal et al., 2001), making it an inadequate measure for computing the difference between feature maps during training. Therefore, similar to Salehi et al. (2021), we propose combining ℓ_2 and cosine distances to account for differences in both the magnitude and direction of the patch features as:

$$\mathcal{L}_{\text{pl}}(\widetilde{\boldsymbol{Z}}', \boldsymbol{Z}) = \frac{1}{k^*} \sum_{k=1}^{k^*} l_v(\widetilde{\boldsymbol{z}}'_k, \boldsymbol{z}_k) + \lambda_l \, l_d(\widetilde{\boldsymbol{z}}'_k, \boldsymbol{z}_k), \tag{3.1}$$

where $\widetilde{\boldsymbol{Z}}' = f(\widetilde{\mathbf{U}}')$, $\boldsymbol{Z} = f(\mathbf{U})$ and $\lambda_l \geq 0$ controls the effect of l_d. Furthermore, $l_v(\widetilde{\boldsymbol{z}}'_k, \boldsymbol{z}_k)$ and $l_d(\widetilde{\boldsymbol{z}}'_k, \boldsymbol{z}_k)$ measure the differences in magnitude and direction between the patch features $\boldsymbol{z}_k$ and $\widetilde{\boldsymbol{z}}'_k$, which is the k-th column of $\boldsymbol{Z}$ and $\widetilde{\boldsymbol{Z}}'$, respectively. The losses are given by

$$l_v(\widetilde{\boldsymbol{z}}'_k, \boldsymbol{z}_k) = \|\widetilde{\boldsymbol{z}}'_k - \boldsymbol{z}_k\|_2^2, \quad \text{and} \quad l_d(\widetilde{\boldsymbol{z}}'_k, \boldsymbol{z}_k) = 1 - \frac{(\widetilde{\boldsymbol{z}}'_k)^T \boldsymbol{z}_k}{\|\widetilde{\boldsymbol{z}}'_k\|_2 \|\boldsymbol{z}_k\|_2}. \tag{3.2}$$

3.2.3 Detecting logical anomalies

Although the feature reconstruction task discussed in Section 3.2.2 allows us to detect structural anomalies, it is not suited for identifying logical anomalies

that violate the logical constraints of normal images. Recall that such violations appear in the form of misplaced, misaligned, or surplus objects found in normal images. If we consider the example of misaligned objects, the previously discussed approach will fail, as it focuses on the individual image patches, which would be normal. It is the overall spatial arrangement of objects in the image which is anomalous. Thus, to identify such anomalies, our goal is to learn the spatial relationships among the objects present in the normal images of the training dataset $\mathcal{D}_N$. We achieve this with the global branch of `ULSAD`, shown in Figure 3.3b, which leverages the entire image and not just its individual patches.

In order to achieve our goal, we start by analysing the feature maps extracted using the feature extractor N_θ, which is a pre-trained CNN. Pre-trained CNNs tend to have similar activation patterns for semantically similar objects (Tung and Mori, 2019; Zagoruyko and Komodakis, 2017). In Figure 3.4, we visualise four self-attention maps computed from the features of a pre-trained Wide-Resnet50-2 network. It can be seen that in the first map, all the items for the semantic class "fruits" receive a high attention score. The remaining attention maps focus on individual semantic concepts like "oranges", "cereal" and "plate", respectively. Based on this observation and inspired by the attention-transfer concept for knowledge distillation (Zagoruyko and Komodakis, 2017; Tung and Mori, 2019), we propose to learn the spatial relationships (Dosovitskiy et al., 2021) among the patch features in $\mathbf{U}$ obtained from normal images. Recall that each patch feature corresponds to a patch in the image. Therefore, learning the spatial relationships among the patch features would allow us to learn the spatial relationships among the patches in the image. This forces `ULSAD` to learn the relative positions of objects in the normal images, thereby enabling it to capture the logical constraints. Starting from $\boldsymbol{Z} = f(\mathbf{U})$, we first compute the self-attention weight matrix $\boldsymbol{W} \in \mathbb{R}^{k^* \times k^*}$ as:

$$\boldsymbol{W}[p,q] = \frac{\exp(\boldsymbol{z}_p^T \ \boldsymbol{z}_q / \sqrt{c^*})}{\sum_{k=1}^{k^*} \exp(\boldsymbol{z}_k^T \ \boldsymbol{z}_q / \sqrt{c^*})}, \tag{3.3}$$

where $\boldsymbol{z}_p$ and $\boldsymbol{z}_q$ refers to the p-th and q-th column of $\boldsymbol{Z}$ with $p, q = \{1, \cdots, k^*\}$. Then, the attention map $\boldsymbol{A} \in \mathbb{R}^{c^* \times k^*}$ is computed as $\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{W}$. For learning the spatial relations using $\boldsymbol{A}$ as our target, we use a convolutional autoencoder-like network $N_\phi = N_{\phi_e} \circ N_{\phi_d}$ where N_{ϕ_e} is the encoder and N_{ϕ_d} is the decoder. Similar to a standard autoencoder, N_{ϕ_e} compresses the input image $\mathbf{X}$ to a lower-dimensional space. However, N_{ϕ_d} maps the encoded representation to the feature space $\mathcal{U}$, which has a lower dimension than the input space $\mathcal{X}$. We denote the output of N_ϕ as $\widetilde{\widetilde{\mathbf{U}}} = N_\phi(\mathbf{X})$.

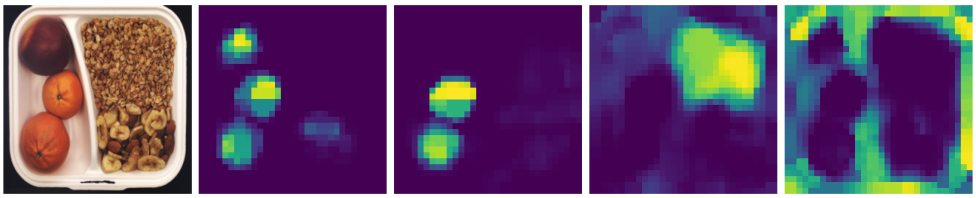


Figure 3.4: (First) Example image belonging to the category "breakfast box" in the MVTecLOCO dataset. (Rest) Visualisation of selected four self-attention maps computed using the intermediate features from a pre-trained Wide-Resnet50-2 model as $\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{W}$ where $\boldsymbol{W}$ is computed using Eq. 3.3.

A direct approach would be to compute the self-attention map for $\widetilde{\widetilde{\mathsf{U}}}$ and minimise its distance from $\boldsymbol{A}$. However, it makes the optimisation problem computationally challenging as each vector in $\widetilde{\mathsf{U}}$ is coupled with every other vector by the network weights N_ϕ (Zhang et al., 2024). To overcome this, we compute the cross-attention map $\widetilde{\widetilde{\boldsymbol{A}}} \in \mathbb{R}^{c^* \times k^*}$ between U and $\widetilde{\widetilde{\mathsf{U}}}$. Given $\widetilde{\widetilde{\mathsf{Z}}} = f(\widetilde{\widetilde{\mathsf{U}}})$, we first compute $\widetilde{\widetilde{\boldsymbol{W}}}$ as:

$$\widetilde{\widetilde{\boldsymbol{W}}}[p,q] = \frac{\exp(\boldsymbol{z}_p^T \, \widetilde{\widetilde{\boldsymbol{z}}}_q / \sqrt{c^*})}{\sum_{k=1}^{k^*} \exp(\boldsymbol{z}_k^T \, \widetilde{\widetilde{\boldsymbol{z}}}_q / \sqrt{c^*})}. \tag{3.4}$$

Then, the attention map $\widetilde{\widetilde{\boldsymbol{A}}}$ can be computed as $\widetilde{\widetilde{\boldsymbol{A}}} = \boldsymbol{Z}\widetilde{\widetilde{\boldsymbol{W}}}$. Given, the self-attention map $\boldsymbol{A}$ and the cross-attention map $\widetilde{\widetilde{\boldsymbol{A}}}$, we define a consistency loss $\mathcal{L}_{\text{pg}}$ as:

$$\mathcal{L}_{\text{pg}}(\widetilde{\widetilde{\boldsymbol{A}}}, \boldsymbol{A}) = \frac{1}{k^*} \sum_{k=1}^{k*} l_v(\widetilde{\widetilde{\boldsymbol{a}}}_k, \boldsymbol{a}_k) + \lambda_g \, l_d(\widetilde{\widetilde{\boldsymbol{a}}}_k, \boldsymbol{a}_k), \tag{3.5}$$

where $\boldsymbol{a}_k = \boldsymbol{A}[:,k]$, $\widetilde{\widetilde{\boldsymbol{a}}}_k = \widetilde{\widetilde{\boldsymbol{A}}}[:,k]$ and $\lambda_{\text{g}} \geq 0$ controls the effect of l_d. A limitation of this approach is that autoencoders usually struggle with generating fine-grained patterns, as also observed by prior works (Dosovitskiy and Brox, 2016; Assran et al., 2023). As a result, the global branch is prone to false positives in the presence of sharp edges or heavily textured surfaces due to the loss of high-frequency details. To address this limitation, we utilise the FRN N_ψ in the local branch to learn the output $\widetilde{\widetilde{\mathsf{U}}}$. Recall that the output of FRN $\widetilde{\mathsf{U}} \in \mathbb{R}^{2c^* \times h^* \times w^*}$ has $2c^*$ number of channels to simultaneously generate two feature maps $\widetilde{\mathsf{U}}'$ and $\widetilde{\mathsf{U}}''$, both having dimension $c^* \times h^* \times w^*$. Out of which, $\widetilde{\mathsf{U}}'$ is used for learning the patch features. Here, we define the loss $\mathcal{L}_{lg}$ to relate

the local feature map $\widetilde{\mathbf{U}}''$ with the global feature map $\widetilde{\widetilde{\mathbf{U}}}$ as:

$$\mathcal{L}_{\text{lg}}(\widetilde{\boldsymbol{Z}}'', \widetilde{\widetilde{\boldsymbol{Z}}}) = \frac{1}{k^*} \sum_{k=1}^{k^*} l_v(\widetilde{\boldsymbol{z}}_k'', \widetilde{\widetilde{\boldsymbol{z}}}_k) + \lambda_g \, l_d(\widetilde{\boldsymbol{z}}_k'', \widetilde{\widetilde{\boldsymbol{z}}}_k), \tag{3.6}$$

where $\widetilde{\boldsymbol{Z}}'' = f(\widetilde{\mathbf{U}}'')$. Therefore, during inference, a difference between the $\widetilde{\mathbf{U}}''$ and $\widetilde{\widetilde{\mathbf{U}}}$ indicates the presence of logical anomalies. The benefits of such a framework are two-fold: (1) it allows for learning the spatial relationships in the normal images while reducing the chance of having false positives, and (2) doubling the channels in the decoder allows sharing the encoder architecture, reducing the computational costs.

3.2.4 `ULSAD` algorithm overview

An overview of `ULSAD` is outlined in Algorithm 1, which can simultaneously detect structural and logical anomalies. Firstly, we pass a normal image $\mathbf{X}$ from the training dataset $\mathcal{D}_N$ through the feature extractor N_θ to obtain feature maps $\mathbf{U}$. We normalize the features (line 4, Algorithm 1) with the channel-wise mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ computed over all the images in $\mathcal{D}_N$. We do not include this step in Algorithm 1 as the calculation is trivial. Instead, we consider the values $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ to be given as input parameters for the sake of simplicity. Secondly, we obtain $\widetilde{\mathbf{U}}$ by passing $\mathbf{U}$ through the feature reconstruction network N_ψ (line 7, Algorithm 1). Recall that, $\widetilde{\mathbf{U}}$ has a dimension $2c^* \times h^* \times w^*$ which can be decomposed into two feature maps $\widetilde{\mathbf{U}}'$ and $\widetilde{\mathbf{U}}''$ each with a dimension $c^* \times h^* \times w^*$. The feature reconstruction loss $\mathcal{L}_{\text{pl}}$ is then computed between $\boldsymbol{Z}$ and $\widetilde{\boldsymbol{Z}}'$, where $\boldsymbol{Z} = f(\mathbf{U})$ and $\widetilde{\boldsymbol{Z}}' = f(\widetilde{\mathbf{U}}')$. Thirdly, we obtain the features $\widetilde{\widetilde{\mathbf{U}}}$ by passing the input sample $\mathbf{X}$ through the autoencoder N_ϕ. Then, for learning the spatial relationships from the normal images, we compute $\mathcal{L}_{\text{pg}}$ between the self-attention map of $\boldsymbol{Z}$ and the cross-attention map between $\boldsymbol{Z}$ and $\widetilde{\widetilde{\mathbf{Z}}} = f(\widetilde{\widetilde{\mathbf{U}}})$ (line 15, Algorithm 1). In the fourth step, we compute the loss $\mathcal{L}_{\text{lg}}$ between $\widetilde{\widetilde{Z}}$ and $\widetilde{\boldsymbol{Z}}'' = f(\widetilde{\mathbf{U}}'')$. Finally, the model parameters ψ and ϕ are updated based on the gradient of the total loss (lines $21 - 22$, Algorithm 1). The end-to-end pipeline is illustrated in Figure 3.2.

3.2.5 Anomaly detection and localisation

After discussing how `ULSAD` is trained to detect structural and logical anomalies, we now focus on the inference process. The first step is to compute an anomaly map $\boldsymbol{M}$ for a given test image $\mathbf{X}$, which assigns a per-pixel anomaly

Algorithm 1: Unified Logical and Structural AD (`ULSAD`) // Local branch Global branch

Require: Training dataset $\mathcal{D}_N$, Feature extractor N_θ, Feature reconstruction network N_ψ, Global autoencoder N_ϕ, Number of epochs e, Learning rate η, Pre-trained feature statictics $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$

for $(\mathbf{X} \in \mathcal{D}_N)$ **do**
- Extract normalised feature maps using the pre-trained network:
  - $\mathbf{U} \leftarrow N_\theta(\mathbf{X})$
  - $\mathbf{U} \leftarrow (\mathbf{U} - \boldsymbol{\mu})/\boldsymbol{\sigma}$
  - $\boldsymbol{Z} \leftarrow f(\mathbf{U})$
- Reconstruct the features maps using the local branch:
  - $\widetilde{\mathbf{U}} \leftarrow N_\psi(\mathbf{U})$
  - $\widetilde{\boldsymbol{Z}} \leftarrow f(\widetilde{\mathbf{U}})$
- Compute local loss (Eq. 3.1):
  - $l_l \leftarrow \mathcal{L}_{\text{pl}}(\widetilde{\boldsymbol{Z}}', \boldsymbol{Z})$
- Obtain the output of the global autoencoder:
  - $\widetilde{\widetilde{\mathbf{U}}} \leftarrow N_\phi(\mathbf{X})$
  - $\widetilde{\widetilde{\boldsymbol{Z}}} \leftarrow f(\widetilde{\widetilde{\mathbf{U}}})$
- Compute consistency loss (Eq. 3.5):
  - $l_g \leftarrow \mathcal{L}_{pg}(\widetilde{\widetilde{\boldsymbol{Z}}}, \boldsymbol{Z})$
- Compute local-global loss (Eq. 3.6):
  - $l_{\text{lg}} \leftarrow \mathcal{L}_{lg}(\widetilde{\widetilde{\boldsymbol{Z}}}, \widetilde{\boldsymbol{Z}}'')$
- Compute overall loss:
  - $l \leftarrow l_l + l_g + l_{\text{lg}}$
- Update model parameters:
  - $\psi \leftarrow \psi - \eta \nabla_\psi l$
  - $\phi \leftarrow \phi - \eta \nabla_\phi l$

end

Return: N_ψ, N_ϕ

score. We begin by calculating the local anomaly map $\boldsymbol{M}^l \in \mathbb{R}^{h^* \times w^*}$ based on the difference between the output of the local branch $\widetilde{\mathbf{U}}'$ and the feature map $\mathbf{U}$, as follows:

$$\boldsymbol{M}^l[h,w] = l_v(\widetilde{\mathbf{U}}'[:,h,w],\ \mathbf{U}[:,h,w]) + \lambda_l\ l_d(\widetilde{\mathbf{U}}'[:,h,w],\ \mathbf{U}[:,h,w]), \tag{3.7}$$

where $\widetilde{\mathbf{U}}' = f^{-1}(\widetilde{\boldsymbol{Z}}')$ and $\mathbf{U} = f^{-1}(\boldsymbol{Z})$. Similarly, the global anomaly map $\boldsymbol{M}^g$ is computed using the output from the global autoencoder $\widetilde{\widetilde{\mathbf{U}}}$ and the local reconstruction branch $\widetilde{\mathbf{U}}''$:

$$\boldsymbol{M}^g[h,w] = l_v(\widetilde{\mathbf{U}}''[:,h,w], \widetilde{\widetilde{\mathbf{U}}}[:,h,w]) + \lambda_g \, l_d(\widetilde{\mathbf{U}}''[:,h,w], \widetilde{\widetilde{\mathbf{U}}}[:,h,w]), \tag{3.8}$$

where $\widetilde{\mathbf{U}}'' = f^{-1}(\widetilde{\boldsymbol{Z}}'')$ and $\widetilde{\widetilde{\mathbf{U}}} = f^{-1}(\widetilde{\widetilde{\boldsymbol{Z}}})$.

Since $\boldsymbol{M}^l$ and $\boldsymbol{M}^g$ may have different ranges of anomaly scores, we normalise each map independently. This normalisation ensures consistent score ranges and prevents noise in one map from overwhelming anomalies detected in the other. Given the variability in anomaly score distributions across datasets, we adopt a quantile-based normalisation method, which makes no assumptions about the underlying score distribution.

To normalize the maps, we generate two sets of anomaly maps: $\mathcal{M}^l = \{\boldsymbol{M}^l \mid \mathbf{X} \in \mathcal{D}_{\text{valid}}\}$ and $\mathcal{M}^g = \{\boldsymbol{M}^g \mid \mathbf{X} \in \mathcal{D}_{\text{valid}}\}$, using images from the validation set $\mathcal{D}_{\text{valid}}$. For each set, we pool together the pixel values from all the anomaly maps in that set to compute the empirical quantiles at significance levels α and β. Specifically, for the local anomaly maps, the quantiles are denoted as q^l_α and q^l_β, while for the global anomaly maps, the quantiles are denoted as q^g_α and q^g_β. Values below q_α are considered normal, while those above q_β are marked as highly abnormal.

Following Batzner et al. (2024), we define linear transformations $t^l(\cdot)$ and $t^g(\cdot)$ for the local and global anomaly maps to map normal pixels to values ≤ 0 and highly anomalous pixels to values ≥ 0.1:

$$t^l(\boldsymbol{M}^l) = 0.1 \left(\boldsymbol{M}^l - \left(\frac{q^l_\alpha}{q^l_\beta - q^l_\alpha} \right) \mathbf{1}_{h^* \times w^*} \right),$$
$$t^g(\boldsymbol{M}^g) = 0.1 \left(\boldsymbol{M}^g - \left(\frac{q^g_\alpha}{q^g_\beta - q^g_\alpha} \right) \mathbf{1}_{h^* \times w^*} \right),$$

where $\mathbf{1}_{h^* \times w^*}$ is a matrix of ones. Mapping the empirical quantiles at α and β to values of 0 and 0.1 helps highlight the anomalous regions on a 0-to-1 colour scale for visualisation. Normal pixels are assigned a score of 0, while pixels with scores between q_α and q_β gradually increase in colour intensity. Pixels with scores exceeding q_β change more rapidly toward 1. Note that this transformation does not affect AUROC scores, as these depend only on the ranking of the scores.

Finally, we compute the overall anomaly map $\boldsymbol{M}$ for the image $\mathbf{X}$ by averaging the normalized local and global maps:

$$\boldsymbol{M} = \frac{t^l(\boldsymbol{M}^l) + t^g(\boldsymbol{M}^g)}{2}.$$

The final anomaly score for $\mathbf{X}$ is the maximum value in the combined anomaly map:

$$s = \max_{h \in \{1,2,\ldots,h^*\}, w \in \{1,2,\ldots,w^*\}} \boldsymbol{M}[h, w].$$

3.3. Experimental evaluation

In this section, we answer the following three questions: (i) How does `ULSAD` perform as compared to the SOTA methods? (ii) How effective is the local and global branch for the detection of *structural* and *logical* anomalies? (iii) How does each component in `ULSAD` impact the overall performance?

3.3.1 Setup

Benchmark datasets. We evaluate our method on the following five IAD benchmarking datasets:

[1] BTAD (Mishra et al., 2021). It comprises real-world images of three industrial products, with anomalies such as body and surface defects. Training data includes 1,799 normal images across the three categories, while the test set contains 290 anomalous and 451 normal images.

[2] MVTec AD (Bergmann et al., 2019). It consists of images from industrial manufacturing across 15 categories, comprised of 10 objects and 5 textures. In totality, it contains 3,629 normal images for training. For evaluation, the test set contains 1,258 anomalous images with varying pixel-level defects and 467 normal images.

[3] MVTec-Loco (Bergmann et al., 2022). An extension of the MVTec dataset, it encompasses both local structural anomalies and logical anomalies violating long-range dependencies. It consists of 5 categories, with 1,772 normal images for training and 304 normal images for validation. It also contains 1568 images, either normal or anomalous, for evaluation.

[4] MPDD (Jezek et al., 2021). It focuses on metal part fabrication defects. The images are captured in variable spatial orientation, position, and distance

of multiple objects concerning the camera at different light intensities and with a non-homogeneous background. It consists of 6 classes of metal parts with 888 training images. For evaluation, the dataset has 176 normal and 282 anomalous images.

[5] VisA (Zou et al., 2022). It contains 10,821 high-resolution images (9,621 normal and 1,200 anomalous images) across 12 different categories. The anomalous images contain different types of anomalies such as scratches, bent, cracks, missing parts or misplacements. For each type of defect, there are 15-20 images, and an image can depict multiple defects.

Evaluation metrics. We measure the image-level anomaly detection performance via the area under the receiver operator curve (AUROC) based on the assigned anomaly score. To measure the anomaly localisation performance, we use pixel-level AUROC and area under per region overlap curve (AUPRO). Furthermore, following prior works (Roth et al., 2022; Gudovskiy et al., 2022; Bergmann et al., 2019), we compute the average metrics over all the categories for each of the benchmark datasets. Moreover, for `ULSAD`, we report all the results over 5 runs with different random seeds.

Baselines. We compare our method with existing state-of-the-art unsupervised AD methods, namely 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), EfficientAD (Batzner et al., 2024) and DFR (Shi et al., 2021). In this study, we only consider baselines that are capable of both anomaly detection and localisation.

Implementation details. `ULSAD` is implemented in PyTorch (Paszke et al., 2019). For the baselines, we follow the implementation in Anomalib (Akcay et al., 2022), a widely used AD library for benchmarking. In `ULSAD`, we use a Wide-ResNet50-2 pre-trained on ImageNet (Zagoruyko and Komodakis, 2016) and extract features from the second and third layers, similar to PathCore (Roth et al., 2022). We use a CNN for the autoencoder N_ϕ in the global branch and the feature reconstruction network N_ψ in the local branch. It consists of convolution layers with LeakyReLU activation in the encoder and deconvolution layers in the decoder. The architecture is provided in the Appendix A.1. Unless otherwise stated, for all the experiments, we consider an image size of 256×256. We train `ULSAD` over 200 epochs for each category using an Adam optimiser with a learning rate of 0.0002 and a weight decay of 0.00002. We set $\alpha = 0.9$ and $\beta = 0.995$ unless specified otherwise based on

empirical analysis. We also provide an ablation study in Section 3.3.3. For the baselines, we use the hyperparameters mentioned in the respective papers.

3.3.2 Evaluation results

We summarise the anomaly detection performance of `ULSAD` in Table 3.1 and the localisation performance in Table 3.2. On the BTAD dataset, we improve over the DFR by approximately 2% in detection. Inspecting the images from the dataset, we hypothesise that the difference stems from the use of a global branch in `ULSAD` as the structural imperfections are not limited to small regions. For localisation, Reverse Distillation performs better owing to the use of anomaly maps computed per layer of the network. We can observe similar improvements over DFR on the MVTec dataset. Although PatchCore provides superior performance on MVTec, it should be noted that even without using a memory bank, `ULSAD` provides comparable results. Then, we focus on more challenging datasets such as MPDD and MVTecLOCO. While MPDD contains varying external conditions such as lighting, background and camera angles, MVTecLOCO contains both logical and structural anomalies. We can observe improvements over DFR ($\sim 12 - 16\%$) in both datasets. This highlights the effectiveness of our method. We visualise the anomaly maps for samples from the "pushpin" and "juice bottle" categories in Figure 3.5. It can be seen that while the global branch is more suited to the detection of logical anomalies, the local branch is capable of detecting localised structural anomalies.

Table 3.1: Average Detection Performance in AUROC (%). Style: **best** and second best

Method	BTAD	MPDD	MVTec	MVTec-LOCO	VisA
PatchCore (Roth et al., 2022)	93.27	93.27	**98.75**	81.49	91.48
CFLOW (Gudovskiy et al., 2022)	93.57	87.11	94.47	73.62	87.77
DRÆM (Zavrtanik et al., 2021)	73.42	74.14	75.26	62.35	77.75
EfficientAD (Batzner et al., 2024)	88.26	85.42	98.23	80.62	91.21
FastFlow (Yu et al., 2021)	91.68	65.03	90.72	71.00	87.49
PaDiM (Defard et al., 2021)	93.20	68.48	91.25	68.38	83.28
Reverse Distillation (Deng and Li, 2022)	83.87	79.62	79.65	61.56	86.24
DFR (Shi et al., 2021)	94.60	79.75	93.54	72.87	85.18
`ULSAD` **(Ours)**	**96.17** ± 0.45	**95.73** ± 0.45	97.65 ± 0.38	**84.1** ± 0.86	**92.46** ± 0.45

Overall, `ULSAD` demonstrates competitive results in anomaly detection compared to the baseline methods across all benchmark datasets. Additionally, the difference in performance between `ULSAD` and the baselines for anomaly localisation is minimal. The most notable difference is in the AUPRO score on the BTAD, MVTec, and VisA datasets. Nonetheless, while the SOTA methods provide slightly better performance in the localisation of structural anomalies,

Table 3.2: Average Segmentation Performance in AUROC (%) and AUPRO (%). Style: **best** and second best

Method	BTAD	MPDD	MVTec	MVTec-LOCO	VisA
PatchCore (Roth et al., 2022)	96.85 | 71.48	**98.07** | 90.84	**97.71** | 91.15	75.77 | 69.09	97.93 | 85.12
CFLOW (Gudovskiy et al., 2022)	96.60 | 73.11	97.42 | 88.56	97.17 | 90.14	76.99 | 66.93	98.04 | 85.29
DRÆM (Zavrtanik et al., 2021)	59.04 | 22.48	86.96 | 70.04	75.01 | 49.72	63.69 | 40.06	71.31 | 54.68
EfficientAD (Batzner et al., 2024)	82.13 | 54.37	97.03 | 90.44	96.29 | 90.11	70.36 | 66.96	97.51 | 84.45
FastFlow (Yu et al., 2021)	96.15 | 75.27	93.60 | 76.89	96.44 | 88.79	75.55 | 53.04	97.32 | 81.70
PaDiM (Defard et al., 2021)	97.07 | 77.80	94.51 | 81.18	96.79 | 91.17	71.32 | 67.97	97.09 | 80.80
Reverse Distillation (Deng and Li, 2022)	**97.85** | **81.47**	97.83 | 91.86	97.25 | **93.12**	68.55 | 66.28	**98.68** | **91.77**
DFR (Shi et al., 2021)	97.62 | 59.06	97.33 | 90.46	94.93 | 89.42	61.72 | 69.78	97.90 | 91.72
ULSAD **(Ours)**	96.73 | 75.41 ± 0.51 | ± 3.95	97.45 | **92.02** ± 0.99 | ± 2.64	97.61 | 91.67 ± 0.64 | ± 1.36	**80.06** | **73.73** ± 0.20 | ± 0.35	98.24 | 87.12 ± 0.20 | ± 0.89

ULSAD provides similar performance across both logical and structural anomalies. We present anomaly maps obtained from different methods in Figure A.1 of Appendix A.2. Extended versions of Tables 3.1 and 3.2 are provided in Appendix A.2. Additionally, we provide the results on MVTecLOCO, split between logical and structural anomalies, in Appendix A.2.1.

3.3.3 Ablation study

In this section, we analyse the impact of the key components of ULSAD, backbone architectures and the choice of α and β for normalisation, using the MVTecLOCO dataset.

Table 3.3: Ablation of the main components of ULSAD.

	Local Branch	Global Branch				Performance (%)
	λ_l	λ_g	$\mathcal{L}_{\text{lg}}$	$\mathcal{L}^d_{\text{pg}}$	$\mathcal{L}_{\text{pg}}$	I-AUROC | P-AUROC | P-AUPRO
1	0.0	-	-	-	-	77.67 | 75.17 | 73.37
2	0.0	0.0	-	✓	-	77.69 | 79.77 | 75.26
3	0.0	0.0	-	-	✓	71.67 | 73.92 | 67.22
4	0.0	0.0	✓	✓	-	81.40 | 82.12 | 77.47
5	0.0	0.0	✓	-	✓	81.08 | 81.97 | 76.45
6	0.5	-	-	-	-	79.14 | 76.57 | 73.41
7	0.5	0.5	-	✓	-	80.50 | 81.85 | 77.35
8	0.5	0.5	-	-	✓	74.51 | 76.59 | 69.01
9	0.5	0.5	✓	✓	-	82.19 | 81.25 | 75.50
10	0.5	0.5	✓	-	✓	**84.10** | **80.06** | **73.73** ± 0.86 | ± 0.20 | ± 0.35

Analysis of main components. We investigate the impact of key components in ULSAD as presented in Table 3.3. Initially, we set both λ_l and λ_g to 0, focusing solely on differences in magnitude when computing $\mathcal{L}_{pg}$, $\mathcal{L}_{lg}$, and

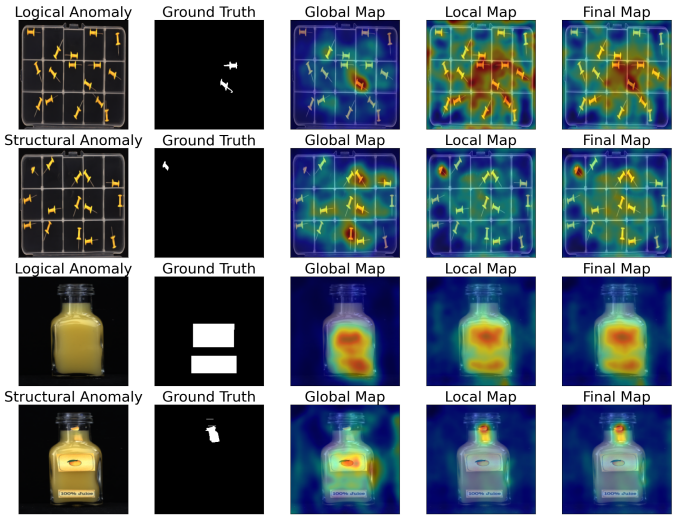
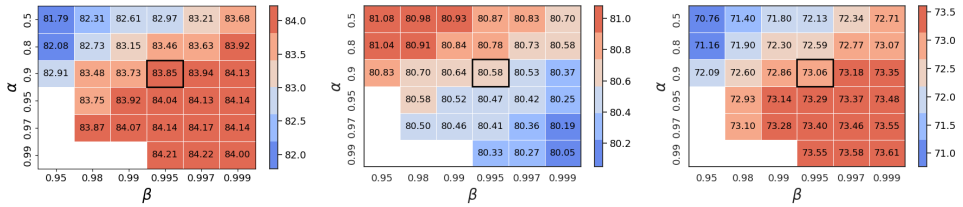


Figure 3.5: Example of anomaly maps obtained from global and local branches along with the combined map.

$\mathcal{L}_{pl}$. The first row corresponds to using only the local branch. In the third row, the consistency loss $\mathcal{L}_{pg}$ is applied to capture spatial relationships for detecting logical anomalies. However, when used in isolation, it limits `ULSAD`'s performance to detecting only logical anomalies and fails to capture localised structural anomalies. Additionally, as discussed in Section 3.2.3, the global branch is prone to false positives in the presence of sharp edges or heavily textured surfaces. When incorporating $\mathcal{L}_{lg}$, which connects the global and local branches, we observe a significant improvement in performance, as shown in the fifth row. For the sake of completeness, we also consider here a variant of the consistency loss $\mathcal{L}_{\text{pg}}$ where we compute the ℓ_2 distance between the feature maps instead of computing the self- and cross-attention maps. We refer to the alternative in the table as $\mathcal{L}^{d}_{\text{pg}}$. We observe that the difference between the two variants becomes negligible when combined with $\mathcal{L}_{lg}$ (row 4 and 5). Further, incorporating differences in direction when computing $\mathcal{L}_{pg}$, $\mathcal{L}_{lg}$, and $\mathcal{L}_{pl}$ leads to improved performances across all settings, as shown in the last five rows. Overall, the best performance is obtained when both $\mathcal{L}_{lg}$ and $\mathcal{L}_{pg}$ are used while considering differences in both direction and magnitude for computing the losses.

Effect of normalization. We analyse the impact of the quantile-based normalisation on the performance metrics by considering multiple values for α and β. The results are shown in Figure 3.6. It can be seen that the final



(a) Image AUROC (b) Pixel AUROC (c) Pixel AUPRO

Figure 3.6: Ablation study of α and β for normalization of anomaly maps with selected value highlighted.

performance is robust to the choice of α and β.

Effect of backbone. We investigate the impact of using different pre-trained backbones in `ULSAD` in Figure 3.7. We can observe that the overall best performance is obtained by using a Wide-ResNet101-2 architecture in both detection and localisation. More specifically, for detection, Wide-ResNet variants are more effective than the ResNet architectures, whereas, for localisation performance measured using Pixel AUROC, the deeper networks such as ResNet152 and Wide-ResNet101-2 seem to have precedence over their shallower counterparts. Overall, we can see that performance is robust to the choice of pre-trained model architecture. In our experiments, we utilise a Wide-ResNet50-2 architecture, which is used by most of our baselines for fair comparison.

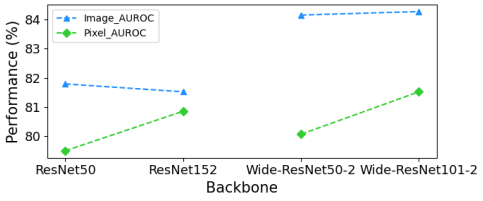


Figure 3.7: Ablation study of the backbone network

3.4. Memory and computational complexity

We report the computational cost and memory requirements of `ULSAD` compared to the baselines in Table 3.4. For this analysis, we ran inference on the test samples in the MVTecLOCO dataset using an NVIDIA A100 GPU. We measured throughput with a batch size of 32, as a measure of computational complexity, following EfficientAD (Batzner et al., 2024). Throughput is defined as the number of images processed per second when processing in batches. `ULSAD` demonstrates higher throughput than most baselines while

Table 3.4: Memory and computational efficiency on MVTecLOCO dataset.

	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
I-AUROC ↑	73.62	62.35	71.00	68.38	<u>81.49</u>	61.56	72.87	80.62	**84.10** ± 0.86
P-AUROC ↑	<u>76.99</u>	63.69	75.55	71.32	75.77	68.55	61.72	70.36	**80.06** ± 0.20
P-AUPRO ↑	66.93	40.06	53.04	67.97	69.09	66.28	<u>69.78</u>	66.96	**73.73** ± 0.35
Throughput (img / s) ↑	11.69	10.06	30.21	<u>33.45</u>	32.70	**34.87**	15.08	23.33	33.42
GPU Memory (GB) ↓	2.57	7.95	**1.69**	<u>1.92</u>	6.80	1.93	5.85	3.48	2.17

maintaining competitive anomaly detection and localisation performance. In addition to throughput, we also report peak GPU memory usage in Table 3.4 to highlight the memory efficiency of ULSAD. It is evident that ULSAD requires approximately one-third of the memory compared to retrieval-based methods such as PatchCore, which is one of the state-of-the-art methods for IAD. For DFR (Shi et al., 2021), we follow the authors' approach by using a multiscale representation, concatenating features from 12 layers of the pre-trained network N_θ for anomaly detection. This approach results in reduced throughput and increased memory usage, as shown in Table 3.4. With our proposed modifications in ULSAD, we achieve superior performance using features from only 2 layers of N_θ, drastically reducing memory requirements to approximately one-third of DFR's and increasing throughput by approximately two times.

3.5. Limitations

For training ULSAD, we follow the common assumption in unsupervised anomaly detection (Ruff et al., 2021; Chandola et al., 2009; Roth et al., 2022; Batzner et al., 2024) that the training dataset is "clean", meaning it contains no anomalous samples. This setup is known in the literature as one-class classification (Ruff et al., 2018). However, this assumption could impact performance in real-world scenarios where anomalies are unknown a priori. Investigating the effects of dataset contamination (Wang et al., 2019; Jiang et al., 2022; Yoon et al., 2022; Perini et al., 2023, 2022) is an active area of research, which is beyond the scope of our current work. We leave for future research the analysis of contamination's impact on ULSAD and the development of strategies to make the learning process robust in the presence of anomalies.

3.6. Conclusion

Our study focuses on Deep Feature Reconstruction (DFR), a memory- and compute-efficient method for detecting structural anomalies. We propose ULSAD, a unified framework that extends DFR to detect both structural and logical

anomalies using a dual-branch architecture. In particular, we enhance the local branch's training objective to account for differences in the magnitude and direction of patch features, thereby improving structural anomaly detection. Additionally, we introduce an attention-based loss in the global branch to capture logical anomalies effectively. Extensive experiments on five benchmark image anomaly detection datasets demonstrate that `ULSAD` achieves competitive performance in anomaly detection and localisation compared to eight state-of-the-art methods. Notably, `ULSAD` also performs well against memory-intensive, retrieval-based methods like PatchCore (Roth et al., 2022). Finally, ablation studies highlight the impact of various components in `ULSAD` and the role of the pre-trained backbone on overall performance.

CHAPTER 4

Risk Estimator-based Semi-supervised AD

A significant limitation of one-class classification AD methods is their reliance on the assumption that unlabeled training data only contains normal instances. To overcome this impractical assumption, we propose two novel classification-based AD methods. Firstly, we introduce a semi-supervised shallow AD method based on an unbiased risk estimator. Secondly, we present a semi-supervised deep AD method utilising a non-negative (biased) risk estimator. We establish estimation error bounds and excess risk bounds for both risk minimisers. Additionally, we propose techniques to select appropriate regularisation parameters that ensure the non-negativity of the empirical risk in the shallow model under specific loss functions.

This chapter is based on the following publication.

- Le Thi Khanh Hien, **Sukanya Patra**, & Souhaib Ben Taieb. Anomaly detection with semi-supervised classification based on risk estimators (2024b). In the *Transactions of Machine Learning Research (TMLR)*.

Le Thi Khanh Hien was responsible for the formalisation of the research problem and the development of the proposed solution. Sukanya Patra contributed to the experimental work involving the deep risk-based AD method and assisted in the preparation of the scientific article. The project was conducted under the supervision of Prof. Souhaib Ben Taieb.

4.1. Introduction

Unsupervised learning, where only unlabeled data is available, represents the most common setting in AD as discussed previously in Section 2.2. However, in real-world scenarios, labelled samples may be available alongside the unlabeled dataset, leading to the development of semi-supervised AD methods (Görnitz et al., 2009; Munoz-Mari et al., 2010; Ruff et al., 2020). Classification methods that learn from positive and unlabeled data have been extensively studied as LPUE or PU learning. Consequently, PU learning methods have also been utilised in semi-supervised classification-based AD methods (Bekker and Davis, 2020; Blanchard et al., 2010; Chandola et al., 2009; Ju et al., 2020) as discussed in Section 2.3. Specifically, in the PU setting (Figure 2.2), we have access to labelled normal samples, which are relatively easier to obtain in practice, along with an unlabelled dataset. However, it is widely recognised that incorporating labelled anomalies, even a few instances, can greatly enhance AD performance (Görnitz et al., 2013; Kiran et al., 2018; Qiu et al., 2022). Thus, semi-supervised AD methods (Han et al., 2022; Ruff et al., 2021, 2020) that leverage labelled anomalous samples have demonstrated highly promising AD performance. Therefore, in our work, we focus on a semi-supervised setting where we have access to both labelled normal and anomalous samples along with a larger dataset of unlabelled samples, also known as the PNU setting discussed in Section 2.2.

Nonetheless, both unsupervised and semi-supervised AD methods do not explicitly handle the unlabeled data, which can contain both normal and anomalous samples. Instead, they predominantly assume that the unlabelled training data consists solely of normal instances (Hodge and Austin, 2004; Pimentel et al., 2014; Zimek et al., 2012; Ruff et al., 2018), which is impractical in

real-world scenarios and can lead to biased AD models. To overcome the impractical assumption of AD methods, we adopt the key concept of risk-based PU learning methods (du Plessis et al., 2014, 2015; Kiryo et al., 2017; Sakai et al., 2017). These methods propose unbiased empirical estimators for the risk associated with the learning problem. It is noteworthy that the estimation of risk in AD is a relatively unexplored subject, characterised by specific features, particularly in terms of error bounds, which are not commonly found in current AD approaches. Our main contributions are summarised as follows.

- Considering AD as a semi-supervised classification problem, we introduce two risk-based AD methods. Specifically, we consider access to both labelled normal and anomalous samples, along with an unlabeled dataset that may also contain anomalous examples. These proposed methods include a shallow AD approach developed using an unbiased risk estimator and a deep AD method based on a nonnegative risk estimator.
- We develop methods to select a suitable regularisation that ensures the nonnegativity of the empirical risk in the proposed shallow AD method. This is crucial as negative empirical risk can lead to significant overfitting issues (Kiryo et al., 2017).
- We additionally establish estimation error bounds and excess risk bounds for the two risk minimisers, building upon the theoretical findings presented in (Kiryo et al., 2017; Niu et al., 2016).
- We conduct extensive experiments on benchmark AD datasets obtained from *Adbench* (Han et al., 2022) to compare the performance of our proposed risk-based AD (rAD) methods against various baseline methods.

4.2. Related work

In the following, we provide a brief overview of the most relevant works related to our proposed risk-based AD methods.

PU learning methods. PU learning methods can be classified into three categories: biased learning, two-step techniques, and class-prior incorporation. Similar to one-class classification-based AD methods, biased PU learning methods make an impractical assumption that all unlabeled instances are negative (Lee and Liu, 2003; Liu et al., 2003). Although PU learning methods using two-step techniques do not make this assumption, they are heuristics, since they first identify "reliable" negative examples and then apply (semi-)supervised

learning techniques to the positive-labelled instances and the reliable negative instances (Li and Liu, 2003; Chaudhari and Shevade, 2012). Furthermore, to provide a theoretical guarantee, the class-prior incorporation methods assume that the class priors are known (du Plessis et al., 2014; Elkan and Noto, 2008; Hsieh et al., 2019). We refer the readers to Bekker and Davis (2020) and the references therein for more details on the three types of PU learning methods. Methods that rely on risk estimators belong to the third category (du Plessis et al., 2014, 2015; Kiryo et al., 2017; Sakai et al., 2017).

Our primary contribution lies in delving deeper into the approach of risk estimation, a technique that remains relatively unexplored within the context of AD. Specifically, we focus on the class-prior incorporation-based approaches for learning an unbiased anomaly detector.

4.3. Background on risk estimators

For our work, we follow the same formulation as discussed in Section 2.1.1. In addition, we consider $\pi_p = \mathbb{P}(Y = +1)$ and $\pi_n = \mathbb{P}(Y = -1)$ to be the class-prior probabilities for the normal and anomalous classes, respectively, with $\pi_p + \pi_n = 1$. Note that π_n is the same as the contamination ratio ϵ defined previously in Section 2.2. We consider the positive $\mathcal{D}_{\mathrm{P}} = \{x_i^p\}_{i=1}^{n_p} \sim \mathrm{P}_X^+$, negative $\mathcal{D}_{\mathrm{N}} = \{x_i^n\}_{i=1}^{n_n} \sim \mathrm{P}_X^-$ and unlabeled $\mathcal{D}_{\mathrm{U}} = \{x_i^u\}_{i=1}^{n_u} \sim \mathrm{P}_x^u$ data are sampled independently, where

$$\mathrm{P}_X^u = \pi_p \mathrm{P}_X^+ + \pi_n \mathrm{P}_X^-. \tag{4.1}$$

Given $\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{N}}$ and $\mathcal{D}_{\mathrm{U}}$, let us consider a binary classification problem from x to y. Here, $g : \mathbb{R}^d \to \mathbb{R}$ is an anomaly detector that needs to be trained from $\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{N}}$ and $\mathcal{D}_{\mathrm{U}}$, using $\ell : \mathbb{R} \times \{+1, -1\} \to \mathbb{R}$ which is a loss function that imposes a cost $\ell(t, y)$ if the predicted output is t and the expected output is y.

Assuming that π_p is known (in practice, π_p can be effectively estimated from $\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{N}}$ and $\mathcal{D}_{\mathrm{U}}$ (du Plessis and Sugiyama, 2014; Saerens et al., 2002)), our goal is to find g that minimizes the risk $\mathcal{R}(g)$, which is defined as

$$\mathcal{R}(g) := \mathbb{E}_{(x,y)\sim P_{X,Y}}[\ell(g(x), y)] = \pi_p \mathcal{R}_p^+(g) + \pi_n \mathcal{R}_n^-(g), \tag{4.2}$$

where $\mathcal{R}_p^+(g) = \mathbb{E}_{x\sim P_X^+}[\ell(g(x), +1)]$ and $\mathcal{R}_n^-(g) = \mathbb{E}_{x\sim P_X^-}[\ell(g(x), -1)]$. Similarly, we can denote

$$\mathcal{R}_p^-(g) = \mathbb{E}_{x\sim P_X^+}[\ell(g(x), -1)], \quad \mathcal{R}_n^+(g) = \mathbb{E}_{x\sim P_X^-}[\ell(g(x), +1)],$$
$$\mathcal{R}_u^+(g) = \mathbb{E}_{x\sim P_X^u}[\ell(g(x), +1)], \quad \mathcal{R}_u^-(g) = \mathbb{E}_{x\sim P_X^u}[\ell(g(x), -1)].$$

In ordinary classification, the optimal classifier minimises the expected misclassification rate that corresponds to using zero-one loss in (4.2), $\ell_{0\text{-}1}(t,y) = 0$ if $ty > 0$ and $\ell_{0\text{-}1}(t,y) = 1$ otherwise. We denote $I(g) = \mathbb{E}_{(x,y)\sim P_{X,Y}}[\ell_{0\text{-}1}(g(x),y)]$.

PN risk estimator. In supervised learning, when we have fully labelled data, $\mathcal{R}(g)$ can be approximated by a PN risk estimator $\hat{\mathcal{R}}_{pn}(g) = \pi_p\hat{\mathcal{R}}_p^+(g) + \pi_n\hat{\mathcal{R}}_n^-(g)$, where

$$\hat{\mathcal{R}}_p^+(g) := \frac{1}{n_p}\sum_{i=1}^{n_p}\ell(g(x_i^p),+1), \quad \hat{\mathcal{R}}_n^-(g) := \frac{1}{n_n}\sum_{i=1}^{n_n}\ell(g(x_i^n),-1). \tag{4.3}$$

PU risk estimator. In PU learning when $\mathcal{D}_{\mathrm{N}}$ is unavailable, du Plessis et al. (2014, 2015); Kiryo et al. (2017) propose methods to approximate $\mathcal{R}(g)$ from $\mathcal{D}_{\mathrm{P}}$ and $\mathcal{D}_{\mathrm{U}}$. From (4.1) we have $\pi_n \mathrm{P}_X^- = \mathrm{P}_X^u - \pi_p \mathrm{P}_X^+$, which implies $\pi_n\mathcal{R}_n^-(g) = \mathcal{R}_u^-(g) - \pi_p\mathcal{R}_p^-(g)$. Using this, we get

$$\mathcal{R}(g) = \pi_p(\mathcal{R}_p^+(g) - \mathcal{R}_p^-(g)) + \mathcal{R}_u^-(g). \tag{4.4}$$

We consider three estimators for (4.4). First, when ℓ satisfies the *symmetric condition* $\ell(t,+1)+\ell(t,-1) = 1$ then we have $\mathcal{R}(g) = 2\pi_p\mathcal{R}_p^+(g) - \pi_p + \mathcal{R}_u^-(g)$, which can be approximated by

$$\hat{\mathcal{R}}_{pu}^{(1)}(g) = 2\pi_p\hat{\mathcal{R}}_p^+(g) - \pi_p + \hat{\mathcal{R}}_u^-(g), \tag{4.5}$$

where $\hat{\mathcal{R}}_p^+(g)$ is defined in (4.3) and $\hat{\mathcal{R}}_u^-(g) = \frac{1}{n_u}\sum_{i=1}^{n_u}\ell(g(x_i^u),-1)$ (du Plessis et al., 2014). Second, when ℓ satisfies the *linear-odd condition* $\ell(t,+1) - \ell(t,-1) = -t$ then $\mathcal{R}(g)$ can be approximated by (du Plessis et al., 2015)

$$\hat{\mathcal{R}}_{pu}^{(2)}(g) = -\pi_p\frac{1}{n_p}\sum_{i=1}^{n_p}g(x_i^p) + \hat{\mathcal{R}}_u^-(g). \tag{4.6}$$

Lastly, Kiryo et al. (2017) proposed a *non-negative* PU risk estimator

$$\hat{\mathcal{R}}_{pu}^{(3)}(g) = \pi_p\hat{\mathcal{R}}_p^+(g) + \max\{0, \hat{\mathcal{R}}_u^-(g) - \pi_p\hat{\mathcal{R}}_p^-(g)\}, \tag{4.7}$$

where $\hat{\mathcal{R}}_p^-(g) = \frac{1}{n_p}\sum_{i=1}^{n_p}\ell(g(x_i^p),-1)$. Note that $\hat{\mathcal{R}}_{pu}^{(3)}(g)$ is a biased estimator.

NU risk estimator. Similarly, considering NU learning when $\mathcal{D}_{\mathrm{P}}$ is unavailable (Sakai et al., 2017), NU risk estimators can be formulated by combining the equation $\pi_p\mathcal{R}_p^+(g) = \mathcal{R}_u^+(g) - \pi_n\mathcal{R}_n^+(g)$ (which is derived from (4.1)) and (4.2) to obtain

$$\mathcal{R}(g) = -\pi_n(\mathcal{R}_n^+(g) - \mathcal{R}_n^-(g)) + \mathcal{R}_u^+(g). \tag{4.8}$$

Here, we consider two estimators for (4.8). First, with a loss satisfying the *symmetric condition*, we have a nonconvex NU risk estimator

$$\hat{\mathcal{R}}_{nu}^{(1)}(g) = 2\pi_n \hat{\mathcal{R}}_n^-(g) - \pi_n + \hat{\mathcal{R}}_u^+(g), \tag{4.9}$$

where $\hat{\mathcal{R}}_n^-(g)$ is defined in (4.3) and $\hat{\mathcal{R}}_u^+(g) = \frac{1}{n_u}\sum_{i=1}^{n_u} \ell(g(x_i^u), +1)$. Second, with a loss satisfying the *linear-odd condition*, we get a convex NU risk estimator

$$\hat{\mathcal{R}}_{nu}^{(2)}(g) = \pi_n \frac{1}{n_n}\sum_{i=1}^{n_n} g(x_i^n) + \hat{\mathcal{R}}_u^+(g). \tag{4.10}$$

Finally, Sakai et al. (2017) proposes to use a linear combination between the PN, the NU, and the PU risk of du Plessis et al. (2014, 2015). However, they only focused on the set of linear classifiers with two specific losses – the (scaled) ramp loss and the truncated (scaled) squared loss (Sakai et al., 2017, Section 4.1). On the contrary, we consider a more general setting and also propose methods to choose appropriate regularisation to avoid negative empirical risks.

4.4. The proposed semi-supervised AD methods

In the previous section, we presented risk estimators for the PU learning problem where $\mathcal{D}_{\mathrm{N}}$ is unavailable. Let us consider the setting where we have access to $\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{N}}$ as well as $\mathcal{D}_{\mathrm{U}}$. Considering semi-supervised AD as a binary classification problem from X to $Y \in \{+1, -1\}$, our goal is to propose risk estimators for the risk in (4.2). Specifically, we propose two risk estimators for semi-supervised AD that lead to two risk-based AD methods.

If we take a *convex combination* of (4.2) and (4.8), we obtain

$$\begin{aligned}\mathcal{R}(g) &= a(-\pi_n(\mathcal{R}_n^+(g) - \mathcal{R}_n^-(g)) + \mathcal{R}_u^+(g)) + (1-a)(\pi_p\mathcal{R}_p^+(g) + \pi_n\mathcal{R}_n^-(g)) \\ &= a\mathcal{R}_u^+(g) + (1-a)\pi_p\mathcal{R}_p^+(g) + \pi_n\mathcal{R}_n^-(g) - a\pi_n\mathcal{R}_n^+(g),\end{aligned} \tag{4.11}$$

where $a \in (0, 1)$.

The empirical version of (4.11) yields the following linear combination of PN and NU risk estimators:

$$\hat{\mathcal{R}}_s^{(2)}(g) = a\hat{\mathcal{R}}_u^+(g) + (1-a)\pi_p\hat{\mathcal{R}}_p^+(g) + \pi_n\hat{\mathcal{R}}_n^-(g) - a\pi_n\hat{\mathcal{R}}_n^+(g). \tag{4.12}$$

Note that $\hat{\mathcal{R}}_s^{(2)}$ may take negative values when ℓ is unbounded due to the negative term $-a\pi_n\hat{\mathcal{R}}_n^+(g)$. Thus, we propose conditions in Theorem 4.4.1) to choose an appropriate regularisation for $\hat{\mathcal{R}}_s^{(2)}$ to avoid negative empirical risks.

Inspired by $\hat{\mathcal{R}}_{pu}^{(3)}(g)$ in (4.7), we also propose the following nonnegative risk estimator:

$$\hat{\mathcal{R}}_s^{(1)}(g) = \pi_n \hat{\mathcal{R}}_n^-(g) + (1-a)\pi_p \hat{\mathcal{R}}_p^+(g) + a \max\{0, \hat{\mathcal{R}}_u^+(g) - \pi_n \hat{\mathcal{R}}_n^+(g)\}, \quad (4.13)$$

where the max term is introduced since $\mathcal{R}_u^+(g) - \pi_n \mathcal{R}_n^+(g) = \pi_p \mathcal{R}_p^+(g)$ must be nonnegative. Note that $\hat{\mathcal{R}}_{pu}^{(3)}(g)$ was designed for the PU learning problem, while we propose $\hat{\mathcal{R}}_s^{(1)}(g)$ for the AD problem, which often assumes anomalies are rare. In other words, we put more emphasis on $\hat{\mathcal{R}}_u^+(g)$ rather than $\hat{\mathcal{R}}_u^-(g)$.

In Section 4.5, we will establish the theoretical estimation error bounds and excess risk bounds for the minimisers of both $\min_{g\in\mathcal{G}} \hat{\mathcal{R}}_s^{(1)}(g)$ and $\min_{g\in\mathcal{G}} \hat{\mathcal{R}}_s^{(2)}(g)$, where $\mathcal{G}$ is some class function. We now present the practical optimisation problems involved when using $\hat{\mathcal{R}}_s^{(1)}(g)$ and $\hat{\mathcal{R}}_s^{(2)}(g)$.

Optimisation problems. Suppose $g(\cdot; w)$ is parameterized by w, which needs to be learned from $\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{N}}$ and $\mathcal{D}_{\mathrm{U}}$. When $\hat{\mathcal{R}}_s^{(1)}(g)$ (4.13) is used, the corresponding optimisation problem for AD is

$$\begin{aligned}\min_w \Big\{ &\frac{\pi_n}{n_n} \sum_{i=1}^{n_n} \ell(g(x_i^n, w), -1) + \frac{(1-a)\pi_p}{n_p} \sum_{i=1}^{n_p} \ell(g(x_i^p, w), +1) \\ &+ a \max\Big\{0, \frac{1}{n_u} \sum_{i=1}^{n_u} \ell(g(x_i^u, w), +1) - \frac{\pi_n}{n_n} \sum_{i=1}^{n_n} \ell(g(x_i^n, w), +1)\Big\} + \lambda \mathbf{R}(w)\Big\},\end{aligned} \quad (4.14)$$

where $\mathbf{R}$ is some regularizer, and $\lambda \geq 0$ is regularization parameter. And when $\hat{\mathcal{R}}_s^{(2)}(g)$ in (4.12) is used, the corresponding optimization problem is

$$\begin{aligned}\min_w \Big\{ &\frac{a}{n_u} \sum_{i=1}^{n_u} \ell(g(x_i^u, w), +1) + \frac{(1-a)\pi_p}{n_p} \sum_{i=1}^{n_p} \ell(g(x_i^p, w), +1) \\ &+ \frac{\pi_n}{n_n} \sum_{i=1}^{n_n} \ell(g(x_i^n, w), -1) - \frac{a\pi_n}{n_n} \sum_{i=1}^{n_n} \ell(g(x_i^n, w), +1) + \lambda \mathbf{R}(w)\Big\}.\end{aligned} \quad (4.15)$$

Unfortunately, the objective of (4.15) is not guaranteed to be nonnegative due to the negative term $-\frac{a\pi_n}{n_n} \sum_{i=1}^{n_n} \ell(g(x_i^n, w), +1)$. As pointed out by Kiryo et al. (2017), this can lead to serious overfitting problems. The following theorem provides methods to choose the regularisation parameters such that the nonnegativity of the objective of (4.15) is guaranteed.

Theorem 4.4.1. *Suppose there exist positive constants b_1, b_2 and b_3 such that*

$$\ell(t,-1)-\ell(t,+1)\geq -b_1|t|, \quad \textit{and} \quad \ell(t,-1)\geq b_2(b_3-|t|). \tag{4.16}$$

(i) We have

$$\frac{\pi_n}{n_n}\sum_{i=1}^{n_n}\ell(g(x_i^n,w),-1)-\frac{a\pi_n}{n_n}\sum_{i=1}^{n_n}\ell(g(x_i^n,w),+1)$$
$$\geq (1-a)\pi_n b_2 b_3-((1-a)b_2+ab_1)\frac{\pi_n}{n_n}\sum_{i=1}^{n_n}|g(x_i^n,w)|.$$

(ii) If we choose λ and $\mathbf{R}$ such that

$$\lambda\mathbf{R}(w)\geq((1-a)b_2+ab_1)\frac{\pi_n}{n_n}\sum_{i=1}^{n_n}|g(x_i^n,w)|-(1-a)\pi_n b_2 b_3 \tag{4.17}$$

then the objective of (4.15) *is always nonnegative.*

(iii) Consider the specific case $g(x)=\langle w,\phi(x)\rangle$, where $\phi:\mathbb{R}^d\to\mathbb{R}^q$ is a feature map transformation. The following choices of λ and $\mathbf{R}$ satisfy (4.17)*.*

- $\mathbf{R}(w)=\|w\|_2^2$ *and* $\lambda\geq\frac{((1-a)b_2+ab_1)^2\pi_n c^2}{4(1-a)b_2b_3}$, *where* $c=\max\{\|\phi(x_i^n)\|_2 : i=1,\ldots,n_n\}$ *(note that, in practice, we can scale the data to have* $c=1$*).*
- $\mathbf{R}(w)=\|w\|_1$ *and* $\lambda\geq c_\infty((1-a)b_2b_3+ab_1)\pi_n$, *where* $c_\infty=\max\{\|\phi(x_i^n)\|_\infty : i=1,\ldots,n_n\}$ *(in practice, we can scale the data to have* $c_\infty=1$*).*

In Table 4.1 we give examples of loss functions that satisfy (4.16). For the proof, we refer the reader to (Hien et al., 2024, Appendix A.1).

We consider both a shallow and a deep implementation of our proposed risk-based AD (rAD) method. In the following, $\hat{\pi}_p$ and $\hat{\pi}_n=1-\hat{\pi}_p$ will denote estimates of the real class-prior probabilities π_p and π_n, respectively.

A shallow rAD method. We plug in $g(x,w)=\langle w,\phi(x)\rangle$ in (4.15) (the empirical version of (4.12)), where $\phi:\mathbb{R}^d\to\mathbb{R}^q$ is a feature map transformation, and choose the regularization method following the conditions proposed in

Table 4.1: Examples of loss functions satisfying (4.16)

Name	$\ell(t,y)=\ell(z)$ with $z=ty$	Bounded	(b_1,b_2,b_3)
Hinge loss	$\max\{0,1-z\}$	×	$(2,1,1)$
Double hinge loss	$\max\{0,(1-z)/2,-z\}$	×	$(1,1/2,1)$
Squared loss	$\frac{1}{2}(z-1)^2$	×	$(2,1/2,1/2)$
Modified Huber loss	$\begin{cases}\max\{0,1-z\}^2 & \text{if } z\geq -1\\ -4z & \text{otherwise}\end{cases}$	×	$(4,1,1/2)$
Logistic loss	$\ln(1+\exp(-z))$	×	$(1,1,\ln 2)$
Sigmoid loss	$1/(1+\exp(z))$	✓	$(1,1/2,1)$
Ramp loss	$\max\{0,\min\{1,(1-z)/2\}\}$	✓	$(1,1/2,1)$

Theorem 4.4.1 (iii). Specifically, we solve the following minimisation problem:

$$\min_{w}\Big\{\frac{a}{n_u}\sum_{i=1}^{n_u}\ell(w^\top\phi(x_i^u),+1)+\frac{(1-a)\hat{\pi}_p}{n_p}\sum_{i=1}^{n_p}\ell(w^\top\phi(x_i^p),+1) + \frac{\hat{\pi}_n}{n_n}\sum_{i=1}^{n_n}\ell(w^\top\phi(x_i^n),-1)-\frac{a\hat{\pi}_n}{n_n}\sum_{i=1}^{n_n}\ell(w^\top\phi(x_i^n),+1)+\lambda\mathbf{R}(w)\Big\}. \tag{4.18}$$

A deep rAD method. We plug in $g(x)=\phi(x;\mathcal{W})$ in (4.14) (the empirical version of (4.13)), where $\mathcal{W}$ is a set of weights of a deep neural network. Specifically, we train a deep neural network by solving the following optimisation problem:

$$\min_{\mathcal{W}}\Big\{\frac{\hat{\pi}_n}{n_n}\sum_{i=1}^{n_n}\ell(\phi(x_i^n;\mathcal{W}),-1)+\frac{(1-a)\hat{\pi}_p}{n_p}\sum_{i=1}^{n_p}\ell(\phi(x_i^p;\mathcal{W}),+1) + a\max\big\{0,\frac{1}{n_u}\sum_{i=1}^{n_u}\ell(\phi(x_i^u;\mathcal{W}),+1)-\frac{\hat{\pi}_n}{n_n}\sum_{i=1}^{n_n}\ell(\phi(x_i^n;\mathcal{W}),+1)\big\}+\lambda\mathbf{R}(\mathcal{W})\Big\}, \tag{4.19}$$

where $\mathbf{R}$ can be any regularizer. Note that we focus on these specific implementations, but it is also possible to consider a deep model with (4.15) or a shallow model with (4.14). In our initial numerical findings, we have observed that the shallow model in (4.18) frequently yields better results compared to the shallow model with (4.14), while the deep model in (4.19) outperforms the deep model with (4.15).

4.5. Risk bounds

In this section, we establish the estimation error bound and the excess risk bound for $\hat{g}^1$ and $\hat{g}^2$ which are the empirical risk minimizers obtained by $\min_{g\in\mathcal{G}} \hat{\mathcal{R}}_s^{(1)}(g)$ and $\min_{g\in\mathcal{G}} \hat{\mathcal{R}}_s^{(2)}(g)$, where $\mathcal{G}$ is a function class.

Let g^* be the true risk minimiser, that is, $g^* = \arg\min_{g\in\mathcal{G}} \mathcal{R}(g)$. Throughout this section, we assume that (i) $\mathcal{G} = \{g \,\big|\, \|g\|_\infty \leq C_g\}$ for some constant C_g, and (ii) there exists $C_\ell > 0$ such that $\sup_{|t|\leq C_g} \max_y \ell(t,y) \leq C_\ell$. It is worth noting that the set of linear classifiers with bounded norms and feature maps is a special case of Condition (i)

$$\mathcal{G} = \{g(x,w) = \langle w, \phi(x)\rangle_{\mathcal{H}} \,\big|\, \|w\|_{\mathcal{H}} \leq C_w, \|\phi(x)\|_{\mathcal{H}} \leq C_\phi\}, \tag{4.20}$$

where $\mathcal{H}$ is a Hilbert space, ϕ is a feature map, and C_w and C_ϕ are positive constants.

Given g, $\hat{\mathcal{R}}_s^{(2)}(g)$ is an unbiased estimator of $\mathcal{R}(g)$ but $\hat{\mathcal{R}}_s^{(1)}$ is a biased estimator. The following proposition estimates the bias of $\hat{\mathcal{R}}_s^{(1)}$ (see Inequality (4.21)) and shows that, for a fixed g, $\hat{\mathcal{R}}_s^{(1)}(g)$ and $\hat{\mathcal{R}}_s^{(2)}(g)$ converge to $\mathcal{R}(g)$ with the rate $O\big(\frac{\pi_n}{\sqrt{n_n}} + \frac{\pi_p}{\sqrt{n_p}} + \frac{a}{\sqrt{n_u}}\big)$ (see Inequality (4.22) and (4.23)).

Proposition 4.5.1. *Consider a fixed classifier g. Suppose there exists $\rho_g > 0$ such that $\mathcal{R}_p^+(g) \geq \rho_g > 0$ and denote $\epsilon_g = a\pi_n C_\ell \exp\Big(-\frac{2\pi_p^2\rho_g^2}{C_\ell^2(1/n_u + \pi_n^2/n_n)}\Big)$. Then the bias of $\hat{\mathcal{R}}_s^{(1)}(g)$ satisfies*

$$0 \leq \mathbb{E}[\hat{\mathcal{R}}_s^{(1)}(g)] - \mathcal{R}(g) \leq \epsilon_g. \tag{4.21}$$

Moreover, for any $\delta > 0$, we have the following inequalities hold with probability at least $1-\delta$

$$|\hat{\mathcal{R}}_s^{(2)}(g) - \mathcal{R}(g)| \leq C_\ell\sqrt{\ln(2/\delta)/2}\big(\frac{(1+a)\pi_n}{\sqrt{n_n}} + \frac{(1-a)\pi_p}{\sqrt{n_p}} + \frac{a}{\sqrt{n_u}}\big), \tag{4.22}$$

and

$$|\hat{\mathcal{R}}_s^{(1)}(g) - \mathcal{R}(g)| \leq C_\ell\sqrt{\ln(2/\delta)/2}\big(\frac{(1+a)\pi_n}{\sqrt{n_n}} + \frac{(1-a)\pi_p}{\sqrt{n_p}} + \frac{a}{\sqrt{n_u}}\big) + \epsilon_g. \tag{4.23}$$

For the proof, we refer the reader to (Hien et al., 2024, Appendix A.2).

Estimation error bound. The Rademacher complexity of $\mathcal{G}$ for a sample of size n drawn from some distribution q (Mohri et al., 2018) is defined by $\mathfrak{R}_{n,q}(\mathcal{G}) := \mathbb{E}_{Z\sim q^n}[\mathbb{E}_\sigma[\sup_{g\in\mathcal{G}}(\frac{1}{n}\sum_{i=1}^n \sigma_i g(Z_i))]]$, where $Z_1,\ldots,Z_n$ are i.i.d random variables following distribution q, $Z=(Z_1,\ldots,Z_n)$, $\sigma_1,\ldots,\sigma_n$ are independent random variables uniformly chosen from $\{-1,1\}$, and $\sigma=(\sigma_1,\ldots,\sigma_n)$. Similarly to Kiryo et al. (2017, Theorem 4), we can establish the following estimation error bound for $\hat{g}^1$.

Theorem 4.5.2 (Estimation error bound for $\hat{g}^1$)**.** *We assume that (i) there exists $\rho>0$ such that $\mathcal{R}_p^+(g)\geq\rho$ for all $g\in\mathcal{G}$, (ii) if $g\in\mathcal{G}$ then $-g\in\mathcal{G}$, and (iii) $t\mapsto\ell(t,1)$ and $t\mapsto\ell(t,-1)$ are L_ℓ-Lipschitz continuous over $\{t:|t|\leq C_g\}$. Denote $\epsilon=a\pi_n C_\ell\exp\Big(-\frac{2\pi_p^2\rho^2}{C_\ell^2(1/n_u+\pi_n^2/n_n)}\Big)$. For any $\delta>0$, the following inequality hold with probability at least $1-\delta$*

$$\mathcal{R}(\hat{g}^1)-\mathcal{R}(g^*)\leq 8(1+a)\pi_n L_\ell\mathfrak{R}_{n_n,P_X^-}(\mathcal{G})+8(1-a)\pi_p L_\ell\mathfrak{R}_{n_p,P_X^+}(\mathcal{G})+ 8aL_\ell\mathfrak{R}_{n_u,P_X^u}(\mathcal{G})+2C_\ell\sqrt{\ln(2/\delta)/2}\Big(\frac{(1+a)\pi_n}{\sqrt{n_n}}+\frac{(1-a)\pi_p}{\sqrt{n_p}}+\frac{a}{\sqrt{n_u}}\Big)+2\epsilon. \tag{4.24}$$

For the proof, we refer the reader to (Hien et al., 2024, Appendix A.3). By using basic uniform deviation bound (Mohri et al., 2018), the McDiarmid's inequality (McDiarmid, 1989), and Talagrand's contraction lemma (Ledoux and Talagrand, 1991), we can prove the following estimation error bound for $\hat{g}^2$.

Theorem 4.5.3 (Estimation error bound for $\hat{g}^2$)**.** *Assume that $t\mapsto\ell(t,1)$ and $t\mapsto\ell(t,-1)$ are L_ℓ-Lipschitz continuous over $\{t:|t|\leq C_g\}$. For any small $\delta>0$, the following inequality hold with probability at least $1-\delta$*

$$\mathcal{R}(\hat{g}^2)-\mathcal{R}(g^*)\leq 4(1-a)\pi_p L_\ell\mathfrak{R}_{n_p,P_X^+}(\mathcal{G})+4(a+1)\pi_n L_\ell\mathfrak{R}_{n_n,P_X^-}(\mathcal{G}) + 4aL_\ell\mathfrak{R}_{n_u,P_X^u}(\mathcal{G})+2C_\ell\sqrt{\ln(6/\delta)/2}\Big(\frac{(1-a)\pi_p}{\sqrt{n_p}}+\frac{(1+a)\pi_n}{\sqrt{n_n}}+\frac{a}{\sqrt{n_u}}\Big). \tag{4.25}$$

For the proof, we refer the reader to (Hien et al., 2024, Appendix A.4). Note that Theorem 4.5.3 explicitly states the error bound for $\hat{g}^2$ with any loss function that satisfies the Lipschitz continuity assumption. The (scaled) ramp loss and the truncated (scaled) squared loss considered in (Sakai et al., 2017) have $L_\ell=1/2$.

Excess risk bound. The excess risk focuses on the error due to the use of surrogates for the 0-1 loss function. Denote $I^* = \inf_{g \in \mathcal{F}} I(g)$ and $\mathcal{R}^* = \inf_{g \in \mathcal{F}} \mathcal{R}(g)$, where $\mathcal{F}$ is the set of all measurable functions. By using (Bartlett et al., 2006, Theorem 1) (see (B.1) in the Appendix), Theorem 4.5.2 and 4.5.3, we can derive the following excess risk bound for $\hat{g}^1$ and $\hat{g}^2$.

Corollary 4.5.4. *If ℓ is a classification-calibrated loss (see Definition B.1.1 in the supp. material), then there exists a convex, invertible, and nondecreasing transformation ψ_ℓ with $\psi_\ell(0) = 0$ and the following inequalities hold with probability at least $1 - \delta$*

$$I(\hat{g}^1) - I^* \leq \psi_\ell^{-1}(B_1 + \mathcal{R}(g^*) - \mathcal{R}^*), \quad I(\hat{g}^2) - I^* \leq \psi_\ell^{-1}(B_2 + \mathcal{R}(g^*) - \mathcal{R}^*),$$

where B_1 and B_2 are the right hand side of (4.24) *and* (4.25)*, respectively.*

4.6. Experiments

4.6.1 Experiments with shallow rAD

Baseline methods and implementation. We compare rAD with OC-SVM (Schölkopf et al., 2001), ECOD (Li et al., 2023b), COPOD (Li et al., 2020), semi-supervised OC-SVM (Munoz-Mari et al., 2010), and the PU methods using the risk estimator $\hat{\mathcal{R}}_{pu}(g)$ given in (4.4). Note that $\hat{\mathcal{R}}_{pu}(g) = \hat{\mathcal{R}}_{pu}^{(1)}(g)$ given in (4.5) if ℓ satisfies the symmetric condition, and $\hat{\mathcal{R}}_{pu}(g) = \hat{\mathcal{R}}_{pu}^{(2)}(g)$ given in (4.6) if ℓ satisfies the linear-odd condition. We implement rAD and PU methods with 3 losses: squared loss, hinge loss, and modified Huber loss. For rAD, we use l_2 regularisation and take $\phi(x) = x$ in (4.18), i.e. no kernel is used. We set $a = 0.1$ and $\hat{\pi}_p = 0.8$ ($\hat{\pi}_n = 0.2$) as default values for both the shallow rAD and the PU methods. Note that the real π_n of the datasets can be different.

Datasets. We test the algorithms on 26 classical AD benchmark datasets from (Han et al., 2022), whose π_n ranges from 0.02 to 0.4. The real π_n of the datasets are given in the first column of Table 4.2. We randomly split each dataset 30 times into train and test data with a ratio of 7:3, i.e. we have 30 trials for each dataset. Then, for each trial, we randomly select 5% of the train data to make the labelled data and keep the remaining 95% as unlabeled data.

Experimental results In Table 4.2, we report the mean and standard error (SE) of the AUC (area under the ROC curve) over 30 trials of the 26 benchmark

Table 4.2: Mean (and SE $\times 10^2$) of the AUC over 30 trials. The best means are highlighted in bold. d, n, and π_n denote the feature dimension, the sample size of the dataset, and the ratio of negative samples in the dataset.

dataset (d, n, π_n)	rAD			PU			OC-SVM	ECOD	COPOD	semi-OC-SVM
	square	hinge	m-Huber	square	hinge	m-Huber				
pendigits (16, 6870, 0.02)	**0.98**(0.22)	**0.98**(0.22)	**0.98**(0.22)	0.78(4.79)	0.78(4.83)	0.78(4.77)	0.86(0.31)	0.92(0.16)	0.90(0.17)	0.82(2.48)
mammography (6, 11 183, 0.02)	**0.91**(0.29)	**0.91**(0.29)	**0.91**(0.29)	0.87(1.49)	0.87(1.49)	0.87(1.48)	0.77(0.47)	**0.91**(0.30)	**0.91**(0.29)	0.61(2.97)
optdigits (64, 5216, 0.03)	**1.00**(0.07)	**1.00**(0.07)	**1.00**(0.06)	0.76(2.93)	0.75(3.02)	0.77(2.89)	0.46(0.53)	0.60(0.40)	0.68(0.35)	0.83(1.82)
Stamps (9, 340, 0.09)	0.90(1.44)	0.90(1.24)	0.90(1.46)	0.76(3.37)	0.77(3.76)	0.71(4.21)	0.65(1.74)	0.88(0.64)	**0.93**(0.44)	0.69(3.85)
cardio (21, 1831, 0.10)	0.92(2.03)	0.89(2.12)	0.93(1.99)	0.83(2.09)	0.81(2.31)	0.84(1.93)	0.87(0.32)	**0.94**(0.23)	0.93(0.21)	0.79(1.32)
InternetAds (1555, 1966, 0.19)	0.73(3.00)	**0.87**(0.49)	0.75(0.92)	0.64(3.45)	0.77(0.57)	0.77(0.66)	0.60(0.54)	0.68(0.46)	0.68(0.46)	0.64(0.97)
Cardiotocography (21, 2114, 0.22)	0.86(1.32)	0.84(1.68)	**0.88**(1.10)	0.81(1.86)	0.79(2.01)	0.82(1.75)	0.72(0.41)	0.78(0.33)	0.66(0.40)	0.81(0.80)
magic.gamma (10, 19 020, 0.35)	**0.78**(0.47)	**0.78**(0.49)	**0.78**(0.45)	**0.78**(0.69)	0.77(0.71)	**0.78**(0.68)	0.56(0.18)	0.64(0.12)	0.68(0.11)	0.54(0.32)
SpamBase (57, 4207, 0.40)	**0.94**(0.15)	**0.94**(0.15)	**0.94**(0.16)	0.93(0.20)	0.93(0.19)	0.93(0.21)	0.54(0.28)	0.66(0.21)	0.69(0.21)	0.64(0.85)
satimage-2 (36, 5803, 0.01)	0.99(0.17)	0.99(0.16)	0.99(0.17)	0.80(4.40)	0.77(4.29)	0.82(4.47)	**1.00**(0.09)	0.96(0.37)	0.97(0.31)	0.51(4.52)
thyroid (6, 3772, 0.02)	**1.00**(0.05)	**1.00**(0.04)	**1.00**(0.05)	0.86(2.95)	0.87(2.95)	0.86(2.89)	0.93(0.31)	0.98(0.07)	0.94(0.15)	0.70(2.25)
vowels (12, 1456, 0.03)	0.85(1.45)	0.82(1.59)	**0.86**(1.42)	0.63(2.59)	0.62(2.45)	0.64(2.66)	0.72(1.40)	0.58(1.20)	0.49(1.12)	0.69(2.55)
Waveform (21, 3443, 0.03)	0.83(1.49)	0.81(1.80)	**0.84**(1.33)	0.66(2.67)	0.66(2.68)	0.67(2.64)	0.67(0.70)	0.61(0.71)	0.74(0.53)	0.78(1.11)
CIFAR10-1 (512, 5263, 0.05)	**0.77**(0.84)	**0.77**(0.84)	**0.77**(0.86)	0.59(1.70)	0.59(1.83)	0.59(1.63)	0.64(0.50)	0.53(0.45)	0.49(0.45)	0.74(0.75)
SVHN-1 (512, 10 000, 0.05)	0.83(0.46)	0.83(0.45)	**0.84**(0.47)	0.69(1.42)	0.69(1.57)	0.69(1.33)	0.66(0.27)	0.65(0.30)	0.63(0.31)	0.71(0.77)
20news-1 (768, 2514, 0.05)	0.64(1.56)	0.61(1.14)	**0.68**(1.70)	0.51(1.57)	0.52(1.21)	0.53(1.57)	0.52(0.71)	0.48(0.76)	0.48(0.71)	0.65(1.54)
agnews-1 (768, 10000, 0.05)	0.97(0.27)	0.93(0.69)	**0.98**(0.18)	0.79(1.28)	0.74(1.53)	0.81(1.12)	0.76(0.25)	0.75(0.24)	0.76(0.24)	0.89(0.40)
amazon (768, 10000, 0.05)	0.80(0.76)	0.76(0.87)	**0.82**(0.69)	0.63(0.98)	0.60(1.06)	0.63(0.95)	0.54(0.40)	0.51(0.39)	0.48(0.39)	0.78(0.56)
imdb (768, 10000, 0.05)	0.82(0.73)	0.77(1.00)	**0.83**(0.65)	0.63(1.30)	0.61(1.22)	0.65(1.35)	0.50(0.43)	0.49(0.42)	0.50(0.42)	0.78(0.60)
yelp (768, 10000, 0.05)	0.89(0.85)	0.83(1.36)	**0.90**(0.73)	0.70(1.63)	0.67(1.67)	0.71(1.55)	0.61(0.31)	0.55(0.32)	0.52(0.33)	0.82(0.63)
mnist (100, 7603, 0.09)	**0.96**(0.14)	**0.96**(0.15)	**0.96**(0.14)	0.92(0.59)	0.92(0.57)	0.92(0.60)	0.80(0.24)	0.75(0.23)	0.78(0.22)	0.85(0.55)
campaign (62, 41 188, 0.11)	**0.85**(0.16)	**0.85**(0.17)	**0.85**(0.16)	0.84(0.30)	0.84(0.30)	0.84(0.30)	0.68(0.12)	0.77(0.09)	0.78(0.09)	0.77(0.41)
vertebral (6, 240, 0.13)	0.72(2.57)	**0.75**(2.64)	0.74(2.58)	0.59(2.60)	0.58(2.68)	0.60(2.65)	0.48(2.18)	0.43(1.38)	0.35(1.08)	0.68(2.59)
landsat (36, 6435, 0.21)	0.73(0.20)	0.73(0.21)	0.73(0.19)	0.70(0.52)	0.70(0.51)	0.71(0.51)	0.35(0.28)	0.36(0.25)	0.42(0.24)	**0.76**(0.60)
satellite (36, 6435, 0.32)	**0.80**(0.22)	**0.80**(0.22)	**0.80**(0.22)	**0.80**(0.26)	**0.80**(0.25)	**0.80**(0.27)	0.55(0.30)	0.59(0.25)	0.64(0.23)	0.67(0.72)
fault (27, 1941, 0.35)	**0.64**(0.87)	0.62(0.76)	**0.64**(0.91)	0.58(1.30)	0.58(1.27)	0.59(1.29)	0.52(0.52)	0.47(0.47)	0.46(0.49)	0.57(0.99)

datasets. We observe that, on average, rAD outperforms the PU methods, OC-SVM methods, ECOD, and COPOD. The difference between the AUC of rAD and that of PU is large on the datasets with $\pi_n \leq 0.2$, but it is small when π_n is larger. We also notice that rAD with modified Huber loss often gives better

Table 4.3: AUC means of shallow rAD over 30 trials for different $\hat{\pi}_p$. The significant changes in the AUC means are highlighted in bold.

Dataset	square/$\hat{\pi}_p$				hinge/$\hat{\pi}_p$				m-Huber/$\hat{\pi}_p$			
	$1-\pi_n$	0.9	0.7	0.6	$1-\pi_n$	0.9	0.7	0.6	$1-\pi_n$	0.9	0.7	0.6
pendigits	0.96	0.98	0.98	0.98	0.94	0.98	0.98	0.98	0.97	0.98	0.98	0.98
mammography	0.90	0.91	0.91	0.91	0.90	0.91	0.91	0.91	0.90	0.91	0.91	0.91
optdigits	0.96	0.99	0.997	0.997	**0.93**	0.99	0.997	0.997	0.98	0.996	0.998	0.998
Stamps	0.80	0.80	0.82	0.82	0.81	0.81	0.81	0.80	0.80	0.80	0.80	0.80
cardio	0.91	0.91	0.92	0.92	0.87	0.88	0.88	0.89	0.92	0.93	0.94	0.94
InternetAds	0.77	0.77	0.70	**0.60**	0.86	0.85	0.86	0.86	0.87	0.87	0.86	0.86
Cardiotocography	0.89	0.88	0.89	0.89	0.87	0.85	0.87	0.87	0.90	0.90	0.90	0.90
magic.gamma	0.78	0.77	0.78	0.78	0.78	0.77	0.78	0.78	0.78	0.78	0.78	0.78
SpamBase	0.94	0.94	0.94	0.94	0.94	0.93	0.94	0.94	0.94	0.94	0.94	0.94

results than rAD with square loss and hinge loss.

Sensitivity analysis for $\hat{\pi}_p$. With $a = 0.1$, we run shallow rAD on the 30 trials for $\hat{\pi}_p \in \{1-\pi_n, 0.9, 0.7, 0.6\}$ (when $\hat{\pi}_p = 1-\pi_n$, no approximation is made). The results are reported in Table 4.3 for 9 benchmark datasets, and the results of the 17 remaining datasets are given in Table B.1 in the supp. material. From Table 4.2– B.1, we can see that we can obtain good results even if $\hat{\pi}_p$ is different from π_p. In fact, with $a = 0.1$, we get worse AUC means when $\hat{\pi}_p$ is close to π_p. The combination $(a, \hat{\pi}_p) = (0.1, 0.8)$ or $(a, \hat{\pi}_p) = (0.1, 0.7)$ seem to be good choices across the datasets. Compared to the other two losses, we found the modified Huber loss to be robust to the values of $\hat{\pi}_p$.

Sensitivity analysis for a. We run shallow rAD (with fixed $\hat{\pi}_p = 0.8$) on the 30 trials of each dataset for $a \in \{0.3, 0.7, 0.9\}$. The results are reported in Table 4.4 for the 9 benchmark datasets, and the results of the 17 remaining datasets are given in Table B.2 in the appendix. From Table 4.2, 4.4 and B.2, we can see that the AUC means do not decrease significantly when we increase a (except for the dataset InternetAds). Hence, shallow rAD with $\hat{\pi}_p = 0.8$ is also robust to different values of a.

4.6.2 Experiments with deep rAD

Baseline methods and implementation. We compare deep rAD with the Latent Outlier Exposure method (LOE) Qiu et al. (2022), the deep semi-supervised AD method (deep SAD) Ruff et al. (2020) and the PU learning method with nonnegative risk estimator and sigmoid loss (nnPU) Kiryo et al.

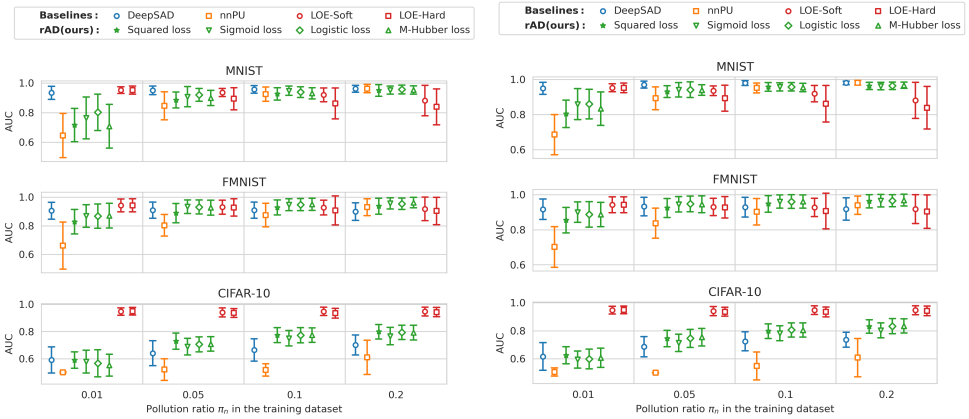
Table 4.4: AUC means of shallow rAD over 30 trials for different a. The significant changes in the AUC means are highlighted in bold.

Dataset	square/a			hinge/a			m-Huber/a		
	0.3	0.7	0.9	0.3	0.7	0.9	0.3	0.7	0.9
pendigits	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
mammography	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
optdigits	0.997	0.995	0.99	0.996	0.995	0.99	0.997	0.996	0.99
Stamps	0.83	0.82	0.82	0.81	0.82	0.81	0.80	0.81	0.81
cardio	0.92	0.91	0.91	0.88	0.87	0.85	0.93	0.93	0.93
InternetAds	0.79	**0.69**	**0.62**	0.87	0.85	**0.77**	0.83	**0.71**	**0.65**
Cardiotocography	0.87	0.87	0.87	0.86	0.85	0.83	0.90	0.89	0.88
magic.gamma	0.78	0.78	0.78	0.78	0.78	0.77	0.78	0.78	0.78
SpamBase	0.94	0.94	0.93	0.94	0.94	0.93	0.94	0.94	0.94

(2017). For deep SAD and nnPU, we use default hyperparameter settings and network architectures as in their original implementation by the authors. We use the same network architectures as deep SAD for experiments on Fashion-MNIST and MNIST datasets. For experiments on CIFAR-10, the network architecture from nnPU is used. In deep rAD, the optimisation problem in (4.19) is solved using ADAM. We implement 4 losses for deep rAD: squared loss, sigmoid loss, logistic loss, and modified Huber loss. We set $a = 0.1$ and $\hat{\pi}_p = 0.8$ (thus $\hat{\pi}_n = 0.2$) as default values for deep rAD.

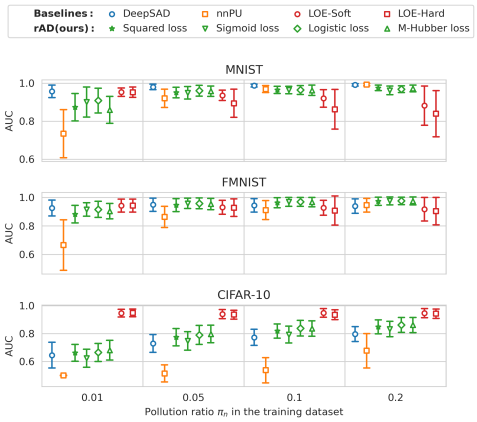
Datasets. We test the algorithms on 3 benchmark k-classes-out datasets: MNIST, Fashion-MNIST, and CIFAR-10 (all have 10 classes). We use AD setups following previous works (Chalapathy et al., 2019; Ruff et al., 2020): for each $\pi_n \in \{0.01, 0.05, 0.1, 0.2\}$, we set one of the ten classes to be a positive class, letting the remaining nine classes be anomalies and maintaining the ratio between normal instances and anomaly instances such that the setup has the required π_n (so we have 10 setups corresponding to 10 classes). We note that the anomalous data in our generation process can originate from more than one of the nine classes (unlike in the setup of deep SAD, where the anomaly is only from one of the nine classes). For each π_n, we repeat this generation process 2 times to get 20 AD setups (or 20 trials). Then, in each trial, we randomly choose γ_l (with $\gamma_l \in \{0.05, 0.1, 0.2\}$) portion of the train data to be labelled and keep the remaining $(1 - \gamma_l)$ portion as unlabeled data. Note that we make the labelled data for nnPU only from normal instances. To make labelled data for deep rAD and deep SAD, $(1 - \pi_n)$ portion is taken

from the nnPU labelled data (which contains only normal instances), and the remaining π_n portion from the anomalous instances. Hence, the number of labelled anomalous instances for deep rAD and deep SAD is about $(\gamma_l \times \pi_n)$ portion of the train data.

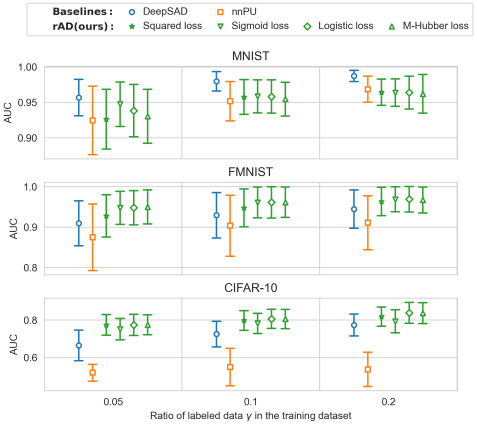


(a) Various π_n with $\gamma_l = 0.05$

(b) Various π_n with $\gamma_l = 0.1$



(c) Various π_n with default $\gamma_l = 0.2$



(d) Various γ_l with $\pi_n = 0.1$

Figure 4.1: AUC mean and std over 20 trials under various conditions.

Experiment results. In Figure 4.1a, we report the mean and standard deviation (std) of the AUC over 20 trials on the datasets with increasing pollution ratio π_n and default $\gamma_l = 0.05$. The results for $\gamma_l \in \{0.1, 0.2\}$ are given in Figures 4.1b and 4.1c. Figures 4.1a, 4.1b and 4.1c show that, on CIFAR-10, LOE performs the best and deep rAD methods on average provide better AUC than deep SAD and nnPU; deep rAD and deep SAD have similar performance

when $\pi_n = 0.01$ but their AUC difference is significant when π_n is increased. On FMNIST, deep rAD methods, on average, are better than the others when π_n is increased, but the AUC improvement is small. On MNIST, deep SAD is best, and when either π_n or γ_l is increased, deep rAD catches up with deep SAD, while LOE gives worse AUC than the others. Deep rAD with quadratic loss underperforms the other rAD methods on MNIST and FMNIST. On average, deep rAD with logistic loss performs best among the rAD methods. It is also interesting to note that in the presence of anomalies from multiple classes, the performance of deep SAD degrades over the performance reported in (Ruff et al., 2020). The degradation is more severe for CIFAR-10.

To observe the impact of the amount of labelled data, we report the results for the datasets with $\pi_n = 0.1$ and $\gamma_l \in \{0.05, 0.1, 0.2\}$ in Figure 4.1d. We observe that all the semi-supervised methods improve when we increase γ_l. From $\gamma_l = 0.05$ to $\gamma_l = 0.1$ (i.e., 5% more labelled data), deep rAD methods show a significant improvement in performance.

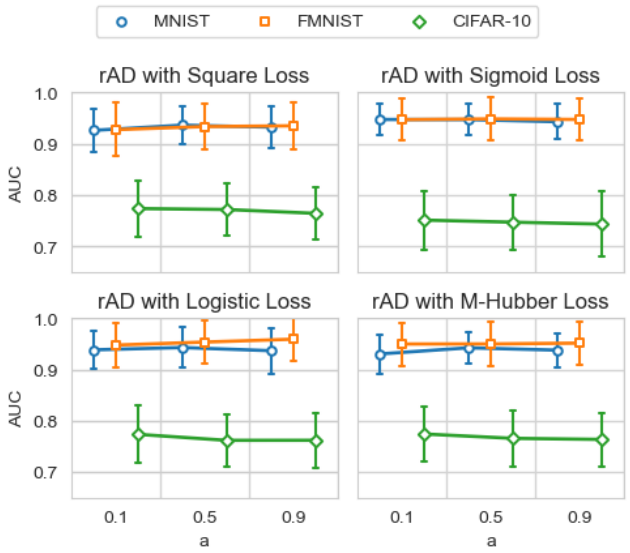


Figure 4.2: AUC mean and std over 20 trials with $\gamma_l = 0.05$ and $\pi_n = 0.1$

Sensitivity analysis for $\hat{\pi}_p$. We run deep rAD with $a = 0.1$ on the 20 trials of each dataset for $\hat{\pi}_p \in \{1 - \pi_n, 0.9, 0.7, \pi_n\}$ (when $\hat{\pi}_p = 1 - \pi_n$, it is an exact estimation of π_p, and when $\hat{\pi}_p = \pi_n$, we can say $\hat{\pi}_p$ is a bad estimation of π_p). We report the result in the appendix Table B.3. Again, we see that $\hat{\pi}_p$ is not necessarily a precise estimation of π_p; and $(a, \hat{\pi}_p) = (0.1, 0.8)$ or

$(a, \hat{\pi}_p) = (0.1, 0.7)$ are good settings. These results are consistent with the results of shallow rAD.

Sensitivity analysis for a. We fix $\hat{\pi}_p = 0.8$ and run deep rAD with additional values of $a \in \{0.5, 0.9\}$ ($a = 0.1$ is the default setting). We report the results for the datasets with $\pi_n = 0.1$ and $\gamma_l = 0.05$ in Figure 4.2. The results for the datasets with other values of π_n and γ_l are given in the supp. material. We observe that on CIFAR-10, AUC decreases when a is increased; however, the difference is not significant. On FMNIST and MNIST, deep rAD with $\hat{\pi}_p = 0.8$ is quite robust to the change of a.

4.7. Discussion

With semi-supervised classification based on risk estimators, we have introduced a shallow AD method equipped with suitable regularisation as well as a deep AD method. Theoretically, we have established the estimation error bounds and the excess risk bounds for the two risk minimisers. Empirically, the shallow AD methods show significant improvement over the baseline methods, while the deep AD methods compete favourably with the baselines.

Limitations. On the implementation side, although the experiments have shown that our rAD methods are quite robust to the changes of the parameters a and $\hat{\pi}_p$, we still have to tune them to obtain the best AD performance. Furthermore, solving the optimisation problem in (4.19) is challenging for a very large-scale dataset since the max operator does not allow parallel computations. On the theoretical side, although the risk bounds are established for the proposed risk minimisers in Section 4.5, we still need the assumption that π_p and π_n are known in advance, which is a limitation.

Future works. One possible research direction is to develop a method that can learn the best combination of $(a, \hat{\pi}_p)$ from the available data. On the other hand, our experiments have shown that using $a = 0.1$, precise estimation of π_p and π_n is not necessarily needed to obtain good accuracy in terms of AUC. Hence, another possible research direction would be to study the theoretical bounds of the risk minimisers with π_p and π_n replaced by some estimates. Finally, investigating effective optimisation techniques to tackle the non-convex Problem (4.19) is also an important research direction aimed at overcoming the difficulties associated with handling exceedingly large-scale datasets.

CHAPTER 5

Evidence-Based Test-time Adaptation Framework

Existing solutions to mitigate the adverse effects of training data contamination on unsupervised AD require access to the training pipelines, data, or prior knowledge of the proportions of anomalies in the data, which limits their real-world applicability. To address this challenge, we propose `EPHAD`*, a simple yet effective inference-time adaptation framework that updates the outputs of AD models trained on contaminated datasets using evidence gathered at inference. Our approach integrates the prior knowledge captured by the AD model trained on contaminated datasets with auxiliary evidence derived from multimodal foundation models like Contrastive Language-Image Pre-training (CLIP), classical AD methods like the Local Outlier Factor or domain-specific knowledge.*

This chapter is based on the following publication.

- **Sukanya Patra** & Souhaib Ben Taieb (2025a). An Evidence-Based Post-Hoc Adjustment Framework for Anomaly Detection Under Data Contamination. In the *Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

5.1. Introduction

Real-world applications of unsupervised AD often violate the widely adopted assumption that training data are "clean", i.e. composed solely of normal samples (Ruff et al., 2021), as discussed previously in Chapter 4. In practice, training datasets are often contaminated with anomalous instances. As in Chapter 4, we therefore adopt a more realistic setting where contamination is present. However, unlike in the previous chapter, our focus here is on the unsupervised setting, in which no labelled samples are available.

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, as discussed in Chapter 4, methods leverage additional labelled datasets containing both normal and anomalous samples (Hien et al., 2024; Ruff et al., 2020). However, their effectiveness diminishes when the anomalous instances encountered during training do not replicate real-world 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, we address the challenging setting in which training pipelines, data, or prior knowledge of the proportions of anomalies cannot be accessed. 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 *test-time adaptation* (TTA) methods (Karmanov et al.,

2024; Zhang et al., 2023; Xiao and Snoek, 2024) , which remain largely unexplored in the context of AD. To address this gap, we introduce the **E**vidence-based **P**ost-**H**oc Adjustment Framework for **A**nomaly **D**etection (`EPHAD`), a simple yet effective method that adjusts the outputs of a pretrained AD model post-hoc, using auxiliary evidence collected at test time.

Notably, we establish conceptual links between `EPHAD` and recent advances in test-time alignment for generative models (Mudgal et al., 2024; Li et al., 2024; Korbak et al., 2022), underscoring its broader significance. `EPHAD` is flexible and can incorporate various forms of evidence, including foundation models like Contrastive Language–Image Pre-training (CLIP) (Zhou et al., 2024; Jeong et al., 2023), classical AD methods such as Local Outlier Factor (LOF) (Breunig et al., 2000), and domain-specific knowledge. Our core contributions are summarised below:

- We introduce `EPHAD`, a *simple yet effective* TTA framework for unsupervised AD models trained on contaminated datasets. Unlike existing approaches, it requires no access to training pipelines, data or prior knowledge of the proportions of anomalies in the data, making it highly practical for real-world deployments.
- `EPHAD` performs TTA by combining the prior knowledge captured by the AD model trained on the contaminated dataset and an auxiliary evidence gathered at test-time. This principled formulation allows for conceptual connections to recent test-time alignment techniques in generative modelling.
- We illustrate the intuition behind `EPHAD` using a carefully designed toy example. Furthermore, extensive experiments across eight visual AD, twenty-six tabular AD datasets, and a real-world industrial AD dataset demonstrate the effectiveness of `EPHAD` across diverse unsupervised AD models, evidence pairs.

5.2. Related work

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.

5.3. Background

For our work, we follow the same formulation as discussed in Section 2.1.1. The training dataset $\mathcal{D}^+_{\text{train}} := \{x_i\}_{i=1}^m$ contains only normal samples (uncontaminated) i.e., $x_i \overset{\text{iid}}{\sim} \mathrm{P}^+_X$. We denote the test dataset as $\mathcal{D}_{\text{test}} := \{(x_i, y_i)\}_{i=1}^n$ which contains both normal and anomalous samples i.e, $(x_i, y_i) \overset{\text{iid}}{\sim} \mathrm{P}_{X,Y}$.

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 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 is typically unknown in practice, density-based methods approximate it using a density estimator.

Score-based anomaly detection. Density estimation poses significant chal-

lenges, particularly in high-dimensional spaces or when data is sparse, 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* $s_{\text{out}}(x) : \mathcal{X} \to \mathbb{R}$, which directly assigns an anomaly score to a data point $x \in \mathcal{X}$, thereby quantifying its *degree of anomalousness* (Ruff et al., 2021). To complement this, the *inlier score function* is defined as $s_{\text{in}}(x) = -s_{\text{out}}(x)$, 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 $s^{+}_{\text{out}}(x)$ using $\mathcal{D}^{+}_{\text{train}}$. Then, we define the anomaly detector as

$$g_{\lambda_s}(x) = \begin{cases} +1, & \text{if } s^{+}_{\text{out}}(x) \leq \lambda_s \\ -1, & \text{if } s^{+}_{\text{out}}(x) > \lambda_s \end{cases} \tag{5.1}$$

where $\lambda_s \geq 0$ is a pre-determined 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 $s^{+}_{\text{out}}(x) = -\phi(f^{+}_{X}(x))$ and $\phi(\cdot)$ is an order-preserving transformation chosen to be the logarithm.

Data contamination. For training the AD method, 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 our dataset $\mathcal{D}^{\pm}_{\text{train}} := \{x_i\}_{i=1}^{m}$ contains both normal and anomalous samples drawn from a mixture distribution $\mathrm{P}^{\pm}_{X}$ with PDF $f^{\pm}_{X}$ (Huber and Ronchetti, 2011; Huber, 1992). Letting $\epsilon = \mathbb{P}(Y = -1)$ denote the *contamination factor*, the data distribution can be written as

$$\mathrm{P}^{\pm}_{X} = \epsilon\, \mathrm{P}^{-}_{X} + (1 - \epsilon)\, \mathrm{P}^{+}_{X}. \tag{5.2}$$

As ϵ increases, the model trained on $\mathcal{D}^{\pm}_{\text{train}}$ becomes biased towards the anomalous regions, reducing its ability to separate normal from anomalous samples (Qiu et al., 2022; Yoon et al., 2022). The existing literature examining the impact of contamination on unsupervised AD methods (Jiang et al., 2022; Qiu et al., 2022; Hien et al., 2024; Perini et al., 2023, 2022) typically considers contamination levels ranging from 0% to 20%. Additionally, an analysis of 57 datasets spanning Natural Language Processing and Computer Vision in ADBench (Han et al., 2022) [Appendix B.2, Figure B1] revealed that nearly 70% of the datasets exhibit anomaly ratios below 10%, with a median of 5%.

5.4. EPHAD: An evidence-based post-hoc adjustment framework

We consider the realistic scenario in which an AD model has already been trained on a possibly contaminated dataset $\mathcal{D}^{\pm}_{\text{train}}$. The goal is to adapt the model's predictions at test-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 novel framework to mitigate the adverse effects of training data contamination using an evidence function at test-time. Here, the *evidence function* $T(x) : \mathcal{X} \to \mathbb{R}$ assigns 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).

For density-based AD, as discussed in Section 5.3, anomalies are identified as samples lying in the low-density regions under the distribution of normal samples P^{+}_{X}. However, due to data contamination, the trained model estimates the PDF $f^{\pm}_{X}$ of the contaminated distribution $\mathrm{P}^{\pm}_{X}$, as defined in (5.2), rather than the inlier PDF f^{+}_{X}. Given an evidence function $T(x)$, EPHAD computes the revised PDF $\check{f}^{\pm}_{X}$ using *exponential tilting* as:

$$\check{f}^{\pm}_{X}(x) = \frac{f^{\pm}_{X}(x)\exp(T(x)/\beta)}{Z^{\beta}_{X}}, \tag{5.3}$$

where $\exp(T(x)/\beta)$ is the evidence scaled by a temperature parameter $\beta \in \mathbb{R}$ and $Z^{\beta}_{X} = \int_{\mathcal{X}} f^{\pm}_{X}(x)\exp(T(x)/\beta)\,dx$ is the normalising constant. The goal is to increase the scores of normal samples relative to those of anomalous samples, increasing the likelihood of an outcome supported by the evidence function. Proposition 5.4.1 provides a condition under which the revised PDF $\check{f}^{\pm}_{X}$ is closer to the PDF of normal samples f^{+}_{X} than the contaminated PDF $f^{\pm}_{X}$, in terms of Kullback–Leibler (KL) divergence.

Proposition 5.4.1. *Let f^{+}_{X}, $f^{\pm}_{X}$, and $\check{f}^{\pm}_{X}$ be PDFs over the same domain $\mathcal{X}$. Then the KL divergence between f^{+}_{X} and $\check{f}^{\pm}_{X}$ is strictly less than the divergence between f^{+}_{X} and $f^{\pm}_{X}$ iff*

$$\mathbb{E}_{x\sim \mathrm{P}^{+}_{X}}\left[\log \frac{\exp(T(x)/\beta)}{Z^{\beta}_{X}}\right] > 0. \tag{5.4}$$

The proof is provided in Appendix C.1.1. Consequently, we expect $\check{f}_X^{\pm}$ will result in an improved AD performance compared to using $f_X^{\pm}$, assuming a well-chosen threshold. It can also be shown that (5.3) is the optimal solution to the KL-regularised objective

$$J_{\mathrm{KL}}(\check{f}_X^{\pm}) := \mathbb{E}_{x\sim\check{f}_X^{\pm}}\left[T(x)\right] - \beta\ \mathrm{KL}(\check{f}_X^{\pm} \| f_X^{\pm}). \tag{5.5}$$

Interestingly, the interpretation in (5.5) highlights a connection with a well-established TTA approach used in generative models (Korbak et al., 2022; Mudgal et al., 2024; Li et al., 2024). For the proof of (5.5), we refer the readers to (Korbak et al., 2022). In this scenario, a generative model is treated as an RL policy and is fine-tuned using a reward function that encodes the desired evidence or alignment criteria. (5.5) also offers a valuable interpretation of `EPHAD` as a form of TTA that shifts the original density $f_X^{\pm}$ toward regions favoured by the evidence function $T(x)$ while maintaining consistency with the original density through KL regularisation. The hyperparameter β provides fine-grained control over this trade-off, recovering the evidence-driven solution in the limit $\beta \to 0$ and reverting to the original model as $\beta \to \infty$.

5.4.1 Extension to score-based anomaly detection

Due to the challenges of estimating the inlier density $f_X^{+}(x)$, particularly in high-dimensional settings, most practical AD methods are *score-based*. Recall that the inlier score function is an order-preserving transformation of the inlier PDF, i.e., $s_{\mathrm{in}}^{+}(x) = \phi(f_X^{+}(x))$, where $\phi(\cdot)$ is a monotonic transformation such as the logarithm. Analogously, when an AD model is trained on a contaminated dataset $\mathcal{D}_{\mathrm{train}}^{\pm}$, it learns a contaminated inlier score $s_{\mathrm{in}}^{\pm}(x) = \phi(f_X^{\pm}(x))$.

Nonetheless, the transformation ϕ is not known in practice. Furthermore, it might not be invertible. Following the approach used in energy-based models (EBMs) (LeCun et al., 2006), we can represent the PDF given an inlier score function $s_{\mathrm{in}}^{\pm}$ as

$$\tilde{f}_X^{\pm}(x) = \frac{\exp(s_{\mathrm{in}}^{\pm}(x))}{Z_X^e}, \tag{5.6}$$

where $Z_X^e = \int_X \exp(s_{\mathrm{in}}^{\pm}(x))$ is the normalising constant. Then, given (5.6), we can perform exponential tilting as in (5.3) where

$$\begin{aligned} \check{s}_{\mathrm{in}}^{\pm}(x) &:= \frac{\tilde{f}_X^{\pm}(x)\exp(T(x)/\beta)}{Z_X^{\beta}} \\ &= \frac{\exp(s_{\mathrm{in}}^{\pm}(x))\exp(T(x)/\beta)}{Z_X^{\beta} Z_X^e}. \end{aligned} \tag{5.7}$$

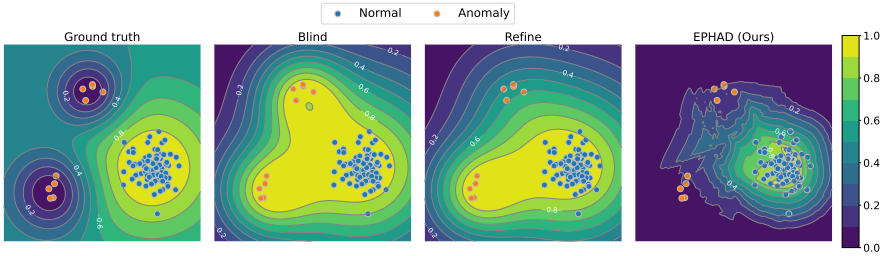


Figure 5.1: DeepSVDD trained on 2D synthetic contaminated training data with different configurations: (I) Supervised AD with ground truth labels for reference, (ii)"Blind" considering all samples as normal, (iii) "Refine" filtering out a fraction of the anomalies, and (iv) `EPHAD` updating the "Blind" anomaly detector using evidence computed on the samples available at test-time.

The value of the constant $Z_X^\beta Z_X^e$ in (5.7) does not affect anomaly detection performance, as detection depends solely on the *relative ranking* of samples rather than their absolute probabilities. Therefore, (5.7) can be equivalently expressed as

$$\check{s}_{\text{in}}^{\pm}(x) \propto \exp(s_{\text{in}}^{\pm}(x)) \exp(T(x)/\beta) = \exp(s_{\text{in}}^{\pm}(x) + T(x)/\beta). \tag{5.8}$$

Given the inlier score $s_{\text{in}}^{+}(x)$ obtained from an AD model trained on the clean dataset $\mathcal{D}_{\text{train}}^{+}$, the corresponding PDF $\tilde{f}_X^+$ can be derived following the same formulation as in (5.6). Under the conditions stated in Proposition 5.4.1, the revised inlier score $\check{s}_{\text{in}}^{\pm}(x)$ is closer, in terms of KL divergence, to $\tilde{f}_X^+$ than the PDF $\tilde{f}_X^{\pm}$ obtained from the same AD model trained on the contaminated dataset $\mathcal{D}^{\pm}$. The anomaly detector in (5.1) can thus be redefined using the corrected score:

$$\check{g}_{\lambda_s}(x) = \begin{cases} +1, & \text{if } \check{s}_{\text{in}}^{\pm}(x) \geq \lambda_s, \\ -1, & \text{otherwise.} \end{cases} \tag{5.9}$$

This extension allows `EPHAD` to operate directly on score-based AD models, enabling post-hoc correction of models trained on contaminated datasets without requiring retraining or access to the original training procedure. In all subsequent experiments, we adopt this score-based formulation of `EPHAD`, reflecting the dominance of score-based methods in modern anomaly detection practice.

5.4.2 An illustrative example

To illustrate the effect of `EPHAD`, we use 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 $f_X^+ = c_1$, with $\mu_1 = [1, 1]^T$ and $\Sigma_1 = 0.07\ \mathbf{I}_2$. Anomalous samples are drawn from a mixture distribution $f_X^- := 0.5c_2 + 0.5c_3$ where $\mu_2 = [-0.25, 2.5]^T, \mu_3 = [-1, 0.5]^T$ and $\Sigma_2 = \Sigma_3 = 0.03\ \mathbf{I}_2$. The extended implementation details is provided in Appendix C.2.2. 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 5.1: (i) "Blind", (ii) "Refine", and (iii) with `EPHAD`. We refer to the baseline model that treats all samples as normal as "Blind", while "Refine" denotes a model that iteratively filters out suspected anomalies during training. As an evidence function in `EPHAD`, LOF (Breunig et al., 2000) is computed on test samples at test time. The results in Figure 5.1 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.

5.4.3 Determining the temperature parameter β

As previously discussed, `EPHAD` has only a single hyperparameter, β, which controls the trade-off between reliance on the original 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 at test time 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`, that determines the optimal β in an unsupervised manner using only test data at test time. This adaptation is inspired by the principle of Entropy Minimisation (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. Specifically, the computation of β depends on the entropy of the inlier probabilities derived from the scores of

both the original model and the evidence function.

Computing inlier probability from the output scores. For computing the inlier probability, we follow the approach of Perini et al. (2021). Given an output score $s \in \mathbb{R}$, the class label can be modelled as a conditional random variable $Y \mid S = s$. Then, Perini et al. (2021)[Equation 2] defined the outlier probability as

$$\mathbb{P}(Y = -1 \mid S = s) := \mathbb{P}(S \leq s). \tag{5.10}$$

Following this, the inlier probability $p_{Y=+1}(s)$ can be expressed as

$$p_{Y=+1}(s) := \mathbb{P}(Y = +1 \mid S = s) = 1 - \mathbb{P}(Y = -1 \mid S = s) = 1 - p_s, \tag{5.11}$$

where $p_s := \mathbb{P}(S \leq s)$. Since p_s is unknown in practice, we 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 $s' \sim S$ by first sampling $x \sim \mathcal{X}$ and then computing the corresponding anomaly score s'. We record a success ($b = 1$) if $s' \leq s$, 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. (2021), 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

$$p_s := \mathbb{E}[P_s] = \frac{1 + t}{2 + n}. \tag{5.12}$$

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 (5.11) and (5.12), we obtain the estimated inlier probability for a data point x with anomaly score s as

$$p_{Y=+1}(s) = 1 - p_s = 1 - \frac{1 + t}{2 + n}. \tag{5.13}$$

Finally, using (5.13), we compute the inlier probabilities from the scores of the original model and the evidence function as $p^p_{Y=+1}(x) := p_{Y=+1}(s^{\pm}_{\text{out}}(x))$ and $p^e_{Y=+1}(x) := p_{Y=+1}(-T(x))$, respectively.

Computing the value of the hyperparameter β. We define the empirical entropy of the binary predictive PMF $p_Y(x)$ as

$$H(p_Y) = -\sum_{x \in \mathcal{D}_{\text{test}}} \left[p_{Y=+1}(x) \log p_{Y=+1}(x) + p_{Y=-1}(x) \log p_{Y=-1}(x) \right]. \tag{5.14}$$

The adaptive temperature parameter is then defined as

$$\beta_{\text{ada}} = \frac{H(p_Y^e)}{H(p_Y^o) + \delta}, \tag{5.15}$$

where $\delta > 0$ is a small constant introduced to ensure numerical stability.

A low $H(p_Y^o)$ indicates that the original AD model produces confident (low-entropy) predictions, suggesting that a higher value of β should be used to place greater trust in this model. Conversely, a lower $H(p_Y^e)$ implies higher confidence in the evidence function, motivating a smaller β. Through this formulation, `EPHAD-Ada` enables unsupervised, test-time determination of β, thereby improving practicality and eliminating the need for labelled validation data.

5.5. Experiments

We evaluate the effectiveness of `EPHAD` for unsupervised AD across a range of datasets, including visual AD datasets (Section 5.5.1), tabular AD datasets (Section 5.5.2), and an industrial AD use case (Section 5.5.3). To systematically investigate the impact of contamination at different levels in a rigorous and reproducible way, we introduce controlled contamination into the data, adhering to the experimental design employed in several prior studies (Jiang et al., 2022; Wang et al., 2025; Zhou and Wu, 2024). The evidence functions employed in the experiments 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 a contamination factor of $\epsilon = 0.1$ and a parameter $\beta = 0.5$. An ablation study on different values of ϵ and β is presented in Section 5.5.4. For image and tabular datasets, we evaluate performance using the AUROC. Following prior work (Roth et al., 2022; Gudovskiy et al., 2022), AUROC is averaged across all categories for each dataset.

5.5.1 Experiments on visual AD datasets

Benchmark datasets. We assess the effectiveness of `EPHAD` in both sensory and semantic anomaly detection. Sensory AD focuses on detecting physical defects or imperfections, such as a broken capsule or a cut in a carpet, while semantic AD identifies anomalies belonging to a different semantic class—for instance, treating cats as normal and any other animal as anomalous. For sensory AD in industrial contexts, we evaluate performance using four well-established benchmark datasets: MVTecAD (Bergmann et al., 2019), MPDD

(Jezek et al., 2021), ViSA (Zou et al., 2022), and RealIAD (Wang et al., 2024). For semantic AD, we utilise four commonly used datasets, including CIFAR-10, Fashion-MNIST, MNIST, and SVHN. Following the one-vs-rest protocol (Qiu et al., 2022), we construct k AD tasks per dataset, where k corresponds to the number of classes. For MVTecAD, ViSA, MPDD and RealIAD, 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. Our implementation is built upon the publicly available codebase from Jiang et al. (2022). Additional details are provided in Appendix C.2.1.

Baseline AD methods. We evaluate the performance of several state-of-the-art unsupervised anomaly detection 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`. Implementations for all methods, except ULSAD, are based on the Anomalib library (Akcay et al., 2022), while ULSAD is implemented using its official public code. Since, to the best of our knowledge, no existing AD method with contaminated data offers post-hoc adaptation in the same manner as `EPHAD`, our primary objective is to demonstrate the effectiveness of `EPHAD` by comparing its relative performance against the AD model and the evidence function alone. For completeness, we also provide comparative analyses with three existing frameworks "Refine" (Yoon et al., 2022), Latent Outlier Exposure (LOE) (Qiu et al., 2022), and SoftPatch (Jiang et al., 2022) in Appendix C.3.2.

Evidence function. For the experiments, we employ Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) as the evidence function for image-based datasets, following the anomaly detection approach as in WinCLIP (Jeong et al., 2023). We use CLIP as the 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. These templates are dataset-dependent, reflecting subjectivity (e.g., "missing wire" as anomalous for cables). For each label, compute the mean of text embeddings t_N and t_A. Finally, given an input image x, the evidence $T(x)$

at test-time is computed as:

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

Additional implementation details are provided in Appendix C.2.3.1.

While CLIP has been previously applied as a standalone zero-shot anomaly detector, our methodology leverages it differently: we employ CLIP not as a complete detection system, but as an auxiliary source of evidence integrated into a more general and flexible framework. Importantly, `EPHAD` is not limited to foundation models such as CLIP; it can seamlessly incorporate domain-specific knowledge as well (see Section 5.5.3), thereby broadening its applicability across diverse domains.

One potential concern when using pre-trained models like CLIP is the overlap between their training data and the test samples encountered in downstream tasks. Such overlap could challenge the assumption that test-time statistics are based solely on test data. However, Radford et al. (2021) provides an extensive analysis of this issue and shows that excluding all overlapping samples from CLIP's pre-training corpus leads to only a negligible performance drop. This result suggests that CLIP's effectiveness stems primarily from its generalisation ability rather than memorisation. Accordingly, our experiments emphasise this generalisation property, ensuring that the use of CLIP within our framework remains valid.

Results. In our experiments, as we adopt CLIP in the same manner as WinCLIP (Jeong et al., 2023), the baseline CLIP results reported here directly correspond to the standalone performance of WinCLIP. In Table 5.1, 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 SVHN, we observe a decline in performance when integrating `EPHAD` compared to the standalone AD method.

Table 5.1: 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	**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)
+ 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	**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)
+ 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	**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)
+ EPHAD-Ada	82.83 (± 0.44)	92.10 (± 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	87.50 (± 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)
+ EPHAD-Ada	**87.56** (± 0.23)	92.87 (± 0.02)	90.23 (± 0.67)	**57.09** (± 1.05)	79.56 (± 0.28)	86.10 (± 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	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)
+ 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)	48.60 (± 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	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)
+ 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	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)
+ 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)

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 Section 5.5.4.

Using the adaptive variant, `EPHAD-Ada`, we observe further improvements in certain settings, such as with PatchCore and DREAM on the RealIAD dataset. Interestingly, in cases where the default value of $\beta = 0.5$ led to decreased performance (e.g., ULSAD on SVHN or MPDD), `EPHAD-Ada` manages to overcome the problem, highlighting its effectiveness. Nevertheless, while `EPHAD-Ada` offers an unsupervised mechanism for determining β, its performance is often comparable to, or slightly below, that of `EPHAD` with the default value for β. We hypothesise that this behaviour arises as the inlier probability estimated from anomaly scores is uncalibrated. Investigating principled approaches to selecting β remains an interesting direction for future research.

5.5.2 Experiments on tabular AD datasets

Benchmark datasets. We evaluate our proposed framework on 26 classical benchmark datasets from ADBench (Han et al., 2022)[Table B1]. The classical

Table 5.2: Performance on tabular AD benchmarking datasets with 10% contamination ratio. Style: AUROC % (± SE). Best in **bold**. † represents transductive inference.

Dataset	aloi	cover	glass	ionosphere	letter	pendigits	vowels	wine
LOF†	72.64 (± 0.1)	52.12 (± 0.1)	77.52 (± 0.93)	82.43 (± 0.16)	83.15 (± 0.73)	47.21 (± 0.12)	89.1 (± 0.67)	97.57 (± 1.46)
COPOD	51.46 (± 0.05)	78.7 (± 0.03)	76.11 (± 0.77)	79.42 (± 1.03)	56.71 (± 0.12)	**88.44** (± 0.2)	56.1 (± 0.32)	80.51 (± 1.36)
+ EPHAD	**57.74** (± 0.26)	77.96 (± 0.09)	**84.82** (± 0.77)	**85.57** (± 0.24)	**78.06** (± 0.93)	80.72 (± 0.54)	**84.48** (± 0.71)	**97.12** (± 1.25)
+ EPHAD-Ada	53.65 (± 0.17)	**79.57** (± 0.01)	81.77 (± 1.28)	84.15 (± 0.38)	71.03 (± 0.99)	87.09 (± 0.22)	75.39 (± 0.88)	93.96 (± 1.66)
DeepSVDD	54.06 (± 0.54)	75.11 (± 11.37)	64.52 (± 6.87)	83.09 (± 0.57)	50.51 (± 2.54)	**74.87** (± 9.91)	64.47 (± 2.55)	82.26 (± 2.29)
+ EPHAD	**71.98** (± 0.08)	71.92 (± 8.84)	79.67 (± 1.91)	84.5 (± 0.32)	**75.14** (± 1.93)	68.43 (± 7.41)	**87.64** (± 0.8)	**96.72** (± 1.81)
+ EPHAD-Ada	70.67 (± 0.22)	**75.58** (± 10.82)	**80.94** (± 2.52)	**85.03** (± 0.25)	65.9 (± 2.88)	74.08 (± 9.18)	82.12 (± 0.9)	93.96 (± 1.77)
ECOD	53.14 (± 0.03)	85.34 (± 0.02)	67.65 (± 0.44)	73.04 (± 0.84)	56.41 (± 0.29)	**90.63** (± 0.17)	54.29 (± 0.06)	67.12 (± 2.04)
+ EPHAD	**59.67** (± 0.29)	83.13 (± 0.12)	**81.59** (± 0.8)	**80.88** (± 0.35)	**77.03** (± 1.08)	83.78 (± 0.6)	**84.64** (± 0.78)	**95.59** (± 2.02)
+ EPHAD-Ada	55.47 (± 0.18)	**85.45** (± 0.01)	78.43 (± 1.72)	78.14 (± 0.49)	70.15 (± 1.15)	89.66 (± 0.2)	75.39 (± 0.91)	89.27 (± 2.95)
IForest	54.05 (± 0.21)	72.59 (± 1.59)	78.5 (± 1.47)	89.58 (± 1.57)	59.84 (± 0.64)	**81.86** (± 1.48)	66.01 (± 0.57)	80.4 (± 3.42)
+ EPHAD	**61.77** (± 0.26)	72.99 (± 1.43)	**83.15** (± 0.91)	88.66 (± 0.79)	**77.45** (± 0.7)	73.52 (± 1.1)	**86.29** (± 0.62)	**96.67** (± 1.59)
+ EPHAD-Ada	57.49 (± 0.31)	**73.15** (± 1.57)	**83.15** (± 1.86)	**90.05** (± 1.22)	71.38 (± 0.86)	79.5 (± 1.5)	80.76 (± 0.6)	93.56 (± 2.15)
LOF	73.57 (± 0.1)	22.44 (± 0.1)	71.79 (± 1.08)	**94.64** (± 0.52)	**85.74** (± 0.54)	14.87 (± 0.18)	**93.04** (± 0.54)	**99.94** (± 0.05)
+ EPHAD	73.6 (± 0.1)	**44.15** (± 0.22)	**76.36** (± 0.65)	89.7 (± 0.61)	84.76 (± 0.36)	**37.27** (± 0.89)	91.01 (± 0.24)	99.1 (± 0.6)
+ EPHAD-Ada	**73.85** (± 0.05)	36.78 (± 0.23)	75.67 (± 0.75)	91.85 (± 0.68)	85.31 (± 0.36)	30.16 (± 1.01)	91.85 (± 0.12)	**99.94** (± 0.05)

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 into the train and test sets 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 C.2.3.2.

Results. The experimental results for a subset of the 26 benchmarking datasets are presented in Table 5.2, with the extended version provided in Appendix C.3.1. 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 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 appro-

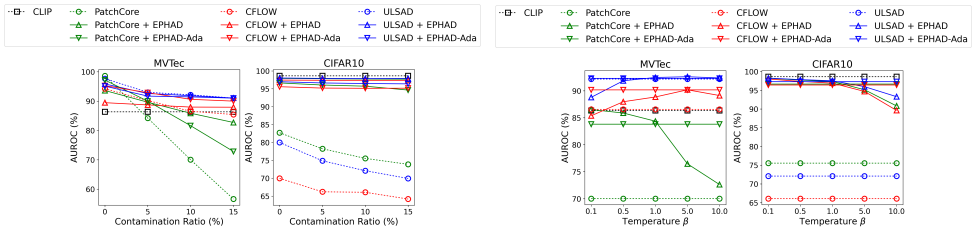
priately tuning β. Similar to the results in the image-based experiments, we observe some improvements when using the adaptive variant `EPHAD-Ada`, such as on the cover dataset. In some scenarios, we also observe that `EPHAD-Ada` avoids the performance drop observed with `EPHAD`, such as with LOF on the ionosphere dataset and with DeepSVDD on the pendigits dataset. Nonetheless, the performance in most cases is similar to `EPHAD` with default value β, suggesting the need for further exploration on how to select the optimal value of β based on the data.

5.5.3 Experiments on industrial use case

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. Foundation models, such as CLIP, which were previously used in our experiments on image datasets, are not applicable in specialised applications, such as detecting anomalous behaviour in solar power plants, due to the lack of semantic content in thermal images. This makes zero-shot methods like WinCLIP and AnoCLIP inapplicable. In contrast, while EPHAD can incorporate evidence from foundation models like CLIP, it also allows the seamless integration of domain-specific knowledge. 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 indi-



(a) Ablation on ϵ. (b) Ablation on β.

Figure 5.2: Ablation on parameters.

cate 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.

Results. The results presented in Table 5.3 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` and further using `EPHAD-Ada`, the performance nearly matches that on the clean dataset. It emphasises the value of leveraging structured, context-aware evidence to enhance the detection of anomalies.

Table 5.3: Performance on CSP plant dataset.

Setting	Method	AUROC (± SE)
Clean	`ForecastAD`	94.91 (±0.09)
Evidence	Rule-based (R1, R2)	69.46 (± 0.0)
Contaminated ($\epsilon = 0.1$)	`ForecastAD`	90.45 (± 0.8)
	+ `EPHAD`	93.51 (± 0.45)
	+ `EPHAD-Ada`	**93.57** (± 0.43)

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.

5.5.4 Ablation study

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 5.2a. 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%. An extended version is provided in Figure C.1.

Effect of temperature parameter β. We also analyse the performance of the `EPHAD` by varying the temperature parameter β. In Figure 5.2b, 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. An extended version is provided in Figure C.2.

5.6. Conclusion

Limitations and future work. While existing AD methods can serve as domain-agnostic evidence functions within `EPHAD`, the full potential of our framework is best realised by designing evidence functions that incorporate domain-specific knowledge. Exploring the interplay between datasets, AD methods, and evidence functions remains an open direction for future work. Another limitation concerns the parameter β, which has a significant influence on overall performance, as demonstrated in our experiments. Although we introduced an unsupervised strategy for estimating β in `EPHAD-Ada`, this approach does not always lead to performance improvements. We hypothesize that this may stem from uncalibrated inlier probability. Future work should thus investigate more reliable approaches for inferring β based on the anomaly scores and the underlying distributions of normal and anomalous samples in the test set. Finally, integrating explainability techniques into `EPHAD` represents an interesting direction for future research, as it could provide deeper insights and enhance the interpretability of results in real-world applications.

Concluding remarks. Unsupervised AD methods typically assume anomaly-free training data, yet real-world datasets often contain undetected or mislabeled anomalies, leading to significant 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 test-time. Extensive experiments demonstrate the effectiveness of `EPHAD` across diverse sources of evidence, multiple AD methods, and various datasets. Additionally, ablation studies analyse the impact of hyperparameters and varying contamination levels, highlighting the robustness of `EPHAD`.

CHAPTER 6

Detecting Anomalies in Irregular Image Sequences

AD in dynamic environments requires careful modelling of the temporal features of normal data, where statistical properties evolve over time. We explore this by considering the application of AD for detecting anomalous behaviours of a solar power plant based on thermal images collected from an operational plant. The infrared cameras mounted on the solar receivers capture these images at irregular intervals throughout the day. Our goal is to develop a method that extracts meaningful features for AD from high-dimensional thermal images, considering the temporal aspects such as irregular sampling, temporal dependency between images and non-stationarity. To achieve this, we propose a forecasting-based AD method that predicts future thermal images from past sequences and timestamps via a deep sequence model. The proposed approach effectively captures temporal patterns for AD. Furthermore, it can distinguish between challenging instances, such as low-temperature anomalies that resemble normal instances from the start and the end of the operational period.

This chapter is based on the following two publications.

- **Sukanya Patra**, Nicolas Sournac, & Souhaib Ben Taieb. Detecting Abnormal Operations in Concentrated Solar Power Plants from Irregular Sequences of Thermal Images (2024c). In the *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.
- **Sukanya Patra**, Le Thi Khanh Hien, & Souhaib Ben Taieb (2023). Anomaly detection in irregular image sequences for concentrated solar power plant. In the *European Symposium on Artificial Neural Networks (ESANN)*.

For the first publication, Sukanya Patra was responsible for the development of the proposed solution, experimental work and the preparation of the scientific article. Nicolas Sournac contributed to the experimental work, annotation of the real-world dataset, preparation of the synthetic dataset, and assisted in preparing the scientific article. The project was conducted under the supervision of Souhaib Ben Taieb.

For the second publication, Sukanya Patra contributed to the development of the proposed solution, experimental work and the preparation of the scientific article. Le Thi Khanh Hien contributed to ideation and assisted in preparing the scientific article. The project was conducted under the supervision of Souhaib Ben Taieb.

6.1. Introduction

The focus on renewable energies to counteract climate change has intensified recently. However, a critical challenge in adopting renewable energy sources is ensuring on-demand generation and dispatchability. A promising solution to this challenge is the integration of Thermal Energy Storage (TES) facilities, which temporarily store energy by heating or cooling a storage medium, such as water or molten salt. CSP plants effectively utilise TES for storing energy by heating the medium with an array of mirrors focused on solar receivers atop a central tower (Zhang et al., 2013). These solar receivers are composed of vertical heat exchanger tubes arranged in panel form, allowing the medium to flow through them.

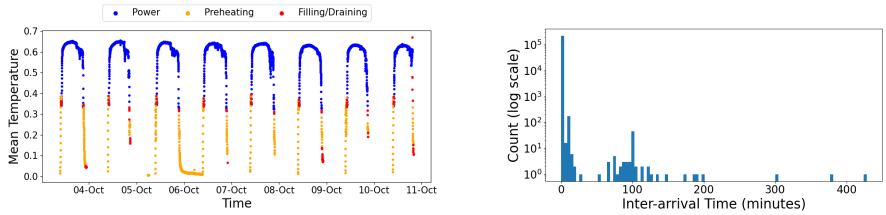
Operating at extreme temperatures, these systems are prone to adverse effects, including the freezing of the medium (affecting a subset of vertical tubes with

significantly higher temperatures), damage to heat-resistant coatings, and deformation and corrosion of the heat exchanger tubes. Therefore, meticulous monitoring of the process is crucial. Given the vast amount of data generated from multiple sensors, manually detecting abnormal behaviours becomes impractical. This necessitates an automated system capable of immediately identifying abnormal behaviours. The advantages of such a system are twofold: it ensures smooth operation and uninterrupted power generation by minimising downtime, and it reduces the risk of further equipment damage by allowing for prompt failure responses. This approach also leads to an extended operational lifetime for the CSP plant.

In this chapter, our goal is to develop a deep image-based anomaly detection (Ruff et al., 2021; Pang et al., 2021) method to identify abnormal behaviours in sequences of thermal images collected over a span of one year from an operational CSP plant. These images are captured at irregular intervals ranging from one to five minutes throughout the day by infrared cameras mounted on solar receivers. Our problem is related to data-driven Predictive Maintenance (PdM), where the state of equipment in industrial processes is monitored to predict future failures (Tang et al., 2020).

Specifically, we aim to extract useful representations for anomaly detection from high-dimensional thermal images. It should be able to handle temporal features of the data, which include irregular sampling, temporal dependency between images and non-stationarity due to a strong daily seasonal pattern. An additional challenge is the coexistence of low-temperature anomalies that resemble low-temperature normal images from the start and the end of the operational cycle alongside high-temperature anomalies. This necessitates learning the current state of the operational cycle to correctly identify anomalies.

We first examine the performance of state-of-the-art (SOTA) deep AD methods that have been successful in extracting useful image representations for anomaly detection, such as CFlow (Gudovskiy et al., 2022), PatchCore (Roth et al., 2022), and DRÆM (Zavrtanik et al., 2021). Our experiments confirm that neglecting the temporal features of the data leads to low accuracy, especially in distinguishing low-temperature normal samples from anomalies. Then, we explore a new forecasting-based AD method, `ForecastAD`, which predicts the image for a given future time based on a sequence of past observed images and their timestamps using a deep sequence model. `ForecastAD` extracts relevant representations from the high-dimensional images and captures the normal behaviour of the solar receivers, taking into account the temporal features of the data. An anomaly is then defined as a significant deviation from



(a) Average image temperature for multiple days.

(b) Histogram of inter-arrival times

Figure 6.1: Visualisation of different properties of the data

the learned normal behaviour. Our experiments demonstrate the effectiveness of `ForecastAD` compared to multiple SOTA baselines across various evaluation metrics. We have also successfully deployed our solution on five months of unseen data, providing critical insights for the maintenance of the CSP plant.

We first present the case study on detecting anomalous behaviours in CSP plants in Section 6.2. Then, we explain `ForecastAD`, our forecasting-based approach, in Section 6.3. Finally, in Section 6.4, we discuss our empirical results before providing our concluding remarks in Section 6.10.

6.2. A case study on detecting anomalous behaviours in CSP plants

A CSP plant consists of two main components, namely: (i) the Thermal Solar Receiver and (ii) the Steam Generator. The Thermal Solar Receiver, placed on top of a central tower at the plant, acts as a solar furnace. On the ground surrounding the tower, an array of flat, movable mirrors called heliostats concentrates the sun rays on the solar receiver. The receiver consists of vertical heat exchanger tubes through which the heat transfer medium flows, absorbing the heat from the concentrated sun rays. Then, the absorbed thermal energy is utilised to generate superheated steam, which runs the Steam Generator for the production of energy. In this work, we focus on detecting anomalous behaviours of the Thermal Solar Receiver using data obtained from an operational plant.

CSP plants utilise high-capacity fluids like molten salts as the heat transfer medium, which are stored in TES facilities for future use. This allows for the on-demand generation of energy, making CSP plants a viable alternative to fossil fuel-based energy plants. However, due to operation in extreme temperatures, the solar thermal receivers are adversely impacted in several ways:

[i] Blocked tubes. The molten salts passing through the heat exchanger tubes tend to freeze in localised zones when the temperature falls below a certain threshold, blocking them.

[ii] Deformity. The metal heat exchanger tubes in the receiver tend to expand due to the high temperatures. Uneven dilation of the tubes could eventually lead to deformity.

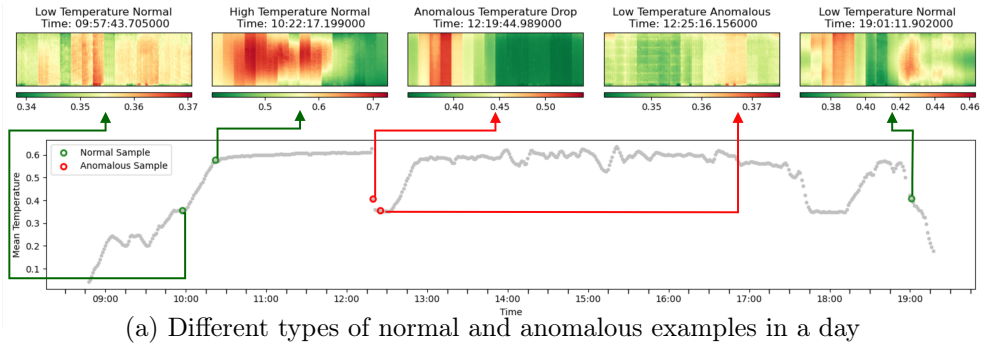
[iii] Stress and metal fatigue. The metal tubes in the receiver undergo expansion when exposed to high temperatures during regular operation and contraction when the operation ends. Such repeated changes lead to metal fatigue. Additionally, the pressure generated from the flowing molten salts exerts stress on the tubes.

[iv] Corrosion. Due to the interaction of the metal with the molten salt flowing through the tubes, it tends to deteriorate over time. These reactions are further accelerated due to the high temperatures in the receiver.

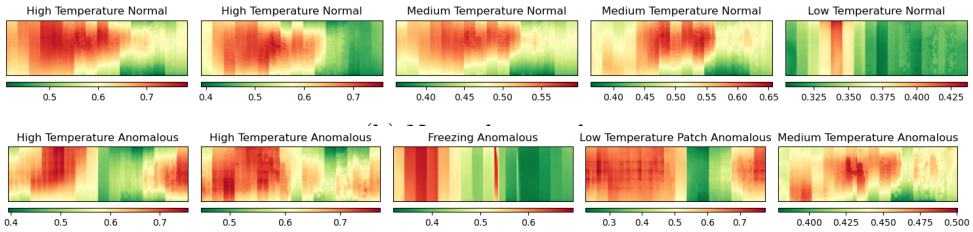
Hence, CSP plants require close monitoring to guarantee seamless operation and continuous power generation. Achieving this requires the analysis of data collected by numerous sensors installed on the Solar Receiver. Yet, the vast volume of data generated renders manual inspection unfeasible, thus underscoring the need for an automated, data-driven monitoring system.

6.2.1 Data description

The Thermal Solar Receiver is composed of several panels, each featuring vertical heat exchanger tubes. These tubes allow the heat transfer medium to flow through, effectively absorbing heat from the concentrated sunlight. Infrared (IR) cameras, strategically positioned around the solar receiver, capture the surface temperature, producing thermal images with dimensions of 184×608. These images are captured approximately every one to five minutes, with each image's timestamp recorded. During normal operations, the temperature of the heat transfer medium gradually increases as it traverses the vertical tubes from one end of the panel to the other, a direct result of absorbing heat from the concentrated sunlight. Consequently, the surface temperature patterns recorded by the IR cameras are anticipated to exhibit a smooth gradient, aligning with the medium's flow direction. Our dataset covers a year of operational data



(a) Different types of normal and anomalous examples in a day



(c) Anomalous examples

Figure 6.2: Examples of different types of normal and anomalous images

without ground truth labels for the images (normal or abnormal), making it an unsupervised anomaly detection problem. Note that throughout this work, we provide normalised images from the dataset for the sake of confidentiality.

Operation in CSP plants occurs across three distinct phases as depicted in Figure 6.1a: (i) *Preheating*, (ii) *Filling/Draining*, and (iii) *Power*. The molten salt used in CSP plants freezes when the temperature drops below a certain threshold. To avoid this, solar panels are initially heated during the *Preheating* phase. Then, in the subsequent *Filling/Draining* phase, molten salt is circulated within the panels. The *Power* phase initiates as the molten salt absorbs heat from sunlight, facilitating power generation. As operations conclude, the molten salt is drained from the panels during the *Filling/Draining* phase. Consequently, the panels commence cooling down, transitioning back to the *Preheating*. Our work focuses solely on the *Power* phase, as it is crucial for power generation and susceptible to damage from prolonged exposure to high temperatures.

Data characteristics and modelling challenges. Through extensive data analysis, we identified the following additional challenges, which are essential for modelling the solar receiver data:

[i] Non-stationarity. Figure 6.1a presents the average surface temperature across a specific week, highlighting temporal variations in the mean image temperature and demonstrating a clear pattern of daily seasonality in the data.

[ii] Irregular sampling. The images were captured at irregular time intervals, as illustrated in Figure 6.1b, which depicts the distribution of inter-arrival times. Additionally, the dataset lacks data for the extended periods when the plant was not operational.

[iii] Temporal dependence. The images exhibit a strong temporal dependence, influenced significantly by weather conditions.

[iv] High dimensionality. Anomalous characteristics often stay hidden and unnoticed due to data sparsity in high-dimensional spaces. Identifying features that capture the essential high-order, non-linear interactions needed for AD is thus challenging.

[v] Large volume of data. The dataset comprises images captured throughout a year of operation at approximately one to five-minute intervals, leading to a vast volume of data.

[vi] Unlabeled data. Our dataset lacks ground truth labels for the images, whether they are normal or abnormal, classifying our task as an unsupervised anomaly detection problem.

6.2.2 Data labelling

To effectively assess the performance of various AD methods, we have labelled a subset of data from the CSP plant. This endeavour is notably complex due to the plant's operation across multiple phases, each characterised by unique temperature ranges. Consequently, this diversity leads to a range of normal and anomalous sample types, as depicted in Figure 6.2. The challenge of identifying anomalies through the plant's operational phases is evident in Figure 6.2a. Notably, normal images with low temperatures at the operation's start and end (the left-most and right-most images in Figure 6.2a) closely resemble low-temperature anomalies (the second image from the right in Figure 6.2a). The distinction between these samples relies heavily on context.

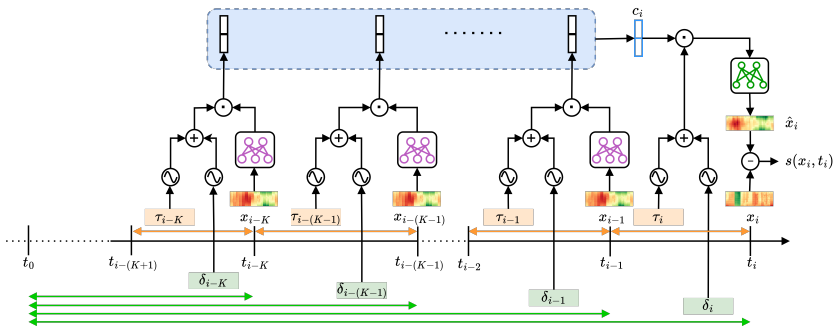


Figure 6.3: Illustration of the end-to-end architecture of `ForecastAD`. The model is trained to forecast the next image in the sequence given a context embedding c_i of K prior data points obtained using a sequence-to-sequence model. For $(x_{i-k}, t_{i-k}, y_{i-k}) \in \mathcal{D}$ in the context, we sum the embeddings of inter-arrival time τ_{i-k} and interval since the start of operation δ_{i-k} and concatenate it with the image embedding. The anomaly score $s(x_i, t_i)$ is computed as the difference between the forecasted and original image.

Moreover, the variable nature of anomalies adds a layer of complexity to the labelling process. Our approach to this challenge is informed by a deep understanding of the CSP plant's operations and expert insights from the field. We categorise the *Power* phase into three distinct segments: (i) Starting (S), where the solar receiver's mean temperature begins to rise; (ii) Middle (M), where it reaches and maintains its peak; and (iii) Ending (E), where it declines as the day concludes. In our preprocessing, we exclude days with significantly few samples or with a consistently low temperature throughout the M segment, likely indicative of sensor or system failures. For the S and E segments, samples showing a consistent temperature increase ($>$ 5°C) or decrease ($<$ -5°C), respectively, are deemed normal, whereas those displaying contrary trends are marked as anomalous. In the M segment, we apply the following four rules for labelling:

[**R1**.] **Difference between consecutive images**. During the M segment of the *Power* phase, we expect a stable temperature. Significant deviations from the preceding observation indicate an abnormality. To detect such anomalies, we first compute the pixel-wise squared differences between every two consecutive images. For each pair, we select the 95th percentile of these pixel-wise differences as our *score*. Samples are then labelled as anomalous if their score

exceeds the 99.9th percentile of the scores for all samples in the dataset.

[**R2**.] **Difference from average daily temperature**. Samples with average temperatures that significantly deviate from the daily average temperature are labelled as anomalous. To identify these anomalies, we first compute the daily mean temperature. Then, we calculate the difference between each image's average temperature and the mean temperature of the corresponding day, which serves as our *score*. Finally, samples are labelled as anomalous if their score falls below the 1st percentile of the distribution of scores across all samples in the dataset.

[**R3**.] **Difference with specific daily normal samples.** Rules **R1** and **R2** are limited to the detection of low-temperature anomalous samples. To address this, we select the first five images from the M segment of the *Power* phase of each day to serve as a set of templates for that day. We then employ a similar methodology as in Rule **R1**, but instead of comparing an image to just the prior image, we compute the mean difference between the image and all five templates of the corresponding day. Applying this rule allows us to obtain sets of high-temperature normal and abnormal samples, along with a diverse set of low-temperature abnormal samples.

[**R4**.] **Freezing statistics.** To identify the anomalous samples with characteristics such as freezing and low-temperature patches, we compute row-wise and column-wise differences within each image. First, we calculate the maximum value of the element-wise differences between two consecutive rows, which we term the *horizontal score*. Next, we compute the element-wise differences between consecutive columns and apply a Sobel filter (Sobel, 1968) to detect vertical edges. The mean value of the elements detected by the Sobel filter across all columns is referred to as the *vertical score*. An image is labelled as anomalous if either the horizontal or the vertical score exceeds a predefined threshold.

Given the labelling rules, we first apply them to obtain an initial set of labels. Then, in collaboration with domain experts, we conduct a visual analysis of the labelled thermal images. For the visual analysis, we also perform clustering on the labelled samples and inspect the cluster centres in addition to analysing each image individually. This thorough inspection leads to subsequent refinements of the labelled set, enhancing the reliability of the labels for accurately assessing anomaly detection methods.

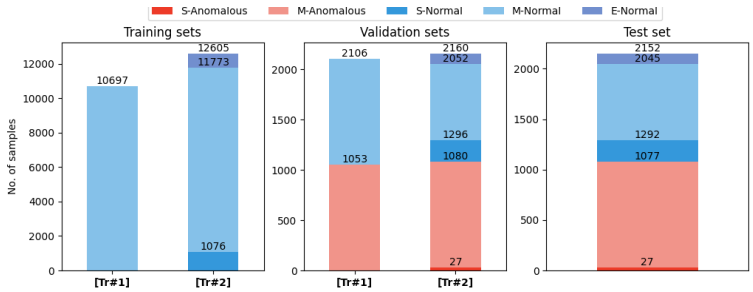


Figure 6.4: Dataset split for two different training setups and the test set

6.2.3 General problem formulation

In this chapter, we follow the notations from the paper (Patra et al., 2024). Consider a dataset $\mathcal{D} = \{(x_i, t_i, y_i)\}_{i=1}^{n}$ consisting of $n = 16,917$ triplets. Each $x_i \in \mathcal{X} = \mathbb{R}_+^d$ corresponds to a thermal image with dimension $d = H \times W$, where the height H is 184 and the width W is 608. These images were captured at times $t_i \in \mathbb{R}_+$, and each $y_i \in \{0, 1\}$ denotes the corresponding label, with 0 representing the normal class and 1 representing the anomalous class.

Let $\mathcal{D}_N$, $\mathcal{D}_V$ and $\mathcal{D}_T$ denote disjoint training, validation and test sets, respectively, with $\mathcal{D}_N \cup \mathcal{D}_V \cup \mathcal{D}_T = \mathcal{D}$. $\mathcal{D}_N$ is exclusively composed of normal samples, i.e., $y_i = 0$ for all $(x_i, t_i, y_i) \in \mathcal{D}_N$. $\mathcal{D}_V$ and $\mathcal{D}_T$ include both normal and anomalous samples.

Using the training set $\mathcal{D}_N$, the AD methods aim to learn a scoring function $s(\cdot, \cdot) : \mathbb{R}_+^d \times \mathbb{R}_+ \to \mathbb{R}$ that assigns an anomaly score $s(x, t)$ to any given point (x, t). By using a threshold $\lambda \in \mathbb{R}$, this anomaly score can then be converted into a predicted label $\hat{y}$ as follows:

$$\hat{y} = \begin{cases} 1, & \text{if } s(x,t) \geq \lambda; \\ 0, & \text{if } s(x,t) < \lambda. \end{cases} \tag{6.1}$$

6.3. Forecasting-based AD model

We present a new forecasting-based AD method, denoted `ForecastAD`, to detect anomalous operations in the Thermal Solar Receiver of a CSP plant from irregular sequences of thermal images. The proposed method builds a forecasting model to reconstruct the thermal images using past observations as context. Images that are hard to reconstruct are considered anomalous. For a given image, our procedure can be summarized in the following steps: (i)

extract feature embeddings for that image, (ii) use the previous K images as *context* and encode them using a deep sequence model, and (iii) using the context, reconstruct the image with a decoder forecasting model, then assign an anomaly score based on the reconstruction error between the original and predicted image. We provide an overview of the architecture of `ForecastAD` in Figure 6.3 and summarise it in Algorithm 2.

6.3.1 Image encoder

Using the training data, $\mathcal{D}_N$, we pre-train an encoder network to capture the inherent structure of our dataset's images. The image encoder, denoted by $\phi_e(\cdot; W_e) : \mathcal{X} \to \mathcal{Z}$, transforms images from the high-dimensional input space $\mathcal{X}$ to a compact latent space $\mathcal{Z} = \mathbb{R}^{d'}$, significantly reducing dimensionality where $d' << d$. We use an autoencoder framework for image reconstruction, with a decoder network $\phi_d(\cdot; W_d) : \mathcal{Z} \to \mathcal{X}$ to project images from the latent space $\mathcal{Z}$ back to the original input space $\mathcal{X}$. The autoencoder is given by $\phi = \phi_e \circ \phi_d$, with $\circ$ indicating function composition. Given the high dimensionality of the input image, we opt for a multi-layer deep convolutional network as the image encoder, exploiting its effectiveness in extracting meaningful representations directly from the data (Bishop and Nasrabadi, 2006). We calculate the reconstruction loss between original data points x_i and their reconstructions $\tilde{x}_i = \phi(x_i)$ as:

$$\mathcal{L}_{\text{pre-train}} = \frac{1}{|\mathcal{D}_N|} \sum_{i=1}^{|\mathcal{D}_N|} \|x_i - \tilde{x}_i\|_F^2, \tag{6.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

6.3.2 Context encoder

To handle the irregular inter-arrival times $\tau_i = t_i - t_{i-1}$ between successive (i)-th and $(i-1)$-th images, our deep sequence model incorporates both the image sequences and their associated irregular inter-arrival times. We employ a sinusoidal encoding $\psi_i = f_{\sin}(\tau_i)$, inspired by the positional encoding technique in transformer models (Vaswani et al., 2017). This method aligns with strategies used in Neural Temporal Point Processes (Enguehard et al., 2020).

In addition to the inter-arrival times between consecutive images, we also embed the relative time since the start of the operation t_0, i.e., $\delta_i = t_i - t_0$, which provides information about the position of an image x_i within the operational cycle. Such temporal context helps in detecting challenging temperature-based

anomalies, as it helps distinguish between low-temperature anomalies occurring mid-cycle and low-temperature normal images at the start of the operation. We use the same sinusoidal encoding for the interval δ_i as $\Psi_i = f_{\sin}(\delta_i)$. The sum of the two time embeddings ψ_i and Ψ_i is combined with the image embedding to obtain the final embedding $\hat{z}_i = [z_i \oplus (\psi_i + \Psi_i)]$, where $z_i = \phi_e(x_i)$ represents the image embedding and $\oplus$ denotes the concatenation operator.

We compactly encode the embeddings of the K samples preceding the image at timestep t_i into a fixed-dimensional vector c_i, termed the context vector for the i-th image. This can be accomplished with a deep sequence model. In our work, we opt for an LSTM $\varphi(\cdot; W_c)$, parameterized by W_c. Given a context sequence $\mathcal{C}_i = \{\hat{z}_{i-K}, \cdots, \hat{z}_{i-1}\}$, the hidden state is recursively updated from previous states as $c_i = \varphi(c_{i-1}, \hat{z}_{i-1}; W_c)$, starting from a random state.

6.3.3 Image decoder

To predict the i-th image, we use c_i, the past context encoding, ψ_i, the embedding of the next inter-arrival time τ_i, as well as Ψ_i, the embedding of the time duration since the start of the operation δ_i. Specifically, we compute $\hat{x}_i = \phi_d([c_i \oplus (\psi_i + \Psi_i)]; W_d)$ where $\phi_d(\cdot; W_d)$ is the decoder network. Note that the decoder network is pre-trained along with the image encoder using the image reconstruction task on the images from the training set $\mathcal{D}_N$. The prediction error is computed as the Frobenius norm difference between the original and the forecasted image. The total training loss is obtained by averaging the prediction errors over all the training examples:

$$\mathcal{L}_{\text{train}} = \frac{1}{|\mathcal{D}_N|} \sum_{i=1}^{|\mathcal{D}_N|} \|x_i - \hat{x}_i\|_F^2 \tag{6.3}$$

Finally, the anomaly score of a new point (x, y, t) is defined as the associated prediction error, i.e. $s(x, t) = \|x - \hat{x}\|_F^2$. Algorithm 2 summarises the different steps of our `ForecastAD` method.

6.4. Experimental setup

Baselines. We first compare `ForecastAD` against simple methods, which detect anomalies based on statistical features extracted from the images. These features include the corresponding time of day, as well as the mean, maximum, and standard deviation of the temperature, to distinguish between normal and

Algorithm 2: Training process of `ForecastAD`

Require: Training dataset $\mathcal{D}_N$, Sinusoidal encoder $f_{\sin}$
Image encoder ϕ_e, Image decoder ϕ_d, Number of epochs e
Learning rate η, Context length K

for (epoch $= 1, 2, \cdots, e$) and $((x_i, t_i, y_i) \in \mathcal{D}_N)$ **do**
  Initialize context embedding:
    $c_i \leftarrow$ `random()`
  for $k = K, K-1, \cdots, 1$ **do**
    Calculate the time embeddings:
      $\psi_{i-k} \leftarrow f_{\sin}(\tau_{i-k})$
      $\Psi_{i-k} \leftarrow f_{\sin}(\delta_{i-k})$
    Create joint embedding:
      $\hat{z}_{i-k} \leftarrow [\phi_e(x_{i-k}; W_e) \oplus (\psi_{i-k} + \Psi_{i-k})]$
    Update context embedding
      $c_i \leftarrow \varphi(c_i, \hat{z}_{i-k}; W_c)$
  end
  Encode target time embeddings:
    $\psi_i \leftarrow f_{\sin}(\tau_i)$, $\Psi_i \leftarrow f_{\sin}(\delta_i)$
  Predict the next data point:
    $\hat{x}_i \leftarrow \phi_d([c_i \oplus (\psi_i + \Psi_i)]; W_d)$
  Update the model parameters W_e, W_d and W_c by minimising the loss $\mathcal{L}_{\text{train}}$ (Eq. 6.3)
end

Return: $\phi_e(\cdot; W_e)$, $\phi_d(\cdot; W_d)$, $\varphi(\cdot; W_c)$

abnormal samples. We also evaluate against deep AD methods, namely, autoencoder, FastFlow (Yu et al., 2021), PatchCore (Roth et al., 2022), PaDiM (Defard et al., 2021), DRÆM (Zavrtanik et al., 2021), CFlow (Gudovskiy et al., 2022), and Reverse Distillation (Deng and Li, 2022). Deep methods have been shown to be more effective than shallow ones for image AD (Ruff et al., 2021), leveraging the deep neural networks' capability to extract representative features through multiple layers of abstraction.

Network architectures and hyperparameters. Except for the autoencoder, the baselines follow the implementation from Anomalib (Akcay et al., 2022), which is a widely used library for benchmarking AD methods. Based on our experiments, we opted for a Deep Convolutional Autoencoder (DCAE) with a latent dimension of $d' = 128$. Detailed architectural specifications are provided in Appendix D.1. The image encoder employed in `ForecastAD` mir-

rors the structure of the downsampling branch in DCAE. In `ForecastAD`, we adopt a 4-layer LSTM network with a hidden dimension of 128 to serve as the context encoder φ. For time encoding, the sinusoidal embedding has a dimension of 16. We adhere to the hyperparameters mentioned by the authors for the baseline methods. For `ForecastAD`, we use MSE and train using an Adam optimiser with a learning rate of 0.001 and weight decay of 0.00001. We use a pre-processing step for all the experiments where the images in the dataset are resized to 256×256 to be compatible with the baselines. Unless otherwise specified, we use a sequence length of $K = 30$.

Dataset. Our labelled dataset comprises days, which are segmented into training, validation, and test sets. Days featuring exclusively normal samples are allocated across these three sets, while those with anomalous samples are included in both the validation and test sets. To underscore the challenges presented by low-temperature samples, we adopt two training setups: (i) **[Tr#1]**, incorporating training and validation samples solely from the M phase, and (ii) **[Tr#2]**, comprising training and validation samples from the S, M, and E phases. Importantly, the test set in both scenarios consists of samples spanning the S, M, and E phases. The distribution of normal and anomalous samples across S, M, and E phases for these setups is depicted in Figure 6.4. For `ForecastAD`, we generate a sequence for each data point by selecting K preceding samples. If there are less than K prior samples, we duplicate the corresponding day's first data point to form a K-length sequence. Lastly, for the first data point captured each day, we set the τ and δ to a small positive value $\epsilon = 1e - 5$.

Model evaluation. We evaluate the models based on the Area under the Receiver Operating Characteristics curve (AUROC) and the Area under the Precision-Recall curve (AUPR). To highlight the effectiveness of each model in distinguishing between low-temperature normal and anomalous behaviours, we utilise three test setups containing: (i) test samples in M **[Ts#1]**, (ii) test samples in S-E **[Ts#2]**, and (iii) test samples in S-M-E **[Ts#3]**. For the experiments below, we report the mean over 5 runs along with one standard error.

6.5. Results and discussion

We summarise the results over five runs for different training setups in Table 6.1. For the training setup **[Tr#1]**, `ForecastAD` provides competitive results when compared to image-based SOTA models as measured by both AUROC

Table 6.1: Anomaly detection performance. Style: best in bold and second best using underline

Train Setting	Model	AUROC (%)			AUPR (%)		
		[Ts#1]	[Ts#2]	[Ts#3]	[Ts#1]	[Ts#2]	[Ts#3]
[Tr#1]	Autoencoder	98.05 (± 0.74)	46.43 (± 1.61)	87.87 (± 0.26)	98.50 (± 0.54)	6.62 (± 0.19)	81.46 (± 0.55)
	CFlow (Gudovskiy et al., 2022)	94.68 (± 1.26)	39.99 (± 2.33)	82.91 (± 1.08)	96.28 (± 0.92)	5.94 (± 0.24)	76.11 (± 1.04)
	DRÆM (Zavrtanik et al., 2021)	97.70 (± 0.77)	40.48 (± 2.15)	87.38 (± 0.61)	97.97 (± 0.92)	6.13 (± 0.27)	82.05 (± 0.57)
	FastFlow (Yu et al., 2021)	99.83 (± 0.03)	47.32 (± 0.29)	**91.36 (± 0.25)**	99.87 (± 0.02)	**9.42 (± 1.18)**	86.02 (± 0.52)
	PaDiM (Defard et al., 2021)	99.85 (± 0.02)	49.86 (± 0.47)	91.23 (± 0.10)	**99.89 (± 0.01)**	7.73 (± 0.18)	**87.25 (± 0.21)**
	PatchCore (Roth et al., 2022)	99.23 (± 0.08)	**50.58 (± 0.37)**	89.04 (± 0.30)	99.43 (± 0.05)	7.25 (± 0.08)	83.41 (± 0.27)
	Reverse Distillation (Deng and Li, 2022)	93.88 (± 1.13)	41.31 (± 2.19)	84.61 (± 1.54)	95.47 (± 0.79)	6.08 (± 0.30)	80.70 (± 1.37)
	`ForecastAD`	**99.86 (± 0.05)**	46.22 (± 1.06)	89.89 (± 0.35)	**99.89 (± 0.04)**	6.57 (± 0.09)	85.75 (± 0.65)
[Tr#2]	Autoencoder	96.67 (± 0.77)	45.92 (± 2.47)	85.45 (± 1.18)	96.91 (± 0.93)	6.69 (± 0.35)	78.61 (± 1.30)
	CFlow (Gudovskiy et al., 2022)	84.91 (± 2.72)	42.90 (± 2.71)	77.38 (± 2.98)	88.18 (± 2.02)	6.51 (± 0.39)	74.80 (± 3.24)
	DRÆM (Zavrtanik et al., 2021)	93.52 (± 0.52)	40.51 (± 1.33)	85.71 (± 0.78)	94.56 (± 0.44)	7.62 (± 1.01)	83.36 (± 1.08)
	FastFlow (Yu et al., 2021)	92.38 (± 0.72)	52.51 (± 1.09)	89.92 (± 0.68)	93.46 (± 0.60)	8.87 (± 0.46)	88.76 (± 0.56)
	PaDiM (Defard et al., 2021)	95.99 (± 0.37)	58.14 (± 1.00)	92.28 (± 0.32)	96.77 (± 0.32)	11.50 (± 0.86)	90.73 (± 0.42)
	PatchCore (Roth et al., 2022)	**96.78 (± 0.57)**	60.15 (± 1.82)	91.38 (± 0.42)	**97.57 (± 0.37)**	9.77 (± 0.79)	88.92 (± 0.72)
	Reverse Distillation (Deng and Li, 2022)	87.19 (± 0.99)	57.22 (± 5.77)	84.04 (± 1.64)	86.09 (± 1.34)	10.64 (± 1.55)	78.83 (± 2.66)
	`ForecastAD`	94.78 (± 1.09)	**85.81 (± 1.23)**	**92.53 (± 0.81)**	96.92 (± 0.57)	**28.73 (± 1.70)**	**92.97 (± 0.36)**

and AUPR metrics over the test samples in **[Ts#3]**. Additionally, we observe good performance for image-based SOTA approaches in **[Ts#1]**. This performance can be attributed to the training exclusively on samples from M, which predominantly fall within the high-temperature region where temporal context is less critical. However, since the models are not trained on low-temperature normal samples from the start and end of the operational cycle, their performance in **[Ts#2]** naturally declines. Specifically, the AUPR score is significantly low in **[Ts#2]**, as the models tend to assign very high anomaly scores to most low-temperature samples found in S-E.

Considering the setup **[Tr#2]**, we observe a drop in performance for the SOTA methods compared to **[Tr#1]**. This observation can be attributed to the fact that when the model is exposed to a limited number of low-temperature normal samples, it struggles to learn from them. Instead, these samples act as contamination, diminishing performance over the high-temperature samples in **[Ts#1]**. Additionally, baselines fail to distinguish between low-temperature normal and anomalous samples in **[Ts#2]**, as they do not incorporate temporal features. `ForecastAD` significantly outperforms all baselines by approx. 25% in **[Ts#2]**, while maintaining competitive performance across all test samples in **[Ts#3]**. In Appendix D.4, we provide an extended version of Table 6.1.

6.6. Ablation study

Importance of time-embedding and pre-training. Table 6.2 shows the results of an ablation study to understand the importance of τ and δ in

`ForecastAD`. In all configurations, we always keep the image encoding as part of the input. Firstly, we observe the lowest AUROC and AUPR scores in **[Ts#2]** when the context has only the encodings of K-prior images. It emphasises the need to address the challenge posed by irregular sequences and the co-occurrence of low-temperature normal and anomalous samples. Then, on considering either τ or δ, we observe a significant improvement in **[Ts#2]**. Furthermore, incorporating both τ and δ yields the best performance, highlighting that both τ and δ are necessary for reliable detection of anomalies. Lastly, we also empirically validate the impact of pre-training the image encoder and decoder using the image reconstruction task. Using the pre-trained models offers substantial enhancements in performance when compared to a randomly initialised backbone.

Table 6.2: Ablation of time-embedding and pre-training.

Pre-train	τ	δ	AUROC (%)			AUPR (%)		
			[Ts#1]	[Ts#2]	[Ts#3]	[Ts#1]	[Ts#2]	[Ts#3]
-	✓	✓	94.60 (± 1.60)	75.30 (± 5.89)	90.58 (± 1.00)	96.78 (± 0.81)	23.54 (± 4.17)	89.93 (± 1.29)
✓	-	-	**97.12 (± 0.44)**	72.94 (± 7.15)	92.74 (± 1.26)	98.06 (± 0.29)	21.32 (± 4.29)	91.41 (± 1.88)
✓	✓	-	94.59 (± 0.93)	84.08 (± 3.83)	92.49 (± 0.79)	96.56 (± 0.49)	28.83 (± 4.17)	92.67 (± 0.57)
✓	-	✓	92.71 (± 1.32)	82.71 (± 3.09)	91.12 (± 1.09)	95.58 (± 0.95)	26.31 (± 3.60)	92.15 (± 0.96)
✓	✓	✓	94.78 (± 1.09)	**85.81 (± 1.23)**	**92.53 (± 0.81)**	**96.92 (± 0.57)**	**28.73 (± 1.70)**	**92.97 (± 0.36)**

Effect of context length (K). We report the AD performance of `ForecastAD` with varying context lengths K in Table 6.3. For context lengths $K \leq 20$, we do not observe any correlation between performance and context length. However, larger sequence lengths of 30 or 40 yield better performance. To limit computational demands, we did not consider larger sequence lengths and chose a sequence length of 30 for all our experiments.

Table 6.3: Ablation of K

K	AUROC (%)			AUPR (%)		
	[Ts#1]	[Ts#2]	[Ts#3]	[Ts#1]	[Ts#2]	[Ts#3]
1	88.85 (± 2.55)	78.26 (± 1.86)	83.64 (± 1.72)	92.68 (± 1.49)	23.14 (± 2.63)	83.22 (± 1.01)
5	91.21 (± 0.94)	**87.83 (± 1.45)**	89.25 (± 0.81)	94.44 (± 0.51)	**32.24 (± 2.15)**	89.82 (± 0.49)
10	94.02 (± 1.81)	78.13 (± 3.91)	89.82 (± 0.77)	96.40 (± 0.93)	23.05 (± 2.07)	89.29 (± 0.86)
20	92.64 (± 1.21)	83.22 (± 2.90)	90.46 (± 1.31)	95.65 (± 0.60)	27.20 (± 3.45)	91.39 (± 0.93)
30	**94.78 (± 1.09)**	85.81 (± 1.23)	**92.53 (± 0.81)**	**96.92 (± 0.57)**	28.73 (± 1.70)	**92.97 (± 0.36)**
40	92.66 (± 1.46)	85.87 (± 0.92)	91.13 (± 1.26)	95.59 (± 0.73)	29.24 (± 1.49)	91.88 (± 0.83)
50	93.44 (± 0.72)	87.29 (± 1.77)	92.09 (± 0.45)	96.06 (± 0.34)	31.51 (± 1.52)	92.62 (± 0.31)
60	93.36 (± 0.75)	81.03 (± 4.25)	91.03 (± 1.51)	95.95 (± 0.44)	28.46 (± 3.83)	91.45 (± 1.58)

Effect of different architecture. In Figure 6.5, we analyse the effect of the number of layers in LSTM and the latent dimension d' on the AUROC and AUPR scores. Firstly, we observe that larger latent dimensions lead to higher scores in most cases, regardless of the number of layers in the LSTM.

Secondly, `ForecastAD` performs better on **[Ts#1]** and **[Ts#3]** with a 4-layer LSTM, while a 2-layer LSTM yields better results on **[Ts#2]**. Based on this empirical observation, we chose a 4-layer LSTM with latent dimension $d' = 128$, which has the highest scores in **[Ts#1]** and **[Ts#3]** while having a comparable performance with the best configuration on **[Ts#2]**.

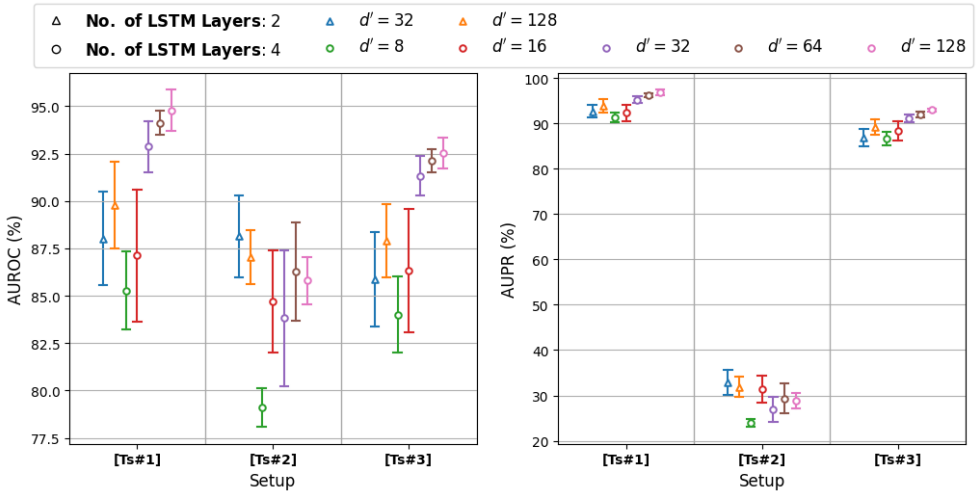


Figure 6.5: Ablation of different architectures

6.7. Interpretability of `ForecastAD`

Interpretability of deep learning models is critical for high-risk applications to enhance transparency and trustworthiness. Therefore, we extract anomaly maps from `ForecastAD` corresponding to each image during inference. Recall that `ForecastAD` is trained with pixel-wise regression loss, and thus, the anomaly map can be computed as the difference between the original and forecasted images. Based on recent works on IAD (Roth et al., 2022; Defard et al., 2021), we smoothed the anomaly maps using a Gaussian filter and normalized it using the minimum and maximum anomaly scores for the normal samples in the validation set. In Figure 6.6, we show the anomaly maps of 4 normal and 4 anomalous test samples, along with the image for reference. It can be seen that for specific types of anomalies, such as freezing, where we observe high-temperature streaks, `ForecastAD` assigns high anomaly scores to those regions. Therefore, it aids the interpretability of the results from `ForecastAD`. To further enhance the understanding, the anomaly maps can be complemented by plots of mean temperature to show the sudden drops or rises in temperature

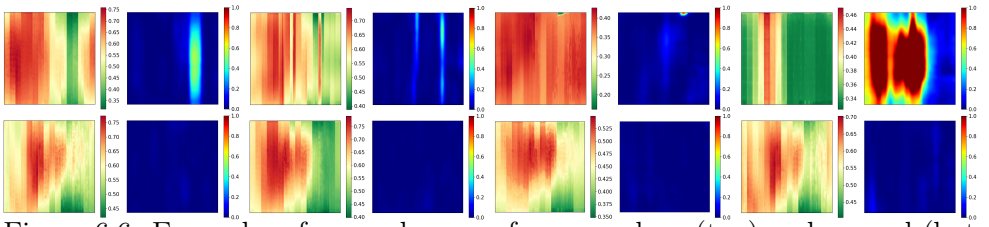


Figure 6.6: Examples of anomaly maps for anomalous (top) and normal (bottom) images.

resulting in the samples being anomalous.

6.8. Simulated dataset

We have prepared a simulated dataset to ensure reproducibility and validation of the results. We use a variational autoencoder to generate the data. Additional details about the data generation are deferred to Appendix D.2. The distribution of normal and anomalous samples across S, M, and E phases for these setups is depicted in Figure 6.7. We have also compared our method to the baselines on the simulated dataset. The results are reported in Table 6.4 for training setup **[Tr#2]** and test setup **[Ts#3]**, which are the main focus of our work. The results highlight the effectiveness of `ForecastAD`, similar to our results on the original dataset as reported in Table 6.1 of the paper. Furthermore, in Appendix D.2, we provide qualitative evidence to support the validity of the simulated dataset by visualising generated and original images for a random set of timestamps.

Table 6.4: Anomaly detection performance on simulated data.

Model	**AUROC** (%)			**AUPR** (%)		
	[Ts#1]	**[Ts#2]**	**[Ts#3]**	**[Ts#1]**	**[Ts#2]**	**[Ts#3]**
Autoencoder	87.97 (± 4.08)	66.34 (± 2.49)	82.00 (± 1.58)	94.04 (± 1.99)	24.72 (± 6.51)	83.46 (± 1.61)
CFlow (Gudovskiy et al., 2022)	83.42 (± 2.97)	51.32 (± 4.16)	70.30 (± 2.67)	90.67 (± 1.97)	10.46 (± 0.80)	69.42 (± 2.14)
DRÆM (Zavrtanik et al., 2021)	98.11 (± 0.81)	61.89 (± 5.32)	89.02 (± 0.81)	99.02 (± 0.40)	25.90 (± 4.32)	88.52 (± 0.75)
FastFlow (Yu et al., 2021)	97.24 (± 0.54)	52.23 (± 3.63)	87.98 (± 0.67)	98.43 (± 0.26)	9.49 (± 0.63)	87.76 (± 0.93)
PaDiM (Defard et al., 2021)	97.93 (± 0.56)	56.04 (± 0.42)	88.97 (± 0.44)	98.76 (± 0.31)	9.89 (± 0.07)	88.25 (± 0.25)
PatchCore (Roth et al., 2022)	98.28 (± 0.29)	66.42 (± 1.96)	92.31 (± 0.31)	98.81 (± 0.20)	21.57 (± 2.79)	92.28 (± 0.26)
Reverse Distillation (Deng and Li, 2022)	75.80 (± 5.53)	57.59 (± 4.25)	65.60 (± 4.60)	86.23 (± 3.10)	12.14 (± 1.95)	63.36 (± 3.35)
`ForecastAD`	**98.73 (± 0.64)**	**98.61 (± 0.43)**	**97.84 (± 0.65)**	**99.30 (± 0.33)**	**88.03 (± 3.18)**	**97.96 (± 0.54)**

6.9. Deployment

We have tested `ForecastAD` over five months of data from an operational CSP plant. A freshly labelled dataset was curated by initially applying a prede-

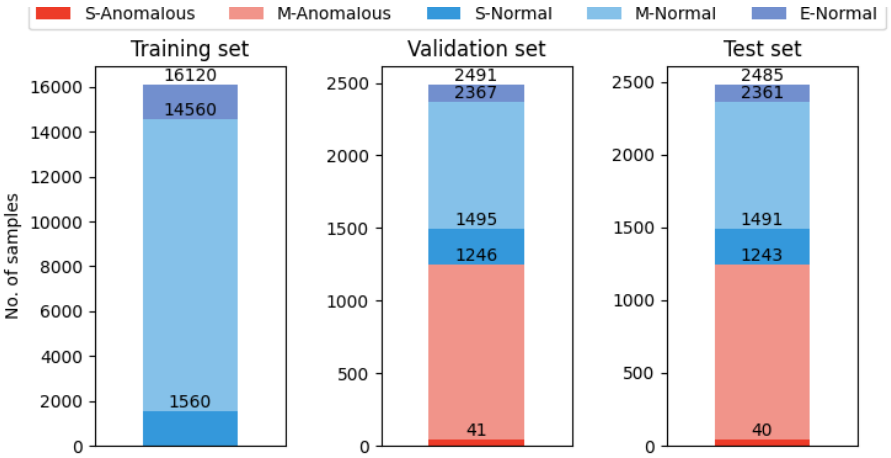


Figure 6.7: Data split for simulated dataset

fined set of labelling rules, followed by a meticulous review and cleanup of the dataset with guidance from domain experts. The performance metrics of `ForecastAD` on this labelled set containing 8373 normal and 1,321 abnormal samples are detailed in Table 6.5. Furthermore, the deployment results are broken down per month over different operating stages. Please note that for some months, we could not compute the performance metrics as there are no anomalous samples present in the dataset. Such cases are marked as "–" in the table. It is important to note that there is variability in this data, such as different stages of operations (starting, ending, and middle) and varying external weather conditions. We can observe that `ForecastAD` is fairly robust in the detection of anomalies over this period. The actionable insights derived from `ForecastAD` contribute to the strategic maintenance planning of the CSP plant, thereby enhancing the durability of its equipment.

Table 6.5: Deployment performance

Month	**AUROC (%)**			**AUPR (%)**		
	[Ts#1]	**[Ts#2]**	**[Ts#3]**	**[Ts#1]**	**[Ts#2]**	**[Ts#3]**
1	0.89	0.71	0.89	0.66	0.17	0.62
2	0.86	0.94	0.85	0.82	0.70	0.79
3	0.96	0.93	0.95	0.91	0.19	0.87
4	0.91	-	0.91	0.75	-	0.63
5	0.85	-	0.85	0.60	-	0.57
Overall	0.88	0.81	0.88	0.72	0.25	0.69

6.10. Conclusion

We address the problem of anomaly detection in irregular sequences of thermal images collected from IR cameras in an operational CSP plant. Extensive analysis of our dataset reveals distinctive temporal characteristics, setting it apart from established AD industrial image benchmark datasets like MVTec (Bergmann et al., 2019). We empirically demonstrate that image-based SOTA AD methods underperform, especially when context is critical for anomaly detection. We also introduce a forecasting-based AD method, `ForecastAD`, that predicts future thermal images from past sequences and timestamps using a deep sequence model. This method effectively captures specific temporal data features and distinguishes between difficult-to-detect temperature-based anomalies. Experimental results demonstrate the effectiveness of `ForecastAD`, outperforming existing SOTA methods as measured by AUROC and AUPR. Notably, `ForecastAD` exhibits significant enhancements in detecting anomalous behaviours, particularly among low-temperature samples. Furthermore, `ForecastAD` has been successfully deployed, providing critical insights for the maintenance of the CSP plant to our industry partner. For future work, we aim to further study the role of context and sequence lengths in anomaly detection performance. We also aim to extend our model to be more robust to distribution shifts inherent in industrial processes, notably by considering probabilistic forecasting models.

CHAPTER 7

Risk-Based Thresholding for Reliable Anomaly Detection

Anomalous images can be detected by thresholding an anomaly score, where the threshold is chosen to optimise metrics such as the F1-score on a validation set. This work proposes a framework, using risk control, for generating more reliable decision thresholds with finite-sample coverage guarantees on any chosen risk function. Our framework also incorporates an abstention mechanism, allowing high-risk predictions to be deferred to domain experts. Focusing on the detection of anomalous behaviours in CSP plants, as introduced in the previous chapter, we propose a density forecasting method to estimate the likelihood of an observed image given a sequence of previously observed images. The estimated likelihood is used as the anomaly score within the proposed framework to obtain reliable decision thresholds. We empirically evaluated the effectiveness of our proposed approach across multiple training configurations over several months of operational data from two CSP plants.

This chapter is based on the following publication.

- Yorick Estievenart, **Sukanya Patra** & Souhaib Ben Taieb (2025b). Risk-Based Thresholding for Reliable Anomaly Detection in Concentrated Solar Power Plants. In the *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.

Yorick Estievenart led the development of the proposed framework to compute reliable decision thresholds for AD and conducted the experimental studies. He also collaborated with Sukanya Patra on the design and implementation of the proposed density-based AD method. The scientific article was primarily written by Sukanya Patra, with assistance from Yorick Estievenart, particularly in the section addressing reliable threshold estimation for AD. The project was conducted under the supervision of Souhaib Ben Taieb.

7.1. Introduction

Despite significant progress in both deep and shallow AD research (Liu et al., 2024; Ruff et al., 2021), existing image- and video-based approaches fall short in addressing the problem of detecting anomalous behaviours of operational CSP plants as discussed previously in Chapter 6, Section 3) due to three key challenges. First, unlike classical image- and video-based AD data, CSP plant monitoring involves thermal images without semantic content, lacks a fixed frame rate, and exhibits significant non-stationarity and temporal dependencies due to pronounced daily seasonal patterns. As a result, conventional image- and video-based anomaly detection methods developed for images or videos with semantic content are not applicable. A recent forecasting-based AD method, `ForecastAD` (Patra et al., 2024), attempts to address these challenges by computing the anomaly score as the per-pixel errors between predicted and observed thermal images. However, reconstruction-based AD methods suffer a critical flaw where models trained on normal data can inadvertently reconstruct and misclassify anomalous images as normal (Bouman and Heskes, 2025; Moore and Morelli, 2024), leading to unreliable detection. Second, given only the anomaly score from an AD method, its lack of interpretability hinders decision-making in high-stakes applications without an appropriate thresholding strategy (Perini et al., 2021). Traditional approaches rely on performance metrics such as F1-score or GMean to determine thresholds depending on the available labelled samples. These methods do not provide any statistical guar-

antees on the errors. Moreover, they assume that all CSP plants define risk similarly and follow the same operational strategies. In reality, this often differs (e.g., deploying a maintenance team may be preferable to replacing a tower component). Third, deep learning-based AD models are often perceived as unreliable (Perini et al., 2021) due to the uncertainty in predictions stemming from their inability to properly estimate the decision boundary, particularly when training data is limited. Thus, practitioners are hesitant to use such predictions even when the associated uncertainty is minimal, severely limiting their adoption in real-world applications.

To overcome these limitations, we propose a principled, reliable AD framework tailored for CSP plant monitoring. First, we introduce a risk-controlling thresholding strategy for anomaly scores that satisfies finite-sample coverage guarantees on any chosen risk function (e.g., false positive rate or F1-score)—a critical requirement for reliable predictive maintenance (PdM) in industrial settings. To enhance trust and adoption, we integrate a machine-learning-with-abstention framework (Perini and Davis, 2023) with adaptive thresholds that account for the overlap between normal and anomalous score distributions. This approach defers high-risk predictions to domain experts, ensuring human intervention when uncertainty is high. Furthermore, we propose an AD method based on density forecasting, `DensityAD`, which leverages conditional normalizing flows to model the likelihood of an observed sample being normal, given past thermal images and timestamps. This approach mitigates the limitations of reconstruction-based methods and enables likelihood-based thresholding for more effective anomaly detection. Our key contributions are:

- We propose a framework for computing reliable anomaly detection thresholds with finite-sample coverage guarantees for any chosen risk function. The framework includes an abstention mechanism that defers decisions to domain experts under high uncertainty.
- We develop an unsupervised AD method that computes anomaly scores using density forecasting by estimating the conditional likelihood of an observed infrared image given a sequence of previously observed images.
- We conduct an extensive deployment analysis of our framework across multiple real-world scenarios over several months, using data from two CSP plants. This analysis provides valuable insights to our industry partner for maintenance operations.
- As the data from the operational CSP is confidential, to foster reproducibility, we release a simulated dataset by leveraging recent advance-

ments in generative modelling to create diverse infrared images that emulate the real-world dataset.

Our work not only advances the state of anomaly detection in renewable energy systems but also serves as an important milestone for future research in reliable, data-driven PdM strategies for critical infrastructure monitoring.

7.2. Background

Notations. In this chapter, we follow the notations from the paper (Estievenart et al., 2026). We consider an unsupervised AD setting, where the training dataset, denoted as $\mathcal{D}_N = \{x_i\}_{i=1}^n$, consists of n *unlabeled* samples. Each sample $x_i = (y_i, t_i) \in \mathcal{X}$ is a tuple, where $\mathcal{X} = \mathbb{R}_+^d \times \mathbb{R}_+$. The first component, $y_i \in \mathcal{Y}$, represents a thermal image of dimension $d = H \times W$, where H and W denote the height and width, respectively, with $\mathcal{Y} = \mathbb{R}_+^d$. The second component, $t_i \in \mathbb{R}_+$, corresponds to the timestamp at which the thermal image y_i was captured. Following prior works (Roth et al., 2022), we assume that the training dataset $\mathcal{D}_N$ predominantly contains normal samples. Additionally, we consider another *labelled* dataset, $\mathcal{D}_R = \{(x_i, z_i)\}_{i=1}^{n_R}$, consisting of n_R labeled pairs, where $n_R \ll n$. Each label $z_i \in \mathcal{Z}$, where $\mathcal{Z} = \{0, 1\}$, indicates whether the corresponding sample is normal ($z_i = 0$) or anomalous ($z_i = 1$). Section 6.2 contains details on how the dataset is labelled. Furthermore, the dataset $\mathcal{D}_R$ is partitioned into three disjoint subsets: validation ($\mathcal{D}_V$), calibration ($\mathcal{D}_C$), and test ($\mathcal{D}_T$).

Unsupervised AD. As discussed previously in Chapter 2, unsupervised AD is to estimate an anomaly score function $s(\cdot) : \mathcal{X} \to \mathbb{R}$ using $\mathcal{D}_N$, such that normal samples receive lower scores. A label (0 for normal or 1 for anomalous) is then assigned to a new test sample $x \in \mathcal{X}$ by thresholding its anomaly score:

$$\hat{z} = h(x) = \begin{cases} 0, & \text{if } s(x) \leq \lambda, \\ 1, & \text{if } s(x) > \lambda, \end{cases} \tag{7.1}$$

where $h : \mathcal{X} \to \{0, 1\}$ is the labelling function and $\lambda \in \mathbb{R}$ is a threshold to be determined, whose optimal value depends on the proportion of anomalies in the test set (Perini et al., 2023, 2022). However, since the true proportion is unknown in practice, existing methods rely on test performance metrics to select a threshold $\lambda \in \Lambda$ from a set of thresholds $\Lambda \subset \mathbb{R}$. Commonly adopted approaches include:

→ **F1-score (Akcay et al., 2022)**. The threshold λ_{F} yields the highest F1-score:

$$\lambda_{\text{F}} = \arg\max_{\lambda \in \Lambda} \text{F1-Score}\big(\mathcal{H}_V\big), \tag{7.2}$$

where $\mathcal{H}_V = \{(h(x), z) \mid (x, z) \in \mathcal{D}_V\}$. Given set $\mathcal{H}_V$, the function F1-Score$(\cdot)$ first determines the number of true positives, false positives, and false negatives and then computes the F1-score according to the formula defined in Section 2.1.6.

→ **G-Mean (Kubat et al., 1997).** The threshold λ_{G} maximizes the G-Mean:

$$\lambda_{\text{G}} = \arg\max_{\lambda \in \Lambda} \text{G-Mean}\big(\mathcal{H}_V\big). \tag{7.3}$$

Here, G-Mean$(\cdot)$ first determines the number of true positives, false positives, and false negatives given the set $\mathcal{H}_V$, then computes the geometric mean of precision and recall.

→ **Z-score**. Let $\mathcal{S}_V$ be the set of anomaly scores for normal samples in $\mathcal{D}_V$, defined as $\mathcal{S}_V = \{s(x) \mid (x, 0) \in \mathcal{D}_V\}$ with the corresponding mean and standard deviation denoted by μ_{S_V} and σ_{S_V}, respectively. The threshold λ_{z} is set k standard deviations above μ_{S_V}. Unlike Eq. 7.1, where the threshold is applied directly to $s(x)$, here it is applied to the z-scores, defined as $s_z(x) = \left|\frac{s(x)-\mu_{S_V}}{\sigma_{S_V}}\right|$.

For a comprehensive discussion on existing methods for selecting λ, we refer to (Perini et al., 2023). A key limitation of these approaches is that they do not account for uncertainty when the anomaly score distributions of normal and anomalous samples overlap, as illustrated in Figure 7.1. However, given the high-risk nature of AD applications, it is essential to abstain from assigning labels under high uncertainty. This allows domain experts to intervene, reducing the risk of incorrect classifications and ensuring more reliable decision-making.

Unsupervised AD with abstention. To enable abstention from labelling under high uncertainty, we augment the labelling function with an abstention label (Ⓡ) and introduce two thresholds (λ^l and λ^h), reformulating $h(x)$ as:

$$\hat{z} = h(x) = \begin{cases} 0, & \text{if } x \in \hat{\mathcal{N}}, \quad \hat{\mathcal{N}} = \{x' \in \mathcal{X} \mid s(x') \leq \lambda^l\}, \\ 1, & \text{if } x \in \hat{\mathcal{A}}, \quad \hat{\mathcal{A}} = \{x' \in \mathcal{X} \mid s(x') \geq \lambda^h\}, \\ \text{Ⓡ}, & \text{if } x \in \hat{\mathcal{U}}, \quad \hat{\mathcal{U}} = \{x' \in \mathcal{X} \mid \lambda^l < s(x') < \lambda^h\}, \end{cases} \tag{7.4}$$

where $\hat{\mathcal{N}}$ is the *normal prediction region*, $\hat{\mathcal{A}}$ is the *anomalous prediction region*, and $\hat{\mathcal{U}}$ is the *abstention region*, where the model refrains from making a decision.

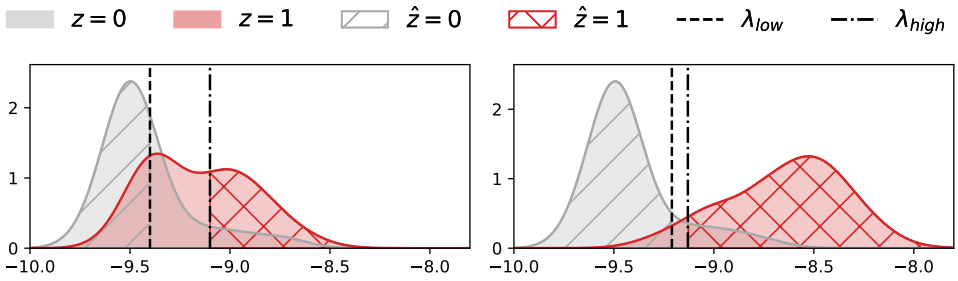


Figure 7.1: Illustration of AD with abstention under high (left) and low (right) overlap in anomaly score distributions of normal and anomalous samples.

Figure 7.1 illustrates two examples of this decision-making process. The parameters λ^l and λ^h define the normal and anomalous prediction regions while also regulating the abstention region, thereby controlling the abstention rate.

Ideally, the pair of thresholds (λ^l, λ^h) should adapt to the anomaly score distribution, effectively capturing the overlap between normal and anomalous scores. A trivial yet uninformative approach is to set $\lambda^l = -\infty$ and $\lambda^h = +\infty$, which results in abstaining from prediction for all samples. We aim to propose a principled method for selecting a reliable pair of thresholds.

7.3. Reliable decision thresholds for AD

Recall that a Risk-Controlling Prediction Set (RCPS) $\widehat{C}_\lambda$ for a given threshold $\lambda \in \Lambda$ can be defined as in Definition 2.4.5. One method for constructing an RCPS is *conformal risk control*, an extension of conformal prediction (CP) (Angelopoulos et al., 2023) designed to control the expected value of a risk function, assuming it is monotonically non-increasing with respect to a single threshold λ. However, this approach is limited to a single-parameter setting, as in (7.1), and relies on a restrictive assumption about the risk function.

To overcome these limitations, we propose leveraging the *Learn then Test* (LTT) procedure (Angelopoulos et al., 2025). We consider the unsupervised AD problem with abstention, as defined in (7.4). Our objective is to determine a pair of reliable thresholds (λ^l, λ^h) that define a RCPS $\hat{C}_{(\lambda^l, \lambda^h)}(X) = h(X)$ with finite-sample coverage guarantees for any given risk function (e.g., the false positive rate). Additionally, we seek to adapt the abstention rate based on the complexity of the risk function.

Our LTT procedure for reliable threshold selection. We propose an extension of the LTT procedure, denoted as `xLTT`, which generalises the framework to consider a pair of thresholds (λ^l, λ^h) instead of a single threshold λ. The procedure begins by defining a set of paired threshold values, $\Lambda = \{(\lambda^l_{(a)}, \lambda^h_{(b)}) \mid a, b \in \{1, \ldots, m\}, \lambda^l_{(a)} \leq \lambda^h_{(b)}\}$. Next, we define the null hypotheses $\mathcal{H}_j : \hat{R}_{n_C}(\hat{C}_{(\lambda^l_j, \lambda^h_j)}) > \alpha$ for each $(\lambda^l_j, \lambda^h_j) \in \Lambda$, $j \in \{1, \ldots, |\Lambda|\}$ and $\alpha \in [0, 1]$, where $\hat{R}_{n_C}$ is an empirical risk function computed on the calibration set $\mathcal{D}_C$. Accepting $\mathcal{H}_j$ indicates that $(\lambda^l_j, \lambda^h_j)$ does not control the risk. To decide whether to accept or reject $\mathcal{H}_j$ and thus verify whether the risk is controlled for a given pair $(\lambda^l_j, \lambda^h_j)$, we compute a valid p-value p_j for every $\mathcal{H}_j$ using α. This is achieved via a concentration inequality (e.g., the Hoeffding-Bentkus inequality (Bates et al., 2021)). Based on the set of p-values $\mathcal{P} = \{p_j\}_{j \in \{1, \ldots, |\Lambda|\}}$, we then select the threshold pairs for which the risk is controlled. Since multiple comparisons increase the likelihood of false positives, a correction function $\mathcal{F} : \mathcal{P} \to \mathcal{P}'$ with $\mathcal{P}' \subseteq \mathcal{P}$ is required to maintain the desired risk control. For example, we define the set $\mathcal{O} = \mathcal{F}(\mathcal{P}) \subset \Lambda$ using Bonferroni correction as $\mathcal{F}(\mathcal{P}) = \{(\lambda^l_j, \lambda^h_j) \mid p_j \leq \frac{\delta}{|\Lambda|}, p_j \in P\}$. If $\mathcal{O} = \emptyset$, we set $\mathcal{O} = \{(-\infty, \infty)\}$. Finally, any pair $(\lambda^l, \lambda^h) \in \mathcal{O}$ ensures that $\hat{C}_{(\lambda^l, \lambda^h)}$ forms a RCPS. This method enables the use of any risk function in a post-hoc manner (i.e., without requiring retraining of a given anomaly detector), making it particularly valuable for AD in CSP plants with diverse and evolving requirements.

Optimal threshold selection for AD. Now that we have obtained the set $\mathcal{O}$ of threshold pairs that control the risk, our next objective is to (1) avoid trivial selections where $\lambda^l = -\infty$ and $\lambda^h = \infty$, and (2) minimize false positives and false negatives while keeping the abstention rate as low as possible. Let $\mathcal{I}_1 = \{i \mid (x_i, z_i) \in \mathcal{D}_V, z_i = 1\}$ and $\mathcal{I}_0 = \{i \mid (x_i, z_i) \in \mathcal{D}_V, z_i = 0\}$ be the set of indices for anomalous and normal points, respectively. Let $\hat{z}_i$ be the predicted labels computed using (7.4), with $i = 1, \ldots, |\mathcal{D}_V|$. We propose selecting the optimal thresholds λ^l_* and λ^h_* by computing:

$$\lambda^l_*, \lambda^h_* = \arg\min_{\lambda^l, \lambda^h \in \mathcal{O}} \underbrace{\frac{|\{i \in \mathcal{I}_1 \mid \hat{z}_i = 0\}|}{|\mathcal{I}_1|}}_{\text{False Negative Rate (FNR)}} + \underbrace{\frac{|\{i \in \mathcal{I}_0 \mid \hat{z}_i = 1\}|}{|\mathcal{I}_0|}}_{\text{False Positive Rate (FPR)}} + \underbrace{\frac{|\{i \mid \hat{z}_i = \varnothing\}|}{|\mathcal{D}_V|}}_{\text{Abstention Rate}}.$$

Density-Based anomaly score functions. Recent work (Novello et al., 2025) examined the intrinsic connection between anomaly detection and conformal prediction, demonstrating how insights from each field can mutually enhance the other. Building on this perspective, we leverage recent advancements in CP (Dheur et al., 2025) to develop novel anomaly score functions $s(\cdot)$[1] for the labeling function in (7.4). These score functions are further integrated with the reliable threshold selection procedure `xLTT`.

Our framework is based upon an invertible, conditional generative model (e.g., normalizing flows) $\hat{g} : \mathcal{V} \times \mathcal{C} \times \mathbb{R}_+ \to \mathcal{Y}$, where $\mathcal{V}$ is a latent space with a known distribution and $\mathcal{C}$ is the space of the conditioning variable. We defer the discussion of the exact model used to Section 7.4. Formally, $\hat{g}(\hat{g}^{-1}(y; c, t); c, t) = y$ for any $c \in \mathcal{C}$, $y \in \mathcal{Y}$ and $t \in \mathbb{R}_+$. The invertibility allows us to compute the exact density $\hat{f}(y \mid c, t)$ via the change of variables formula. For a test observation $x = (y, t)$, and given $\hat{g}$, we consider the following two approaches:

- `DR-xLTT`. The negative log-likelihood is the score function:

$$s_{\mathrm{DR}}(x; c) = -\log(\hat{f}(y \mid c, t)). \tag{7.5}$$

- `L-xLTT`. The second approach is based on an invertible model, following the L-CP method introduced by Dheur et al. (2025). Unlike the output space $\mathcal{Y}$, we expect the latent space $\mathcal{V}$ to be more structured, where normal samples are ideally clustered near the origin. Consequently, in `L-xLTT`, we frame the decision-making process as a one-class classification problem in the latent space. Assuming the latent variable follows a standard normal distribution, we use the ℓ_2 distance of the latent representation from the origin as the anomaly score for a test point x:

$$s_{\mathrm{L}}(x; c) = \|v\|, \quad \text{where } v = \hat{g}^{-1}(y; c, t). \tag{7.6}$$

7.4. Density-based AD model

The most recent AD model for CSP plants, `ForecastAD`, is a reconstruction-based AD methods. However, prior research has shown that anomalies, despite significantly different from normal data, can often be reconstructed in practice (Bouman and Heskes, 2025). For instance, in a bimodal distribution, the distance between the two peaks is greater than the distance between a peak and the local minimum separating them. In such cases, when a prediction aligns

[1] Hereafter, the score function incorporates contextual information c.

with one of the peaks, observations near the local minimum exhibit lower reconstruction errors and thus are incorrectly deemed more likely (Moore and Morelli, 2024). Figure 7.2 presents examples of IR images that are well reconstructed but are anomalous and exhibit empirically low density. To overcome

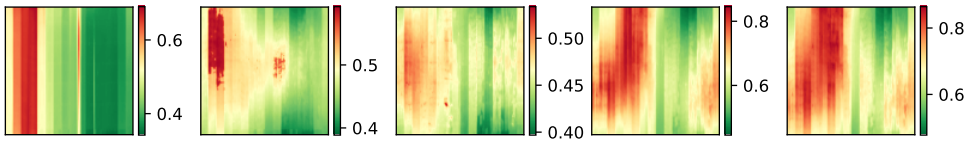


Figure 7.2: Well-reconstructed anomalous images with empirically low density.

such limitations of reconstruction-based approaches, we introduce `DensityAD`, an invertible generative model that directly estimates the density of thermal images given the contextual information from past images. `DensityAD` operates in two main steps: (i) concatenating the K preceding images and their timestamps as a context vector c, and (ii) leveraging this context to estimate the density of the current observation $x = (y, t)$, i.e. $f(y \mid c, t)$.

Context encoding. Building on (Patra et al., 2024), given a test observation $x_i = (y_i, t_i)$, we construct a rich contextual representation c_i for AD by encoding both spatial and temporal information from the preceding K images. First, at each time step t_{i-k}, where $k = 1, \cdots, K$, the corresponding image y_{i-k} is mapped into a lower-dimensional latent space. Specifically, we define an image encoder $\phi_e(\cdot; W_e) : \mathcal{Y} \to \mathcal{V}'$, which transforms images from the high-dimensional input space $\mathcal{Y}$ into a lower dimensional latent space $\mathcal{V}' = \mathbb{R}^{d'}$, where $d' \ll d$. Then, to capture temporal dependencies, we consider two temporal features: the inter-arrival time $\tau_{i-k} = t_{i-k} - t_{i-(k+1)}$, which represents the time elapsed since the previous observation, and the relative time since the start of operation $\gamma_{i-k} = t_{i-k} - t_0$, which situates the observation within the broader operational cycle. These temporal attributes are encoded using a sinusoidal function $\psi(\cdot)$. The final embedding for each data point (y_{i-k}, t_{i-k}) is then constructed by concatenating the temporal encodings with the image embedding as $\hat{c}_{i-k} = \phi_e(y_{i-k}; W_e) \oplus \psi_\tau(\tau_{i-k}) \oplus \psi_\gamma(\gamma_{i-k})$. Lastly, to generate the fixed-dimensional context vector c_i at time step t_i, the embeddings of the past K images are aggregated using a deep sequence model.

Conditional normalizing flow. The conditional PDF $f(y_i \mid c_i, t_i)$ of the current image y_i, given context c_i at timestep t_i, is estimated using a conditional normalizing flow, specifically GLOW (Lu and Huang, 2020). The invertibil-

ity property of normalizing flows (Rezende and Mohamed, 2015b; Kingma and Dhariwal, 2018) enables exact likelihood computation, which is essential for the threshold selection methods discussed in Section 7.3. To model $f(y_i \mid c_i, t_i)$, we apply conditional invertible transformations g, mapping y_i to a latent variable v_i as $v_i = g(y_i; c_i, t_i)$. The conditional log-likelihood is then computed using the change-of-variables formula. For further details, we refer the reader to (Lu and Huang, 2020).

7.5. Experiments

Here, we compare the performance of `DensityAD` against existing baselines and assess the efficacy of our proposed decision thresholds for risk-controlled AD.

7.5.1 Experimental setup

Dataset. We use data from two CSP plants, denoted as A and B. The validation set also serves as a calibration set. For the first data point of each day, both τ and γ are initialized to a small positive value, $\epsilon =$ 1e-5.

Baselines. In our evaluation, we compare the performance of `DensityAD` against deep image-based AD methods, specifically CFlow (Gudovskiy et al., 2022) and DRÆM (Zavrtanik et al., 2021). To extend the comparison to AD approaches that incorporate historical sequences of observations, similar to `DensityAD`, we include a spatiotemporal autoencoder (STAE) architecture (Hasan et al., 2016; Deepak et al., 2021; Sudhakaran and Lanz, 2017) and TimeSformer (Bertasius et al., 2021), a transformer-based video classification framework, as baselines, along with `ForecastAD` (Patra et al., 2024).

Experimental details. To prevent numerical instability during training, images are resized to 64×64, and we employ 3 flows per block across 5 blocks. The model is trained using the Adam optimizer with a learning rate of 0.0001 and a weight decay of 0.00001. Early stopping is applied based on the validation AUPR[2], maintaining a fixed balance between normal and anomalous samples in the validation set during training to mitigate the impact of dataset imbalance. The baseline models are trained following their published training setups. We used TimeSformer as the encoder in an encoder-decoder architecture, using the decoder from `ForecastAD`, and trained with a mean squared error loss. For the decision thresholds, we use $\alpha = \delta = 0.1$. We conduct an ablation study in Section 2 of the supplementary material on the context length

[2]Area Under the Precision-Recall Curve (AUPR)

Table 7.1: AUROC and AUPR performances of `DensityAD` against baseline methods. Style: best in **bold**, and second best underlined.

CSP	Model	AUROC (%) ↑	AUPR (%) ↑
A	CFlow (Gudovskiy et al., 2022)	76.46 ± 0.92	70.32 ± 1.20
	DRÆM (Zavrtanik et al., 2021)	81.55 ± 1.9	74.8 ± 2.79
	STAE	89.47 ± 1.59	87.38 ± 2.4
	TimeSformer (Bertasius et al., 2021)	87.8 ± 2.46	83.36 ± 3.15
	`ForecastAD` (Patra et al., 2024)	86.28 ± 1.74	87.57 ± 1.38
	`DensityAD`	**94.25 ± 0.2**	**93.88 ± 0.48**
B	CFlow (Gudovskiy et al., 2022)	55.8 ± 5.47	57.56 ± 4.85
	DRÆM (Zavrtanik et al., 2021)	78.82 ± 5.72	71.75 ± 8.56
	STAE	89.9 ± 1.18	88.98 ± 1.68
	TimeSformer (Bertasius et al., 2021)	88.59 ± 2.14	89.84 ± 1.29
	`ForecastAD` (Patra et al., 2024)	81.76 ± 0.7	82.88 ± 1.39
	`DensityAD`	**91.93 ± 0.52**	**90.66 ± 0.46**

K and the importance of time embeddings τ and γ. Based on the analysis, we opt for the sequence length $K = 30$ and only consider τ in `DensityAD` for modelling the temporal dynamics.

Evaluation metrics. We evaluate `DensityAD` using two primary metrics: the AUROC[3] and the AUPR. Additionally, we assess the proposed thresholding methods by reporting the risk, along with the F1-score and the corresponding abstention rate for two controlled risk measures relevant to our context: the FPR and the F1-score. These choices are not fixed—any risk function can be selected to meet the specific requirements of a CSP plant. We also report these risk measures for existing threshold selection methods. For all experiments, we present the mean over three runs along with one standard error.

7.5.2 Results and discussion

AD models. Table 7.1 presents the performance of `DensityAD` for both CSP plants. The results indicate that `DensityAD` consistently outperforms all baseline methods on both datasets. While STAE, `ForecastAD`, and TimeSformer perform well, they still fall short of the performance achieved by our `DensityAD`.

Anomaly scores. Figure 7.3 shows the distributions of normal and anomalous scores for test samples on CSP A, using the proposed scores, introduced in Section 7.3 (i.e., s_{DR} and s_{L}) and the reconstruction score s_{REC} from

[3]Area Under the Receiver Operating Characteristic Curve (AUROC)

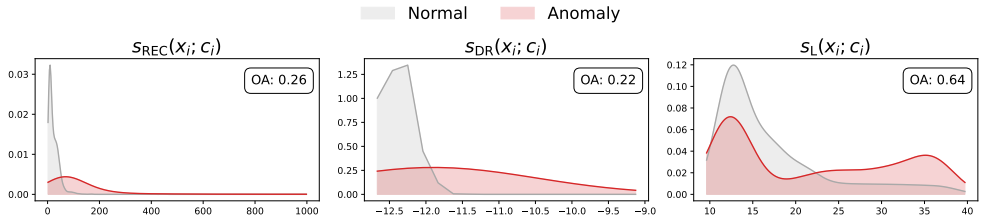


Figure 7.3: Empirical score distributions of normal and anomalous test samples from CSP A for our proposed score functions and the one used by `ForecastAD`, with the overlapping area (OA) between both distributions in the top right corner.

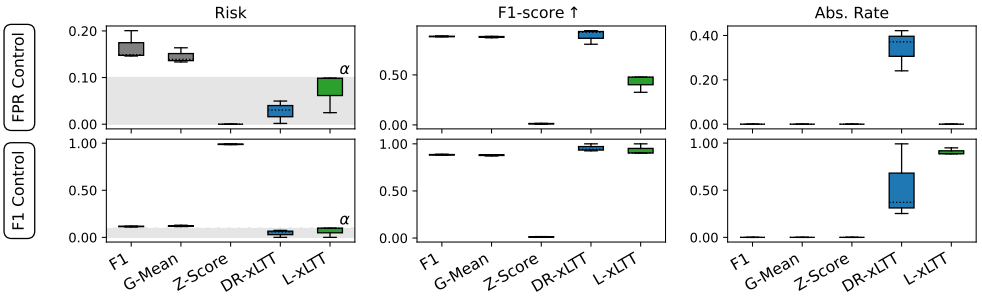
Table 7.2: Total CPU time and memory used for training the models.

	`ForecastAD`		`DensityAD`	
	CSP A	CSP B	CSP A	CSP B
Training time (s)	4151 ± 327	2204 ± 334	3760 ± 836	3248 ± 411
Memory used (Gb)	1.63 ± 0.13	1.73 ± 0.05	1.73 ± 0.02	1.63 ± 0.02
Inference time (ms)	194 ± 6.73	177 ± 1.44	201.3 ± 17.41	178 ± 2.37

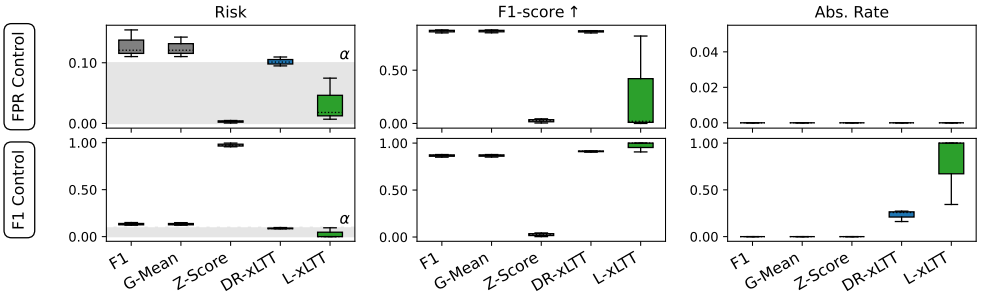
`ForecastAD`. In this example, s_{DR} and s_{REC} scores effectively distinguish normal from anomalous samples, as shown by the overlapping area (OA) between both distributions.

Anomaly threshold selection. Figure 7.4 provides an overview of the threshold selection approaches. The results clearly show that the proposed methods effectively control risk for both risk functions, whereas existing methods do not offer such guarantees. The `DR-xLTT` methods demonstrate strong performance, balancing risk control with a high F1-score while maintaining a low abstention rate. Notably, they outperform approaches that select the maximum validation set value. Furthermore, these methods adapt to the complexity of the risk function, recognising that controlling the F1-score presents greater predictive challenges than the FPR. They also fully adjust to user requirements, increasing the abstention rate when constraints are too stringent (e.g., attempting to control the F1-score with a weak underlying model).

Computation requirements. The models are trained using a single NVIDIA A100 GPU with 40 GB of memory, along with 8 CPU cores and 20 GB of RAM. Table 7.2 presents the training times (excluding the risk-control), memory usage and inference time for a single test point. As shown in Table 7.1,



(a) For CSP A



(b) For CSP B

Figure 7.4: Risk control over FPR (top row) and F1-score (bottom row) for existing and proposed methods. The risk is FPR (top row) and 1−F1 (bottom row).

`DensityAD` performs better than `ForecastAD` while using similar resources.

7.6. Deployment

We deployed our threshold selection methods using `DensityAD` on 5 and 6 months of anonymised data from CSP plants A and B, respectively. Figure 7.5 presents the thresholding results, where the FPR is used as the controlled risk. The results demonstrate that risk is effectively controlled in deployment, with `DR-xLTT` emerging as the most consistent method across both CSP plants. All methods maintain a low abstention rate, making them well-suited for deployment. Additionally, the deployment results align closely with those observed during testing. Performance fluctuations across months can be attributed to variations in the frequency and complexity of anomalies, with some months exhibiting a higher occurrence or more challenging cases.

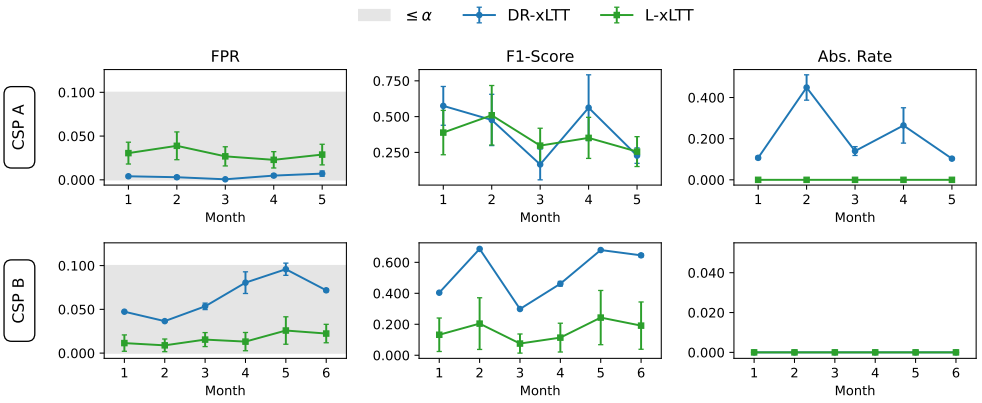


Figure 7.5: FPR control for the proposed approaches in a deployment setting over multiple months, for the two CSP plants.

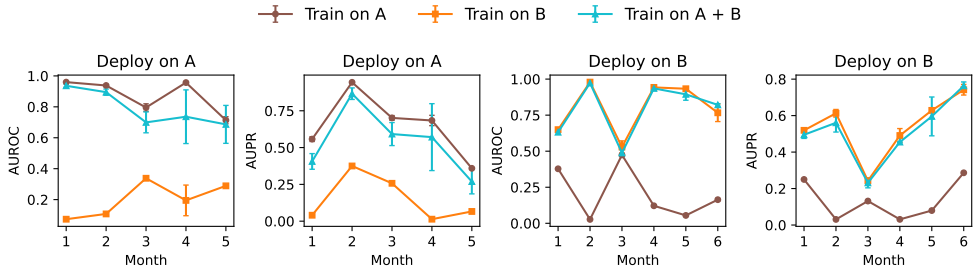


Figure 7.6: AUROC and AUPR for the two CSP plants over multiple months using three different training settings (i.e. training on A, B, and $A + B$).

Figure 7.6 evaluates the performance of `DensityAD` in deployment under three training configurations: training on CSP A, training on CSP B, and training on a combination of data from both CSP plants. As expected, deploying a model trained on a different tower results in a performance decline. Furthermore, training on data from both plants does not yield any performance improvement, suggesting that information from one tower does not generalise well to another. Although thermal flow patterns are similar across CSPs, anomaly definitions vary due to site-specific factors such as geographic location and operational context. This limits cross-site generalisation, indicating the need for per-site models or fine-tuning. Future work could address this through domain adaptation techniques. Although originally not designed for this purpose, `DensityAD` offers a general framework that can be extended to

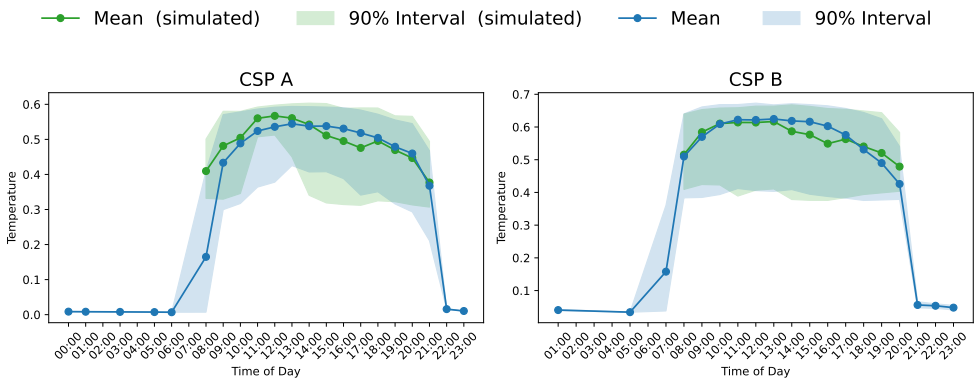


Figure 7.7: Mean daily temperature for the original and simulated datasets. The shaded area represents the 90% temperature interval.

multivariate time series anomaly detection. In this work, we focus on its application to anomaly detection in CSPs, where the anomaly score is computed and subsequently processed through a thresholding mechanism. Finally, the results suggest that the proposed method supports practical deployment by enabling control over operational risk. For instance, it allows organisations to meet predefined detection targets—such as identifying 90% of anomalies—thereby supporting compliance with regulatory or performance requirements.

7.7. Simulated dataset

Building on the methodology described in (Patra et al., 2024), we construct a simulated dataset to facilitate the reproducibility and validation of our results. `DensityAD` enables exact likelihood computation while also allowing sampling from the learned distribution. Leveraging this capability, we generate high-quality samples using our proposed density-based model. The dataset simulates two distinct CSP setups, providing a valuable resource for advancing anomaly detection research in CSP plants. Further details are provided in the supplement.

Due to transformations applied during anonymisation, we assessed the reliability of the generated images by comparing the average daily temperature profiles of both CSP plants. As shown in Figure 7.7, temperature levels during the critical period (08:00 to 20:00) closely match between the simulated and original datasets. Temperatures outside this interval, which are notably lower,

were regarded as trivial outliers and excluded from the simulated data.

7.8. Related work

Unsupervised Video AD. Beyond image-based AD discussed in Section 2.3, prior research also investigated AD in videos (Chandrakala et al., 2023; Le and Kim, 2023; Abdalla et al., 2024), using historical sequences of observations to identify deviations. However, our setting differs in two key ways: (1) our IR images lack the semantic content of typical video frames, and (2) our solar dataset is captured at irregular intervals, while videos are captured at a fixed frame rate. Although Patra et al. (2024) introduced a forecast-based AD approach for CSP plants, it lacks a reliable selection of AD threshold. Moreover, their study is limited to a single CSP plant, whereas our setting involves multiple plants, introducing additional heterogeneity.

Anomaly detection thresholds. To assign labels, AD methods threshold anomaly scores. Commonly used decision thresholds particularly relevant to our use case involve optimising performance metrics, such as the F1-score (Akcay et al., 2022), G-Mean (Kubat et al., 1997), or the area under the Precision-Recall Curve (PRC), on the validation set over a range of possible thresholds. Another class of methods builds on conformal prediction (CP) (Vovk et al., 2005), a distribution-free framework for constructing prediction sets (e.g., those defined in our decision-making process) providing a finite-sample coverage guarantee. Bates et al. (2023) introduces a method for computing conditionally valid conformal p-values for nonparametric outlier detection, framing the problem within a multiple-hypothesis testing context. A key extension of CP, named *conformal risk control* (Angelopoulos et al., 2024), shifts the guarantee from coverage to managing any monotonically non-increasing risk function. The *Learn then Test* procedure (Angelopoulos et al., 2025) further allows us to extend this concept to any risk function, irrespective of its monotonicity, to generate risk-controlled prediction sets (Bates et al., 2021).

7.9. Conclusion

We introduced a principled and reliable framework for anomaly detection designed to monitor CSP plants using infrared imagery captured at irregular intervals throughout the day. Our approach labels images as normal or anomalous by first assigning an anomaly score using a model trained on an unlabeled image dataset, followed by a thresholding procedure. To address the challenges

of unsupervised AD for CSP plants, our contributions are fourfold. First, we proposed a framework for computing reliable anomaly detection thresholds with finite-sample risk coverage guarantees for any chosen risk function while allowing deferral to domain experts under high uncertainty. Second, to compute anomaly scores for an observed image, we developed a density forecasting method that estimates its likelihood conditional on a sequence of previously observed images. Third, we conducted an extensive real-world deployment analysis over several months across two operational CSP plants, providing valuable insights for industrial maintenance. Lastly, we released a simulated dataset leveraging recent advancements in generative modeling, facilitating data-driven predictive maintenance (PdM) for critical infrastructure. By enhancing the reliability of renewable energy systems, our work supports the broader adoption of sustainable energy solutions for a greener future.

CHAPTER 8

Conclusion

AD plays a crucial role in various high-stakes domains, including industrial quality inspection, healthcare, fraud detection, and predictive maintenance. Despite remarkable progress in deep learning, the deployment of deep AD models in practice remains hindered by several fundamental challenges. This thesis advances the field by making four key contributions: (C1) identifying different types of anomalies, (C2) addressing training data contamination, (C3) handling temporal features of the normal data for a real-world industrial application, and (C4) determining risk-controlling anomaly thresholds with finite-sample performance guarantees. Each contribution is motivated by practical limitations of existing methods and systematically addressed by proposing novel methodologies. Below, we provide a summary of contributions and directions for future research.

8.1. Summary of contributions

First, in real-world applications, different types of anomalies often appear simultaneously. This renders existing methods ineffective as they focus on one kind of anomaly at a time. Traditionally, anomalies have been categorised as point, contextual, or group anomalies. In addition, recent studies have further identified structural anomalies (local irregularities in texture or edges) and logical anomalies (semantic or geometrical violations). In our work, we focus on automated industrial inspection, where structural and logical anomalies often appear simultaneously. To address this, in our first contribution (C1), we develop a unified method for the detection of both structural and logical anomalies, building on the Deep Feature Reconstruction (DFR) approach. Our method achieves competitive performance across multiple benchmark datasets, demonstrating the ability to detect co-occurring anomaly types.

Second, contamination of the training dataset undermines the assumption that training datasets are "clean". In practice, contamination is unavoidable in domains such as industrial monitoring, where unnoticed defects are embedded in training data. For our second contribution (C2), we address this problem in two settings. In semi-supervised AD, we formulate the task as a binary classification problem with partially labelled data and introduce two risk-based estimators: a shallow method with a regularised unbiased risk estimator, and a deep method employing a nonnegative risk estimator. Both methods are supported by theoretical bounds on estimation error and excess risk. In the fully unsupervised setting, we develop a test-time adaptation framework that dynamically adjusts model predictions using exponential tilting without requir-

ing additional labelled data. Together, these approaches improve robustness against contamination at both training.

Third, in the context of the John Cockerill use case for CSP plants, the data exhibit several complex temporal features that pose significant challenges for AD. These include non-stationarity, strong daily seasonal patterns, irregular sampling intervals, and temporal dependencies across observations. Accurately capturing these temporal features is essential for AD. Thus, as our third contribution (C3), we propose a forecasting-based AD framework. Here, a deep sequence model is used to predict the following image under the assumption of normal behaviour. Then, anomalies are identified as deviations between predicted and observed thermal images. This approach enables detecting anomalous behaviours of the CSP plant by extracting meaningful representations from thermal images while modelling the temporal features of the data.

Lastly, although deep learning-based AD methods can learning rich representations from high-dimensional data without manual feature engineering, they often yield unreliable and overly optimistic predictions (Nalisnick et al., 2019a). This issue is particularly problematic in safety-critical applications. While uncertainty quantification has been proposed as a solution, uncertainty scores alone are insufficient to guide decision-making. Existing learning-with-reject framework (Perini and Davis, 2023) partially addresses this issue but fails to provide statistical guarantees on user-defined risk functions. To overcome this limitation, as our fourth contribution (C4), we propose a risk-controlling thresholding strategy for anomaly scores that ensures finite-sample performance guarantees for any chosen risk function, including false positive rates and F1-scores. Our approach builds upon the distribution-free Learn then Test framework (Angelopoulos et al., 2025) and is further enhanced by two adaptive thresholds that account for overlap between normal and anomalous score distributions. We additionally propose a density-forecasting-based AD model using conditional normalising flows and likelihood-based anomaly scoring.

Overall, these contributions not only advance the methodological foundations of deep AD but also strengthen its applicability to safety-critical domains, paving the way for more reliable deployment in real-world systems.

8.2. Future research directions

While our work has explored four key challenges in deep AD, we highlight other promising directions focusing on scalability, interpretability, and leveraging

emerging paradigms such as federated learning and foundation models.

8.2.1 Zero-shot anomaly detection using foundation models

The emergence of foundation models in vision, language, and multimodal domains has created exciting new opportunities for anomaly detection. Their large-scale pre-training on massive and diverse datasets provides rich semantic representations that can, in principle, enable zero-shot detection without task-specific supervision (Jeong et al., 2023; Zhou et al., 2024; Li et al., 2023a). However, significant challenges remain before such models can be reliably integrated into AD pipelines.

Li et al. (2025) provided two key insights regarding the challenges of using zero-shot models. First, pre-trained models inevitably inherit the biases of their training data, which may not align with the domain in which AD is deployed. Anomalies are inherently context-dependent, and features that suffice for one domain may be inadequate in another. Furthermore, on examining the use of predictive uncertainty or feature representations from supervised models for AD, it is concluded that these approaches often mis-specify the problem itself. For example, a cat–dog classifier may confidently misclassify a medical pill in an industrial AD setting, simply because the pill shares certain discriminative features with cats or dogs, despite being semantically unrelated. This illustrates a fundamental misalignment between the objectives of classification and those of anomaly detection. While uncertainty-based methods are found to frequently conflate high predictive uncertainty with being anomalous, feature-based methods incorrectly equate large feature-space distances with anomalous behaviour. Interventions such as hybrid feature–logit methods, scaling model or data size, modelling epistemic uncertainty, or leveraging outlier exposure have been explored. However, it was found that they do not fully resolve the underlying misalignment in objectives. Consequently, caution is warranted when applying large-scale pre-trained models to AD, and future approaches must explicitly address the core requirements of the anomaly detection problem rather than relying on proxy signals.

In addition, the substantial computational demands of foundation models present a barrier to deployment in resource-constrained environments, such as embedded monitoring systems or edge devices. Thus, a deeper theoretical understanding of when and why pre-training benefits AD is crucial. Such insights would not only clarify the limits of current foundation-model-based approaches but also inform the design of more effective and resource-efficient anomaly detection methods.

8.2.2 Synthetic anomaly generation

A persistent challenge in anomaly detection research and deployment lies in the scarcity of anomalous data. This limitation not only constrains model training but also hampers reliable evaluation in real-world settings. Synthetic data generation has emerged as a promising avenue to mitigate this issue (Murase and Fukumizu, 2022; Azizmalayeri et al., 2022). Approaches range from physics-based simulations to modern generative techniques, such as Generative Adversarial Networks (GANs) and diffusion models, which can enrich training datasets with artificially constructed anomalies. Beyond augmenting data availability, synthetic anomalies also offer the potential to serve as controlled testbeds for systematically evaluating the robustness of AD methods across diverse and repeatable scenarios.

However, a key unresolved question concerns the fidelity and representativeness of generated anomalies. Artificially simplistic or biased samples risk misleading both model development and evaluation, thereby limiting their practical utility. Addressing this concern, Perini et al. (2025) introduced the *Expected Anomaly Posterior* (EAP), an example-wise score function designed to quantify the quality of auxiliary anomalies. Their framework emphasises two essential criteria for high-quality synthetic anomalies: (i) they must be distinguishable from the in-distribution training data, and (ii) they must remain realistic with respect to anomalies encountered during inference. Striking a balance between these criteria is inherently challenging. Violating the first diminishes AD performance by reducing discriminative power, while violating the second undermines the practical value of the generated anomalies.

Future research should therefore prioritise the development of controllable and domain-informed generative processes, potentially integrating expert knowledge or interactive guidance to better align synthetic samples with realistic failure modes. A promising step in this direction is illustrated by Li and Zhang (2025), who proposed the Hamiltonian Monte Carlo Outlier Synthesis (HamOS) framework. By formulating synthetic generation as a Markov chain sampling process over the in-distribution data, HamOS enables extensive exploration of the feature space. Thus, it facilitates the creation of diverse and plausible anomalous scenarios. Extending such approaches, particularly by embedding domain priors or uncertainty-awareness into the generative process, represents a compelling future direction for advancing both the realism and utility of synthetic anomalies in AD research.

8.2.3 Conformal risk control for anomaly detection

The integration of conformal prediction into anomaly detection is an emerging direction that offers finite-sample statistical guarantees on prediction sets (Bates et al., 2023). In Chapter 7, we developed a risk-controlling thresholding strategy for anomaly scores, ensuring finite-sample guarantees across arbitrary risk measures. Our analysis assumed the availability of a labelled calibration dataset. However, in practical AD applications, collecting additional labelled samples is often costly or infeasible.

In contrast, conformal outlier detection methods (Angelopoulos and Bates, 2021) typically assume access to unlabelled calibration sets composed exclusively of normal instances. This assumption is fragile, as calibration sets are frequently contaminated with anomalous points, which may distort empirical quantiles and compromise statistical validity. Recent work by Bashari et al. (2025) has shown that, under realistic non-adversarial contamination, conformal calibration can still maintain conservative type-I error control, thereby demonstrating robustness. Nonetheless, such conservativeness comes at the expense of reduced power. Parallel investigations have examined the effects of noisy or weak labels on the robustness of conformal prediction (Einbinder et al., 2024; Sesia et al., 2024; Caprio et al., 2025). Building on these insights, an important avenue for future research is to assess the effectiveness of conformal risk control under contaminated or noisy calibration samples, with a particular focus on AD settings.

Another promising extension lies in cross-conformal prediction (Vovk, 2015; Hennhöfer and Preisach, 2024; Gasparin and Ramdas, 2025), which partitions calibration data into multiple folds. This strategy offers the potential for improved stability and reliability of guarantees, particularly in small-sample regimes, which is a scenario common in AD tasks. Extending cross-conformal approaches to control risk in AD explicitly is, therefore, a natural and practically valuable next step.

Taken together, these directions suggest the need for hybrid frameworks that address both contamination and data scarcity. Such methods would yield uncertainty-aware anomaly thresholds that remain valid and powerful under realistic conditions, thereby strengthening the practical utility of conformal risk control in anomaly detection.

8.2.4 Federated anomaly detection

A significant avenue for future research is the development of federated approaches to AD (Khan et al., 2021; Zhang et al., 2021; Kairouz et al., 2021b). In sensitive domains such as healthcare, finance, and industrial monitoring, centralising data is often impractical due to privacy concerns, regulatory restrictions, or the scale of data involved. For example, in the context of John Cockerill CSP plants, centralising data from multiple CSP plants situated across the globe poses a significant challenge in terms of storage and privacy. Federated learning provides a natural paradigm to address these constraints by enabling decentralised training while sharing only model updates.

Despite its promise, federated AD introduces unique challenges. Anomalies may arise only at a subset of clients, leading to highly imbalanced and non-IID data distributions. While federated learning methods are typically designed to mitigate statistical heterogeneity to accelerate convergence and improve aggregate performance, AD requires a shift of focus towards improving the detection of minority-class instances. Moreover, the presence of adversarial participants poses further risks, as malicious clients may deliberately introduce anomalous patterns to bias global aggregation.

Addressing these challenges calls for several research directions. Robust aggregation mechanisms (Al-Sayed et al., 2017; Chen et al., 2019) are needed to ensure resilience against adversarial manipulation, while adaptive personalisation strategies (Tan et al., 2022) can tailor anomaly detectors to client-specific data distributions. Furthermore, communication-efficient training protocols (Sattler et al., 2019) will be essential to reduce overhead in large-scale deployments. Privacy-preserving techniques such as differential privacy (Dwork et al., 2006) and secure aggregation (Kairouz et al., 2021a) will also play a key role in ensuring compliance with data-protection regulations.

Exploring these avenues will be crucial for realising anomaly detection systems that are not only accurate and robust but also privacy-preserving and practically deployable in federated environments.

Bibliography

Abdalla, M., Javed, S., Radi, M. A., Ulhaq, A., and Werghi, N. (2024). Video anomaly detection in 10 years: A survey and outlook.

Abdallah, A., Maarof, M. A., and Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113.

Adler, A., Elad, M., Hel-Or, Y., and Rivlin, E. (2013). Sparse coding with anomaly detection. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In Van den Bussche, J. and Vianu, V., editors, *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ahmed, F. and Courville, A. (2020). Detecting semantic anomalies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3154–3162.

Ahmed, M., Mahmood, A. N., and Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288.

Akcay, S., Ameln, D., Vaidya, A., Lakshmanan, B., Ahuja, N., and Genc, U. (2022). Anomalib: A Deep Learning Library for Anomaly Detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1706–1710. IEEE.

Al-Sayed, S., Zoubir, A. M., and Sayed, A. H. (2017). Robust distributed estimation by networked agents. *IEEE Transactions on Signal Processing*, 65(15):3909–3921.

Amruthnath, N. and Gupta, T. (2018). A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In *5th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 355–361.

Angelopoulos, A. N. and Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.

Angelopoulos, A. N., Bates, S., Candès, E. J., Jordan, M. I., and Lei, L. (2025). Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662.

Angelopoulos, A. N., Bates, S., et al. (2023). Conformal prediction: A gentle introduction. *Foundations and trends® in machine learning*, 16(4):494–591.

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. (2024). Conformal risk control. In *The Twelfth International Conference on Learning Representations*.

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. (2023). Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629.

Azizmalayeri, M., Soltani Moakhar, A., Zarei, A., Zohrabi, R., Manzuri, M., and Rohban, M. H. (2022). Your out-of-distribution detection method is not robust! *Advances in Neural Information Processing Systems*, 35:4887–4901.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

Bashari, M., Sesia, M., and Romano, Y. (2025). Robust conformal outlier detection under contaminated reference data. In *Forty-second International Conference on Machine Learning*.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., and Jordan, M. (2021).

Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6).

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. (2023). Testing for outliers with conformal p-values. *The Annals of Statistics*, 51(1):149 – 178.

Batzner, K., Heckler, L., and König, R. (2024). Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 128–138.

Bekker, J. and Davis, J. (2020). Learning from positive and unlabeled data: A survey. *Machine Learning*, 109:719–760.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., and Steger, C. (2022). Beyond Dents and Scratches: Logical Constraints in Unsupervised Anomaly Detection and Localization. *International Journal of Computer Vision*, 130(4):947–969.

Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. (2019). MVTEC ad-A comprehensive real-world dataset for unsupervised anomaly detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9584–9592.

Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. (2020). Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 813–824. PMLR.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Blanchard, G., Lee, G., and Scott, C. (2010). Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11(99):2973–3009.

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.

Bouman, R. and Heskes, T. (2025). Autoencoders for anomaly detection are unreliable.

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.

Bruckstein, A. M., Donoho, D. L., and Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81.

Cao, Y., Xu, X., Zhang, J., Cheng, Y., Huang, X., Pang, G., and Shen, W. (2024). A survey on visual anomaly detection: Challenge, approach, and prospect. *arXiv preprint arXiv:2401.16402*.

Caprio, M., Stutz, D., Li, S., and Doucet, A. (2025). Conformalized credal regions for classification with ambiguous ground truth. *Transactions on Machine Learning Research*.

Chalapathy, R. and Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.

Chalapathy, R., Menon, A. K., and Chawla, S. (2019). Anomaly detection using one-class neural networks.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3).

Chandrakala, S., Deepak, K., and Revathy, G. (2023). Anomaly detection in surveillance videos: a thematic taxonomy of deep models, review and performance analysis. *Artificial Intelligence Review*, 56(4):3319–3368.

Chaudhari, S. and Shevade, S. (2012). Learning from positive and unlabelled examples using maximum margin clustering. In *Proceedings of the 19th International Conference on Neural Information Processing - Volume Part III*, ICONIP'12, page 465–473, Berlin, Heidelberg. Springer-Verlag.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple frame-

work 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.

Chen, Y., Kar, S., and Moura, J. M. (2019). Resilient distributed parameter estimation with heterogeneous data. *IEEE Transactions on Signal Processing*, 67(19):4918–4933.

Chen, Y.-C., Genovese, C. R., and Wasserman, L. (2017). Density level sets: Asymptotics, inference, and visualization. *Journal of the American Statistical Association*, 112(520):1684–1696.

Choi, H., Kim, D., Kim, J., Kim, J., and Kang, P. (2022). Explainable anomaly detection framework for predictive maintenance in manufacturing systems. *Applied Soft Computing*, 125:109147.

Das, A. S., Pang, G., and Bhuyan, M. (2025). Adaptive deviation learning for visual anomaly detection with data contamination. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, page 8863–8872. IEEE.

Dastidar, K. G., Caelen, O., and Granitzer, M. (2024). Machine learning methods for credit card fraud detection: A survey. *IEEE Access*.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

Deepak, K., Chandrakala, S., and Mohan, C. K. (2021). Residual spatiotemporal autoencoder for unsupervised video anomaly detection. *Signal, Image and Video Processing*, 15(1):215–222.

Defard, T., Setkov, A., Loesch, A., and Audigier, R. (2021). Padim: A patch distribution modeling framework for anomaly detection and localization. In Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G. M., Mei, T., Bertini, M., Escalante, H. J., and Vezzani, R., editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 475–489, Cham. Springer International Publishing.

Deng, H. and Li, X. (2022). Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9737–9746.

Dheur, V., Fontana, M., Estievenart, Y., Desobry, N., and Taieb, S. B. (2025). A unified comparative study with generalized conformity scores for multi-output conformal regression. In *Forty-second International Conference on Machine Learning*.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: Spectral clustering and normalized cuts. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 551–556, New York, NY, USA. Association for Computing Machinery.

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

Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

du Plessis, M. C., Niu, G., and Sugiyama, M. (2014). Analysis of learning from positive and unlabeled data. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

du Plessis, M. C., Niu, G., and Sugiyama, M. (2015). Convex formulation for learning from positive and unlabeled data. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1386–1394, Lille, France. PMLR.

du Plessis, M. C. and Sugiyama, M. (2014). Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Eduardo, S., Nazábal, A., Williams, C. K. I., and Sutton, C. (2020). Robust variational autoencoders for outlier detection and repair of mixed-type data.

In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4056–4066. PMLR.

Einbinder, B.-S., Feldman, S., Bates, S., Angelopoulos, A. N., Gendler, A., and Romano, Y. (2024). Label noise robustness of conformal prediction. *Journal of Machine Learning Research*, 25(328):1–66.

Elkan, C. and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 213–220, New York, NY, USA. Association for Computing Machinery.

Enguehard, J., Busbridge, D., Bozson, A., Woodcock, C., and Hammerla, N. (2020). Neural temporal point processes for modelling electronic health records. In Alsentzer, E., McDermott, M. B. A., Falck, F., Sarkar, S. K., Roy, S., and Hyland, S. L., editors, *Proceedings of the Machine Learning for Health NeurIPS Workshop*, volume 136 of *Proceedings of Machine Learning Research*, pages 85–113. PMLR.

Erfani, S. M., Rajasegarar, S., Karunasekera, S., and Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58:121–134.

Estievenart, Y., Patra, S., and Ben Taieb, S. (2026). Risk-based thresholding for reliable anomaly detection in concentrated solar power plants. In Dutra, I., Pechenizkiy, M., Cortez, P., Pashami, S., Pasquali, A., Moniz, N., Jorge, A. M., Soares, C., Abreu, P. H., and Gama, J., editors, *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track and Demo Track*, pages 111–128, Cham. Springer Nature Switzerland.

Faust, K., Xie, Q., Han, D., Goyle, K., Volynskaya, Z., Djuric, U., and Diamandis, P. (2018). Visualizing histopathologic deep learning classification and anomaly detection using nonlinear feature space dimensionality reduction. *BMC bioinformatics*, 19:1–15.

Fernando, T., Gammulle, H., Denman, S., Sridharan, S., and Fookes, C. (2021). Deep learning for medical anomaly detection – a survey. *ACM Comput. Surv.*, 54(7).

Flach, P. and Kull, M. (2015). Precision-recall-gain curves: Pr analysis done right. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett,

R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Fontana, M., Zeni, G., and Vantini, S. (2023). Conformal prediction: a unified review of theory and new challenges. *Bernoulli*, 29(1):1–23.

Gasparin, M. and Ramdas, A. (2025). Improving the statistical efficiency of cross-conformal prediction. In *Forty-second International Conference on Machine Learning*.

Ghasemi, A., Rabiee, H. R., Manzuri, M. T., and Rohban, M. H. (2012). A Bayesian Approach to the Data Description Problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):907–913.

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., and Hengel, A. v. d. (2019). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Görnitz, N., Kloft, M., and Brefeld, U. (2009). Active and semi-supervised data domain description. In Buntine, W., Grobelnik, M., Mladenić, D., and Shawe-Taylor, J., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 407–422, Berlin, Heidelberg. Springer Berlin Heidelberg.

Görnitz, N., Kloft, M., Rieck, K., and Brefeld, U. (2013). Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262.

Gornitz, N., Lima, L. A., Muller, K. R., Kloft, M., and Nakajima, S. (2018). Support Vector Data Descriptions and k-Means Clustering: One Class? *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):3994–4006.

Goyal, S., Raghunathan, A., Jain, M., Simhadri, H. V., and Jain, P. (2020). DROCC: Deep robust one-class classification. 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 3711–3721. PMLR.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. (2020). Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*.

Gudovskiy, D., Ishizaka, S., and Kozuka, K. (2022). Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 98–107.

Han, S., Hu, X., Huang, H., Jiang, M., and Zhao, Y. (2022). Adbench: Anomaly detection benchmark. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 32142–32159. Curran Associates, Inc.

Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., and Davis, L. S. (2016). Learning temporal regularity in video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hastie, T., Tibshirani, R., Friedman, J., et al. (2009). The elements of statistical learning.

He, F., Liu, T., Webb, G. I., and Tao, D. (2020). Instance-dependent pu learning by bayesian optimal relabeling.

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

Hennhöfer, O. and Preisach, C. (2024). Leave-one-out-, bootstrap-and cross-conformal anomaly detectors. In *2024 IEEE International Conference on Knowledge Graph (ICKG)*, pages 110–119. IEEE.

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*.

Hilal, W., Gadsden, S. A., and Yawney, J. (2022). Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances. *Expert Systems with Applications*, 193:116429.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. (2022). Equivariant diffusion for molecule generation in 3D. 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 8867–8887. PMLR.

Hsieh, Y.-G., Niu, G., and Sugiyama, M. (2019). Classification from positive, unlabeled and biased negative data. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2820–2829. PMLR.

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.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning: with applications in R*, volume 103. Springer.

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. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15433–15445. Curran Associates, Inc.

Ju, H., Lee, D., Hwang, J., Namkung, J., and Yu, H. (2020). Pumad: Pu metric learning for anomaly detection. *Information Sciences*, 523:167–183.

Kairouz, P., Liu, Z., and Steinke, T. (2021a). The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pages 5201–5212. PMLR.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2021b). Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210.

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 (CVPR)*, pages 14162–14171.

Khan, L. U., Saad, W., Han, Z., Hossain, E., and Hong, C. S. (2021). Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 23(3):1759–1799.

Khan, S. S. and Madden, M. G. (2014). One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374.

Kind, A., Stoecklin, M. P., and Dimitropoulos, X. (2009). Histogram-based traffic anomaly detection. *IEEE Transactions on Network and Service Management*, 6(2):110–121.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Kiran, B. R., Thomas, D. M., and Parakkal, R. (2018). An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2).

Kirichenko, P., Izmailov, P., and Wilson, A. G. (2020). Why normalizing flows fail to detect out-of-distribution data. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20578–20589. Curran Associates, Inc.

Kiryo, R., Niu, G., du Plessis, M. D., and Sugiyama, M. (2017). Positive-unlabeled learning with non-negative risk estimator. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

Kubat, M., Matwin, S., et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of the 14th International Conference on Machine Learning*, volume 97, pages 179–186.

Lai, C.-H., Zou, D., and Lerman, G. (2020). Robust subspace recovery layer for unsupervised anomaly detection. In *International Conference on Learning Representations*.

Le, V.-T. and Kim, Y.-G. (2023). Attention-based residual autoencoder for video anomaly detection. *Applied Intelligence*, 53(3):3240–3254.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., et al. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces: Isoperimetry and Processes*. Springer Berlin Heidelberg, Berlin, Heidelberg.

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.

Lee, W. S. and Liu, B. (2003). Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 448–455. AAAI Press.

Leys, C., Klein, O., Dominicy, Y., and Ley, C. (2018). Detecting multivari-

ate outliers: Use a robust variant of the mahalanobis distance. *Journal of Experimental Social Psychology*, 74:150–156.

Li, A., Miao, Z., Cen, Y., and Cen, Y. (2017). Anomaly detection using sparse reconstruction in crowded scenes. *Multimedia Tools Applications*, 76(24):26249–26271.

Li, A., Qiu, C., Kloft, M., Smyth, P., Rudolph, M., and Mandt, S. (2023a). Zero-shot anomaly detection via batch normalization. *Advances in Neural Information Processing Systems*, 36:40963–40993.

Li, H. and Zhang, T. (2025). Outlier synthesis via hamiltonian monte carlo for out-of-distribution detection. In *The Thirteenth International Conference on Learning Representations*.

Li, X. and Liu, B. (2003). Learning to classify texts using positive and unlabeled data. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03): 2003; Acapulco, Mexico*, pages 587–594.

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, Y. L., Lu, D., Kirichenko, P., Qiu, S., Rudner, T. G. J., Bruss, C. B., and Wilson, A. G. (2025). Position: Supervised classifiers answer the wrong questions for OOD detection. In *Forty-second International Conference on Machine Learning Position Paper Track*.

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)*, pages 1118–1123. 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, B., Dai, Y., Li, X., Lee, W., and Yu, P. (2003). Building text classifiers using positive and unlabeled examples. In *Third IEEE International Conference on Data Mining*, pages 179–186.

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.

Lu, Y. and Huang, B. (2020). Structured output learning with conditional generative flows. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5005–5012.

McDiarmid, C. (1989). *On the method of bounded differences*, page 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press.

McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction.

Menon, A., Rooyen, B. V., Ong, C. S., and Williamson, B. (2015). Learning from corrupted binary labels via class-probability estimation. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 125–134, Lille, France. PMLR.

Mishra, P., Verk, R., Fornasier, D., Piciarelli, C., and Foresti, G. L. (2021). Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2 edition.

Moore, A. and Morelli, D. (2024). Condense: Conditional density estimation for time series anomaly detection. *Journal of Artificial Intelligence Research*, 79:801–824.

Muandet, K. and Schölkopf, B. (2013). One-class support measure machines for group anomaly detection. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 449–458, Arlington, Virginia, USA. AUAI Press.

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 Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp,

F., editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 36486–36503. PMLR.

Munoz-Mari, J., Bovolo, F., Gomez-Chova, L., Bruzzone, L., and Camp-Valls, G. (2010). Semisupervised one-class support vector machines for classification of remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(8):3188–3197.

Murase, H. and Fukumizu, K. (2022). Algan: Anomaly detection by generating pseudo anomalous data via latent variables. *Ieee Access*, 10:44259–44270.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2019a). Do deep generative models know what they don't know? In *International Conference on Learning Representations*.

Nalisnick, E., Matsukawa, A., Teh, Y. W., and Lakshminarayanan, B. (2019b). Detecting Out-of-Distribution Inputs to Deep Generative Models Using Typicality. In *4th Workshop on Bayesian Deep Learning (NeurIPS 2019)*.

Ngiam, J., Chen, Z., Koh, P. W., and Ng, A. Y. (2011). Learning deep energy models. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1105–1112.

Nguyen, D. T., Lou, Z., Klar, M., and Brox, T. (2019). Anomaly Detection With Multiple-Hypotheses Predictions. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4800–4809. PMLR.

Niu, G., du Plessis, M. C., Sakai, T., Ma, Y., and Sugiyama, M. (2016). Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Novello, P., Dalmau, J., and Andéol, L. (2025). Exploring the link between out-of-distribution detection and conformal prediction with illustrations of its benefits.

Oza, P. and Patel, V. M. (2019). One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281.

Pang, G., Shen, C., Cao, L., and Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2).

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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 Taieb, S. B. (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. (2024). *Operational, Uncertainty-Aware, and Reliable Anomaly Detection*. PhD thesis, KU Leuven.

Perini, L., Bürkner, P.-C., and Klami, A. (2023). Estimating the contamination factor's distribution in unsupervised anomaly detection. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27668–27679. PMLR.

Perini, L. and Davis, J. (2023). Unsupervised anomaly detection with rejection. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 69673–69691. Curran Associates, Inc.

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. (2021). Quantifying the confidence of anomaly detectors in their example-wise predictions. In Hutter, F., Kersting, K., Lijffijt, J., and Valera, I., editors, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 227–243, Cham. Springer International Publishing.

Perini, L., Vercruyssen, V., and Davis, J. (2022). Transferring the Contamination Factor between Anomaly Detection Domains by Shape Similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):4128–4136.

Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99:215–249.

Press, O., Shwartz-Ziv, R., Lecun, Y., and Bethge, M. (2024). The entropy enigma: Success and failure of entropy minimization. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F., editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 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 Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 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., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Reiss, T., Cohen, N., Bergman, L., and Hoshen, Y. (2021). Panda: Adapting

pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2806–2814.

Reiss, T., Cohen, N., Horwitz, E., Abutbul, R., and Hoshen, Y. (2022). Anomaly detection requires better representations. In *Computer Vision – ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, page 56–68, Berlin, Heidelberg. Springer-Verlag.

Rezende, D. and Mohamed, S. (2015a). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Rezende, D. and Mohamed, S. (2015b). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., and Gehler, P. (2022). Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14318–14328.

Roth, V. (2004). Outlier detection with one-class kernel fisher discriminants. In Saul, L., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Roth, V. (2006). Kernel fisher discriminants for outlier detection. *Neural Computation*, 18(4):942–960.

Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. *Mathematical statistics and applications*, 8(283-297):37.

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

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*.

Ruff, L., Zemlyanskiy, Y., Vandermeulen, R., Schnake, T., and Kloft, M. (2019). Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4061–4071, Florence, Italy. Association for Computational Linguistics.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252.

Sabokrou, M., Khalooei, M., Fathy, M., and Adeli, E. (2018). Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Saerens, M., Latinne, P., and Decaestecker, C. (2002). Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural computation*, 14:21–41.

Sakai, T., du Plessis, M. C., Niu, G., and Sugiyama, M. (2017). Semi-supervised classification based on classification from positive and unlabeled data. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2998–3006. PMLR.

Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M. H., and Rabiee, H. R. (2021). Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14902–14912.

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. (2019). Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413.

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

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

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.

Scott, C. (2015). A Rate of Convergence for Mixture Proportion Estimation, with Application to Learning from Noisy Labels. In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 838–846, San Diego, California, USA. PMLR.

Sesia, M., Wang, Y. R., and Tong, X. (2024). Adaptive conformal classification with noisy labels. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, page qkae114.

Shi, Y., Yang, J., and Qi, Z. (2021). Unsupervised anomaly segmentation via deep feature reconstruction. *Neurocomputing*, 424:9–22.

Sobel, I. (1968). An isotropic 3× 3 image gradient operator, presentation at stanford artificial intelligence project (sail).

Steinwart, I. (2011). Adaptive density level set clustering. In *Proceedings of the 24th annual conference on learning theory*, pages 703–738. JMLR Workshop and Conference Proceedings.

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

Sudhakaran, S. and Lanz, O. (2017). Learning to detect violent videos using convolutional long short-term memory. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.

Tan, A. Z., Yu, H., Cui, L., and Yang, Q. (2022). Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603.

Tang, W., Yang, Q., Xiong, K., and Yan, W. (2020). Deep learning based automatic defect identification of photovoltaic module using electroluminescence images. *Solar Energy*, 201:453–460.

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.

Tibshirani, R. and Hastie, T. (2007). Outlier sums for differential gene expression analysis. *Biostatistics*, 8(1):2–8.

Tung, F. and Mori, G. (2019). Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Van Ryzin, J. (1973). A histogram method of density estimation. *Communications in Statistics*, 2(6):493–506.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vovk, V. (2015). Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74:9–28.

Vovk, V., Gammerman, A., and Saunders, C. (1999). Machine-learning applications of algorithmic randomness. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 444–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world*. Springer.

Wang, C., Jiang, X., Gao, B.-B., Gan, Z., Liu, Y., Zheng, F., and Ma, L. (2025). Softpatch+: Fully unsupervised anomaly classification and segmentation. *Pattern Recognition*, 161:111295.

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 (CVPR)*, 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. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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

Xie, M., Hu, J., and Tian, B. (2012). Histogram-based online anomaly detection in hierarchical wireless sensor networks. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 751–759.

Yan, X., Zhang, H., Xu, X., Hu, X., and Heng, P.-A. (2021). Learning semantic context from normal samples for unsupervised anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3110–3118.

Yang, J., Zhou, K., Li, Y., and Liu, Z. (2024). Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662.

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*.

Yoon, J., Sohn, K., Li, C.-L., Arik, S. O., and Pfister, T. (2023). SPADE: Semi-supervised anomaly detection under distribution mismatch. *Transactions on Machine Learning Research*. Featured Certification.

You, Z., Cui, L., Shen, Y., Yang, K., Lu, X., Zheng, Y., and Le, X. (2022). A unified model for multi-class anomaly detection. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in*

Neural Information Processing Systems, volume 35, pages 4571–4584. Curran Associates, Inc.

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. *arXiv preprint arXiv:2111.07677*.

Zagoruyko, S. and Komodakis, N. (2016). Wide Residual Networks. *British Machine Vision Conference 2016, BMVC 2016*, 2016-September:1–87.

Zagoruyko, S. and Komodakis, N. (2017). Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*.

Zavrtanik, V., Kristan, M., and Skočaj, D. (2021). Dræm - a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8330–8339.

Zenati, H., Foo, C. S., Lecouat, B., Manek, G., and Chandrasekhar, V. R. (2018). Efficient GAN-Based Anomaly Detection. *arXiv preprint*.

Zhai, S., Cheng, Y., Lu, W., and Zhang, Z. (2016). Deep structured energy based models for anomaly detection. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1100–1109, New York, New York, USA. PMLR.

Zhang, B. and Zuo, W. (2009). Reliable negative extracting based on kNN for learning from positive and unlabeled examples. *Journal of Computers*, 4(1):94–101.

Zhang, H. L., Baeyens, J., Degrève, J., and Cacères, G. (2013). Concentrated solar power plants: Review and design methodology. *Renewable and Sustainable Energy Reviews*, 22:466–481.

Zhang, J., Suganuma, M., and Okatani, T. (2024). Contextual affinity distillation for image anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 149–158.

Zhang, T., He, C., Ma, T., Gao, L., Ma, M., and Avestimehr, S. (2021). Federated learning for internet of things. In *Proceedings of the 19th ACM conference on embedded networked sensor systems*, pages 413–419.

Zhang, Y., Wang, X., Jin, K., Yuan, K., Zhang, Z., Wang, L., Jin, R., and Tan, T. (2023). AdaNPC: Exploring non-parametric classifier for test-time adaptation. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41647–41676. PMLR.

Zhou, J. and Wu, Y. (2024). Outlier-probability-based feature adaptation for robust unsupervised anomaly detection on contaminated training data. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(10):10023–10035.

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*.

Zimek, A., Schubert, E., and Kröger, P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5:363–387.

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 Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T., editors, *European Conference on Computer Vision*, pages 392–408, Cham. Springer Nature Switzerland.

Appendices

APPENDIX A

Detecting Logical and Structural Anomalies

A.1. Implementation details

ULSAD is implemented in PyTorch (Paszke et al., 2019). Specifically, we used the Anomalib (Akcay et al., 2022) library by incorporating our code within it. It helps us have a fair comparison as we use the implementations of baselines from Anomalib. Moreover, we used a single NVIDIA A4000 GPU for all the experiments unless mentioned otherwise. The architecture of the global autoencoder-like model and FRN is provided in Table A.1 and A.2, respectively.

Table A.1: Global Autoencoder of ULSAD.

	Layer Name	Stride	Kernel Size	Number of Kernels	Padding	Activation
Encoder	Conv-1	2	4×4	32	1	ReLU
	BatchNorm-1	-	-	-	-	-
	Conv-2	2	4×4	32	1	ReLU
	BatchNorm-2	-	-	-	-	-
	Conv-3	2	4×4	64	1	ReLU
	BatchNorm-3	-	-	-	-	-
	Conv-4	2	4×4	64	1	ReLU
	BatchNorm-4	-	-	-	-	-
	Conv-5	2	4×4	64	1	ReLU
	BatchNorm-5	-	-	-	-	-
	Conv-6	1	8×8	64	1	ReLU
	BatchNorm-6	-	-	-	-	-
Decoder	Interpolate-1 (31, mode= "bilinear")	-	-	-	-	-
	Conv-1	1	4×4	64	2	ReLU
	BatchNorm-1	-	-	-	-	-
	Interpolate-2 (8, mode= "bilinear")	-	-	-	-	-
	Conv-2	1	4×4	64	2	ReLU
	BatchNorm-2	-	-	-	-	-
	Interpolate-3 (16, mode= "bilinear")	-	-	-	-	-
	Conv-3	1	4×4	64	2	ReLU
	BatchNorm-3	-	-	-	-	-
	Interpolate-4 (32, mode= "bilinear")	-	-	-	-	-
	Conv-4	1	4×4	64	2	ReLU
	BatchNorm-4	-	-	-	-	-
	Interpolate-5 (64, mode= "bilinear")	-	-	-	-	-
	Conv-5	1	4×4	64	2	ReLU
	BatchNorm-5	-	-	-	-	-
	Interpolate-6 (32, mode= "bilinear")	-	-	-	-	-
	Conv-6	1	3×3	64	1	ReLU
	BatchNorm-6	-	-	-	-	-
	Conv-7	1	3×3	384	1	ReLU

Table A.2: Feature Reconstruction Network of `ULSAD`.

	Layer Name	**Stride**	**Kernel Size**	**Number of Kernels**	**Padding**	**Activation**
Encoder	Conv-1	2	3×3	768	1	ReLU
	BatchNorm-1	-	-	-	-	-
	Conv-2	2	3×3	1536	1	ReLU
	BatchNorm-2	-	-	-	-	-
	Conv-3	1	3×3	1536	1	ReLU
	BatchNorm-3	-	-	-	-	-
Decoder	ConvTranspose-1	2	4×4	768	1	ReLU
	BatchNorm-4	-	-	-	-	-
	ConvTranspose-2	2	4×4	384	1	ReLU
	BatchNorm-5	-	-	-	-	-
	ConvTranspose-3	1	5×5	384	1	ReLU
	BatchNorm-6	-	-	-	-	-

A.2. Extended results

Extended versions of the Table 3.1 and 3.2 are provided in Tables A.3-A.17. It shows the performance of ULSAD per category of the benchmark datasets for anomaly detection and localisation. Additionally, we provide a visual comparison of the generated anomaly maps using the MVTecLOCO dataset in Figure A.1.

Table A.3: Anomaly detection based on Image AUROC on MVTec dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
bottle	**100.0**	94.6	98.57	99.37	**100.0**	98.10	94.92	**100.0**	**100.0** ± 0.00
cable	93.46	74.29	89.30	86.94	**98.54**	95.41	79.54	94.61	97.92 ± 0.18
capsule	91.74	65.46	86.04	88.43	**97.93**	81.01	96.01	95.57	94.61 ± 0.28
carpet	93.26	57.70	98.76	97.31	97.91	57.95	97.59	**99.70**	98.50 ± 0.17
grid	93.57	76.02	99.08	84.04	97.24	93.07	94.57	**100.0**	92.67 ± 1.20
hazelnut	**100.0**	84.43	81.57	86.07	**100.0**	99.82	**100.0**	93.39	99.93 ± 0.05
leather	99.97	79.31	**100.0**	99.66	**100.0**	42.39	99.46	99.92	**100.0** ± 0.00
metal_nut	**99.76**	45.06	94.53	96.92	99.61	67.20	93.06	99.34	98.88 ± 0.07
pill	90.73	44.65	87.53	88.52	94.35	54.66	92.06	**99.08**	96.17 ± 0.39
screw	88.30	89.38	66.00	75.24	**98.26**	94.57	93.69	98.09	95.20 ± 0.15
tile	**100.0**	90.15	95.42	95.49	98.67	97.37	92.97	99.85	99.99 ± 0.02
toothbrush	83.33	80.28	79.44	93.61	**100.0**	84.72	**100.0**	**100.0**	**100.0** ± 0.00
transistor	91.50	88.37	94.42	92.29	**100.0**	83.29	80.54	96.57	97.65 ± 0.57
wood	98.33	90.96	97.54	98.33	**99.30**	53.16	98.77	98.13	98.81 ± 0.23
zipper	93.07	68.17	92.54	86.48	**99.47**	92.04	89.97	99.22	94.36 ± 0.13
Mean	94.47	75.26	90.72	91.25	**98.75**	79.65	93.54	98.23	97.65 ± 0.38

A.2.1 Performance on MVTecLOCO: logical and structural AD

In Tables A.18, A.19, A.20, A.21, A.22, and A.23, we report the anomaly detection and localisation results on MVTec LOCO separately for structural and logical anomalies. It can be observed that although PatchCore performs slightly better than ULSAD on structural anomalies, ULSAD delivers competitive results on logical anomalies. Moreover, as discussed previously in Section 3.4, the improvement in performance with PatchCore comes with three times the memory requirement. Therefore, we consider ULSAD to be an efficient and effective approach for the detection and localisation of both logical and structural anomalies.

Table A.4: Anomaly segmentation performance based on Pixel AUROC on MVTec dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
bottle	**98.58**	76.53	97.8	98.3	97.98	98.31	90.83	98.31	96.21 ± 2.21
cable	96.1	66.59	95.71	96.81	98.03	96.37	91.37	**98.5**	97.71 ± 0.06
capsule	98.71	86.96	98.37	98.67	98.77	**98.96**	98.46	98.33	98.95 ± 0.03
carpet	98.57	71.95	98.27	98.68	98.67	99.05	98.46	94.83	**99.18** ± 0.06
grid	97.49	53.56	98.39	92.82	97.86	**99.01**	97.41	96.02	95.47 ± 1.09
hazelnut	98.64	84.66	94.79	97.85	98.43	**98.91**	98.53	96.15	98.81 ± 0.03
leather	99.42	63.32	**99.62**	99.30	98.87	99.17	99.33	97.5	98.68 ± 0.01
metal_nut	97.97	80.25	97.01	96.71	**98.51**	97.68	93.02	98.07	97.62 ± 0.03
pill	97.83	77.17	96.38	95.03	97.53	96.96	96.86	**98.63**	96.67 ± 0.09
screw	97.64	83.38	89.87	97.89	99.19	**99.43**	99.07	98.50	99.33 ± 0.01
tile	**96.68**	85.75	93.14	92.42	94.86	95.47	90.82	91.61	95.78 ± 0.05
toothbrush	98.16	90.70	97.50	98.83	98.67	**98.99**	98.49	96.0	98.42 ± 0.02
transistor	89.91	63.23	96.45	96.85	96.84	86.77	79.11	94.77	**98.89** ± 0.05
wood	94.70	71.73	**95.71**	93.83	93.31	95.06	95.36	90.85	95.20 ± 0.31
zipper	97.08	69.31	97.62	97.82	98.06	**98.54**	96.85	96.21	97.24 ± 0.07
Mean	97.17	75.01	96.44	96.79	**97.71**	97.25	94.93	96.29	97.61 ± 0.64

Table A.5: Anomaly segmentation performance based on Pixel AUPRO on MVTec dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
bottle	94.19	50.05	92.0	95.11	92.28	**95.12**	83.14	93.84	90.16 ± 3.24
cable	85.85	28.58	86.65	89.65	90.77	90.32	83.09	**92.53**	88.63 ± 0.48
capsule	90.47	81.11	90.15	92.62	92.4	93.93	**96.33**	91.09	93.77 ± 0.16
carpet	92.64	48.64	94.63	95.59	92.7	**96.41**	95.47	90.99	96.39 ± 0.24
grid	90.52	17.71	93.95	82.52	89.46	**96.39**	91.15	93.14	83.3 ± 3.96
hazelnut	96.12	76.19	93.92	92.95	94.44	96.92	**97.17**	83.25	94.87 ± 0.3
leather	98.39	52.1	**99.06**	97.91	96.33	97.97	98.34	97.32	97.44 ± 0.01
metal_nut	88.97	35.79	85.89	90.45	91.9	**94.4**	87.01	92.97	91.58 ± 0.19
pill	93.67	64.26	91.0	93.88	93.92	94.76	95.86	**95.93**	94.5 ± 0.08
screw	90.25	53.22	68.6	92.14	95.39	**97.05**	95.96	96.04	96.45 ± 0.1
tile	**91.49**	58.48	81.01	78.32	79.64	88.4	79.36	83.54	87.82 ± 0.15
toothbrush	81.05	54.02	80.62	**93.52**	86.48	92.23	92.93	88.61	86.28 ± 0.46
transistor	78.75	51.37	88.92	89.04	**94.06**	75.05	64.25	82.82	91.61 ± 0.68
wood	90.5	45.29	**93.26**	91.39	85.08	92.69	92.48	76.16	91.34 ± 0.29
zipper	89.3	28.98	92.12	92.48	92.43	**95.18**	88.74	93.48	90.89 ± 0.35
Mean	90.14	49.72	88.79	91.17	91.15	**93.12**	89.42	90.11	91.67 ± 1.36

Table A.6: Anomaly detection performance based on Image AUROC on MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	71.86	70.26	74.04	63.66	**85.24**	52.69	65.46	74.80	83.54 ± 0.23
juice_bottle	81.70	62.55	78.03	88.42	92.51	76.28	86.81	98.89	**97.12** ± 0.10
pushpins	73.43	51.32	61.20	61.30	75.54	50.72	72.68	80.58	**86.85** ± 0.94
screw_bag	65.48	59.39	68.04	60.14	69.90	65.15	63.55	67.42	**70.71** ± 1.49
splicing_connectors	75.63	68.25	73.71	68.40	**84.24**	62.95	75.87	81.39	82.30 ± 0.72
Mean	73.62	62.35	71.00	68.38	81.49	61.56	72.87	80.62	**84.1** ± 0.86

Table A.7: Anomaly segmentation performance based on Pixel AUROC on MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	**89.6**	63.61	82.73	87.35	88.53	85.78	76.25	80.76	89.14 ± 0.11
juice_bottle	91.37	80.71	86.33	**91.99**	90.54	90.41	87.06	88.40	89.07 ± 0.10
pushpins	70.66	54.74	**82.94**	40.72	67.67	41.42	29.42	59.96	75.64 ± 0.36
screw_bag	69.94	65.23	58.07	65.35	62.40	67.33	59.74	61.64	**71.35** ± 0.12
splicing_connectors	63.40	54.16	67.69	71.20	69.71	57.82	56.14	61.02	**75.10** ± 0.20
Mean	76.99	63.69	75.55	71.32	75.77	68.55	61.72	70.36	**80.06** ± 0.20

Table A.8: Anomaly segmentation performance based on Pixel AUPRO on MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	67.27	36.11	63.8	**74.28**	73.08	69.67	63.56	58.44	71.36 ± 0.38
juice_bottle	80.75	51.51	77.90	**88.78**	85.42	84.95	82.88	86.51	87.72 ± 0.09
pushpins	61.09	24.68	50.62	52.71	63.52	53.52	59.12	59.25	**68.34** ± 0.55
screw_bag	54.39	31.27	38.1	61.42	56.12	59.66	**71.66**	62.45	66.52 ± 0.33
splicing_connectors	71.15	56.72	34.77	62.64	67.29	63.62	71.67	68.14	**74.70** ± 0.25
Mean	66.93	40.06	53.04	67.97	69.09	66.28	69.78	66.96	**73.73** ± 0.35

Table A.9: Anomaly detection performance based on Image AUROC on MPDD dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
tubes	**99.64**	61.28	89.36	56.48	87.50	89.67	94.47	95.28	93.39 ± 0.57
metal_plate	97.42	80.05	86.92	42.69	99.72	91.87	68.31	**100.0**	93.81 ± 0.28
connector	99.52	83.33	52.38	86.07	**100.0**	93.10	**100.0**	50.00	96.00 ± 0.82
bracket_white	79.89	84.00	50.78	80.33	89.67	83.67	54.55	96.48	**100.0** ± 0.00
bracket_black	**96.48**	65.36	62.83	66.69	86.97	50.73	72.63	85.45	93.09 ± 0.32
bracket_brown	49.70	70.81	47.89	78.66	95.78	68.70	88.54	85.32	**98.08** ± 0.14
Mean	87.11	74.14	65.03	68.48	93.27	79.62	79.75	85.42	**95.73** ± 0.45

Table A.10: Anomaly segmentation performance based on Pixel AUROC on MPDD dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
tubes	**99.15**	76.87	98.44	91.35	98.45	99.08	98.62	98.98	98.55 ± 0.10
metal_plate	**98.56**	96.23	92.99	91.67	98.30	97.50	93.59	96.52	96.77 ± 0.09
connector	97.38	90.01	92.69	97.93	99.11	98.55	98.68	99.32	**99.40** ± 0.18
bracket_white	96.74	86.64	90.23	97.21	97.90	98.13	96.63	97.71	**98.73** ± 0.1
bracket_black	97.68	95.89	94.39	93.79	97.52	96.40	**98.42**	97.17	97.43 ± 0.21
bracket_brown	95.04	76.11	92.84	95.13	97.15	97.34	**98.06**	92.46	93.83 ± 2.41
Mean	97.42	86.96	93.60	94.51	**98.07**	97.83	97.33	97.03	97.45 ± 0.99

Table A.11: Anomaly segmentation performance based on Pixel AUPRO on MPDD dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
tubes	**96.76**	44.84	94.85	71.53	93.83	95.98	95.20	96.27	94.33 ± 0.33
metal_plate	91.53	82.83	74.62	75.47	**92.50**	92.00	83.99	83.59	90.07 ± 0.22
connector	91.32	72.06	76.98	92.74	96.89	95.29	95.60	97.77	**97.98** ± 0.60
bracket_white	78.66	69.13	49.65	81.16	83.13	84.71	77.02	93.27	**95.33** ± 0.37
bracket_black	89.48	93.12	79.65	83.51	93.65	89.14	**95.57**	89.98	90.17 ± 0.67
bracket_brown	83.62	58.29	85.62	82.69	85.07	94.04	**95.37**	81.77	84.24 ± 6.38
Mean	88.56	70.04	76.89	81.18	90.84	91.86	90.46	90.44	**92.02** ± 2.64

Table A.12: Anomaly detection performance based on Image AUROC on BTAD dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
01	98.64	80.17	94.46	99.51	98.09	92.23	99.51	94.15	**100.0** ± 0.00
02	82.12	65.23	84.27	82.17	81.73	61.73	85.68	75.42	**88.5** ± 0.78
03	99.95	74.87	96.3	97.92	**100.0**	97.65	98.62	95.22	**100.0** ± 0.00
Mean	93.57	73.42	91.68	93.20	93.27	83.87	94.60	88.26	**96.17** ± 0.45

Table A.13: Anomaly segmentation performance based on Pixel AUROC on BTAD dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
01	95.44	59.11	93.05	96.54	95.94	**96.98**	96.93	64.59	95.86 ± 0.03
02	94.81	69.29	96.16	95.11	95.18	**96.83**	96.77	85.67	94.76 ± 0.88
03	99.55	48.73	99.25	99.56	99.44	**99.74**	99.12	96.12	99.55 ± 0.02
Mean	96.60	59.04	96.15	97.07	96.85	**97.85**	97.62	82.13	96.73 ± 0.51

Table A.14: Anomaly segmentation performance based on AUPRO on BTAD dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
01	66.79	21.57	60.83	75.76	64.34	79.45	**83.77**	29.75	72.88 ± 0.12
02	54.32	27.64	**67.98**	59.19	52.36	66.05	65.58	44.37	55.16 ± 6.83
03	98.21	18.24	96.99	98.45	97.76	**98.92**	27.83	88.98	98.18 ± 0.08
Mean	73.11	22.48	75.27	77.80	71.48	**81.47**	59.06	54.37	75.41 ± 3.95

Table A.15: Anomaly detection performance based on Image AUROC on VisA dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
candle	94.38	79.43	93.18	86.19	**98.59**	85.54	89.65	80.52	87.11 ± 0.29
capsules	69.9	72.77	81.05	61.72	69.92	**87.37**	76.75	63.73	79.61 ± 0.72
cashew	94.7	95.5	87.78	90.94	**96.29**	85.38	93.80	96.11	94.72 ± 0.16
chewinggum	99.02	83.68	95.18	98.20	99.29	81.92	99.22	98.27	**99.49** ± 0.12
fryum	92.98	70.46	92.60	85.06	93.5	77.94	**96.58**	95.70	95.86 ± 0.14
macaroni1	92.72	72.8	82.48	78.62	91.50	82.06	95.14	**95.23**	90.66 ± 0.76
macaroni2	63.44	47.85	69.75	70.05	71.36	81.75	**86.25**	83.82	82.84 ± 1.05
pcb1	91.06	72.27	88.07	87.59	95.08	92.60	**97.57**	93.78	92.92 ± 0.11
pcb2	79.95	91.17	86.47	83.20	92.46	87.57	91.55	**94.95**	93.67 ± 0.18
pcb3	82.23	81.29	81.47	72.79	92.46	90.87	**97.27**	95.92	93.62 ± 0.16
pcb4	96.29	90.44	95.68	95.67	99.20	96.17	97.62	97.89	**99.43** ± 0.03
pipe_fryum	96.54	75.32	96.16	89.28	98.07	85.68	98.36	98.59	**99.61** ± 0.11
Mean	87.77	77.75	87.49	83.28	91.48	86.24	85.18	91.21	**92.46** ± 0.45

Table A.16: Anomaly segmentation performance based on Pixel AUROC on VisA dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
candle	98.75	83.1	97.24	97.68	98.92	**99.11**	98.41	89.93	97.77 ± 0.03
capsules	96.88	62.39	97.13	90.60	97.62	**99.56**	99.13	96.93	98.31 ± 0.31
cashew	99.25	74.17	98.57	97.45	98.88	97.23	95.63	98.85	**99.49** ± 0.02
chewinggum	99.02	84.11	98.83	98.82	98.72	**99.37**	99.16	98.69	98.10 ± 0.41
fryum	97.08	85.7	93.20	96.20	94.30	96.33	95.45	96.52	**97.38** ± 0.19
macaroni1	98.71	63.95	98.60	97.85	98.13	99.48	**99.73**	99.59	99.00 ± 0.13
macaroni2	97.35	79.02	94.65	95.40	96.79	99.33	**99.43**	98.84	98.20 ± 0.28
pcb1	99.05	27.98	99.29	98.67	99.47	**99.65**	99.30	98.98	99.61 ± 0.01
pcb2	96.40	59.49	97.12	98.12	97.72	98.28	96.13	**98.37**	98.03 ± 0.09
pcb3	97.23	76.43	97.04	98.06	98.13	**98.98**	97.99	98.91	98.45 ± 0.05
pcb4	97.97	83.42	97.51	97.00	97.83	**98.29**	96.58	95.49	95.22 ± 0.27
pipe_fryum	98.79	75.99	98.72	99.19	98.68	98.6	97.97	98.99	**99.29** ± 0.03
Mean	98.04	71.31	97.32	97.09	97.93	**98.68**	97.90	97.51	98.24 ± 0.20

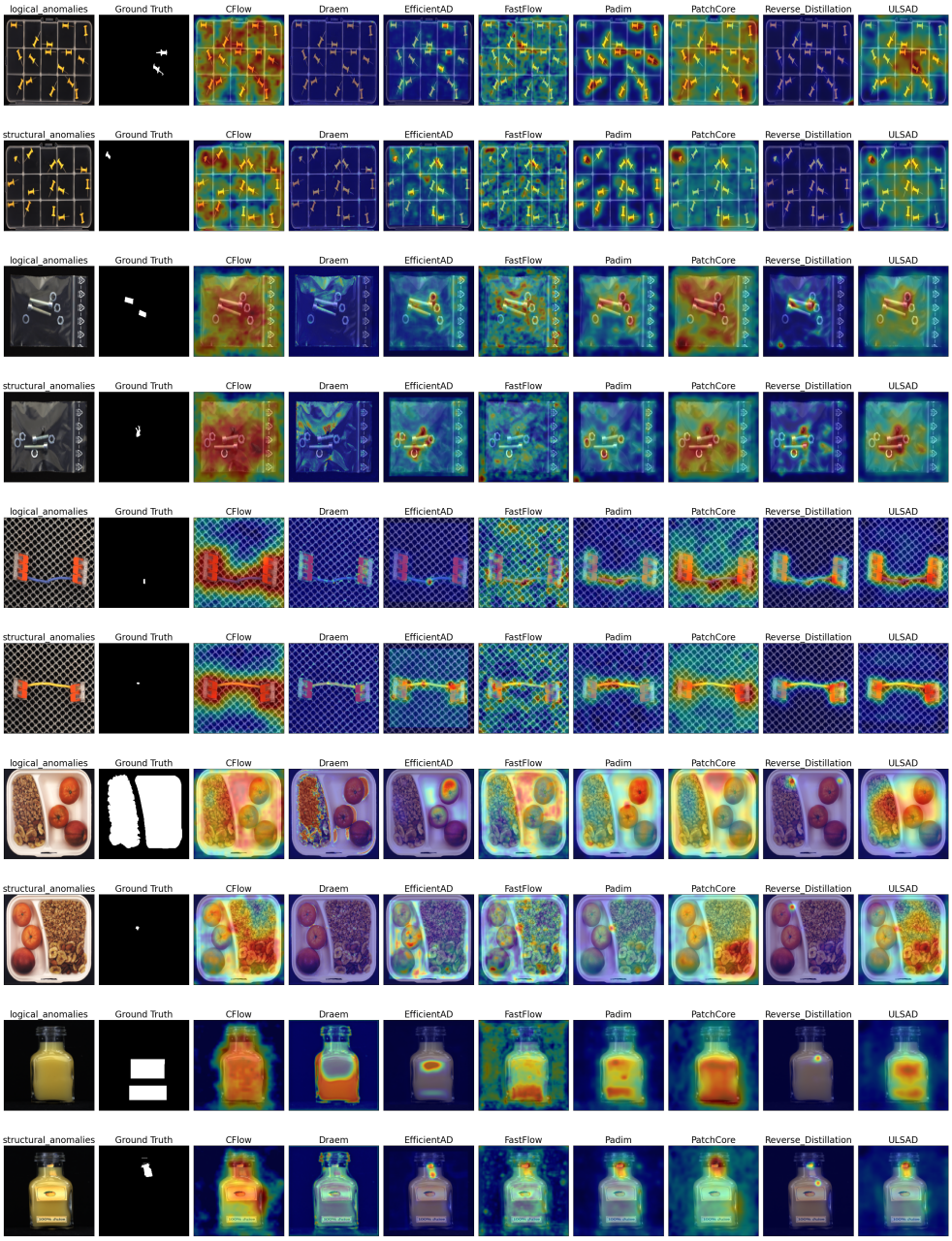


Figure A.1: Visualisation of anomaly maps on anomalous images from MVTecLOCO dataset.

Table A.17: Anomaly segmentation performance based on AUPRO on VisA dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
candle	92.7	80.29	91.65	92.77	94.08	95.30	**95.56**	77.31	92.49 ± 0.15
capsules	74.64	34.4	81.8	48.42	68.88	**92.20**	92.09	83.8	82.76 ± 1.18
cashew	**93.0**	48.33	85.54	82.36	88.01	91.81	90.51	91.57	91.85 ± 1.15
chewinggum	**89.58**	62.66	84.69	84.33	83.86	88.57	85.52	74.87	84.34 ± 1.0
fryum	85.62	71.94	72.39	75.54	78.25	84.8	**92.08**	82.93	85.47 ± 0.66
macaroni1	89.46	63.37	91.89	88.55	91.74	95.53	**97.59**	96.06	92.8 ± 0.74
macaroni2	78.74	56.69	71.94	75.76	87.49	94.01	**94.23**	89.74	88.29 ± 1.89
pcb1	87.24	27.43	85.89	86.39	89.07	**95.0**	93.55	90.53	90.22 ± 0.28
pcb2	77.83	33.99	77.99	83.68	83.00	89.17	87.26	**90.43**	84.53 ± 0.53
pcb3	75.03	71.9	71.33	81.37	79.69	90.89	**92.48**	92.08	85.86 ± 0.38
pcb4	86.53	73.26	83.41	82.47	84.91	**89.17**	84.38	75.25	73.37 ± 0.81
pipe_fryum	93.14	31.86	81.89	87.99	92.42	94.76	**95.46**	68.77	93.49 ± 0.08
Mean	85.29	54.68	81.70	80.80	85.12	**91.77**	91.72	84.45	87.12 ± 0.89

Table A.18: Anomaly detection performance based on Image AUROC on **structural anomalies** of MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	62.3	75.37	71.67	64.17	**84.3**	48.74	58.67	69.04	81.94 ± 0.74
juice_bottle	73.45	52.18	76.52	86.17	96.47	78.29	62.75	**99.71**	95.53 ± 0.45
pushpins	71.03	66.51	60.17	72.4	77.42	50.17	40.06	**92.07**	85.17 ± 0.65
screw_bag	78.0	69.78	75.07	67.7	**86.79**	80.35	57.76	82.2	83.01 ± 1.56
splicing_connectors	74.35	80.33	70.31	66.85	88.67	63.04	55.84	**90.2**	80.59 ± 0.35
Mean	71.83	68.83	70.75	71.46	**86.73**	64.12	55.02	86.64	85.25 ± 0.86

Table A.19: Anomaly segmentation performance based on Pixel AUROC on **structural anomalies** of MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	94.72	61.6	84.86	88.41	94.31	91.75	82.79	67.36	**95.35** ± 0.16
juice_bottle	88.81	83.75	85.67	90.73	95.98	89.57	80.74	**97.24**	89.67 ± 0.23
pushpins	91.4	46.14	85.54	90.82	**95.16**	87.83	48.52	89.48	88.38 ± 0.22
screw_bag	95.37	72.27	91.74	94.97	97.45	**97.78**	66.67	97.37	97.64 ± 0.24
splicing_connectors	98.3	93.03	95.79	96.25	**98.82**	97.25	91.82	98.55	98.3 ± 0.1
Mean	93.72	71.36	88.72	92.24	**96.34**	92.84	74.12	90.0	93.87 ± 0.2

Table A.20: Anomaly segmentation performance based on Pixel AUPRO on **structural anomalies** of MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	70.0	41.92	71.29	**81.97**	78.71	76.67	26.67	65.87	75.43 ± 0.67
juice_bottle	80.56	50.44	87.61	92.64	**95.1**	90.81	61.67	92.08	90.1 ± 0.31
pushpins	68.34	20.04	61.72	70.01	75.88	62.63	45.30	**78.62**	68.34 ± 0.84
screw_bag	82.07	39.41	73.69	85.1	88.92	**93.77**	53.95	91.38	91.98 ± 0.73
splicing_connectors	87.75	68.61	67.8	82.35	91.69	83.57	63.44	**94.19**	90.84 ± 0.35
Mean	77.74	44.08	72.42	82.41	**86.06**	81.49	50.21	84.43	83.34 ± 0.62

Table A.21: Anomaly detection performance based on Image AUROC on **logical anomalies** of MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	76.86	68.71	75.81	62.0	83.43	55.27	61.55	**83.54**	83.33 ± 1.63
juice_bottle	80.52	70.09	80.89	92.94	94.61	76.4	77.07	**99.12**	98.82 ± 0.14
pushpins	71.15	37.9	58.81	50.72	74.53	52.98	60.27	72.0	**85.23** ± 0.72
screw_bag	60.72	52.34	64.01	55.29	59.36	55.09	63.64	58.8	**66.33** ± 0.9
splicing_connectors	67.5	56.15	74.54	69.43	80.27	61.74	53.10	75.64	**86.27** ± 1.33
Mean	71.35	57.04	70.81	66.08	78.44	60.3	63.13	77.82	**84.0** ± 1.07

Table A.22: Anomaly segmentation performance based on Pixel AUROC on **logical anomalies** of MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	90.43	65.5	85.35	89.23	90.32	87.35	74.55	85.58	**91.41** ± 0.1
juice_bottle	92.56	80.49	90.23	**95.29**	93.97	92.97	86.95	91.98	93.74 ± 0.08
pushpins	70.38	56.06	**82.91**	41.95	69.39	41.86	67.93	61.12	75.96 ± 1.11
screw_bag	67.75	65.43	58.93	66.02	63.81	68.4	**75.75**	61.85	72.3 ± 0.41
splicing_connectors	60.8	52.1	66.82	69.83	68.06	55.5	57.67	59.01	**74.19** ± 0.19
Mean	76.38	63.92	76.85	72.46	77.11	69.22	72.57	71.91	**81.52** ± 0.54

Table A.23: Anomaly segmentation performance based on Pixel AUPRO on **logical anomalies** of MVTecLOCO dataset.

Category	CFLOW (2021)	DRÆM (2021)	FastFlow (2021	PaDiM (2021)	PatchCore (2022)	RD (2022)	DFR (2020)	EffAD (2024)	ULSAD (Ours)
breakfast_box	68.79	32.23	64.39	69.74	72.27	68.77	43.60	52.19	**73.97** ± 0.8
juice_bottle	79.67	52.72	77.9	91.69	87.88	84.22	62.87	87.82	**91.36** ± 0.12
pushpins	59.07	26.21	47.71	51.93	63.47	53.84	41.80	58.36	**68.78** ± 0.98
screw_bag	50.7	26.98	25.58	53.92	46.8	48.72	54.59	52.8	**61.48** ± 0.45
splicing_connectors	65.86	53.71	26.66	57.66	62.21	58.85	34.72	62.7	**72.66** ± 0.27
Mean	64.82	38.37	48.45	64.99	66.53	62.88	47.52	62.77	**73.65** ± 0.61

A.3. Extended ablations

In this section, we provide additional ablations on the local branch in Table A.24 and the total architecture in Table A.25. Lastly, in Table A.26 we provide the per-category results for the ablation on the pre-trained backbone which is summarized in Figure 3.7.

Table A.24: Ablations for local branch. Style: I-AUROC | P-AUROC | P-AUPRO.

category	$\lambda_l = 0$	$\lambda_l = 0.01$	$\lambda_l = 0.5$	$\lambda_l = 0.9$	$\lambda_l = 1.0$
breakfast_box	78.64 | 88.28 | 74.22	79.2 | **88.51** | **74.35**	77.86 | 87.79 | 71.27	77.95 | 86.89 | 67.06	**79.44** | 86.96 | 65.36
juice_bottle	**97.82** | 92.14 | 89.24	97.76 | **92.23** | 89.38	97.56 | 88.78 | 88.16	97.36 | 84.39 | 84.63	97.08 | 83.61 | 83.47
pushpins	72.4 | 69.81 | 65.69	72.77 | 69.84 | 65.68	**79.92** | **74.49** | **69.03**	76.98 | 74.35 | 63.17	76.53 | 73.69 | 65.18
screw_bag	66.42 | 66.6 | 64.39	67.18 | 68.47 | 65.92	**68.06** | 69.13 | **66.22**	67.56 | **69.33** | 63.61	66.34 | 69.31 | 62.09
splicing_connectors	**73.05** | 59.04 | 73.3	72.84 | 59.15 | 73.29	72.29 | 62.66 | 72.39	72.79 | 64.33 | 70.74	72.36 | **64.5** | 70.57
Mean	77.67 | 75.17 | 73.37	77.95 | 75.64 | **73.72**	**79.14** | **76.57** | 73.41	78.53 | 75.86 | 69.84	78.35 | 75.61 | 69.33

Table A.25: Ablations for total architecture. Style: I-AUROC | P-AUROC | P-AUPRO.

category	$\mathcal{L}_{pg}^{d}$	$\mathcal{L}_{pg}^{d}$; $\mathcal{L}_{lg}$	$\mathcal{L}_{pg}$	$\mathcal{L}_{pg}$; $\mathcal{L}_{lg}$
		$\lambda_l = \lambda_g = 0.0$		
breakfast_box	77.29 | **90.41** | 77.16	82.82 | 89.85 | 76.92	66.01 | 87.36 | 67.5	**85.08** | 90.19 | 75.36
juice_bottle	96.48 | **92.01** | 88.82	**97.93** | 91.82 | **89.26**	91.2 | 91.98 | 85.38	97.29 | 92.0 | 89.18
pushpins	70.89 | 80.89 | 70.67	**78.66** | **88.09** | **79.11**	74.67 | 77.37 | 58.75	74.61 | 85.86 | 76.33
screw_bag	65.48 | 65.87 | 64.04	63.02 | 68.13 | **65.51**	61.14 | 59.51 | 57.75	**66.93** | **68.67** | 65.35
splicing_connectors	78.29 | 69.68 | 75.61	**84.55** | 72.71 | **76.54**	65.31 | 53.4 | 66.74	81.5 | **73.11** | 76.01
Mean	77.69 | 79.77 | 75.26	**81.4** | **82.12** | **77.47**	71.67 | 73.92 | 67.22	81.08 | 81.97 | 76.45
		$\lambda_l = \lambda_g = 0.5$		
breakfast_box	79.22 | **90.87** | **78.32**	82.45 | 88.49 | 71.03	71.36 | 87.64 | 67.38	**83.36** | 89.34 | 72.36
juice_bottle	96.3 | 91.17 | **88.89**	**98.08** | 87.06 | 88.05	91.14 | **91.84** | 85.92	97.46 | 88.81 | 87.71
pushpins	79.86 | 84.89 | 77.97	82.46 | **87.62** | **80.49**	78.64 | 81.14 | 65.12	**88.07** | 74.22 | 66.45
screw_bag	66.58 | 68.83 | 66.11	65.11 | 70.04 | 62.81	62.09 | 67.32 | 62.92	**70.6** | **71.58** | **67.01**
splicing_connectors	80.53 | 73.47 | **75.44**	**82.85** | 73.03 | 75.13	69.33 | 55.02 | 63.69	81.27 | **74.94** | 74.89
Mean	80.5 | **81.85** | **77.35**	82.19 | 81.25 | 75.5	74.51 | 76.59 | 69.01	**84.15** | 79.78 | 73.68

Table A.26: Ablations for backbone on MvTec-LOCO. Style: I-AUROC | P-AUROC | P-AUPRO.

Class	ResNet50	ResNet152	Wide-ResNet50-2	Wide-ResNet100-2
breakfast_box	82.41 | 89.74 | 73.09	**85.11** | **91.15** | 72.0	84.46 | 89.18 | 72.21	82.37 | 89.25 | **74.02**
juice_bottle	96.9 | 92.23 | 89.38	97.64 | 91.66 | 89.78	97.11 | 88.9 | 87.94	**98.74** | **92.87** | **91.07**
pushpins	81.79 | **79.49** | **75.15**	73.28 | 76.49 | 63.71	**85.48** | 75.46 | 67.82	82.48 | 76.55 | 70.67
screw_bag	66.46 | 69.14 | **67.01**	68.06 | 68.63 | 66.97	71.14 | **71.57** | 66.82	**73.1** | 68.99 | 66.88
splicing_connectors	80.88 | 72.76 | 74.39	83.53 | 76.31 | 77.24	82.59 | 75.21 | 75.05	**84.71** | **79.95** | **78.65**
Mean	81.79 | 79.49 | 75.15	81.52 | 80.85 | 73.94	84.16 | 80.06 | 73.97	**84.28** | **81.52** | **76.26**

APPENDIX B

Risk estimator-based Semi-supervised Anomaly Detection

B.1. Some definitions

Definition B.1.1. A loss ℓ is said to be classification-calibrated if, for any $\eta \neq \frac{1}{2}$, we have $H_\ell^-(\eta) > H_\ell(\eta)$, where

$$H_\ell(\eta) = \inf_{\alpha\in\mathbb{R}} (\eta\ell(\alpha,+1) + (1-\eta)\ell(\alpha,-1)),$$
$$H_\ell^-(\eta) = \inf_{\alpha\in\mathbb{R}:\alpha(\eta-\frac{1}{2})\leq 0} (\eta\ell(\alpha,+1) + (1-\eta)\ell(\alpha,-1))$$

Examples of classification-calibrated loss include the scaled ramp loss, the hinge loss, and the exponential loss. (Bartlett et al., 2006, Theorem 1) shows that if ℓ is a classification-calibrated loss, then there exists a convex, invertible and nondecreasing transformation ψ_ℓ with $\psi_\ell(0) = 0$ and $\psi_\ell(I(g) - I^*) \leq \mathcal{R}(g) - \mathcal{R}^*$, which implies that

$$I(g) - I^* \leq \psi_\ell^{-1}(\mathcal{R}(g) - \mathcal{R}^*) = \psi_\ell^{-1}(\mathcal{R}(g) - \mathcal{R}(g^*) + \mathcal{R}(g^*) - \mathcal{R}^*). \quad \text{(B.1)}$$

B.2. Additional experiments

B.2.1 Additional experiments for shallow rAD

Table B.1 reports the mean of the AUC of shallow rAD over the 30 trials for different values of $\hat{\pi}_p$.

Table B.2 reports the mean of the AUC of shallow rAD over the 30 trials for different values of a.

B.2.2 Additional experiments for deep rAD

Sensitivity analysis for $\hat{\pi}_p$ Table B.3 reports the mean and the standard error of the AUC of deep rAD over the 20 trials for different values of $\hat{\pi}_p$.

Sensitivity analysis for a Fixing $\hat{\pi}_p = 0.8$, Figure B.1–B.3 show AUC mean and std of deep rAD with additional values of $a \in \{0.5, 0.9\}$ ($a = 0.1$ is the default setting) on the datasets with $\gamma_l = 0.05$ and $\pi_n = \in \{0.01, 0.05, 0.2\}$

ROC curves Figure B.4 shows representative ROC curves obtained by a trial of running the methods (with default settings) on the datasets with $\gamma = 0.05$ and $\pi_n = 0.1$.

Table B.1: AUC means of shallow rAD over 30 trials for different $\hat{\pi}_p$. The significant changes in the AUC means are highlighted in bold.

Dataset	square/$\hat{\pi}_p$				hinge/$\hat{\pi}_p$				m-Huber/$\hat{\pi}_p$			
	$1-\pi_n$	0.9	0.7	0.6	$1-\pi_n$	0.9	0.7	0.6	$1-\pi_n$	0.9	0.7	0.6
thyroid	0.98	0.995	0.996	0.996	0.97	0.994	0.996	0.996	0.99	0.996	0.996	0.996
Waveform	**0.74**	0.82	0.84	0.84	**0.70**	**0.78**	0.83	0.83	**0.77**	0.84	0.85	0.85
mnist	0.96	0.96	0.97	0.97	0.96	0.96	0.96	0.96	0.97	0.97	0.97	0.97
campaign	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
landsat	0.74	0.74	0.74	0.74	0.74	0.73	0.74	0.74	0.74	0.74	0.74	0.74
satellite	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.81	0.80	0.80	0.80
satimage-2	0.97	0.98	0.98	0.98	**0.93**	0.98	0.98	0.98	0.98	0.99	0.99	0.99
vowels	**0.77**	0.85	0.87	0.87	**0.69**	**0.77**	0.85	0.85	0.85	0.88	0.88	0.88
CIFAR10-1	**0.69**	0.73	0.77	0.77	**0.66**	**0.71**	0.76	0.76	**0.71**	0.74	0.77	0.77
SVHN-1	0.80	0.82	0.84	0.84	**0.79**	0.82	0.84	0.84	0.80	0.83	0.84	0.84
20news-1	**0.64**	0.67	0.70	0.70	**0.56**	**0.59**	0.65	0.66	0.72	0.75	0.75	0.75
agnews-1	0.94	0.96	0.97	0.97	**0.88**	0.91	0.95	0.96	0.96	0.98	0.98	0.98
amazon	**0.72**	0.78	0.82	0.82	**0.66**	0.72	0.77	0.77	**0.76**	0.80	0.84	0.84
imdb	**0.75**	0.80	0.83	0.83	**0.69**	0.74	0.79	0.80	**0.78**	0.82	0.85	0.85
yelp	**0.82**	0.87	0.90	0.90	**0.74**	0.80	0.85	0.86	**0.85**	0.89	0.92	0.92
vertebral	0.71	0.71	0.72	0.72	0.71	0.70	0.70	0.71	0.72	0.73	0.73	0.72
fault	0.65	0.63	0.65	0.65	0.63	0.60	0.63	0.64	0.66	0.65	0.66	0.66

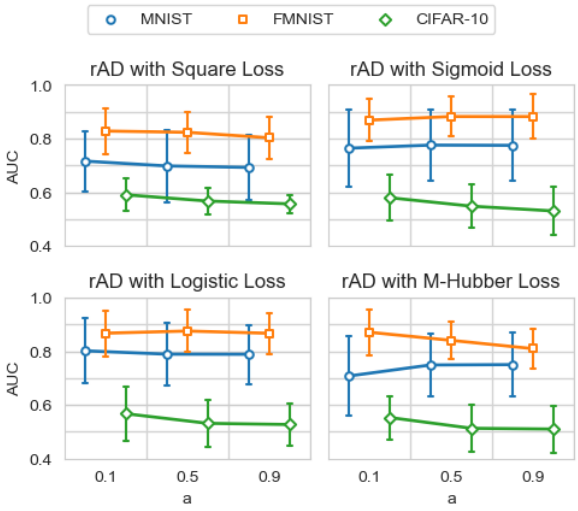


Figure B.1: AUC mean and std over 20 trials at various a for the datasets with $\gamma_l = 0.05$ and $\pi_n = 0.01$.

Table B.2: AUC means of shallow rAD over 30 trials for different a. The significant changes in the AUC means are highlighted in bold.

Dataset	square/a			hinge/a			m-Huber/a		
	0.3	0.7	0.9	0.3	0.7	0.9	0.3	0.7	0.9
thyroid	0.996	0.99	0.99	0.996	0.99	0.99	0.996	0.99	0.99
Waveform	0.84	0.81	0.80	0.82	0.80	**0.77**	0.85	0.83	0.81
mnist	0.97	0.96	0.96	0.96	0.96	0.96	0.97	0.96	0.96
campaign	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
landsat	0.74	0.74	0.73	0.74	0.74	0.73	0.74	0.74	0.74
satellite	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.81	0.81
satimage-2	0.98	0.99	0.98	0.98	0.99	0.98	0.99	0.99	0.98
vowels	0.87	0.85	0.83	0.85	0.81	0.78	0.88	0.87	0.86
CIFAR10-1	0.76	0.73	0.70	0.75	0.74	0.72	0.76	0.72	**0.69**
SVHN-1	0.84	0.83	0.82	0.83	0.83	0.82	0.84	0.83	0.81
20news-1	0.71	0.69	**0.65**	0.63	0.61	0.60	0.76	0.70	**0.66**
agnews-1	0.98	0.97	0.96	0.94	0.94	0.94	0.98	0.98	0.97
amazon	0.81	0.80	0.77	0.77	0.75	0.76	0.83	0.81	0.79
imdb	0.82	0.80	0.78	0.78	0.77	0.75	0.84	0.81	0.79
yelp	90	0.88	0.86	0.84	0.83	0.82	0.91	0.89	0.87
vertebral	0.72	0.73	0.73	0.73	0.74	0.73	0.74	0.75	0.74
fault	0.65	0.65	0.65	0.63	0.64	0.64	0.66	0.66	0.66

Table B.3: AUC means (and standard error) of deep rAD over 20 trials for different $\hat{\pi}_p$. The significant changes in the AUC means are highlighted in bold.

Dataset	Loss	$\hat{\pi}_p = 1 - \pi_n$	$\hat{\pi}_p = 0.9$	$\hat{\pi}_p = 0.8$	$\hat{\pi}_p = 0.7$	$\hat{\pi}_p = \pi_n$
MNIST ($\pi_n = 0.01$)	square	**0.66**(0.04)	0.70(0.03)	0.72(0.02)	0.68(0.03)	**0.65**(0.03)
	sigmoid	**0.68**(0.03)	0.76(0.03)	0.76(0.03)	0.77(0.03)	0.77(0.03)
	logistic	**0.67**(0.03)	0.76(0.03)	0.80(0.03)	0.77(0.03)	0.77(0.03)
	m-Huber	**0.68**(0.03)	0.74(0.03)	0.71(0.03)	0.72(0.03)	0.73(0.03)
MNIST ($\pi_n = 0.05$)	square	0.85(0.02)	0.87(0.01)	0.89(0.01)	0.89(0.01)	0.86(0.01)
	sigmoid	0.88(0.01)	0.91(0.01)	0.91(0.01)	0.93(0.01)	0.93(0.01)
	logistic	0.87(0.02)	0.89(0.01)	0.92(0.01)	0.92(0.01)	0.91(0.01)
	m-Huber	0.86(0.01)	0.88(0.01)	0.90(0.01)	0.90(0.01)	0.87(0.01)
MNIST ($\pi_n = 0.1$)	square	0.92(0.01)	0.92(0.01)	0.93(0.01)	0.93(0.01)	**0.89**(0.01)
	sigmoid	0.94(0.01)	0.94(0.01)	0.95(0.01)	0.95(0.01)	0.94(0.01)
	logistic	0.93(0.01)	0.93(0.01)	0.94(0.01)	0.94(0.01)	0.93(0.01)
	m-Huber	0.93(0.01)	0.93(0.01)	0.93(0.01)	0.94(0.01)	0.92(0.01)
MNIST ($\pi_n = 0.2$)	square	0.95(0.01)	0.95(0.01)	0.95(0.01)	0.96(0.01)	0.94(0.01)
	sigmoid	0.95(0.01)	0.95(0.01)	0.95(0.01)	0.95(0.01)	0.95(0.01)
	logistic	0.96(0.01)	0.96(0.01)	0.96(0.01)	0.96(0.01)	0.96(0.01)
	m-Huber	0.95(0.01)	0.94(0.01)	0.95(0.01)	0.95(0.01)	0.94(0.01)
F-MNIST ($\pi_n = 0.01$)	square	**0.76**(0.02)	0.80(0.01)	0.83(0.02)	0.84(0.02)	**0.78**(0.02)
	sigmoid	0.86(0.02)	0.87(0.02)	0.87(0.02)	0.88(0.02)	0.88(0.02)
	logistic	0.85(0.02)	0.87(0.02)	0.87(0.02)	0.88(0.02)	0.87(0.02)
	m-Huber	**0.82**(0.02)	0.88(0.02)	0.87(0.02)	0.87(0.02)	**0.83**(0.02)
F-MNIST ($\pi_n = 0.05$)	square	**0.84**(0.01)	0.86(0.01)	0.89(0.01)	0.91(0.01)	0.91(0.01)
	sigmoid	0.93(0.01)	0.93(0.01)	0.93(0.01)	0.94(0.01)	0.95(0.01)
	logistic	0.91(0.01)	0.92(0.01)	0.93(0.01)	0.93(0.01)	0.95(0.01)
	m-Huber	0.92(0.01)	0.94(0.01)	0.93(0.01)	0.93(0.01)	0.93(0.01)
F-MNIST ($\pi_n = 0.1$)	square	**0.88**(0.01)	**0.88**(0.01)	0.93(0.01)	0.94(0.01)	0.94(0.01)
	sigmoid	0.94(0.01)	0.94(0.01)	0.95(0.01)	0.95(0.01)	0.96(0.01)
	logistic	0.94(0.01)	0.94(0.01)	0.95(0.01)	0.95(0.01)	0.96(0.01)
	m-Huber	0.95(0.01)	0.95(0.01)	0.95(0.01)	0.95(0.01)	0.95(0.01)
F-MNIST ($\pi_n = 0.2$)	square	0.94(0.01)	0.92(0.01)	0.94(0.01)	0.95(0.01)	0.96(0.01)
	sigmoid	0.96(0.01)	0.95(0.01)	0.96(0.01)	0.96(0.01)	0.96(0.01)
	logistic	0.95(0.01)	0.94(0.01)	0.95(0.01)	0.96(0.01)	0.97(0.01)
	m-Huber	0.96(0.01)	0.94(0.01)	0.96(0.01)	0.96(0.01)	0.96(0.01)
CIFAR-10 ($\pi_n = 0.01$)	square	0.60(0.01)	0.60(0.01)	0.59(0.01)	0.60(0.01)	0.59(0.01)
	sigmoid	0.58(0.01)	0.58(0.02)	0.58(0.02)	0.57(0.02)	0.55(0.02)
	logistic	0.60(0.02)	0.58(0.02)	0.57(0.02)	0.56(0.02)	**0.53**(0.02)
	m-Huber	0.61(0.02)	**0.55**(0.02)	**0.55**(0.02)	**0.55**(0.02)	0.60(0.02)
CIFAR-10 ($\pi_n = 0.05$)	square	0.73(0.01)	0.72(0.01)	0.73(0.01)	0.72(0.01)	0.73(0.01)
	sigmoid	0.66(0.02)	0.68(0.01)	0.69(0.01)	0.67(0.01)	0.69(0.02)
	logistic	0.71(0.01)	0.71(0.02)	0.71(0.01)	0.70(0.01)	0.69(0.01)
	m-Huber	0.69(0.01)	0.70(0.01)	0.71(0.01)	0.72(0.01)	0.71(0.01)
CIFAR-10 ($\pi_n = 0.1$)	square	0.77(0.01)	0.77(0.01)	0.77(0.01)	0.78(0.01)	0.77(0.01)
	sigmoid	0.75(0.01)	0.75(0.01)	0.75(0.01)	0.76(0.01)	0.73(0.01)
	logistic	0.77(0.01)	0.77(0.01)	0.77(0.01)	0.77(0.01)	0.76(0.01)
	m-Huber	0.77(0.01)	0.77(0.01)	0.77(0.01)	0.78(0.01)	0.77(0.01)
CIFAR-10 ($\pi_n = 0.2$)	square	0.80(0.01)	0.80(0.01)	0.80(0.01)	0.80(0.01)	0.80(0.01)
	sigmoid	0.77(0.01)	0.74(0.01)	0.77(0.01)	0.77(0.01)	0.77(0.01)
	logistic	0.79(0.01)	0.78(0.01)	0.79(0.01)	0.80(0.01)	0.79(0.01)
	m-Huber	0.79(0.01)	0.79(0.01)	0.79(0.01)	0.80(0.01)	0.80(0.01)

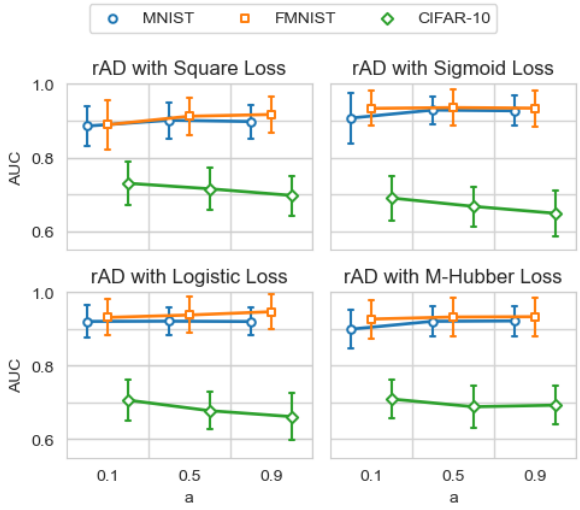


Figure B.2: AUC mean and std over 20 trials at various a for the datasets with $\gamma_l = 0.05$ and $\pi_n = 0.05$.

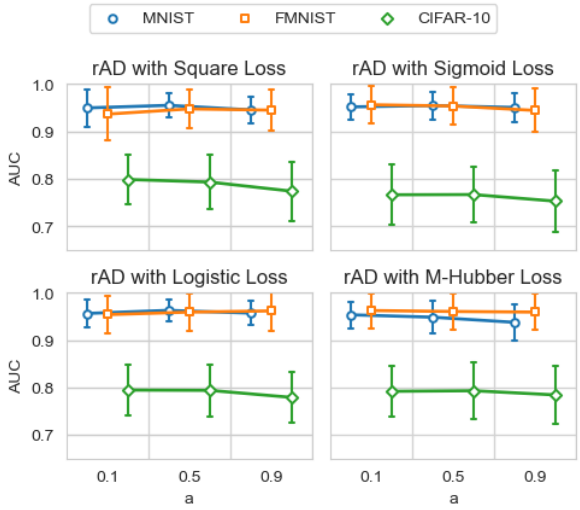


Figure B.3: AUC mean and std over 20 trials at various a for the datasets with $\gamma_l = 0.05$ and $\pi_n = 0.2$.

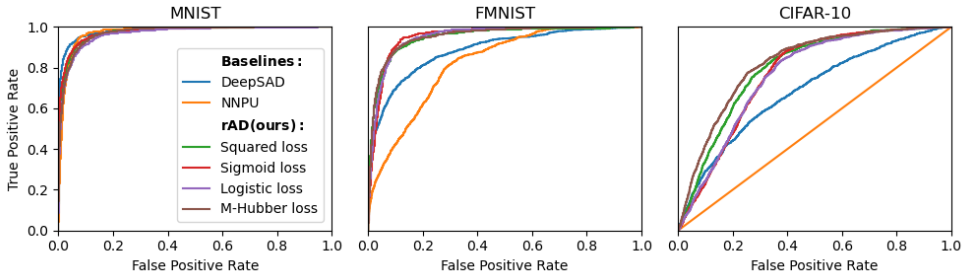


Figure B.4: Representative ROC curves for different datasets with $\gamma = 0.05$ and $\pi_n = 0.1$.

APPENDIX C

Evidence-Based Test-time Adaptation Framework

C.1. Proofs

C.1.1 Proof of Proposition 5.4.1

Proof. From (5.2), we have

$$f_X^{\pm}(x) = \epsilon f_X^{-}(x) + (1-\epsilon) f_X^{+}(x).$$

Additionally, from (5.3), we have

$$\check{f}_X^{\pm}(x) = \frac{f_X^{\pm}(x)\exp(T(x)/\beta)}{Z_X^{\beta}}$$

Then,

$$\begin{aligned}
\mathrm{KL}(f_X^{+} \| \check{f}_X^{\pm}) &= \mathbb{E}_{x\sim \mathrm{P}_X^{+}}\left[\log \frac{f_X^{+}(x)}{\check{f}_X^{\pm}(x)}\right] \\
&= \mathbb{E}_{x\sim \mathrm{P}_X^{+}}\left[\log f_X^{+}(x) - \log \check{f}_X^{\pm}(x)\right] \\
&= \mathbb{E}_{x\sim \mathrm{P}_X^{+}}\left[\log f_X^{+}(x) - \log \frac{f_X^{\pm}(x)\exp(T(x)/\beta)}{Z_X^{\beta}}\right] \\
&= \mathbb{E}_{x\sim \mathrm{P}_X^{+}}\left[\log f_X^{+}(x) - \log f_X^{\pm}(x)\exp(\frac{T(x)}{\beta}) + \log Z_X^{\beta}\right] \\
&= \mathbb{E}_{x\sim \mathrm{P}_X^{+}}\Big[\log f_X^{+}(x) - \log f_X^{\pm}(x) - \log\exp(\frac{T(x)}{\beta}) \\
&\qquad\qquad + \log Z_X^{\beta}\Big] \\
&= \mathrm{KL}(f_X^{+} \| f_X^{\pm}) - \mathbb{E}_{x\sim \mathrm{P}_X^{+}}\left[\log\exp(\frac{T(x)}{\beta}) - \log Z_X^{\beta}\right] \\
&= \mathrm{KL}(f_X^{+} \| f_X^{\pm}) - \mathbb{E}_{x\sim \mathrm{P}_X^{+}}\left[\log \frac{\exp(T(x)/\beta)}{Z_X^{\beta}}\right]
\end{aligned}$$

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

$$\mathbb{E}_{x\sim \mathrm{P}_X^{+}}\left[\log \frac{\exp(T(x)/\beta)}{Z_X^{\beta}}\right] \geq 0. \tag{C.1}$$

□

C.2. Additional implementation details

C.2.1 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.

C.2.2 Details of the experiment using synthetic data

The synthetic dataset is generated using a 2D Gaussian mixture model with three components. Normal samples are drawn from $f_X^+(x) := \mathcal{N}([1,1]^T, 0.07\mathbf{I}_2)$, while anomalous samples are sampled from $f_X^-(x) := \mathcal{N}([-0.25, 2.5]^T, 0.03\mathbf{I}_2) + \mathcal{N}([-1, 0.5]^T, 0.03\mathbf{I}_2)$. 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 optimized during training. The RBF network outputs a 1D scalar obtained as a linear combination of the outputs from the hidden layer. The centre is initialized 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 optimizer with a learning rate of 0.01, 200 epochs, and a mini-batch size of 25.

C.2.3 Computing evidence functions

`EPHAD` relies on an evidence function $T(x)$, computed during test-time, 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 C.2.3.1) and tabular datasets (Section C.2.3.2). 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 5.5, 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.

C.2.3.1 Evidence for visual 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

Table C.1: 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`			

texts $\mathcal{C} = \{c_1, \ldots, c_k\}$

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

where $\langle\cdot,\cdot\rangle$ 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 C.1. 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 test-time is computed as:

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

C.2.3.2 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.

C.2.4 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.

C.3. Extended results

C.3.1 Additional experiments on tabular datasets

Table C.2, C.3, C.4, and C.5 summarise the results on a larger set of tabular datasets from ADBench. Each experiment is repeated with three seeds. We can observe that in most cases 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.

Table C.2: Performance of `EPHAD` 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)

C.3.2 Comparison against LOE and SoftPatch

To ensure a comprehensive evaluation, we compare the performance of our proposed post-hoc framework against SoftPatch (Jiang et al., 2022) and both variants of LOE (Qiu et al., 2022). However, it is important to note that, unlike our approach, both SoftPatch and 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 C.3: Performance of EPHAD 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)

First, 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 C.6, show that given a good evidence function, i.e. the performance of the evidence is better than the "Blind" configuration, our simple test-time framework outperforms LOE.

Table C.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

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 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

Table C.4: Performance od `EPHAD-Ada` 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-Ada`	Blind	+ `EPHAD-Ada`	Blind	+ `EPHAD-Ada`	Blind	+ `EPHAD-Ada`	Blind	+ `EPHAD-Ada`
aloi	72.64 (± 0.1)	51.46 (± 0.05)	**53.65** (± 0.17)	54.06 (± 0.54)	**70.67** (± 0.22)	53.14 (± 0.03)	**55.47** (± 0.18)	54.05 (± 0.21)	**57.49** (± 0.31)	73.57 (± 0.1)	**73.85** (± 0.05)
annthyroid	68.53 (± 0.12)	73.45 (± 0.08)	**73.91** (± 0.06)	62.69 (± 3.33)	**69.27** (± 0.94)	76.05 (± 0.11)	**76.23** (± 0.06)	71.39 (± 0.34)	**72.24** (± 0.36)	**72.12** (± 0.57)	71.79 (± 0.39)
backdoor	70.43 (± 0.08)	**75.06** (± 0.07)	75.04 (± 0.07)	**78.34** (± 1.21)	78.31 (± 1.21)	**83.0** (± 0.09)	82.99 (± 0.09)	**51.29**(± 1.29)	**51.29** (± 1.29)	46.65 (± 0.26)	**61.97** (± 1.74)
breastw	46.31 (± 0.92)	**99.46** (± 0.06)	97.73 (± 0.06)	**98.65** (± 0.05)	92.94 (± 1.83)	**99.01** (± 0.04)	96.87 (± 0.26)	**99.46** (± 0.04)	97.71 (± 0.21)	**73.39** (± 1.35)	66.12 (± 1.49)
celeba	41.45 (± 0.32)	**72.09** (± 0.01)	70.85 (± 0.07)	**67.51** (± 3.07)	64.46 (± 3.06)	**73.99** (± 0.01)	72.89 (± 0.06)	**40.09** (± 0.83)	37.54 (± 0.87)	**42.97** (± 0.23)	40.96 (± 0.34)
cover	52.12 (± 0.1)	78.7 (± 0.03)	**79.57** (± 0.01)	75.11 (± 11.37)	**75.58** (± 10.82)	85.34 (± 0.02)	**85.45** (± 0.01)	72.59 (± 1.59)	**73.15** (± 1.57)	22.44 (± 0.1)	**36.78** (± 0.23)
fault	55.0 (± 0.53)	45.69 (± 0.58)	**45.79** (± 0.58)	47.34 (± 0.99)	**50.25** (± 0.65)	**47.0** (± 0.4)	46.81 (± 0.39)	**58.08** (± 0.94)	57.61 (± 0.9)	**64.41** (± 1.35)	61.29 (± 0.9)
fraud	45.75 (± 0.13)	**94.39** (± 0.0)	94.38 (± 0.01)	**89.98** (± 0.97)	89.93 (± 0.97)	**93.86** (± 0.0)	93.84 (± 0.01)	**92.95** (± 0.29)	92.94 (± 0.29)	33.92 (± 0.34)	**43.01** (± 0.84)
glass	77.52 (± 0.93)	76.11 (± 0.77)	**81.77** (± 1.28)	64.52 (± 6.87)	**80.94** (± 2.52)	67.65 (± 0.44)	**78.43** (± 1.72)	78.5 (± 1.47)	**83.15** (± 1.86)	71.79 (± 1.08)	**75.67** (± 0.75)
http	37.65 (± 0.09)	**94.91** (± 0.01)	**94.91** (± 0.01)	**99.17** (± 0.08)	**99.17** (± 0.08)	**92.35** (± 0.02)	**92.35** (± 0.02)	**96.82** (± 0.37)	**96.82** (± 0.37)	17.85 (± 2.03)	**18.31** (± 1.96)
ionosphere	82.43 (± 0.16)	79.42 (± 1.03)	**84.15** (± 0.38)	83.09 (± 0.57)	**85.03** (± 0.25)	73.04 (± 0.84)	**78.14** (± 0.49)	89.58 (± 1.57)	**90.05** (± 1.22)	**94.64** (± 0.52)	91.85 (± 0.68)
letter	83.15 (± 0.73)	56.71 (± 0.12)	**71.03** (± 0.99)	50.51 (± 2.54)	**65.9** (± 2.88)	56.41 (± 0.29)	**70.15** (± 1.15)	59.84 (± 0.64)	**71.38** (± 0.86)	**85.74** (± 0.54)	85.31 (± 0.36)
lymphography	99.44 (± 0.26)	99.52 (± 0.22)	**99.84** (± 0.13)	98.57 (± 0.74)	**99.45** (± 0.23)	99.6 (± 0.23)	**99.84** (± 0.13)	99.76 (± 0.19)	**99.92** (± 0.07)	98.57 (± 0.59)	**99.28** (± 0.39)
mammography	67.29 (± 0.19)	**89.29** (± 0.05)	89.23 (± 0.05)	**87.23** (± 0.95)	87.11 (± 0.95)	**89.38** (± 0.06)	89.32 (± 0.06)	**80.44** (± 0.29)	80.37 (± 0.29)	69.7 (± 0.36)	**73.82** (± 0.06)
mnist	59.63 (± 0.19)	**75.87** (± 0.03)	74.27 (± 0.12)	**74.26** (± 4.38)	61.3 (± 0.69)	**72.62** (± 0.05)	70.83 (± 0.19)	**71.27** (± 0.7)	70.59 (± 0.56)	**94.55** (± 0.36)	88.51 (± 0.49)
musk	39.44 (± 0.57)	**91.95** (± 0.32)	85.69 (± 0.87)	**88.57** (± 5.4)	81.67 (± 7.04)	**71.84** (± 0.34)	65.84 (± 0.57)	**89.39** (± 1.88)	82.04 (± 3.01)	20.17 (± 0.48)	**28.5** (± 0.74)
optdigits	59.58 (± 0.26)	62.26 (± 0.24)	**65.13** (± 0.22)	40.01 (± 10.2)	**58.64** (± 0.91)	54.04 (± 0.21)	**58.99** (± 0.28)	40.87 (± 4.5)	**48.26** (± 3.08)	18.45 (± 0.59)	**42.52** (± 0.89)
pendigits	47.21 (± 0.12)	**88.44** (± 0.2)	87.09 (± 0.22)	**74.87** (± 9.91)	74.08 (± 9.18)	**90.63** (± 0.17)	89.66 (± 0.2)	**81.86**(± 1.48)	79.5 (± 1.5)	14.87 (± 0.18)	**30.16** (± 1.01)
satellite	52.9 (± 0.31)	64.33 (± 0.25)	**66.71** (± 0.3)	60.59 (± 1.77)	**63.44** (± 1.53)	57.57 (± 0.16)	**59.69** (± 0.2)	**76.31** (± 0.7)	76.08 (± 0.46)	61.01 (± 0.29)	**66.71** (± 0.29)
satimage-2	52.8 (± 0.15)	97.03 (± 0.06)	**98.53** (± 0.07)	92.65 (± 0.46)	**96.14** (± 0.32)	94.21 (± 0.03)	**96.41** (± 0.07)	**98.91** (± 0.09)	98.16 (± 0.3)	24.52 (± 0.87)	**41.8** (± 0.72)
shuttle	55.54 (± 0.11)	**99.26** (± 0.0)	**99.26** (± 0.01)	**97.83** (± 0.91)	89.97 (± 1.95)	**98.82** (± 0.01)	98.8 (± 0.01)	**99.57** (± 0.02)	**99.57** (± 0.02)	99.21 (± 0.01)	**99.82** (± 0.02)
smtp	89.77 (± 0.55)	79.64 (± 0.01)	**79.69** (± 0.01)	**84.05** (± 0.57)	83.73 (± 0.4)	87.98 (± 0.02)	**88.0** (± 0.03)	**89.27** (± 0.88)	**89.27** (± 0.88)	43.01 (± 1.57)	**63.18** (± 2.15)
thyroid	75.91 (± 0.79)	88.45 (± 0.35)	**88.54** (± 0.25)	**86.73** (± 3.72)	85.53 (± 3.6)	**94.91** (± 0.14)	94.06 (± 0.1)	**93.67** (± 0.27)	93.11 (± 0.19)	73.59 (± 1.69)	**76.74** (± 0.69)
vowels	89.1 (± 0.67)	56.1 (± 0.32)	**75.39** (± 0.88)	64.47 (± 2.55)	**82.12** (± 0.9)	54.29 (± 0.06)	**75.39** (± 0.91)	66.01 (± 0.57)	**80.76** (± 0.6)	**93.04** (± 0.54)	91.85 (± 0.12)
wilt	64.63 (± 0.72)	33.45 (± 0.11)	**38.4** (± 0.73)	35.79 (± 1.97)	**59.53** (± 1.51)	38.06 (± 0.13)	**42.06** (± 0.48)	42.92 (± 1.11)	**47.27** (± 0.39)	**81.09** (± 0.41)	76.62 (± 0.9)
wine	97.57 (± 1.46)	80.51 (± 1.36)	**93.96** (± 1.66)	82.26 (± 2.29)	**93.96** (± 1.77)	67.12 (± 2.04)	**89.27** (± 2.95)	80.4 (± 3.42)	**93.56** (± 2.15)	**99.94** (± 0.05)	**99.94** (± 0.05)

model by adjusting β.

Table C.7: Comparison with SoftPatch

	Method	Sensory AD		
		MVTec	MPDD	ViSA
	CLIP	86.34	60.02	74.47
PatchCore	Blind	70.02	51.41	19.91
PatchCore	SoftPatch	**90.40**	**67.00**	**86.54**
PatchCore	`EPHAD`	86.45	60.58	62.94

Now, we compare it against SoftPatch, an approach built upon PatchCore (Roth et al., 2022). SoftPatch enhances PatchCore by incorporating traditional anomaly detection (AD) techniques to refine the memory bank, specifically by identifying and re-weighting patches based on their outlier scores during training. While this strategy improves performance, it introduces a strong dependency on the choice of AD method and increases the computational burden of the training pipeline. For a fair comparison, we adopt the Local Outlier Factor (LOF) as the AD method, as it has been empirically found to be the most effective for SoftPatch. As shown in Table C.7, our method, `EPHAD`, achieves competitive results despite being a fully post-hoc approach that requires no modification to the training process. Crucially, while SoftPatch is tailored for memory-bank-based methods, `EPHAD` is inherently model-agnostic and can be seamlessly applied to any combination of a pre-trained model and an evidence function. This versatility highlights `EPHAD`'s broad applicability and practical utility across a diverse range of settings.

Table C.5: Performance od `EPHAD-Ada` 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-Ada	Blind	+ EPHAD-Ada	Blind	+ EPHAD-Ada	Blind	+ EPHAD-Ada	Blind	+ EPHAD-Ada
aloi	54.18 (± 0.31)	51.46 (± 0.05)	**52.42** (± 0.1)	54.06 (± 0.54)	**54.21** (± 0.32)	53.14 (± 0.03)	**53.72** (± 0.11)	54.05 (± 0.21)	**54.27** (± 0.12)	**73.57** (± 0.1)	62.1 (± 0.6)
annthyroid	78.62 (± 1.01)	73.45 (± 0.08)	**77.13** (± 0.43)	62.69 (± 3.33)	**77.91** (± 0.79)	76.05 (± 0.11)	**77.84** (± 0.5)	71.39 (± 0.34)	**75.66** (± 0.69)	72.12 (± 0.57)	**77.98** (± 0.7)
backdoor	67.83 (± 1.69)	**75.06** (± 0.07)	73.35 (± 0.54)	**78.34** (± 1.21)	72.05 (± 1.0)	**83.0** (± 0.09)	78.27 (± 0.46)	51.29 (± 1.29)	**61.25** (± 0.57)	46.65 (± 0.26)	**67.81** (± 1.68)
breastw	97.97 (± 0.14)	**99.46** (± 0.06)	99.29 (± 0.03)	98.65 (± 0.05)	**98.85** (± 0.06)	99.01 (± 0.04)	**99.1** (± 0.06)	**99.46** (± 0.04)	99.17 (± 0.08)	73.39 (± 1.35)	**92.91** (± 0.52)
celeba	66.62 (± 1.04)	**72.09** (± 0.01)	69.51 (± 0.51)	67.51 (± 3.07)	**68.83** (± 1.07)	**73.99** (± 0.01)	70.52 (± 0.49)	40.09 (± 0.83)	**54.01** (± 0.64)	42.97 (± 0.23)	**60.04** (± 0.95)
cover	86.11 (± 1.6)	78.7 (± 0.03)	**84.05** (± 1.15)	75.11 (± 11.37)	**84.6** (± 3.81)	85.34 (± 0.02)	**86.56** (± 0.94)	72.59 (± 1.59)	**80.23** (± 1.78)	22.44 (± 0.1)	**80.33** (± 2.26)
fault	52.02 (± 0.18)	45.69 (± 0.58)	**48.69** (± 0.37)	47.34 (± 0.99)	**51.44** (± 0.25)	47.0 (± 0.4)	**49.3** (± 0.31)	**58.08** (± 0.94)	55.16 (± 0.61)	**64.41** (± 1.35)	52.81 (± 0.11)
fraud	94.87 (± 0.11)	94.39 (± 0.0)	**94.81** (± 0.07)	89.98 (± 0.97)	**94.84** (± 0.11)	93.86 (± 0.0)	**94.63** (± 0.09)	92.95 (± 0.29)	**94.32** (± 0.09)	33.92 (± 0.34)	**94.86** (± 0.1)
glass	77.6 (± 1.77)	76.11 (± 0.77)	**77.78** (± 1.44)	64.52 (± 6.87)	**76.33** (± 2.04)	67.65 (± 0.44)	**74.08** (± 1.16)	**78.5** (± 1.47)	77.96 (± 1.6)	71.79 (± 1.08)	**83.73** (± 1.51)
http	99.99 (± 0.0)	94.91 (± 0.01)	**99.45** (± 0.03)	99.17 (± 0.08)	**99.52** (± 0.01)	92.35 (± 0.02)	**99.25** (± 0.04)	96.82 (± 0.37)	**99.37** (± 0.01)	17.85 (± 2.03)	**99.98** (± 0.0)
ionosphere	81.8 (± 0.28)	79.42 (± 1.03)	**81.84** (± 0.61)	83.09 (± 0.57)	**83.72** (± 0.5)	73.04 (± 0.84)	**78.3** (± 0.51)	**89.58** (± 1.57)	86.62 (± 0.87)	**94.64** (± 0.52)	89.88 (± 0.6)
letter	61.76 (± 0.26)	56.71 (± 0.12)	**59.61** (± 0.28)	50.51 (± 2.54)	**56.79** (± 1.64)	56.41 (± 0.29)	**59.37** (± 0.28)	59.84 (± 0.64)	**60.93** (± 0.4)	**85.74** (± 0.54)	79.09 (± 0.62)
lymphography	99.92 (± 0.07)	99.52 (± 0.22)	**99.76** (± 0.19)	98.57 (± 0.74)	**99.92** (± 0.07)	99.6 (± 0.23)	**99.76** (± 0.19)	**99.76**(± 0.19)	**99.76** (± 0.19)	98.57 (± 0.59)	**99.84** (± 0.13)
mammography	83.98 (± 0.32)	**89.29** (± 0.05)	87.06 (± 0.13)	**87.23** (± 0.95)	84.27 (± 0.27)	**89.38** (± 0.06)	87.51 (± 0.12)	80.44 (± 0.29)	**82.57** (± 0.09)	69.7 (± 0.36)	**83.91** (± 0.31)
mnist	75.5 (± 0.08)	75.87 (± 0.03)	**76.47** (± 0.02)	74.26 (± 4.38)	**76.04** (± 0.24)	72.62 (± 0.05)	**74.8** (± 0.04)	71.27 (± 0.7)	**73.83** (± 0.38)	**94.55** (± 0.36)	90.56 (± 0.41)
musk	99.29 (± 0.33)	91.95 (± 0.32)	**97.37** (± 0.45)	88.57 (± 5.4)	**97.34** (± 1.28)	71.84 (± 0.34)	**90.7** (± 1.37)	89.39 (± 1.88)	**96.77** (± 0.15)	20.17 (± 0.48)	**77.73** (± 2.96)
optdigits	58.65 (± 3.55)	**62.26** (± 0.24)	60.85 (± 1.93)	40.01 (± 10.2)	**58.15** (± 3.71)	54.04 (± 0.21)	**57.14** (± 2.15)	40.87 (± 4.5)	**50.42** (± 1.43)	18.45 (± 0.59)	**38.14** (± 3.6)
pendigits	92.04 (± 0.23)	88.44 (± 0.2)	**90.69** (± 0.26)	74.87 (± 9.91)	**90.09** (± 2.25)	90.63 (± 0.17)	**92.11** (± 0.18)	81.86 (± 1.48)	**88.36** (± 0.42)	14.87 (± 0.18)	**79.82** (± 0.64)
satellite	64.44 (± 0.57)	64.33 (± 0.25)	**64.72** (± 0.3)	60.59 (± 1.77)	**62.43** (± 0.92)	57.57 (± 0.16)	**61.03** (± 0.21)	**76.31** (± 0.7)	69.99 (± 0.41)	61.01 (± 0.29)	**72.04** (± 0.39)
satimage-2	99.43 (± 0.07)	97.03 (± 0.06)	**98.75** (± 0.06)	92.65 (± 0.46)	**98.19** (± 0.24)	94.21 (± 0.03)	**97.87** (± 0.08)	98.91 (± 0.09)	**99.31** (± 0.07)	24.52 (± 0.87)	**95.39** (± 0.14)
shuttle	98.97 (± 0.08)	99.26 (± 0.0)	**99.42** (± 0.05)	97.83 (± 0.91)	**98.98** (± 0.09)	98.82 (± 0.01)	**99.08** (± 0.06)	**99.57** (± 0.02)	99.56 (± 0.02)	99.21 (± 0.01)	**99.79** (± 0.03)
smtp	90.95 (± 0.28)	79.64 (± 0.01)	**88.06** (± 0.17)	84.05 (± 0.57)	**91.05** (± 0.32)	87.98 (± 0.02)	**90.21** (± 0.17)	89.27 (± 0.88)	**90.49** (± 0.4)	43.01 (± 1.57)	**90.9** (± 0.22)
thyroid	96.65 (± 0.26)	88.45 (± 0.35)	**94.35** (± 0.26)	86.73 (± 3.72)	**95.8** (± 0.48)	94.91 (± 0.14)	**96.2** (± 0.21)	93.67 (± 0.27)	**95.5** (± 0.14)	73.59 (± 1.69)	**95.67** (± 0.1)
vowels	72.73 (± 0.8)	56.1 (± 0.32)	**65.07** (± 0.52)	64.47 (± 2.55)	**70.74** (± 1.46)	54.29 (± 0.06)	**64.37** (± 0.67)	66.01 (± 0.57)	**69.74** (± 0.81)	**93.04** (± 0.54)	90.01 (± 0.24)
wilt	42.57 (± 1.63)	33.45 (± 0.11)	**37.63** (± 0.95)	35.79 (± 1.97)	**41.82** (± 1.71)	38.06 (± 0.13)	**39.62** (± 0.84)	**42.92** (± 1.11)	42.76 (± 1.28)	**81.09** (± 0.41)	61.95 (± 2.3)
wine	58.98 (± 0.68)	**80.51** (± 1.36)	74.58 (± 1.48)	**82.26** (± 2.29)	73.34 (± 2.89)	**67.12** (± 2.04)	63.73 (± 0.96)	**80.4** (± 3.42)	73.62 (± 3.11)	**99.94** (± 0.05)	97.29 (± 0.64)

C.3.3 Ablation on ϵ and β

Extended ablation on ϵ and β can be found in Figure C.1, C.2. We can make similar conclusions as discussed above in Section 5.5.4.

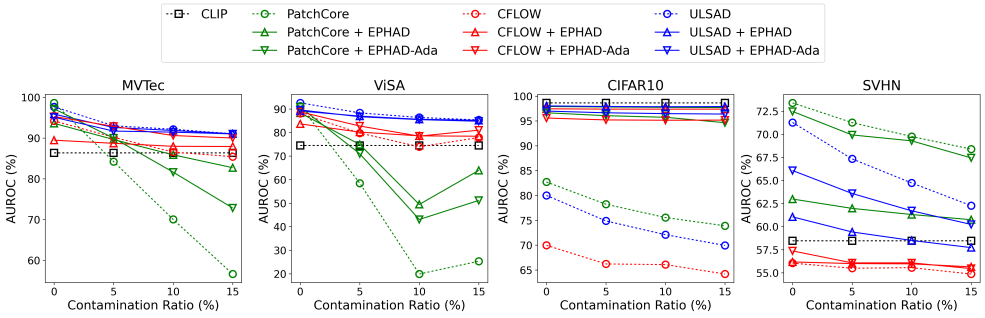


Figure C.1: Ablation on ϵ.

C.3.4 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

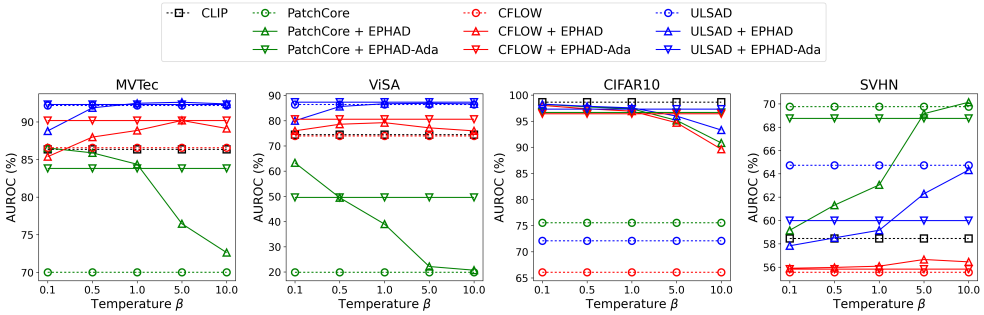


Figure C.2: Ablation on β.

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 C.3, we analyse the impact of varying the proportion of anomalies in the test set, which exhibits consistent improvements across all tested settings.

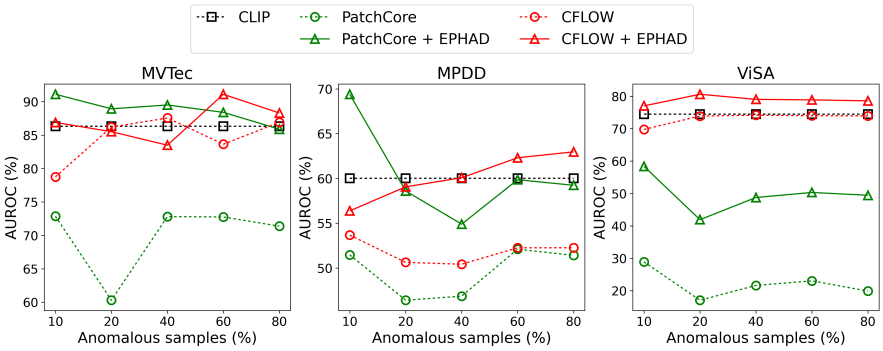


Figure C.3: Ablation on varying proportion of anomalies in the test set.

APPENDIX D

Detecting Anomalies in Irregular Image Sequences

D.1. Network architecture

`ForecastAD` is implemented using the PyTorch framework (Paszke et al., 2019). The architecture of the encoder ϕ_e and ϕ_d is presented in Tables D.1 and D.2, respectively. The input dimension of the encoder network is $(3 \times 256 \times 256)$. The period of the sinusoidal encoding is 1000.

Table D.1: Encoder architecture

Layer
Conv2d-1(3, 32, 5, padding=2, stride=2)
BatchNorm2d-1(32, eps=1e-04)
MaxPool2d-1(2,2)
Conv2d-2(32, 64, 5, padding=2, stride=2)
BatchNorm2d-2(64, eps=1e-04)
MaxPool2d-2(2,2)
Conv2d-3(64, 128, 5, padding=2, stride=2)
BatchNorm2d-3(128, eps=1e-04)
MaxPool2d-3(2,2)
Conv2d-4(128, 128, 5, padding=2, stride=2)
BatchNorm2d-4(128, eps=1e-04)
MaxPool2d-4(2,2)
Linear-1(128, 128, padding=2, stride=2)
BatchNorm2d-1(128, eps=1e-04)

D.2. Data generation

We generate a public dataset mirroring the properties of our private dataset using a variational autoencoder (VAE). For daily sequence generation, we condition the VAE on the time of day, class (Positive, Negative, Unlabelled), and phase (Start, Middle, End). A standard convolutional neural network (CNN) with three convolution blocks with an increasing number of feature maps is used to encode images, and a multilayer perceptron (MLP) is used to encode conditioning variables. Then, a two-layer LSTM network produces the context embedding. Finally, a deconvolutional CNN, mirroring the encoder's architecture, reconstructs the original image from a latent vector sampled from the

Table D.2: Decoder architecture

Layer
ConvTranspose2d-1(128, 64, 5, padding=2)
BatchNorm2d-1(64, eps=1e-04)
ConvTranspose2d-2(64, 64, 5, padding=2)
BatchNorm2d-2(64, eps=1e-04)
ConvTranspose2d-3(64, 32, 5, padding=2)
BatchNorm2d-3(32, eps=1e-04)
ConvTranspose2d-4(32, 3, 5, padding=2)

latent distribution. We use a multivariate Gaussian distribution with a diagonal covariance matrix as our latent distribution. We train the VAE over 20 epochs using Adam optimizer, a learning rate of 1e-4, and a batch size of 16. Figure D.1 allows us to compare the generated images with the original images from the private dataset. From the figure, we can observe that the VAE can generate normal images and a diverse set of anomalies which are visually similar to images in the original dataset.

D.3. Sensitivity to data labelling

Some images in the deployment set are wrongly labelled as normal. Such inconsistencies result in a distribution shift between the labelled training and deployment sets, leading to significant degradation of model performance. Thus, we cleaned the deployment set by calculating the distance of each labelled normal image x_i in the deployment set to the images in the training set $\mathcal{D}_N$. For this, we first obtain the context embedding c_i for each image x_i using the context encoder of `ForecastAD`. We then compute the distance between context embeddings as $\xi_i = \min_{j=1,\ldots,|\mathcal{D}_N|} \|c_i - c_j\|_2$. Images from the deployment set for which the distance d_i exceeds a predetermined threshold are removed. Figure D.2 shows the UMAP projection (McInnes et al., 2018) of the context embeddings corresponding to the samples in the training set $\mathcal{D}_N$ and the samples from the deployment set that are removed during the cleaning process.

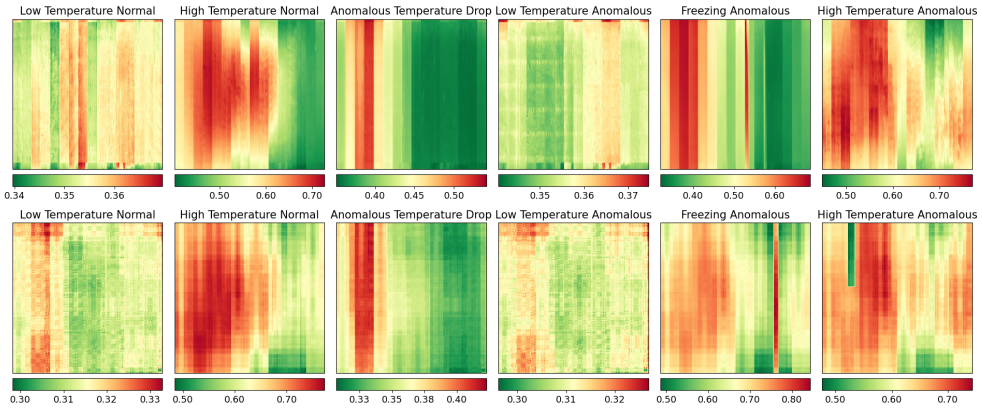


Figure D.1: Examples of different types of images in Original (top) and simulated (bottom) dataset

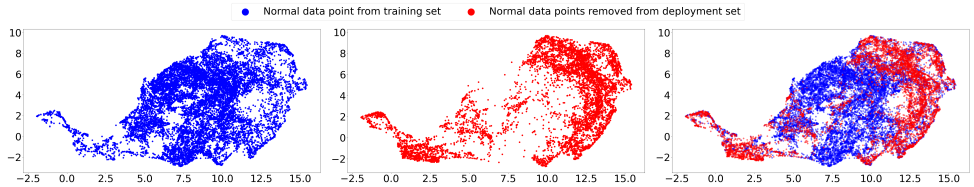


Figure D.2: UMAP plot of training set normal images (left), removed deployment normal images (middle) and the both (right).

D.4. Additional results

Simple baselines. We also consider four simple baselines which compute thresholds based on the features extracted from the samples in the validation set to detect anomalies. The extracted features can be seen as the anomaly scores. In this study, we consider the following features:

- Time of day: the time when the image is recorded.
- Negative mean: negative of the average pixel value of the image.
- Negative max: negative of the maximum pixel value of the image.
- Negative Std: negative of the standard deviation of the pixel value in the image.

Threshold selection. Given the anomaly scores assigned to the labelled

Table D.3: Extended Anomaly Detection Performance. Style: **best** and second best

Train Setting	Model	AUROC (%)			Accuracy (%)						F1-score (%)					
		[Tr#1]	[Tr#2]	[Tr#3]	[Tr#1]		[Tr#2]		[Tr#3]		[Tr#1]		[Tr#2]		[Tr#3]	
					λ_f	λ_g	λ_f	λ_g	λ_f	λ_g	λ_f	λ_g	λ_f	λ_g	λ_f	λ_g
[Tr#1]	Time of day	86.99 (± 0.00)	41.44 (± 0.00)	79.97 (± 0.00)	83.03 (± 0.00)	82.86 (± 0.00)	61.60 (± 0.00)	61.60 (± 0.00)	79.55 (± 0.00)	79.41 (± 0.00)	85.34 (± 0.00)	85.17 (± 0.00)	0.00 (± 0.00)	0.00 (± 0.00)	80.20 (± 0.00)	80.02 (± 0.00)
	Negative Mean	81.67 (± 0.00)	46.73 (± 0.00)	71.51 (± 0.00)	70.49 (± 0.00)	76.43 (± 0.00)	10.89 (± 0.00)	13.47 (± 0.00)	60.83 (± 0.00)	66.22 (± 0.00)	74.40 (± 0.00)	76.91 (± 0.00)	14.33 (± 0.00)	12.72 (± 0.00)	65.46 (± 0.00)	66.76 (± 0.00)
	Negative STD	79.31 (± 0.00)	40.76 (± 0.00)	70.35 (± 0.00)	73.60 (± 0.00)	73.60 (± 0.00)	34.10 (± 0.00)	34.10 (± 0.00)	67.19 (± 0.00)	67.19 (± 0.00)	72.49 (± 0.00)	72.49 (± 0.00)	14.18 (± 0.00)	14.18 (± 0.00)	64.66 (± 0.00)	64.66 (± 0.00)
	Negative Max	77.61 (± 0.00)	44.85 (± 0.00)	68.64 (± 0.00)	75.10 (± 0.00)	75.37 (± 0.00)	14.33 (± 0.00)	14.90 (± 0.00)	65.24 (± 0.00)	65.57 (± 0.00)	75.29 (± 0.00)	75.47 (± 0.00)	11.89 (± 0.00)	11.87 (± 0.00)	65.31 (± 0.00)	65.49 (± 0.00)
	Autoencoder	98.05 (± 0.74)	46.43 (± 1.61)	87.87 (± 0.26)	94.50 (± 1.22)	94.49 (± 1.22)	36.96 (± 3.72)	37.19 (± 3.68)	85.17 (± 0.46)	85.20 (± 0.47)	95.27 (± 1.06)	95.26 (± 1.06)	12.58 (± 0.64)	12.62 (± 0.63)	86.45 (± 0.54)	86.46 (± 0.55)
	CFlow (Gudovskiy et al., 2022)	94.68 (± 1.26)	39.99 (± 2.33)	82.91 (± 1.08)	87.07 (± 1.67)	86.98 (± 1.77)	31.69 (± 1.70)	32.84 (± 1.40)	78.09 (± 1.45)	78.20 (± 1.47)	88.50 (± 1.50)	88.31 (± 1.64)	12.75 (± 0.30)	12.55 (± 0.43)	79.53 (± 1.37)	79.42 (± 1.49)
	Deep SVDD (one-class) (Ruff et al., 2018)	52.76 (± 8.71)	49.22 (± 2.66)	51.85 (± 6.15)	61.35 (± 1.95)	58.34 (± 5.62)	20.00 (± 7.87)	54.10 (± 8.04)	54.65 (± 2.86)	57.65 (± 3.52)	70.79 (± 1.77)	54.97 (± 6.57)	13.79 (± 0.87)	13.33 (± 1.53)	64.12 (± 1.60)	50.54 (± 5.49)
	Deep SVDD (soft-boundary) (Ruff et al., 2018)	30.22 (± 6.77)	49.68 (± 4.26)	35.45 (± 3.84)	58.18 (± 0.04)	43.04 (± 5.65)	7.79 (± 0.06)	68.42 (± 9.38)	50.01 (± 0.03)	47.16 (± 3.99)	73.56 (± 0.03)	33.14 (± 7.34)	14.37 (± 0.01)	11.66 (± 1.73)	66.67 (± 0.02)	31.14 (± 6.59)
	DRAEM (Zavrtanik et al., 2021)	97.70 (± 0.77)	40.48 (± 2.15)	87.38 (± 0.61)	91.70 (± 1.08)	91.81 (± 1.05)	30.49 (± 3.85)	30.95 (± 3.68)	81.78 (± 1.39)	81.94 (± 1.35)	92.97 (± 0.83)	93.04 (± 0.81)	12.14 (± 0.92)	11.94 (± 1.07)	83.66 (± 0.89)	83.75 (± 0.87)
	FastFlow (Yu et al., 2021)	99.83 (± 0.03)	47.32 (± 0.29)	91.36 (± 0.25)	97.39 (± 0.29)	97.38 (± 0.29)	42.12 (± 1.27)	42.18 (± 1.29)	88.43 (± 0.38)	88.43 (± 0.38)	97.72 (± 0.26)	97.71 (± 0.26)	13.22 (± 0.26)	13.24 (± 0.25)	89.17 (± 0.35)	89.17 (± 0.35)
	PaDiM (Defard et al., 2021)	99.85 (± 0.02)	49.86 (± 0.47)	91.23 (± 0.10)	96.92 (± 0.72)	96.45 (± 0.58)	43.44 (± 1.91)	44.76 (± 1.28)	88.24 (± 0.34)	88.07 (± 0.33)	97.28 (± 0.65)	96.86 (± 0.53)	13.76 (± 0.29)	13.46 (± 0.10)	88.92 (± 0.43)	88.66 (± 0.39)
	PatchCore (Roth et al., 2022)	99.23 (± 0.08)	50.58 (± 0.37)	89.04 (± 0.30)	95.50 (± 0.25)	95.52 (± 0.25)	31.29 (± 1.54)	31.46 (± 1.65)	85.08 (± 0.41)	85.13 (± 0.43)	96.21 (± 0.22)	96.23 (± 0.21)	15.16 (± 0.16)	15.31 (± 0.19)	86.77 (± 0.34)	86.81 (± 0.35)
	Reverse Distillation (Deng and Li, 2022)	93.88 (± 1.13)	41.31 (± 2.19)	84.61 (± 1.54)	87.01 (± 1.68)	86.66 (± 1.52)	35.01 (± 2.90)	39.03 (± 2.51)	78.58 (± 1.64)	78.93 (± 1.58)	89.27 (± 1.44)	88.66 (± 1.31)	12.91 (± 0.36)	12.81 (± 0.74)	81.17 (± 1.43)	80.86 (± 1.40)
	ForecastAD	99.86 (± 0.05)	46.22 (± 1.06)	89.89 (± 0.35)	97.73 (± 0.34)	97.65 (± 0.27)	36.10 (± 1.19)	36.62 (± 1.35)	87.73 (± 0.29)	87.75 (± 0.26)	98.04 (± 0.29)	97.97 (± 0.24)	14.02 (± 0.50)	14.13 (± 0.50)	88.76 (± 0.25)	88.75 (± 0.22)
[Tr#2]	Time of day	86.99 (± 0.00)	41.44 (± 0.00)	79.97 (± 0.00)	83.03 (± 0.00)	82.86 (± 0.00)	61.60 (± 0.00)	61.60 (± 0.00)	79.55 (± 0.00)	79.41 (± 0.00)	85.34 (± 0.00)	85.17 (± 0.00)	0.00 (± 0.00)	0.00 (± 0.00)	80.20 (± 0.00)	80.02 (± 0.00)
	Negative Mean	81.67 (± 0.00)	46.73 (± 0.00)	71.51 (± 0.00)	71.16 (± 0.00)	77.43 (± 0.00)	9.46 (± 0.00)	43.27 (± 0.00)	61.15 (± 0.00)	71.89 (± 0.00)	76.17 (± 0.00)	75.96 (± 0.00)	14.59 (± 0.00)	13.91 (± 0.00)	67.24 (± 0.00)	68.54 (± 0.00)
	Negative STD	79.31 (± 0.00)	40.76 (± 0.00)	70.35 (± 0.00)	66.00 (± 0.00)	73.60 (± 0.00)	9.17 (± 0.00)	45.56 (± 0.00)	56.78 (± 0.00)	69.05 (± 0.00)	74.45 (± 0.00)	70.98 (± 0.00)	12.67 (± 0.00)	12.84 (± 0.00)	66.33 (± 0.00)	64.16 (± 0.00)
	Negative Max	77.61 (± 0.00)	44.85 (± 0.00)	68.64 (± 0.00)	76.37 (± 0.00)	74.71 (± 0.00)	31.81 (± 0.00)	49.86 (± 0.00)	71.14 (± 0.00)	70.68 (± 0.00)	77.19 (± 0.00)	72.26 (± 0.00)	14.13 (± 0.00)	14.63 (± 0.00)	68.62 (± 0.00)	65.87 (± 0.00)
	Autoencoder	96.67 (± 0.71)	45.92 (± 2.47)	85.45 (± 1.18)	93.09 (± 1.21)	93.16 (± 1.40)	26.02 (± 3.28)	28.83 (± 2.91)	82.21 (± 1.27)	82.72 (± 1.19)	94.15 (± 1.02)	94.10 (± 1.22)	13.99 (± 0.36)	13.69 (± 0.52)	84.26 (± 1.01)	84.40 (± 1.22)
	CFlow (Gudovskiy et al., 2022)	84.91 (± 2.72)	42.90 (± 2.71)	77.38 (± 2.98)	76.78 (± 3.03)	77.15 (± 2.29)	29.86 (± 2.77)	39.77 (± 4.33)	69.17 (± 2.77)	71.09 (± 2.36)	81.93 (± 2.03)	80.16 (± 1.93)	12.83 (± 0.93)	12.35 (± 1.22)	74.43 (± 1.83)	73.19 (± 1.94)
	Deep SVDD (one-class) (Ruff et al., 2018)	45.93 (± 5.07)	52.24 (± 2.24)	46.56 (± 4.80)	58.11 (± 0.07)	54.04 (± 4.88)	7.85 (± 0.07)	61.60 (± 2.26)	49.96 (± 0.06)	55.27 (± 4.34)	73.51 (± 0.06)	47.40 (± 4.76)	14.38 (± 0.01)	12.36 (± 1.67)	66.63 (± 0.06)	44.36 (± 4.31)
	Deep SVDD (soft-boundary) (Ruff et al., 2018)	29.21 (± 15.05)	52.48 (± 2.83)	35.44 (± 11.72)	62.34 (± 4.10)	38.41 (± 11.14)	17.65 (± 9.91)	65.04 (± 4.93)	55.09 (± 5.05)	42.73 (± 9.28)	74.76 (± 1.16)	46.42 (± 8.16)	14.06 (± 0.30)	14.09 (± 0.89)	68.12 (± 1.42)	44.14 (± 7.40)
	DRAEM (Zavrtanik et al., 2021)	93.52 (± 0.52)	40.51 (± 1.33)	85.71 (± 0.78)	85.00 (± 1.41)	87.14 (± 0.98)	28.54 (± 4.62)	35.70 (± 3.60)	75.85 (± 1.66)	78.80 (± 0.98)	88.07 (± 1.00)	88.98 (± 0.88)	11.84 (± 0.36)	11.40 (± 0.36)	79.62 (± 1.04)	80.67 (± 0.85)
	FastFlow (Yu et al., 2021)	92.38 (± 0.72)	52.51 (± 1.09)	89.92 (± 0.68)	88.04 (± 0.78)	87.83 (± 0.87)	56.62 (± 3.06)	59.03 (± 2.49)	82.95 (± 1.02)	83.16 (± 1.05)	90.15 (± 0.61)	89.86 (± 0.68)	13.39 (± 0.46)	13.18 (± 0.38)	84.49 (± 0.78)	84.45 (± 0.84)
	PaDiM (Defard et al., 2021)	95.99 (± 0.37)	58.14 (± 1.00)	92.28 (± 0.32)	88.77 (± 0.26)	87.58 (± 0.26)	46.25 (± 3.26)	65.73 (± 1.90)	81.88 (± 0.48)	84.03 (± 0.38)	90.92 (± 0.24)	88.87 (± 0.27)	14.81 (± 0.39)	17.14 (± 1.42)	84.09 (± 0.21)	84.07 (± 0.27)
	PatchCore (Roth et al., 2022)	96.78 (± 0.57)	60.15 (± 1.82)	91.38 (± 0.42)	90.67 (± 1.24)	90.34 (± 1.30)	55.82 (± 4.20)	56.85 (± 4.14)	85.02 (± 0.61)	84.91 (± 0.63)	91.84 (± 1.20)	91.49 (± 1.26)	17.86 (± 0.88)	17.86 (± 0.97)	85.71 (± 0.79)	85.49 (± 0.83)
	Reverse Distillation (Deng and Li, 2022)	87.19 (± 0.99)	57.22 (± 5.77)	84.04 (± 1.64)	84.39 (± 1.15)	82.48 (± 1.14)	44.41 (± 6.44)	57.02 (± 5.75)	77.91 (± 1.16)	78.36 (± 0.97)	87.32 (± 0.91)	84.92 (± 1.12)	14.91 (± 1.50)	15.50 (± 1.41)	80.60 (± 0.71)	79.53 (± 0.91)
	ForecastAD	94.78 (± 1.09)	85.81 (± 1.23)	92.53 (± 0.81)	88.82 (± 1.33)	88.75 (± 1.28)	76.68 (± 2.13)	78.40 (± 1.36)	86.85 (± 0.89)	87.07 (± 0.98)	90.03 (± 1.24)	89.88 (± 1.19)	35.40 (± 1.59)	36.29 (± 1.40)	86.83 (± 1.01)	86.89 (± 1.06)

validation samples, we explore two different threshold selection approaches:

- **F1-score**. We compute F1 scores by considering various thresholds. Given the F1 scores, we select the optimal threshold λ_f as the one that corresponds to the maximum F1 score.
- **G-Mean**. Similar to λ_f, we calculate the G-Mean value by applying multiple thresholds The optimal threshold λ_g corresponds to the threshold where we obtain the highest G-Mean value. G-Mean is the geometric mean of specificity and recall.

$$\text{G-Mean} = \sqrt{\text{Specificity} \cdot \text{Recall}} = \sqrt{(1 - FPR) \cdot TPR}$$

Based on the selected threshold, Accuracy and F1-score are reported for **[Tr#1]** and **[Tr#2]** over three test setups in Table D.3.

APPENDIX E

Risk-Based Thresholding for Reliable Anomaly Detection

E.1. Ablation study

Impact of image reduction. In our methodology, we resized the images 256×256 pixels to 64×64 to stabilize training. To assess information loss, we evaluate it on a subset of 1000 samples from our dataset. Specifically, we compute the Mean Squared Error (MSE) relative to the original 256×256 images, along with the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR). Table E.1 shows that despite the MSE increasing significantly at lower resolutions, the SSIM remains strong, implying that even low-resolution images retain recognizable structural features. We select 64×64 as an optimal balance between computational efficiency and image quality.

Table E.1: Loss of information after reducing the original image (256×256).

Image Size	MSE	SSIM	PSNR
128×128	26.0474	1.0000	76.9805
64×64	86.8460	0.9999	71.8129
32×32	209.0905	0.9998	68.0073
16×16	619.8050	0.9994	63.3100

Importance of sequence length. The performance of `DensityAD` for different sequence lengths K is shown in Table E.2. All tested values of K yield strong results. We choose $K = 30$ as the baseline for our experiments, as it provides an optimal balance between capturing sufficient information and avoiding unnecessary complexity.

Table E.2: Ablation of K. Style: best in **bold**.

CSP	K	AUROC (%)	AUPR (%)
A	1	92.46 ± 1.01	91.41 ± 1.42
	10	93.99 ± 0.36	93.54 ± 0.51
	20	93.62 ± 0.16	93.2 ± 0.24
	30	92.89 ± 0.41	91.72 ± 0.65
	40	**94.62 ± 0.48**	**94.3 ± 0.6**
B	1	91.43 ± 0.31	90.41 ± 0.34
	10	91.08 ± 0.28	90.52 ± 0.17
	20	91.3 ± 0.28	90.39 ± 0.62
	30	**91.49 ± 0.1**	**90.52 ± 0.24**
	40	90.36 ± 0.56	89.47 ± 1.05

Importance of time-embedding. Table E.3 provides insights into the necessity of using both τ and γ during training, or whether using only one of them would suffice. It is evident that utilizing only τ improves performance, and therefore we selected this approach.

Table E.3: Ablation of time-embeddings. Style: best in **bold**.

CSP	τ	γ	AUROC (%)	AUPR (%)
A	✓	-	**94.25 ± 0.2**	**93.88 ± 0.48**
	-	✓	93.05 ± 0.58	92.44 ± 0.64
	✓	✓	94.0 ± 0.15	93.69 ± 0.22
B	✓	-	**91.93 ± 0.52**	**90.66 ± 0.46**
	-	✓	90.54 ± 0.63	90.05 ± 0.72
	✓	✓	90.95 ± 0.13	90.58 ± 0.23

Effect of α on controlled risk. Figure E.1 shows the FPR and F1 control for various α values across the proposed threshold selection methods, with $\delta = 0.1$. The results clearly demonstrate that the proposed methods effectively control risk regardless of the α value, whereas the existing methods do not.

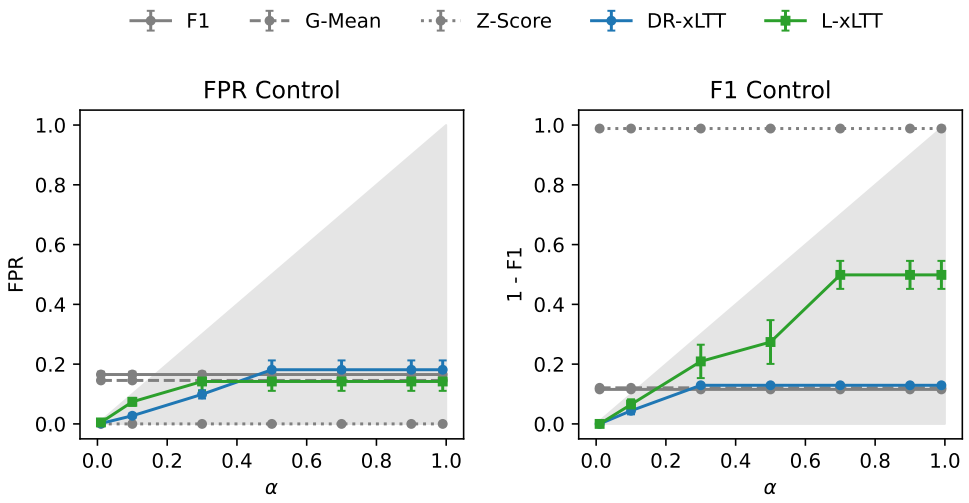


Figure E.1: Controlled risks (i.e., FPR and 1−F1) across multiple α values, with $\delta = 0.1$.

E.2. Details on dataset simulation

In this study, having access to multiple CSP plants allowed us to extend the simulated dataset proposed by Patra et al. (2024) to include thermal images from various CSP facilities. This section presents details on this simulated dataset. Each folder corresponds to a CSP plant (A or B), with individual samples stored as pickle files named after their respective timestamps. Each file contains a thermal image, its label, and the associated setting (i.e., *Starting* (S), where the mean temperature of the solar receiver begins to rise, *Middle* (M), where it reaches and maintains its peak, or *Ending* (E), where it declines as the day concludes.).

E.2.1 Description and performances.

The resulting dataset contains 10001 samples for CSP A and 10001 samples for CSP B, totalling 20002 samples. The distribution of samples across the *Starting*, *Middle*, and *Ending* phases is illustrated in Figure E.2, while examples of simulated samples for each CSP plant are shown in Figure E.3. The samples closely resemble the real dataset for both CSP plants. However, for anonymity purposes, the data has been resized to 64×64, normalised between 0 and 1, and anonymised timestamps are used. Additionally, the generation model may still produce some blurry samples. These limitations should be considered when modelling.

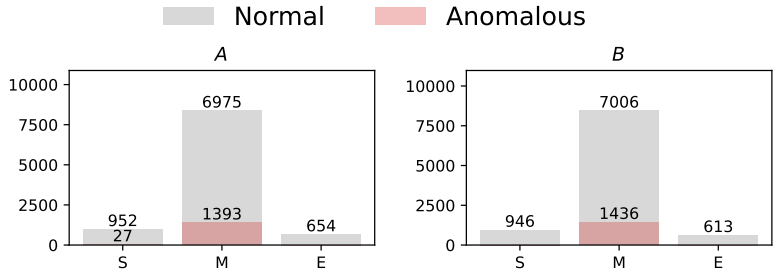


Figure E.2: Spread of normal and anomalous samples across each phase for the simulated data of each CSP plant.

The results of applying `DensityAD` to this simulated dataset are presented in Figure E.4. The dataset was divided into training, validation, and test samples. Due to the limited number of samples, we reduced the number of blocks to 3

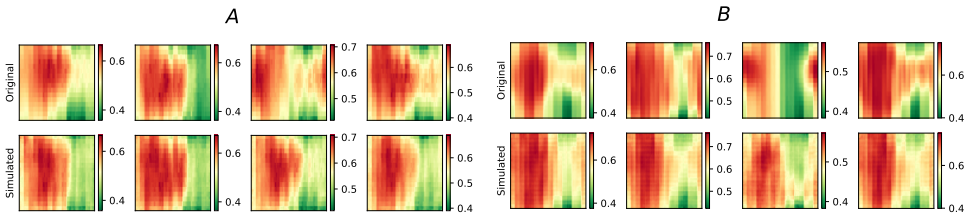


Figure E.3: Example of simulated thermal images for the two CSP plants (i.e. A and B), along with the original image from which it has been sampled.

to mitigate training instability. The method achieves strong performance on CSP B, while performance drops on CSP A. This decline can be attributed to the reduced number of blocks and the greater diversity of samples in CSP A.

Table E.4: AUROC and AUPR of `DensityAD` on the simulated dataset.

A		B	
AUROC (%)	AUPR (%)	AUROC (%)	AUPR (%)
75.13 ± 1.23	74.45 ± 0.66	88.36 ± 0.14	91.10 ± 0.09

List of Figures

1.1 (Left) Illustration of a Concentrated Solar Power plant. (Right) Zoomed-in image of a thermal solar receiver with the thermal image captured using infrared cameras on its surface[1]. Our goal is to detect anomalous behaviours using such thermal images. . 3

2.1 Illustration of different types of anomalies. 16
2.2 Illustration of common data settings in anomaly detection. . . . 21

3.1 Types of anomalies for the categories "breakfast box" and "push-pin" in the MVTecLOCO dataset. Two normal samples (Left) along with structural (Middle) and logical anomalies (Right) where the anomalous regions are highlighted in blue and red, respectively. 39
3.2 Overview of the end-to-end architecture of `ULSAD`. 42
3.3 Overview of the local and global branch of `ULSAD`. 44
3.4 (First) Example image belonging to the category "breakfast box" in the MVTecLOCO dataset. (Rest) Visualisation of selected four self-attention maps computed using the intermediate features from a pre-trained Wide-Resnet50-2 model as $\boldsymbol{A} = \boldsymbol{ZW}$ where $\boldsymbol{W}$ is computed using Eq. 3.3. 46
3.5 Example of anomaly maps obtained from global and local branches along with the combined map. 54
3.6 Ablation study of α and β for normalization of anomaly maps with selected value highlighted. 55
3.7 Ablation study of the backbone network 55

4.1 AUC mean and std over 20 trials under various conditions. . . . 73
4.2 AUC mean and std over 20 trials with $\gamma_l = 0.05$ and $\pi_n = 0.1$. 74

5.1 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 the samples available at test-time. 83
5.2 Ablation on parameters. 92

6.1 Visualisation of different properties of the data 98
6.2 Examples of different types of normal and anomalous images . . 100
6.3 Illustration of the end-to-end architecture of `ForecastAD`. The model is trained to forecast the next image in the sequence given a context embedding c_i of K prior data points obtained using a sequence-to-sequence model. For $(x_{i-k}, t_{i-k}, y_{i-k}) \in \mathcal{D}$ in the context, we sum the embeddings of inter-arrival time τ_{i-k} and interval since the start of operation δ_{i-k} and concatenate it with the image embedding. The anomaly score $s(x_i, t_i)$ is computed as the difference between the forecasted and original image. . . 102
6.4 Dataset split for two different training setups and the test set . 104
6.5 Ablation of different architectures 111
6.6 Examples of anomaly maps for anomalous (top) and normal (bottom) images. 112
6.7 Data split for simulated dataset 113

7.1 Illustration of AD with abstention under high (left) and low (right) overlap in anomaly score distributions of normal and anomalous samples. 120
7.2 Well-reconstructed anomalous images with empirically low density. 123
7.3 Empirical score distributions of normal and anomalous test samples from CSP A for our proposed score functions and the one used by `ForecastAD`, with the overlapping area (OA) between both distributions in the top right corner. 126
7.4 Risk control over FPR (top row) and F1-score (bottom row) for existing and proposed methods. The risk is FPR (top row) and 1−F1 (bottom row). 127
7.5 FPR control for the proposed approaches in a deployment setting over multiple months, for the two CSP plants. 128

7.6 AUROC and AUPR for the two CSP plants over multiple months using three different training settings (i.e. training on A, B, and $A + B$). 128
7.7 Mean daily temperature for the original and simulated datasets. The shaded area represents the 90% temperature interval. . . . 129

A.1 Visualisation of anomaly maps on anomalous images from MVTecLOCO dataset. 172

B.1 AUC mean and std over 20 trials at various a for the datasets with $\gamma_l = 0.05$ and $\pi_n = 0.01$. 179
B.2 AUC mean and std over 20 trials at various a for the datasets with $\gamma_l = 0.05$ and $\pi_n = 0.05$. 182
B.3 AUC mean and std over 20 trials at various a for the datasets with $\gamma_l = 0.05$ and $\pi_n = 0.2$. 182
B.4 Representative ROC curves for different datasets with $\gamma = 0.05$ and $\pi_n = 0.1$. 183

C.1 Ablation on ϵ. 193
C.2 Ablation on β. 194
C.3 Ablation on varying proportion of anomalies in the test set. . . 194

D.1 Examples of different types of images in Original (top) and simulated (bottom) dataset 198
D.2 UMAP plot of training set normal images (left), removed deployment normal images (middle) and the both (right). 198

E.1 Controlled risks (i.e., FPR and 1−F1) across multiple α values, with $\delta = 0.1$. 202
E.2 Spread of normal and anomalous samples across each phase for the simulated data of each CSP plant. 203
E.3 Example of simulated thermal images for the two CSP plants (i.e. A and B), along with the original image from which it has been sampled. 204

List of Tables

2.1 Confusion matrix 18
2.2 Summary of outcomes when testing m null hypotheses. 30

3.1 Average Detection Performance in AUROC (%). Style: **best** and second best 52
3.2 Average Segmentation Performance in AUROC (%) and AUPRO (%). Style: **best** and second best 53
3.3 Ablation of the main components of `ULSAD`. 53
3.4 Memory and computational efficiency on MVTecLOCO dataset. 56

4.1 Examples of loss functions satisfying (4.16) 66
4.2 Mean (and SE $\times 10^2$) of the AUC over 30 trials. The best means are highlighted in bold. d, n, and π_n denote the feature dimension, the sample size of the dataset, and the ratio of negative samples in the dataset. 70
4.3 AUC means of shallow rAD over 30 trials for different $\hat{\pi}_p$. The significant changes in the AUC means are highlighted in bold. . 71
4.4 AUC means of shallow rAD over 30 trials for different a. The significant changes in the AUC means are highlighted in bold. . 72

5.1 Performance on both sensory and semantic AD benchmarking datasets with 10% contamination ratio. Style: AUROC % ($\pm$ SE). Best in **bold**. 89
5.2 Performance on tabular AD benchmarking datasets with 10% contamination ratio. Style: AUROC % ($\pm$ SE). Best in **bold**. † represents transductive inference. 90
5.3 Performance on CSP plant dataset. 92

6.1 Anomaly detection performance. Style: best in bold and second best using underline 109
6.2 Ablation of time-embedding and pre-training. 110
6.3 Ablation of K 110
6.4 Anomaly detection performance on simulated data. 112
6.5 Deployment performance 113

7.1 AUROC and AUPR performances of `DensityAD` against baseline methods. Style: best in **bold**, and second best underlined. 125
7.2 Total CPU time and memory used for training the models. . . . 126

A.1 Global Autoencoder of `ULSAD`. 165
A.2 Feature Reconstruction Network of `ULSAD`. 166
A.3 Anomaly detection based on Image AUROC on MVTec dataset. 167
A.4 Anomaly segmentation performance based on Pixel AUROC on MVTec dataset. 168
A.5 Anomaly segmentation performance based on Pixel AUPRO on MVTec dataset. 168
A.6 Anomaly detection performance based on Image AUROC on MVTecLOCO dataset. 169
A.7 Anomaly segmentation performance based on Pixel AUROC on MVTecLOCO dataset. 169
A.8 Anomaly segmentation performance based on Pixel AUPRO on MVTecLOCO dataset. 169
A.9 Anomaly detection performance based on Image AUROC on MPDD dataset. 169
A.10 Anomaly segmentation performance based on Pixel AUROC on MPDD dataset. 170
A.11 Anomaly segmentation performance based on Pixel AUPRO on MPDD dataset. 170
A.12 Anomaly detection performance based on Image AUROC on BTAD dataset. 170
A.13 Anomaly segmentation performance based on Pixel AUROC on BTAD dataset. 170
A.14 Anomaly segmentation performance based on AUPRO on BTAD dataset. 171
A.15 Anomaly detection performance based on Image AUROC on VisA dataset. 171
A.16 Anomaly segmentation performance based on Pixel AUROC on VisA dataset. 171

A.17 Anomaly segmentation performance based on AUPRO on VisA dataset. 173
A.18 Anomaly detection performance based on Image AUROC on **structural anomalies** of MVTecLOCO dataset. 173
A.19 Anomaly segmentation performance based on Pixel AUROC on **structural anomalies** of MVTecLOCO dataset. 173
A.20 Anomaly segmentation performance based on Pixel AUPRO on **structural anomalies** of MVTecLOCO dataset. 174
A.21 Anomaly detection performance based on Image AUROC on **logical anomalies** of MVTecLOCO dataset. 174
A.22 Anomaly segmentation performance based on Pixel AUROC on **logical anomalies** of MVTecLOCO dataset. 174
A.23 Anomaly segmentation performance based on Pixel AUPRO on **logical anomalies** of MVTecLOCO dataset. 174
A.24 Ablations for local branch. Style: I-AUROC | P-AUROC | P-AUPRO. 175
A.25 Ablations for total architecture. Style: I-AUROC | P-AUROC | P-AUPRO. 175
A.26 Ablations for backbone on MvTec-LOCO. Style: I-AUROC | P-AUROC | P-AUPRO. 176

B.1 AUC means of shallow rAD over 30 trials for different $\hat{\pi}_p$. The significant changes in the AUC means are highlighted in bold. . 179
B.2 AUC means of shallow rAD over 30 trials for different a. The significant changes in the AUC means are highlighted in bold. . 180
B.3 AUC means (and standard error) of deep rAD over 20 trials for different $\hat{\pi}_p$. The significant changes in the AUC means are highlighted in bold. 181

C.1 Prompts for CLIP where `"c"` denotes the category. 188
C.2 Performance of `EPHAD` on tabular datasets with 10% contamination ratio and LOF as evidence function. Style: AUROC % (± SE). Best in **bold**. † represents transductive inference. 190
C.3 Performance of `EPHAD` on tabular datasets with 10% contamination ratio and IForest as evidence function. Style: AUROC % (± SE). Best in **bold**. † represents transductive inference. . . . 191
C.6 Comparison with LOE (AUROC %) 191
C.4 Performance od `EPHAD-Ada` on tabular datasets with 10% contamination ratio and LOF as evidence function. Style: AUROC % (± SE). Best in **bold**. † represents transductive inference. . . 192

C.7 Comparison with SoftPatch 192
C.5 Performance od `EPHAD-Ada` on tabular datasets with 10% contamination ratio and IForest as evidence function. Style: AUROC % (± SE). Best in **bold**. † represents transductive inference. 193

D.1 Encoder architecture 196
D.2 Decoder architecture 197
D.3 Extended Anomaly Detection Performance. Style: **best** and second best 199

E.1 Loss of information after reducing the original image (256×256). 201
E.2 Ablation of K. Style: best in **bold**. 201
E.3 Ablation of time-embeddings. Style: best in **bold**. 202
E.4 AUROC and AUPR of `DensityAD` on the simulated dataset. . . 204