A Study on Enhancing Deep Generative Models Using Adaptive Threshold Technique for Identifying Outliers

Seoyoung Cho

Department of Statistics

The Graduate School of Sungshin Women's University

A Study on Enhancing Deep Generative Models Using Adaptive Threshold Technique for Identifying Outliers

A Master's Thesis
Submitted to the
Graduate School of Sungshin Women's University

in partial fulfillment of the requirements
for the degree of
Master of Statistics

Seoyoung Cho

November, 2024

This is to certify that we have examined the
Master's Thesis of
Seoyoung Cho
Submitted to Department of Statistics

Approved as to style and content:

Thesis Advisor Dongha Kim

Committee Chairman Hohyun Jung

Committee Member Joonho Shin

Committee Member Kwan-Young Bak

The Graduate School of Sungshin Women's University

Abstract

Outlier detection (OD) involves identifying unusual observations (or outliers) within a dataset by learning the distinct patterns of normal observations (or inliers). Recently, a study introduced a robust unsupervised OD (UOD) method based on a new observation in deep generative models, known as the *inlier-memorization (IM) effect*, which indicates that generative models tend to memorize inliers before outliers during the early stages of training. In this study, we aim to develop a theoretically grounded approach to address UOD tasks by *maximally utilizing the IM effect*. We start by noting that the IM effect becomes more pronounced when the training data contains fewer outliers. This insight suggests that enhancing the IM effect in UOD scenarios is possible if outliers can be effectively excluded from mini-batches during loss function design. To achieve this, we introduce two key techniques: 1) gradually increasing the mini-batch size as training progresses, and 2) employing an adaptive threshold for the truncated loss function. We theoretically demonstrate that these techniques effectively filter out outliers from the truncated loss function, enabling us to fully exploit the IM effect. Together with an ensemble technique, we propose a method called *Adaptive Loss Truncation with Batch Increment (ALTBI)*. Extensive experimental results show that ALTBI outperforms other recent methods in identifying outliers, even with significantly lower computational costs. Moreover, we demonstrate that our method maintains robust performance when combined with privacy-preserving algorithms.

Keywords : outlier detection, deep generative model, IM-effect

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Outlier detection

Outlier detection (OD) is an important task across various domains, with the goal of identifying data that are noticeably different from the normal pattern, known as outliers. This process begins by analyzing and learning the unique patterns of normal data points, known as inliers, and developing methods to assign scores to each data point that help distinguish whether it fits within the normal range or stands out as an outlier. The ability to accurately detecting outliers is essential because it directly impacts the quality and reliability of data used in various critical applications such as fraud detection, network security, and fault diagnosis.

Outlier detection tasks can be categorized into three scenarios based on the availability of information regarding anomalies in the given training data. Supervised Outlier Detection (SOD) uses labeled data to classify each sample as either an outlier or not. In this approach, the training dataset includes labels for each data point, indicating whether it is normal or an outlier. The model is trained using this labeled data to learn the distinction between inliers and outliers. Semi-Supervised Outlier Detection (SSOD), also referred to as Out-of-Distribution (OOD) Detection, assumes that training dataset is composed entirely of normal data and builds models using only these inliers. The goal is to distinguish between in-distribution data and out-of-distribution data during the testing phase. Unsupervised Outlier Detection (UOD) is a technique used to accurately identify outliers in a dataset that does not have any labels indicating which data points are outliers and which are not. In UOD

tasks, the dataset is assumed to potentially contain both normal data and outliers, but there is no prior information or labels to distinguish between them. In general, many real-world anomaly detection tasks belong to UOD because outliers in large datasets are typically not known in advance. Therefore, we focus on addressing UOD problems in this study.

Recent advancements in machine learning have significantly enhanced UOD methods, particularly through the use of deep generative models (DGMs). These models, such as VAE (Kingma and Welling, 2013), IWAE (Burda et al., 2016), and GLOW (Kingma and Dhariwal, 2018) have been employed to learn the complex distributions of normal data and develop unique scoring mechanisms to identify outliers. DGMs use their generative capabilities to assign scores based on how well a data point fits the learned normal distribution. Surprisingly, conventional likelihood was not utilized, as likelihood can sometimes assign high probabilities to outliers, thereby confusing them with inliers in fully trained models (Nalisnick et al., 2019b,a; Lan and Dinh, 2021).

A recent study has highlighted the potential of using likelihood values from DGMs in UOD, leveraging the *inlier-memorization (IM) effect* (Kim et al., 2024). This effect suggests that, in the early stages of DGM training, the loss values (negative log-likelihoods) for inliers decrease more quickly than those for outliers. This indicates that likelihood values from *under-fitted DGMs* can serve as a reliable measure for identifying outliers. Building on this observation, Kim et al. (2024) introduced ODIM, a novel UOD method that utilize this early learning behavior. ODIM is designed to capitalize on the rapid memorization of inliers by DGMs, enabling efficient and accurate outlier detection even with limited training. The method has shown to be computationally efficient and felxible across different types of datasets, offering a robust solution for identifying inliers and outliers. This approach not only improves the accuracy of outlier detection but also addresses the limitations of conventional likelihood-based methods, which can sometimes misclassify outliers as inliers in fully trained models. The ability to effectively utilize under-fitted models is the essential

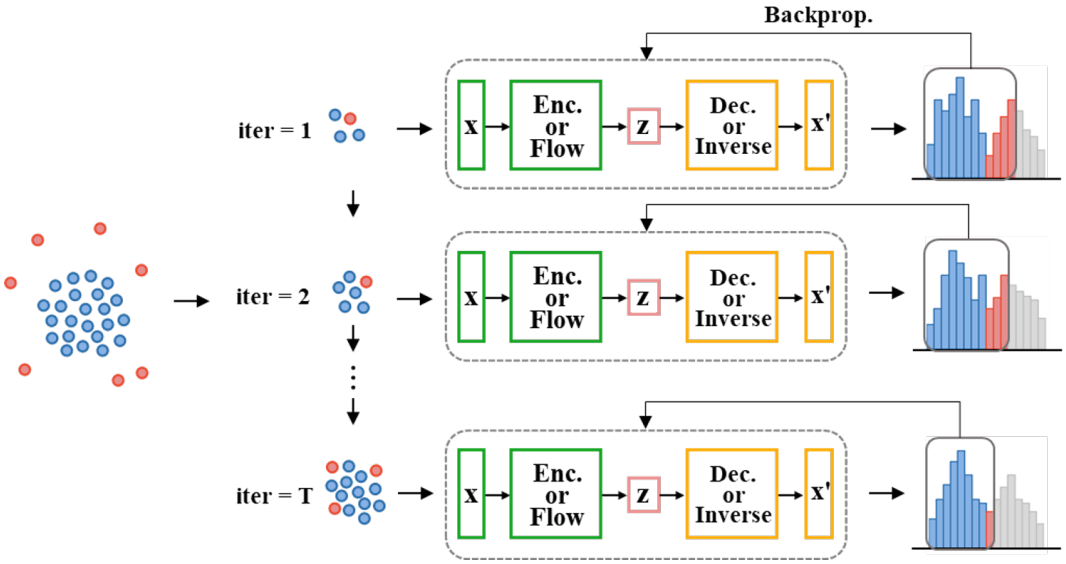

Figure 1: An illustration of ALTBI.

advancement of ODIM, making it a powerful tool for detecting anomalies across a wide range of applications.

1.2 Improvement of IM effect

Building on the principles introduced by ODIM, this study seeks to develop an enhanced UOD method that fully leverages the IM effect. The study starts with the critical observation that the clarity of the IM effect becomes more pronounced as the number of outliers in the training data decreases. This observation is visually represented in Figure 2 and highlights that by effectively separating outliers from inliers early in the training process, the IM effect can be amplified. By excluding outliers when constructing loss functions during these initial stages, the IM effect can be maximized, leading to improved outlier detection. This approach illustrates the importance of strategically managing outliers to enhance the model's ability to distinguish between normal and anomalous data points.

To this end, we introduce two critical techniques aimed at maximizing the IM effect

for more precise outlier detection. First, we propose gradually increasing the mini-batch size during training. Second, we implement an adaptive threshold to truncate the loss function, ensuring that outliers have less influence on the model as training progresses. These approaches allows the model to adjust progressively, improving its ability to distinguish between inliers and outliers. We provide theoretical results showing that these techniques effectively reduce the proportion of outliers in the truncated loss function, approaching zero as training advances. This means that as the model continues to train, it becomes increasingly focused on inliers, enhancing its detection accuracy. Furthermore, we incorporate an ensemble strategy within a single DGM by utilizing various updates of loss values. This strategy aims to improve the performance and stability of our method without introducing additional computational or resource costs.

By integrating all these elements, we develop a comprehensive framework for addressing UOD tasks, which we call *Adaptive Loss Truncation with Batch Increment (ALTBI)*. This approach is visually presented in Figure 1. ALTBI offers several distinct advantages over existing UOD methods. Firstly, it consistently outperforms other techniques in detecting outliers across various datasets. Through extensive experiments on 57 datasets, we demonstrate that ALTBI achieves state-of-the-art performance in identifying outliers.

Moreover, ALTBI is highly efficient, requiring only a simple, under-fitted likelihood-based DGM. It can be trained using models like Variational Autoencoders (VAE, Kingma and Welling (2013)) or Normalizing Flows (NF, Dinh et al. (2017)) with just a few hundred mini-batch updates. This efficiency translates into significantly reduced computational costs compared to other recent methods. Our results suggest that ALTBI is a highly promising approach for delivering effective and efficient UOD solutions in various practical applications.

The reminder of our paper is organized as follows. We first provide a brief review of related research on OD problems, primarily focusing on SSOD and UOD, and some studies on DGMs. Then, we offer detailed descriptions of ALTBI along with its motivations,

followed by its theoretical discussions. The results of various experiments are presented, including performance tests, ablation studies, and further discussions related to data privacy. Finally, concluding remarks are provided. The key contributions of our work are:

- We find that the IM effect is observed more apparently when the training data have fewer outliers.
- We develop a theoretically well-grounded and powerful UOD solution called ALTBI, using truncated loss functions with incrementally increasing mini-batch sizes.
- We empirically validate the superiority of ALTBI in detecting outliers by analyzing 57 datasets.

Chapter 2

Related works

2.1 Outlier detection methods

We first review studies dealing with UOD problems. Numerous traditional approaches have been proposed to address UOD problems. LOF (Breunig et al., 2000a) identifies local outliers in a dataset by analyzing density. This concept is expanded in CBLOF (He et al., 2003), which evaluates an outlier's significance based on the size of the cluster it belongs to and its distance from the nearest cluster. The CBLOF algorithm uses the Squeezer algorithm to partition the dataset into clusters, assigning data points to clusters based on similarity and forming new clusters when the similarity threshold is not met. After clustering, the CBLOF value is calculated for each data point, considering the size of its cluster and its proximity to the nearest large cluster. MCD (Fauconnier and Haesbroeck, 2009) is a robust method for detecting outliers in multivariate data by selecting a subset with the smallest covariance determinant, which reduces the influence of outliers. Outliers are then identified based on their robust distances relative to a cutoff value. Although a high breakdown point ensures robustness, it may decrease efficiency, which can be improved by using reweighted MCD estimators. IF (Liu et al., 2008) detects anomalies by isolating data points through tree structures.

Various deep learning techniques have been developed to address UOD problems. RDA (Zhou and Paffenroth, 2017) enhances deep autoencoders by adding robustness to outliers, while DSEBM (Zhai et al., 2016) generates an energy function as the output of a deterministic deep neural network. ODIM (Kim et al., 2023) utilizes the Inlier-Memorization

Effect observed in the early training phases of deep generative models to efficiently and accurately identify outliers. IM effect indicates that deep generative models memorize inliers before outliers during the early training phases. Also, ODIM is a new computationally efficient and domain-agnostic method that involves training a DGM for a few updates and identifying samples with large loss values as outliers. DTE (Livernoche et al., 2023) estimates the distribution of diffusion time for a given input, using the mode or mean of this distribution as an anomaly score. It explores variations of diffusion modeling for unsupervised and semi-supervised anomaly detection, simplifying the computational complexity of Denoising Diffusion Probability Models (DDPM) (Ho et al., 2020) while maintaining competitive performance and improving inference efficiency. The method assumes a data point produced through the diffusion process and identifies its diffusion time distribution, which resembles an inverse Gamma distribution. It uses a non-parametric model to approximate parameters with k-nearest neighbors and a parametric model with deep neural networks to estimate the posterior distribution.

We also review research related to SSOD problems. SVDD (Tax and Duin, 2004) employs kernel functions to create a boundary around a dataset for outlier detection, while DeepSVDD (Ruff et al., 2018) builds on SVDD by using a deep autoencoder to map data into a feature space where normal instances fall within the boundary and anomalies lie outside. DeepSAD (Ruff et al., 2020) further extends DeepSVDD by incorporating scenarios where a small number of labeled outliers are available.

Self-supervised learning has been widely used to address SSOD tasks (Tack et al., 2020; Golan and El-Yaniv, 2018). For instance, SimCLR (Chen et al., 2020) uses contrastive learning to generate high-quality feature representations of inliers. Moreover, ICL employs two mappings to maximize mutual information between masked and unmasked segments of tabular data through contrastive loss. This technique scores test samples by analyzing contrastive loss, which aids in detecting anomalies by identifying deviations from

the training class distribution. It adopts one-class classification in a semi-supervised manner and enhances anomaly detection by leveraging subsets of consecutive variables to increase mutual information.

2.2 Deep generative models

We also examine various studies on deep generative models. Glow (Kingma and Dhariwal, 2018) is a flow-based generative model designed to efficiently model complex data distributions by combining key innovations. It uses ActNorm to apply an affine transformation with learnable scale and bias parameters per channel, ensuring stable and consistent training through data-dependent initialization. To capture interactions between image channels, it replaces fixed permutations with an invertible 1×1 convolution, leveraging LU decomposition for efficiency and invertibility. By utilizing invertible 1×1 convolution layers, which effectively capture interactions between image channels, the model guarantees both efficiency and invertibility. Additionally, its Affine Coupling Layer transforms data with learnable scaling and translation functions, maintaining invertibility and enabling exact likelihood calculations. With a multi-scale architecture that compresses data hierarchically, Glow is well-suited for tasks like image generation, anomaly detection, and data compression, offering a powerful and efficient approach to generative modeling. Trained on large-scale image datasets, Glow produces realistic, high-resolution images while achieving lower computational costs and superior image quality compared to other generative models. IWAE (Burda et al., 2016) enhances the training of autoencoder-based generative models by introducing an augmented variational inference method that facilitates the learning of more precise probabilistic models. Importance Weighting improves the sampling accuracy by assigning weights to each sampled latent variable, thereby reinforcing the ELBO of the

standard Variational Autoencoder (Kingma and Welling, 2013) and yielding more accurate probabilistic estimates. This method is particularly effective in preserving high performance with high-dimensional data, offering a superior generative model for complex datasets.

Chapter 3

Detailed description of ALTBI

3.1 Preliminaries

3.1.1 Notations and definitions

We introduce notations and definitions frequently used throughout this paper. Let $X_1, \ldots, X_n (\in \mathcal{X} \subset \mathbb{R}^D) \sim P_*$ be n independent random input vectors following the true distribution P_*. Since training data contain outliers as well as inliers in the UOD regime, we assume that P_* is a mixture of two distributions, i.e., $P_* = (1-\alpha)P_i + \alpha P_o$, where P_i, P_o represent the inlier and outlier distributions, respectively, and $\alpha \in (0, 1)$ is the outlier ratio. And we define the support of P_i and P_o as $\mathcal{X}_i$ and $\mathcal{X}_o$, respectively (hence $\mathcal{X} = \mathcal{X}_i \cup \mathcal{X}_o$). Since inliers and outliers do not share their supports, we can obviously assume $\mathcal{X}_i \cap \mathcal{X}_o = \emptyset$.

We denote a training dataset comprising n observations by $\mathcal{D}^{\text{tr}} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. For a given sample $\boldsymbol{x}$, a per-sample loss function with a given DGM is defined as $l(\theta; \boldsymbol{x})$, where $\theta \in \Theta$ represents the parameters for constructing the DGM. Since we consider likelihood-based DGMs such as VAE-based ones (Kingma and Welling, 2013; Burda et al., 2016; Kim et al., 2020), or NF-based ones (Dinh et al., 2015, 2017; Kingma and Dhariwal, 2018), $l(\boldsymbol{x}; \theta)$ would be the negative log-likelihood (or ELBO). Without loss of generality, we assume that $l(\theta; \boldsymbol{x})$ is differentiable and bounded by $[0, 1]$ for any $\boldsymbol{x} \in \mathcal{X}$ and $\theta \in \Theta$.

The risk function calculated over a distribution P is denoted by $L(\theta, P) = E_{X \sim P}[l(\theta; X)]$. We respectively abbreviate $L(\theta, P_i)$ and $L(\theta, P_o)$ as $L_i(\theta)$ and $L_o(\theta)$. Finally, we denote the minimizer of the inlier risk as θ_*, i.e., $\theta_* = \text{argmin}_\theta L_i(\theta)$. We assume $L_i(\theta_*) = 0$.

3.1.2 Brief review of ODIM

ODIM is a novel approach in the field of UOD that leverages the Inlier-Memorization (IM) effect. This effect describes a phenomenon observed when training DGMs on datasets that may contain outliers. The main idea is that during the early stages of training, the model tends to memorize inliers data points that represent the normal, majority pattern before memorizing outliers, which are anomalous or rare data points. This effect occurs because inliers are more prevalent and densely clustered within the data distribution, making it more efficient for the model to reduce the per-sample loss values of inliers first. The model can more effectively minimize the overall loss function during initial training stages, providing an intuitive explanation for the IM effect.

ODIM takes advantage of this effect by training likelihood-based DGMs, such as VAE (Kingma and Welling, 2013) or IWAE (Burda et al., 2016), over a predefined number of updates. During training, ODIM uses the per-sample loss values as indicators or scores to identify outliers. The insight is that during the early training updates, inliers will show lower loss values, while outliers will exhibit higher loss values due to the model's focus on inliers first.

To find the optimal number of updates where the IM effect is most pronounced, ODIM calculates the degree of bi-modality in the distribution of per-sample loss values at each update. Bi-modality in this context refers to a distribution with two distinct peaks, where one peak represents inliers and the other represents outliers. To achieve this, ODIM fits a two-cluster Gaussian mixture model to the loss distribution after each update and then measures the dissimilarity between the two clusters using metrics like the Wasserstein distance. By monitoring these bi-modality measures across all updates, ODIM selects the update point with the highest measure as the optimal moment to distinguish between inliers and outliers.

To further enhance outlier detection performance, ODIM also incorporates an ensemble technique. In this approach, multiple under-fitted DGMs are independently trained, each

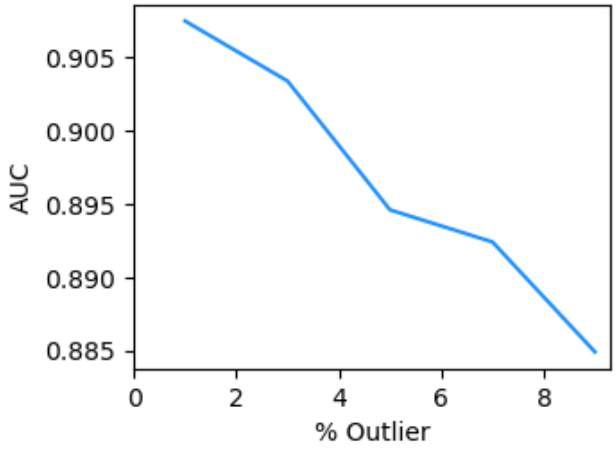

Figure 2: Relationship between the outlier ratio in training data and IM effect.

from a different initialization. The final outlier score for a given data point $\boldsymbol{x}$ is then calculated as the average of the per-sample loss values across these multiple models. Formally, the score is given by:

$$s^{\text{ODIM}}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} l(\theta^{(b)}; \boldsymbol{x}),$$

where $\theta^{(b)}, b = 1, \ldots, B$ represent the parameters estimated from the b-th model out of the total B models. In this formulation, a data point $\boldsymbol{x}$ is considered an outlier if it receives a high score, and conversely, it is regarded an inlier if the score is low. This ensemble approach enhances the robustness and reliability of the outlier detection process, making ODIM a powerful tool in UOD.

3.2 Relationship between IM effect and outlier ratio

The IM effect suggests that distinguishing between inliers and outliers is feasible by utilizing per-sample loss values from an under-fitted DGM. In this section, we argue that the clarity of the IM effect improves as the proportion of outliers in the training data decreases.

To support this, we conducted an experiment on the `PageBlocks` dataset with varying outlier ratios from 1% to 9%. In each scenario, a VAE is trained for 100 mini-batch updates, with a batch size of 128. We then assess AUC values of the training data to evaluate the effectiveness of outlier detection based on per-sample loss values.

The result, illustrated in Figure 2, clearly indicates that the IM effect becomes increasingly evident as the outlier ratio decreases, thereby strongly supporting our hypothesis. This observation suggests that by effectively filtering out outliers during the construction of loss values within mini-batch updates, the IM effect can be enhanced, leading to improved outlier detection performance. This insight forms the primary motivation for our study, leading us to explore methods to strengthen the IM effect by refining the handling of outliers during training.

3.3 Proposed method: ALTBI

3.3.1 Mini-batch increment and adaptive threshold

We again note that the goal of the proposed method is to maximize the effectiveness of the IM effect during the early stages of training. To achieve this, we introduce two simple but key strategies to obtain a refined loss function: *1) using a mini-batch size that gradually increases as training proceeds and 2) utilizing a truncated loss function with an adaptive threshold.*

To be more specific, as a *warm-up* phase, for given integers n_0 and T_0, the process begin with a *warm-up* phase where a DGM is trained using a conventional loss function. During this phase, for two given integers n_0 and T_0, mini-batches with a fixed size n_0 are used for a specific number of updates T_0. The purpose of this phase is to estimate the parameters where the IM effect begins to appear. By beginning with a conventional loss function and a fixed mini-batch size, the model is given the opportunity to stabilize and

start identifying inliers, effectively setting the stage for more advanced and refined training in the following phases.

After warm-up stage, as the second phase, we apply the mini-batch increment and loss truncation strategies. At each update iteration t, we apply the mini-batch size which is systematically increased with each update iteration. Specifically, the size of the mini-batch $\mathcal{D}_t \subset \mathcal{D}^{\text{tr}}$ is calculated using an exponential function relative to the iteration t. The formula used is $|\mathcal{D}_t| = n_0\gamma^{t-1}(=: n_t)$ where n_0 represents the initial mini-batch size, and γ is a constant greater than 1. This gradual increase in mini-batch size allows the model to progressively adjust as training progresses, thereby enhancing its learning efficiency. Additionally, Instead of using the entire mini-batch included in $\mathcal{D}_t$ to calculate the loss, the method employs a *truncated loss function*.

$$\hat{L}(\theta, \tau_t; \mathcal{D}_t) = \frac{\sum_{\boldsymbol{x}\in\mathcal{D}_t} l(\theta; \boldsymbol{x}) \cdot I(l(\theta; \boldsymbol{x}) \leq \tau_t)}{\sum_{\boldsymbol{x}'\in\mathcal{D}_t} I(l(\theta; \boldsymbol{x}') \leq \tau_t)}, \tag{3.1}$$

where $\tau_t > 0$ is an adaptive threshold. This function only considers a subset of data points within the mini-batch, specifically those with lower loss values, allowing the model to concentrate its learning on the most representative inliers.

Theoretically, we set τ_t as the inlier risk, defined as $\tau_t = L_i(\theta_{t-1})$, where θ_{t-1} represents the estimated parameter from the $(t-1)$th update using an SGD-based optimizer. However, calculating $\tau_t = L_i(\theta_{t-1})$ is impractical in real scenarios because we lack prior knowledge about which training samples are anomalies. Instead, we introduce a quantile-based approach for determining τ_t. Specifically, for a pre-specified value $0 < \rho < 1$ we filter out the top $(100 \times (1-\rho))\%$ of samples in the mini-batch that have the highest per-sample loss values. This method allows us to approximate τ_t without requiring explicit anomaly information, ensuring that the model focuses on the most representative inliers while excluding potential outliers during training.

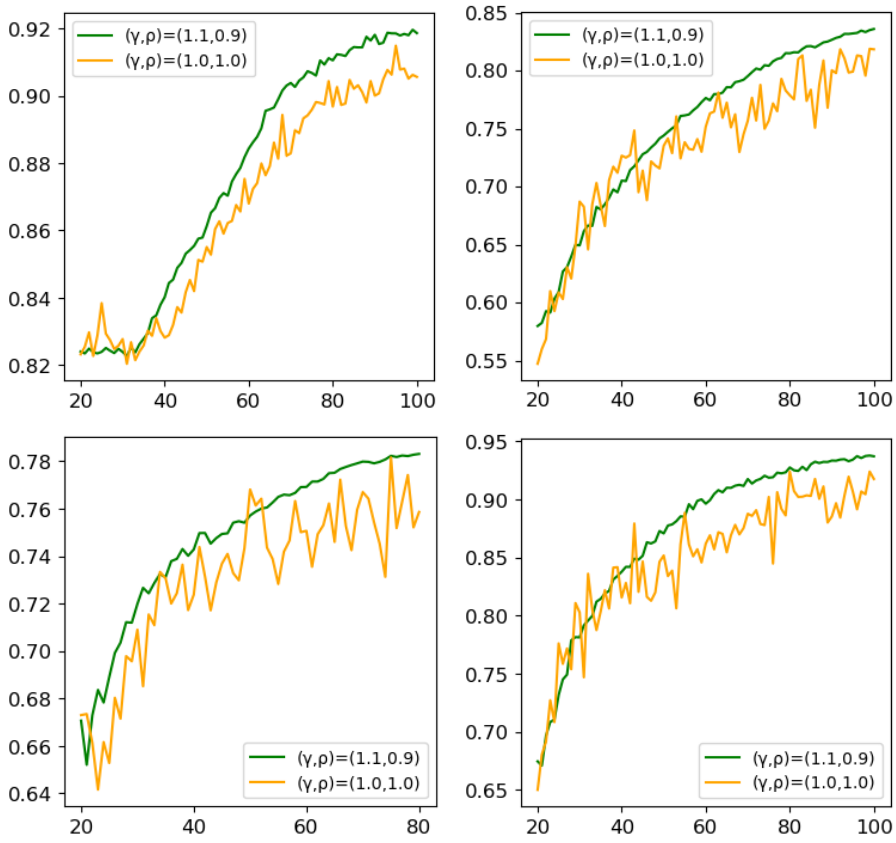

Figure 3: Outlier detection AUC values for DGMs with and without applying mini-batch increment and adaptive threshold, coloured as green and orange, respectively. (**Upper left to clockwise**) We analyze `Ionosphere`, `Letter`, `Vowels`, and `MagicGamma` datasets.

We conduct an experiment to empirically validate the impact of the mini-batch increment and adaptive threshold strategies. VAEs are trained under two different scenarios: one incorporating these strategies and the other without them. For the first scenario, parameters were set as $(T_0, n_0, \gamma, \rho) = (10, 128, 1.1, 0.9)$. The outlier detection AUC results across four datasets: `Ionosphere`, `Letter`, `Vowels`, and `MagicGamma`, as shown in Figure 3, indicate that the IM effect is more pronounced when using these strategies, resulting in superior outlier detection performance. Additionally, the increasing mini-batch size re-

duces fluctuations during updates, leading to a more stable model. The theoretical properties of these adaptive techniques will be discussed in the following section.

Remark 1. *Increasing the mini-batch size and using truncated loss function during training were first proposed in Xu et al. (2021). They utilized these techniques to develop enhanced classifiers in semi-supervised learning tasks. Our proposed method has its own contribution in that we find the close connection between the IM effect and these two techniques in the UOD regime and apply them to train DGMs with high outlier detection performance.*

3.3.2 Ensemble within a single model

In ODIM, the degree of bi-modality in per-sample loss distributions is used to identify the optimal update where the IM effect is most pronounced. However, this approach has limitations, as it can sometimes fail to identify the optimal update, even selecting an update where the IM effect is absent. Additionally, ODIM uses multiple under-fitted models to enhance performance, which significantly increases computation time and resource usage.

To overcome these limitations, we propose a different approach that does not measure bi-modality or require multiple models. Instead, we implement an ensemble strategy *within a single model*. Specifically, for given two integers T_1, T_2 with $T_1 < T_2$,we average the per-sample loss values over updates from $T_1 + 1$ to T_2. This approach is more computationally efficient while still leveraging the benefits of ensemble learning. That is, for a given input $\boldsymbol{x}$, we compute its outlier score as

$$s^{\text{ALTBI}}(\boldsymbol{x}) = \frac{1}{T_2 - T_1} \sum_{t=T_1+1}^{T_2} l(\theta_t; \boldsymbol{x}), \tag{3.2}$$

where θ_t is the estimated parameter at the t-th update.

By adopting this ensemble method, we reduce the computational burden while improving the stability and effectiveness of the outlier detection process. The result is a more robust

and efficient algorithm that addresses the shortcomings of ODIM, particularly in terms of computational efficiency and model performance. We demonstrate that this approach not only improves performance but also provides stability, as reported in the experimental section.

We combine the above three techniques—1) mini-batch increment, 2) truncated loss, and 3) loss ensemble at various updates—to propose our method, which we term *Adaptive Loss Truncation with Batch Increment (ALTBI)*. The pseudo algorithm of ALTBI is presented in Algorithm 1.

3.3.3 Choice of DGM framework

There are numerous DGMs that focus on maximizing likelihood, each with distinct approaches. VAE-based models, such as those introduced by Kingma and Welling (2013) and Burda et al. (2016), are designed to optimize the Evidence Lower Bound (ELBO). By optimizing the ELBO, these models effectively approximate complex data distributions and learn compact, meaningful latent representations that capture the underlying structure of the data. This allows VAEs and their extensions to generate new samples that are similar to the original data and enables them to be used in a wide range of applications, including unsupervised learning, anomaly detection, and data generation.

NF-based models (Dinh et al., 2017; Kingma and Dhariwal, 2018), use Normalizing Flows to transform simple base distributions into complex ones through a series of invertible mappings, enabling the computation of exact log-likelihoods.

Additionally, score-based models, such as those by Ho et al. (2020) and Song et al. (2021), use score matching and denoising techniques to generate data. However, these models typically involve larger neural networks and require substantial computational resources, making them less practical for tasks where efficiency is crucial.

Given the focus on computational efficiency in our method, we decided to utilize IWAE

(Burda et al., 2016) and GLOW (Kingma and Dhariwal, 2018). These models are well-known representatives of VAE and NF approaches, respectively. The loss functions used in these models correspond to the ELBO-like upper bound in IWAE and the exact log-likelihood in GLOW. We excluded score-based DGMs due to their large model sizes, which are less compatible with our objective of maintaining computational efficiency.

Algorithm 1 ALTBI

In practice, we set
$(n_0, \gamma, \rho, T_0, T_1, T_2) = (128, 1.03, 0.92, 10, 60, 80)$.

Input: Training data: $\mathcal{D}^{\text{tr}} = \{\boldsymbol{x}_j\}_{j=1}^{n}$, parameters of a given DGM :θ, initial mini-batch size: n_0, mini-batch increment: γ, quantile value: ρ, learning rate: η, three time steps: T_0, T_1, and T_2.

Initialize θ_0.
Phase 1: Warm-up
for $(t = 1$ **to** $T_0)$ **do**
    Draw a mini-batch with the fixed size of n_0, $\mathcal{D}_t = \{\boldsymbol{x}_j^{\text{mb}}\}_{i=1}^{n_0}$, from $\mathcal{D}^{\text{tr}}$.
    Calculate the loss function:
    $\hat{L}(\theta_0; \mathcal{D}_t) = \frac{1}{n_0} \sum_{j=1}^{n_0} l(\theta_0; \boldsymbol{x}_j^{\text{mb}})$.
    Update θ_0:
    $\theta_0 \leftarrow \theta_0 - \eta \cdot \nabla_\theta \hat{L}(\theta_0; \mathcal{D}_t)$.
end for
Phase 2: Enhancement of IM effect
for $(t = 1$ **to** $T_2)$ **do**
    Draw a mini-batch with a size of $n_t = n_0 \gamma^{t-1}$, $\mathcal{D}_t = \{\boldsymbol{x}_j^{\text{mb}}\}_{j=1}^{n_t}$ from $\mathcal{D}^{\text{tr}}$.
    Set the threshold $\tau_t = L_i(\theta_{t-1})$. // In practice, we choose τ_t as $(100 \times \rho)$-percentile of $\{l(\theta_{t-1}; \boldsymbol{x}_j)\}_{j=1}^{n_t}$.
    Compute truncated loss $\hat{L}(\theta_{t-1}, \tau_t)$ as (3.1)
    Update θ_t:
    $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \nabla_\theta \hat{L}(\theta_{t-1}, \tau_t)$.
    if $(t > T_1)$ **then**
        Incorporate the per-sample loss values to the final ALTBI scores as (3.2)
    end if
end for

Output: ALTBI scores of training data: $\left\{s^{\text{ALTBI}}(\boldsymbol{x}_j)\right\}_{j=1}^{n}$

Chapter 4

Theoretical analysis

In this section, we offer theoretical insights to demonstrate how increasing the mini-batch size and applying a truncated loss function enhance the IM effect. We begin by presenting a precise definition of the IM effect.

Assumption 1 (IM effect)**.** *There exist* $0 < a_1 < a_2 < 1$ *and* $a_3 \in (0, 1 - a_2)$ *such that for any parameter* θ *satisfying* $L_i(\theta) \in [a_1, a_2]$*,* $L_o(\theta) - L_i(\theta) \geq a_3$*.*

Assumption 1 refers to the property that, as training progresses, inliers tend to gradually have lower loss values than outliers, which is crucial for distinguishing between the two. The assumption mathematically captures the IM effect, where inliers are learned by the model earlier than outliers, creating a significant difference in their respective loss values. This difference can then be leveraged by the model to enhance the accuracy of outlier detection. A couple of additional yet reasonable assumptions about the gradient are required, which are almost the same as those in Xu et al. (2021).

Assumption 2 (Bounded and smooth gradient)**.** *Denote the gradients of* $l(\theta; \boldsymbol{x})$ *and* $L_i(\theta)$ *as* $\nabla_\theta l(\theta; \boldsymbol{x})$ *and* $\nabla_\theta L_i(\theta)$*, respectively. Then the followings conditions are satisfied:*

(i) For any $\boldsymbol{x} \in \mathcal{X}$ *and* $\theta \in \Theta$*, there exists a constant* $G > 0$*, such that*

$$\|\nabla_\theta l(\theta; \boldsymbol{x})\| \leq G.$$

(ii) $L_i(\theta)$ *is smooth with a L-Lipschitz continuous gradient, i.e., there exists a constant*

$L > 0$ *such that*

$$\|\nabla_\theta L_i(\theta) - \nabla_\theta L_i(\theta')\| \leq L\|\theta - \theta'\|, \quad \forall \theta, \theta' \in \Theta,$$

(iii) There exists $\mu > 0$ *such that for any* $\theta \in \Theta$*,*

$$2\mu(L_i(\theta) - L_i(\theta_*)) = 2\mu L_i(\theta) \leq \|\nabla_\theta L_i(\theta)\|^2,$$

where θ_* *is the minimizer of* $L_i(\theta)$*.*

The first and second assumptions in Assumption 2 refer to the properties that the loss function and inlier risk are smooth. The last assumption is known as the Polyak-Łojasiewicz condition (Polyak, 1964), which is widely considered in the literature related to SGD with deep learning (Yuan et al. (2019) and references therein).

We finally introduce a technical assumption about the loss distribution. We note that this condition is quite weak and can be satisfied in general situations.

Assumption 3 (Loss distribution)**.** *There is a constant* $0 < c < 1$ *such that, for any* θ*, the following inequality holds:*

$$\left[E_{P_i}\sqrt{l(\theta; X)}\right]^2 \leq (1 - c)L_i(\theta).$$

Then we have the following proposition, which asserts that if we apply the mini-batch increment and threshold to truncate the loss function, the ratio of outliers included in the truncated loss becomes small. The proof of Proposition 1 is provided in the Appendix A.

Proposition 1. *At the* t*-th update, we suppose that the current parameter* θ_{t-1} *satisfies* $a_1 \leq L_i(\theta_{t-1}) \leq a_2\gamma^{-(t-1)}$*. For a mini-batch* $\mathcal{D}_t$*, we denote the inlier set which are included in the truncated loss as* $\mathcal{A}_t^\tau$*. Similarly, we can define* $\mathcal{B}_t^\tau$ *for outliers. Then, under*

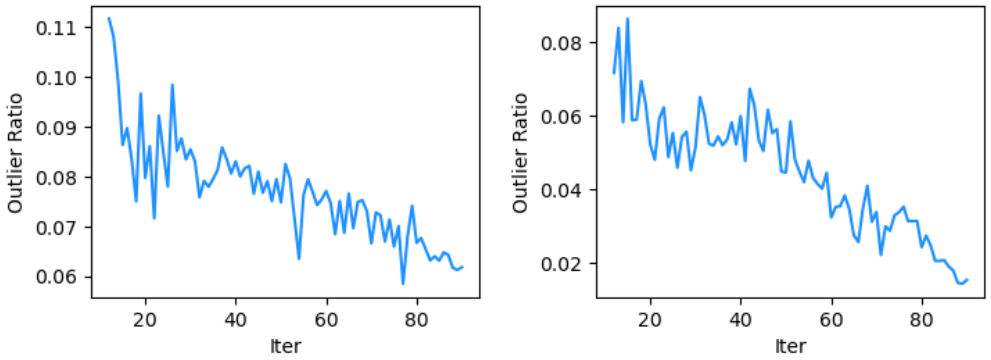

Figure 4: Trace plot of outlier ratio in truncated samples over various iterations. We visualize two datasets: (**Left**) `Cardio` and (**Right**) `Shuttle`.

Assumptions 1 to 3 and for a given $\delta > 0$*, there exists positive constants* c_1 *and* c_2 *not depending on* t *such that the following two inequalities holds:*

$$|\mathcal{A}_t^\tau| \geq c_1 n_t \quad \textit{and} \quad |\mathcal{B}_t^\tau| \leq c_2 n_0,$$

with a probability at least $1 - \delta$*.*

Considering $n_t = n_0 \gamma^{t-1}$, Proposition 1 indicates that at the t-th update, the number of inliers included in the truncated loss increase exponentially over iteration, while the number of outliers remains upper bounded by a constant. Additionally, the ratio of outliers included in the truncated loss cannot exceed $(c_2/c_1) \cdot \gamma^{-(t-1)}$, which decreases toward zero as the update step t increases as long as the IM effect persists at each update. Therefore, our proposed method gradually refines samples in the loss function, leading to the clearer IM effect.

We visualize whether the ratio of outliers actually decreases as the updates proceed. The same learning framework and hyperparameter settings from Figure 2 are considered, and two datasets are analyzed: `Cardio` and `Shuttle`. Figure 4 shows that the outlier ratio in the truncated loss function tends to decrease over updates, providing empirical evidence

for Proposition 1.

We note that Proposition 1 holds provided that the inlier risk is sufficiently small, i.e., smaller than $a_2\gamma^{-(t-1)}$. Next theoretical result deals with the guarantee that the inlier risk indeed decreases over updates with high probability. The proof is provided in Appendix A.

Proposition 2. *At the t-th update, we suppose that all the assumptions considered in Proposition 1 hold. Then for a given $\delta > 0$, there exists a learning rate $\eta > 0$ such that $L_i(\theta_t) \leq a_2\gamma^{-t}$ with a probability at least $1 - \delta$.*

The above proposition implies that if the previously estimated DGM satisfies the IM effect with a small inlier risk, then the subsequent estimated DGM has a smaller inlier risk with a factor of γ.

Suppose that the IM effect starts to be observed with the estimated parameter after warm-up step, i.e., $a_1 \leq L_i(\theta_0) \leq a_2$. Then, Proposition 2 suggests that with a carefully chosen learning rate, ideally, we can observe an enhanced IM effect up to $\lfloor(\log(a_1/a_2))/(\log(1/\gamma))\rfloor$ updates.

Chapter 5

Experiments

We demonstrate the superiority of our proposed method through an extensive analysis of 57 datasets, including image, text, and tabular data. Our results confirm that ALTBI outperforms existing methods across various data types, establishing it as the state-of-the-art solution. Additionally, ALTBI achieves this high level of performance with greater efficiency and significantly lower computational costs compared to other recent approaches. Additionally, we explore an extension of ALTBI for contexts where differential privacy needs to be ensured. In each experiments, we report the averaged results based on three trials with random parameter initializations. We use the `Pytorch` framework to run our algorithm using a single NVIDIA TITAN XP GPU.

5.1 Dataset description

We evaluate all 57 benchmark datasets from `ADBench` (Han et al., 2022), covering tabular, image, and text data. As in Kim et al. (2024), we apply min-max scaling to preprocess each dataset before conducting our analysis. We first consider 46 widely used tabular datasets that are frequently utilized in the analysis of outlier detection. These datasets cover various application domains including healthcare, finance, and astronautics. Additionally, we incorporate five benchmark datasets that are commonly used in natural language processing (NLP): `20news`, `Agnews`, `Amazon`, `IMDB`, and `Yelp`. For these datasets, we utilize their embedding features generated via BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019), both of which are publicly available in `ADBench`.

We finally analyze six image datasets: CIFAR10, MNIST-C, MVTec-AD, SVHN, MNIST, and FMNIST. These datasets are analyzed using the embedding features extracted by the ViT (Dosovitskiy et al., 2021), also available in ADBench. The detailed information about all the datasets is provided in the Appendix B and Han et al. (2022).

5.2 Baseline

We mainly compare our method with ODIM (Kim et al., 2024), and also consider other baselines compared in the study. These methods contain traditional machine-learning-based approaches whose implementations are provided in ADBench, including kNN (Ramaswamy et al., 2000), LOF (Breunig et al., 2000b), OCSVM (Schölkopf et al., 2001), CBLOF (He et al., 2003), PCA (Shyu et al., 2003), FeatureBagging (Lazarevic and Kumar, 2005), IForest (Liu et al., 2008), MCD (Fauconnier and Haesbroeck, 2009), HBOS (Goldstein and Dengel, 2012), LODA (Pevnỳ, 2016), COPOD (Li et al., 2020), and ECOD (Li et al., 2022).

And we also consider two deep learning-based UOD methods, DAGMM (Zong et al., 2018) and DeepSVDD (Ruff et al., 2018), both of which can be implemented through ADBench. Additionally, we evaluate our method against more recent deep learning approaches beyond ADBench, such as DROCC (Goyal et al., 2020), ICL (Shenkar and Wolf, 2022), GOAD (Bergman and Hoshen, 2020), DTE (Livernoche et al., 2023), and ODIM (Kim et al., 2024).

5.3 Implementation details

As mentioned previously, we use two likelihood-based DGM frameworks: 1) IWAE (Burda et al., 2016), an ELBO-based model and 2) GLOW (Nalisnick et al., 2019b), an NF-based model. IWAE uses multiple latent vectors to make the objective function tighter

than the standard ELBO.

IWAE extends the standard Variational Autoencoder (VAE) by using multiple importance-weighted samples from the latent space, which provides a tighter bound on the Evidence Lower Bound (ELBO) compared to traditional VAEs. This enhancement allows IWAE to more accurately approximate the true data distribution. For our implementation, we adopt the same Deep Neural Network (DNN) architecture for IWAE as described in Kim et al. (2024). We set the number of latent samples K to two, balancing computational efficiency with model accuracy. This setup allows IWAE to better capture complex data structures, making it well-suited for outlier detection tasks. Detailed descriptions of the architectures and loss functions are presented in Appendix B. GLOW (Kingma and Dhariwal, 2018) is a type of normalizing flow model that introduces invertible 1×1 convolutional filters to enhance the flexibility and expressiveness of the model. These invertible convolutions allow the model to effectively transform the data while preserving the ability to calculate exact likelihoods. This makes GLOW particularly powerful for high-dimensional data, as it can model complex distributions with a high degree of precision. The invertibility of the 1×1 convolutions also ensures that the model can be efficiently trained and used for tasks such as outlier detection and image generation. The architecture considered in Nalisnick et al. (2019b) is used, and we reshape each dataset into a squared form to apply this architecture.

For the optimizer, we use Adam (Kingma and Ba, 2014) with a learning rate of $1e-3$. Throughout our experimental analysis, we fix the hyperparameters, necessary for our proposed method–$(n_0, \gamma, \rho, T_0, T_1, T_2)$. We fix the initial mini-batch size to 128, and the number of iterations for warm-up stage to 10. For maximizing IM-effect we set the mini-batch size to increase by a factor of 1.03 with each update, and we use only the top 92 percentile of the loss values from the training process. After the warm-up stage, we conducted a total of 80 updates, with the final 20 updates incorporating ensemble methods within a single DGM. Performance results for other hyperparameter values are provided in the ablation studies.

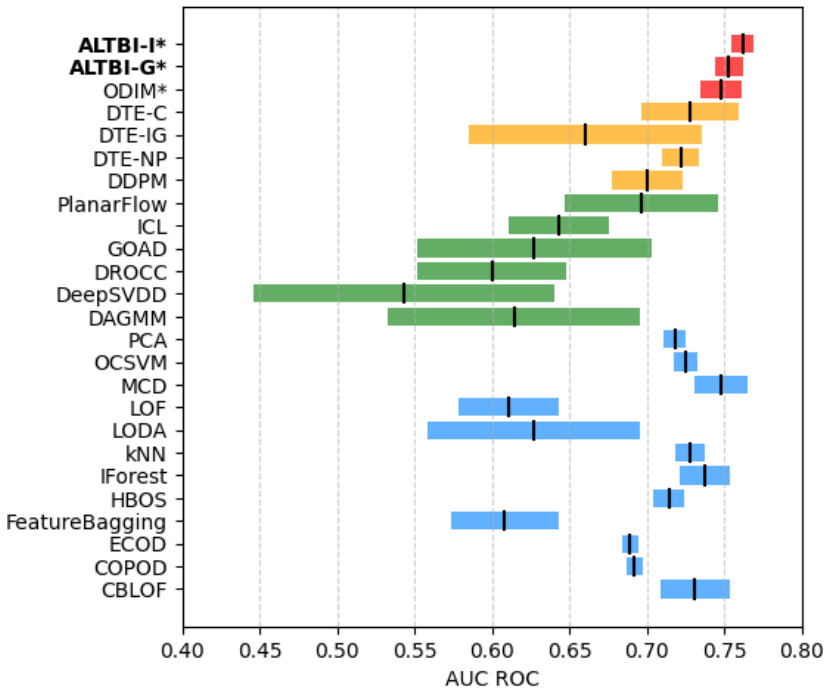

Figure 5: Averaged AUC results, including means and standard deviations, across 57 datasets from ADBench over three different implementations. We mark an asterisk (*) next to methods for our own implementations. Color scheme: red (IM-based), orange (diffusion-based), green (deep-learning-based), blue (machine-learning-based).

5.4 Performance results

We assess the outlier detection performance of ALTBI using both IWAE (ALTBI-I) and GLOW (ALTBI-G) against other baseline models to evaluate its performance in identifying outliers within training datasets. For each dataset, we measure the mean and standard deviation of the area under the receiver operating characteristic curve (AUC) and the area under the precision-recall curve (PRAUC) across three implementations. The averaged AUC results, including their standard deviations, are presented in Figure 5. Detailed results, including PRAUC for each dataset, are summarized in Appendix B. We acknowledge that we implemented ALTBI and ODIM ourselves, while all other baseline results are referenced

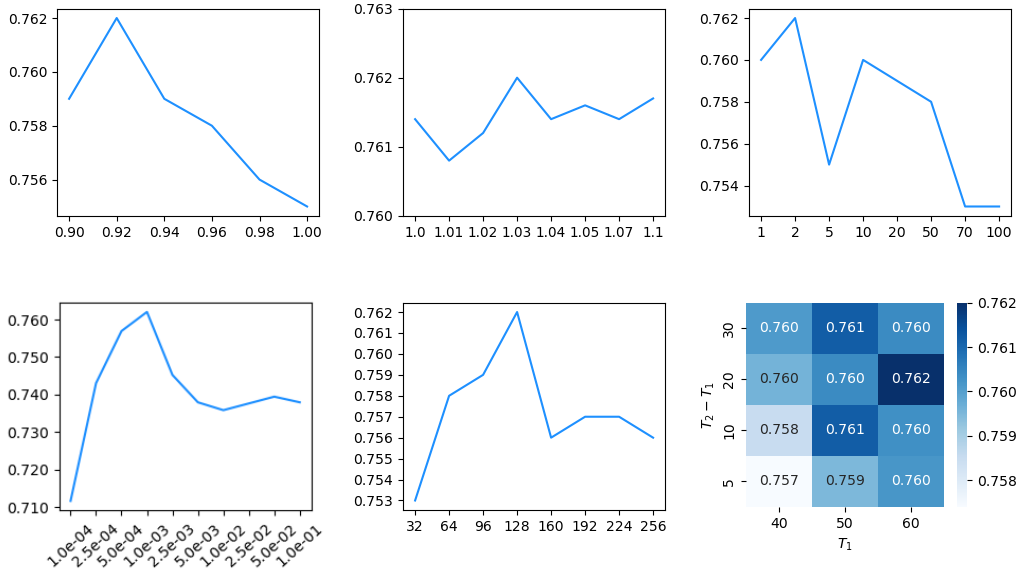

Figure 6: (**From top left to bottom right**) 1) AUC scores with various values of ρ. 2) AUC scores with various values of γ. 3) AUC scores with various values of K in IWAE. 4) AUC scores with various values of learning rate. 5) AUC scores with various values of n_0. 6) Heatmap of AUC scores for ensembling with various values of T_1 and $T_1 - T_2$.

from the Appendix in Livernoche et al. (2023).

Figure 5 demonstrates that ALTBI-I achieves the highest performance, followed by ALTBI-G, both outperforming all other baselines. Considering the computational efficiency of IWAE compared to GLOW, ALTBI-I presents itself as a more favorable off-the-shelf method for UOD. Additionally, our method indicates lower standard deviations in comparison to the other baselines, suggesting that ALTBI not only delivers superior performance but also maintains stability across various data types. This further highlights ALTBI as a reliable, off-the-shelf method for UOD.

5.5 Ablation studies

We perform further experiments to explore the impact of hyperparameter choices on ALTBI's performance across the `ADBench` datasets and the results are presented in Figure 6. Detailed results can be found in the Appendix B.

① We evaluated the optimal percentage of loss to utilize during the ALTBI process to determine the most effective level for achieving the best results. The results improved progressively as we reduced the amount of loss used from 100% to 92%, but beyond this point, the performance started to decline. Hence, we set the loss usage percentage as 92.

② We explored the optimal increase in mini-batch size at each update to achieve the best performance. While varying the γ values showed no significant impact on scores, we found that increasing the mini-batch size by a factor of γ = 1.03 at each update yielded the most effective results.

③ We examined the results with various numbers of samples in IWAE, ranging from 1 to 100. Note that the IWAE with $K = 1$ equals the original VAE. We observed that as the value of K increased, the performance generally declined and saturated. The best performance was achieved when $K = 2$.

④ The performance of ALTBI was investigated with various learning rates using the Adam optimizer, ranging from $1e-4$ to $1e-1$. It was observed that when the learning rate exceeded $1e-3$, the model's performance began to deteriorate and then stabilized.

⑤ We found that increasing n_0 generally enhances performance, reaching its peak at 128, before gradually declining and stabilizing as n_0 surpasses 192. Nevertheless, the performance of ALTBI remains largely unaffected by variations in the choice of n_0.

⑥ Finding appropriate values of T_1 and T_2 for ensembling within a DGM single model affects ALTBI's performance but the impact is not significant. We compare three values of T_1 and four values of $T_2 - T_1$, then we can find that low values of T_1 and T_2 achieve lower

scores.

⑦ Additionally, we empirically find that ALTBI achieve near state-of-the-art performance in solving SSOD tasks as well. We provide verification of this in Appendix B.

5.6 Further discussions: Robustness of ALTBI in DP

A representative method to ensure that a given algorithm satisfies differential privacy (DP) is by training with the DP-SGD algorithm (Abadi et al., 2016) instead of conventional SGDs. This involves clipping the gradient norm for each per-sample loss and adding Gaussian noise. For a given loss function $\tilde{l}(\theta; \boldsymbol{x})$, this operation can be formularized as

$$\text{Clip}\big(\nabla_\theta \tilde{l}(\theta, \boldsymbol{x}); C\big) + \mathcal{N}(0, \sigma^2 C^2 I),$$

where $C > 0$ is a clipping constant and $\sigma > 0$ controls the noise amount.

We note that ALTBI utilizes $\tilde{l}(\theta; \boldsymbol{x}) = l(\theta; \boldsymbol{x}) I(l(\theta; \boldsymbol{x}) \leq \tau)$. When a sample is filtered out from the truncated loss, its gradient is already clipped to have a norm of zero. Since outliers are mostly excluded by the truncated loss, leading to their gradients being clipped to zero, we can infer that incorporating DP-SGD into ALTBI preserves the inliers' information relative to outliers, making ALTBI inherently robust when implementing DP-SGD.

To validate our claim, we conduct an additional experiment by analyzing 20 tabular datasets. We consider two versions of ALTBI: one that applies mini-batch increment and truncated loss, i.e., $(\gamma, \rho) = (1.03, 0.92)$, and one that does not,i.e., $(\gamma, \rho) = (1.0, 1.0)$. As a measure of DP, we adopt (ϵ, δ)-DP, and with a fixed $\delta = 1e - 5$, we train them using a DP-SGD algorithm until the cumulative privacy budget $\epsilon \leq 10$ holds. Then we compare the outlier detection AUC values. The modified ALTBI for DP and detailed results are provided in Appendix B.

Figure 7 shows that increasing mini-batch sizes and using the truncated loss function

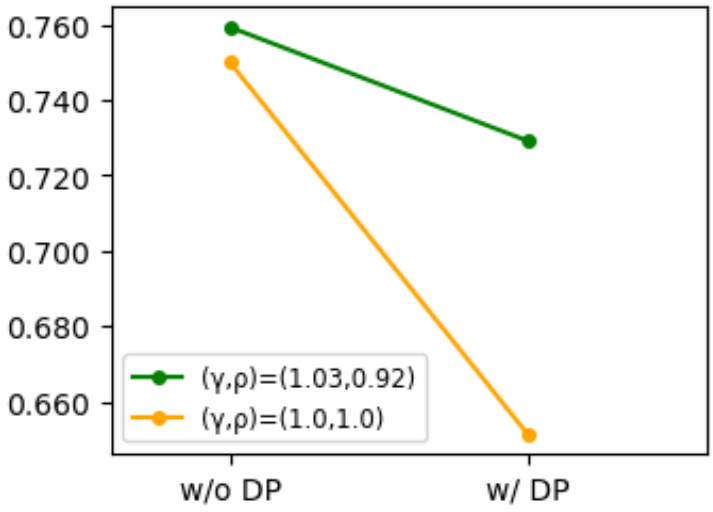

Figure 7: The impact of mini-batch increment and loss truncation when applying an DP-SGD algorithm.

yields more robust performance when applying the DP algorithm, indicating that combining ALTBI and DP has a synergistic effect.

Chapter 6

Concluding remarks

In this study, we presented ALTBI as a powerful and efficient framework for UOD. By leveraging the IM effect through innovative techniques such as incremental mini-batch size increases and adaptive loss truncation, ALTBI consistently demonstrates superior performance across a wide range of datasets. The efficiency of the method is particularly outstanding, as it achieves state-of-the-art results with significantly reduced computational costs. The ability to balance performance and resource utilization makes it a promising off-the-shelf solution for UOD tasks in various practical applications.

The study further confirms the robustness of ALTBI by evaluating its performance across 57 various datasets, showing that ALTBI not only outperforms existing methods but also maintains stability with lower standard deviations in results. This consistency highlights the method's reliability in different data environments, whether addressing tabular, image, or text data. Additionally, the use of under-fitted DGMs, like IWAE and GLOW, further enhances the applicability, and provides a effective approach to outlier detection.

The ensemble strategies within a single DGM during the later stages of training adds another layer of robustness to the method, ensuring that the model's performance remains strong even as training progresses. The findings suggest that ALTBI is not only a powerful tool for current UOD tasks but also a flexible and scalable approach that can adapt to future challenges in anomaly detection. Several studies have extended outlier detection tasks to scenarios where a few outliers with known outlier information are accessible (Ruff et al., 2020; Kim et al., 2024). Applying our method to the more complex case where some labeled outliers are wrongly annotated would be an interesting direction for future work.

Appendix

A. Proof of Theoretical results

Our theoretical results and their proofs are similar to Theorem 1 in Xu et al. (2021). Before starting to prove Proposition 1&2, we first state three lemmas that are used throughout our proofs.

Lemma 1. *If the IM assumption is satisfied, there exists $a_4 > 0$ such that the following inequality holds:*

$$P_o\left(l(\theta; X) \leq L_i(\theta)\right) \leq a_4 \cdot L_i(\theta). \tag{6.1}$$

proof) We have the following inequalities:

$$\begin{aligned} P_o(l(\theta; X) \leq L_i(\theta)) \quad &= \quad P_o(l(\theta; X) - L_o(\theta) \leq -(L_o(\theta) - L_i(\theta)) \\ &\leq \quad P_o(|l(\theta; X) - L_o(\theta)| \geq (L_o(\theta) - L_i(\theta)) \\ &\overset{\text{(Markov's Ineq.)}}{\leq} \quad \frac{\mathrm{E}_o|l(\theta; X) - L_o(\theta)|}{L_o(\theta) - L_i(\theta)} \\ &\leq \quad \frac{\mathrm{E}_o|l(\theta; X)| + L_o(\theta)}{L_o(\theta) - L_i(\theta)} \leq \frac{2L_o(\theta)}{a_3}. \end{aligned}$$

Since $a_1 \leq L_i(\theta)$, we have $L_o(\theta) \leq 1 \leq L_i(\theta)/a_1$. Therefore, we have

$$P_o(l(\theta; X) \leq L_i(\theta)) \leq \frac{2L_o(\theta)}{a_3} \leq \frac{2}{a_1 a_3} L_i(\theta),$$

and the proof is completed with $a_4 = 2/(a_1 a_3)$. □

Lemma 2 (Conditional version of Theorem 3.6 in Chung and Lu (2006))**.** *Suppose that*

Y_i for $i \in [n]$ are random variables satisfying $Y_i \leq M$ and $\mathcal{H}$ is a given σ-algebra. Let assume that the conditional expectations $\mathrm{E}(Y_i|\mathcal{H})$s are independent. Let $Y = \sum_{i=1}^n Y_i$ and $||Y||_{\mathcal{H}} = \sqrt{\sum_{i=1}^n \mathrm{E}(Y_i^2|\mathcal{H})}$. Then, for any $\lambda > 0$, we have

$$P\left(Y \geq \mathrm{E}(Y|\mathcal{H}) + \lambda \Big| \mathcal{H}\right) \leq \exp\left(-\frac{\lambda^2}{2(||Y||_{\mathcal{H}}^2 + M\lambda/3)}\right) \text{ a.s..}$$

Lemma 3 (Conditional version of Lemma 4 in Ghadimi et al. (2016))**.** *Suppose that Y_is, $i \in [n]$, are random variables with mean zero and $\mathcal{H}$ is a given σ-algebra. Let us assume that there exist positive values $\sigma_i^2 > 0$, $i \in [n]$, such that $E(Y_i)^2 \leq \sigma_i^2$. We also assume that the conditional expectations $\mathrm{E}(Y_i|\mathcal{H})$s are independent. Then for any $\lambda > 0$, the following holds:*

$$P\left(\left\|\sum_{i=1}^n Y_i^2\right\| \geq \lambda \sum_{i=1}^n \sigma_i^2 \Big| \mathcal{H}\right) \leq \frac{1}{\lambda} \text{ a.s..}$$

A.1 Proof of Proposition 1.

※We prove this proposition with a probability of $1 - 4\delta$. Transforming $1 - 4\delta$ to $1 - \delta$ is trivial by substituting δ with $\delta/4$.

Before we carry out our analysis, we define a few important constants

$$\begin{aligned}
C_1 &= \sqrt{\frac{\log(2/\delta)}{2c^2(1-\alpha)n_0}}, \\
C_2 &= \max\left(\sqrt{\frac{\log(2/\delta)}{2(1-\alpha))^2 n_0}}, \sqrt{\frac{\log(2/\delta)}{2\alpha^2 n_0}}\right), \\
C_3 &= c(1-\alpha)(1-C_1)(1-C_2), \\
C_4 &= \left(4\alpha(1+C_2)a_2 a_4 + \frac{1}{3}\log(1/\delta)\right),
\end{aligned}$$

where a_4 is the same constant as in Lemma 1. For the mini-batch $\mathcal{D}_t = \{X_1, \ldots, X_{n_t}\}$

with $n_t = n_0\gamma^{t-1}$ training examples sampled from $\mathcal{D}$, we divide it into two sets for the analysis use only, i.e. set $\mathcal{A}_t$ that includes examples sampled from $\mathcal{P}_i$ and set $\mathcal{B}_t$ that includes examples sampled from $\mathcal{P}_o$. We furthermore denote by $\mathcal{A}_t^\tau$ and $\mathcal{B}_t^\tau$ the subset of examples in $\mathcal{A}_t$ and $\mathcal{B}_t$ whose loss is smaller than the given threshold τ_t, i.e.

$$\mathcal{A}_t^\tau = \{X \in \mathcal{A}_t : l(\theta_{t-1}; X) \leq \tau_t\},$$
$$\mathcal{B}_t^\tau = \{X \in \mathcal{B}_t : l(\theta_{t-1}; X) \leq \tau_t\},$$

where $\tau_t = L_i(\theta_{t-1})$ and θ_{t-1} is the currently estimated parameter at the $(t-1)$-th update. We assume that $L_i(\theta_{t-1}) \leq a_2\gamma^{-(t-1)}$. Evidently, the samples used for computing $\mathbf{g}_t$ are the union of $\mathcal{A}_t^\tau$ and $\mathcal{B}_t^\tau$. The following result bounds the size of $\mathcal{A}_t^\tau$ and $\mathcal{B}_t^\tau$. With a probability $1 - 4\delta$. We have

$$|\mathcal{A}_t^\tau| \geq C_3 n_t = C_3 n_0 \gamma^{t-1}, \;\; |\mathcal{B}_t^\tau| \leq C_4 n_0,$$

where C_3 and C_4 are defined above.

(i) Lower bound of $\mathcal{A}_t^\tau$ By Hoeffding's inequality, for any $t > 0$, we have

$$P_*(||\mathcal{A}_t| - (1-\alpha)n_t| \geq -t) \geq P_*(||\mathcal{A}_t| - (1-\alpha)n_t| \leq t) \geq 1 - 2\exp(-2t^2/n_t).$$

By substituting $t = \sqrt{\frac{n_t \log(2/\delta)}{2}}$ with $\delta > 0$, we have

$$|\mathcal{A}_t| \geq (1-\alpha)n_t \left(1 - \sqrt{\frac{\log(2/\delta)}{2(1-\alpha)^2 n_t}}\right) \geq (1-\alpha)(1-C_2)n_t, \tag{6.2}$$

with a probability at least $1 - \delta$.

By using the general Markov inequality,

$$P_i(l(\theta_{t-1};X) \le \tau_t) = 1 - P_i(l(\theta_{t-1};X) \ge \tau_t) \ge 1 - \frac{\mathrm{E}_{P_i}[\sqrt{l(\theta_{t-1};X)}]}{\sqrt{L_i(\theta_{t-1})}} \ge c,$$

where the last inequality holds due to Jensen's inequality.

Let $\mathcal{F}_t := \mathcal{F}\left(I(X_j \in \mathcal{X}_i), j \in [n_t]\right)$. We apply the conditional Hoeffding's inequality to achieve the following: for any $t > 0$,

$$\begin{aligned} P_*\Big(|\mathcal{A}_t^\tau| - c|\mathcal{A}_t| \ge -t\Big|\mathcal{F}_t\Big) &\ge P\Big(\big||\mathcal{A}_t^\tau| - \mathrm{E}(|\mathcal{A}_t^\tau|)\big| \le t\Big|\mathcal{F}_t\Big) \\ &\ge 1 - 2\exp\big(-\frac{2t^2}{|\mathcal{A}_t|}\big) \text{ a.s.} \end{aligned}$$

With $t = \sqrt{\frac{|\mathcal{A}_t|}{2}\log(2/\delta)}$,

$$P_*\left(|\mathcal{A}_t^\tau| \ge c|\mathcal{A}_t|\left(1 - \sqrt{\frac{\log(2/\delta)}{2c^2|\mathcal{A}_t|}}\right)\Big|\mathcal{F}_t\right) \ge 1 - \delta \text{ a.s.},$$

and hence

$$P_*\left(|\mathcal{A}_t^\tau| \ge c|\mathcal{A}_t|\left(1 - \sqrt{\frac{\log(2/\delta)}{2c^2|\mathcal{A}_t|}}\right)\right) \ge 1 - \delta. \tag{6.3}$$

Combining (6.2) and (6.3), we have

$$\begin{aligned} |\mathcal{A}_t^\tau| &\ge c|\mathcal{A}_t|\left(1 - \sqrt{\frac{\log(2/\delta)}{2c^2|\mathcal{A}_t|}}\right) \\ &\ge c(1-\alpha)(1-C_2)n_t\left(1 - \sqrt{\frac{\log(2/\delta)}{2c^2(1-\alpha)(1-C_2)n_t}}\right) \\ &\ge c(1-\alpha)(1-C_2)(1-C_1)n_t \\ &= C_3 n_t, \end{aligned} \tag{6.4}$$

with a probability at least $1 - 2\delta$.

(ii) Upper bound of $\mathcal{B}_t^\tau$ And by Hoeffiding's inequality, we also have

$$P_*(|\mathcal{B}_t| - \alpha n_t \leq t) \geq P_*(||\mathcal{B}_t| - \alpha n_t| \geq t) \geq 1 - 2\exp(-2t^2/n_t).$$

By substituting $t = \sqrt{\frac{n_t \log(2/\delta)}{2}}$, we have

$$|\mathcal{B}_t| \leq \alpha n_t \left(1 + \sqrt{\frac{\log(2/\delta)}{2\alpha^2 n_t}}\right) \leq \alpha(1 + C_2)n_t, \tag{6.5}$$

with a probability at least $1 - \delta$. Also, by using Lemma 1, the following inequalities hold:

$$\begin{aligned} P_o(l(\theta_{t-1}; X) \leq \tau_t) &= P_o(l(\theta_{t-1}; X) \leq L_i(\theta_{t-1})) \\ &\overset{(Lem.1)}{\leq} a_4 \cdot L_i(\theta_{t-1}) \leq a_2 a_4 \gamma^{-(t-1)}. \end{aligned} \tag{6.6}$$

Let $\mathcal{G}_t := \mathcal{F}\left(I(X_j \in \mathcal{X}_o), j \in [n_t]\right)$. We can bound the expectation of $|\mathcal{B}_t^\tau|$ given $\mathcal{G}_t$, i.e.,

$$\begin{aligned} \mathrm{E}_*\Big[|\mathcal{B}_t^\tau|\Big|\mathcal{G}_t\Big] &= \mathrm{E}_*\left[\sum_{X \in \mathcal{B}_t} I\big(l(\theta_{t-1}; X) \leq \tau_t\big)\Big|\mathcal{G}_t\right] \\ &\leq |\mathcal{B}_t| P_o\big(l(\theta_{t-1}; X) \leq \tau_t\big) \overset{(6.6)}{\leq} |\mathcal{B}_t| a_2 a_4 \gamma^{-(t-1)} \text{ a.s..} \end{aligned} \tag{6.7}$$

Also, by using Lemma 2, for any $\lambda > 0$, the following inequality holds:

$$P_*\left(|\mathcal{B}_t^\tau| \leq \mathrm{E}_*\left[|\mathcal{B}_t^\tau||\mathcal{G}_t\right] + \lambda\Big|\mathcal{G}_t\right) \geq 1 - \exp\left(-\frac{\lambda^2}{2\mathrm{E}_*\Big[|\mathcal{B}_t^\tau|\Big|\mathcal{G}_t\Big] + \lambda/3}\right) \text{ a.s.}$$

For a given $\delta > 0$, by substituting

$\lambda = \frac{1}{3}\log(1/\delta) + \sqrt{\frac{1}{9}\log^2(1/\delta) + 2\log(1/\delta)\mathrm{E}_*\left[|\mathcal{B}_t^\tau|\middle|\mathcal{G}_t\right]}$, we have

$$P_*\left(|\mathcal{B}_t^\tau| \leq \mathrm{E}_*\left[|\mathcal{B}_t^\tau|\middle|\mathcal{G}_t\right] + \frac{1}{3}\log(1/\delta) + \sqrt{\frac{1}{9}\log^2(1/\delta) + 2\log(1/\delta)\mathrm{E}_*\left[|\mathcal{B}_t^\tau|\middle|\mathcal{G}_t\right]}\middle|\mathcal{G}_t\right) \geq 1-\delta \text{ a.s.},$$

and hence

$$P_*\left(|\mathcal{B}_t^\tau| \leq \mathrm{E}_*\left[|\mathcal{B}_t^\tau|\middle|\mathcal{G}_t\right] + \frac{1}{3}\log(1/\delta) + \sqrt{\frac{1}{9}\log^2(1/\delta) + 2\log(1/\delta)\mathrm{E}_*\left[|\mathcal{B}_t^\tau|\middle|\mathcal{G}_t\right]}\right) \geq 1-\delta. \tag{6.8}$$

We combine (6.5), (6.7), and (6.8) to achieve the inequality below:

$$\begin{aligned}
|\mathcal{B}_t^\tau| &\leq \mathrm{E}_*\left[|\mathcal{B}_t^\tau|\middle|\mathcal{G}_t\right] + \frac{1}{3}\log(1/\delta) + \sqrt{\frac{1}{9}\log^2(1/\delta) + 2\log(1/\delta)\mathrm{E}_*\left[|\mathcal{B}_t^\tau|\middle|\mathcal{G}_t\right]} \\
&\leq 4\mathrm{E}_*\left[|\mathcal{B}_t^\tau|\middle|\mathcal{G}_t\right] + \frac{2}{3}\log(1/\delta) \\
&\leq 4|\mathcal{B}_t|a_2a_4\gamma^{-(t-1)} + \frac{2}{3}\log(1/\delta) \\
&\leq 4\alpha(1+C_2)n_0\gamma^{t-1}a_2a_4\gamma^{-(t-1)} + \frac{2}{3}\log(1/\delta) \\
&= 4\alpha(1+C_2)n_0a_2a_4 + \frac{2}{3}\log(1/\delta) \\
&\leq \left(4\alpha(1+C_2)a_2a_4 + \frac{1}{3}\log(1/\delta)\right)n_0 = C_4n_0.
\end{aligned} \tag{6.9}$$

with a probability at least $1-2\delta$.

Therefore, with (6.4) and (6.9), the proof is completed with $c_1 = C_3$ and $c_2 = C_4$. □

A.2 Proof of Proposition 2

※Similar to the proof of Proposition 1, we prove Proposition 2 with a probability of $1-5\delta$. Transforming $1-5\delta$ to $1-\delta$ can be done by using $\delta/5$ instead of δ.

We will demonstrate that, with high probability, $L(\theta_t) \leq a_2\gamma^{-t}$. Let $\mathbf{g}_t = \nabla_\theta \hat{L}(\theta_{t-1}, \tau_t)$. To this end, using the notation of $\mathcal{A}_t^\tau$ and $\mathcal{B}_t^\tau$, we can rewrite $\mathbf{g}_t$ as

$$\mathbf{g}_t = (1 - b_t)\mathbf{g}_t^a + b_t\mathbf{g}_t^b,$$

where $\mathbf{g}_t^a = \frac{1}{|\mathcal{A}_t^\tau|}\sum_{X\in\mathcal{A}_t^\tau} \nabla l(\theta_{t-1}; X)$, $\mathbf{g}_t^b = \frac{1}{|\mathcal{B}_t^\tau|}\sum_{X\in\mathcal{B}_t^\tau} \nabla l(\theta_{t-1}; X)$, and b_t is the proportion of samples from $\mathcal{B}_t^\tau$ that

$$b_t = \frac{|\mathcal{B}_t^\tau|}{|\mathcal{A}_t^\tau| + |\mathcal{B}_t^\tau|} \leq \frac{|\mathcal{B}_t^\tau|}{|\mathcal{A}_t^\tau|} \leq \frac{c_2}{1 + c_1\gamma^{t-1}} < \frac{c_2}{c_1}\gamma^{-(t-1)},$$

with a probability of at least $1 - 4\delta$ by Proposition 1. Following the classical analysis of non-convex optimization, since $L(\theta)$ is L-smooth by Assumption 1 (ii), we have

$$\begin{aligned}
L_i(\theta_t) - L_i(\theta_{t-1}) &\overset{(a)}{\leq} \langle \nabla L_i(\theta_{t-1}), \theta_t - \theta_{t-1}\rangle + \frac{L}{2}\|\theta_t - \theta_{t-1}\|^2 \\
&\overset{(b)}{=} \frac{\eta}{2}\|\nabla L_i(\theta_{t-1}) - \mathbf{g}_t\|^2 - \frac{\eta}{2}\left(\|\nabla L_i(\theta_{t-1})\|^2 + (1 - \eta L)\|\mathbf{g}_t\|^2\right) \\
&\overset{(c)}{\leq} \frac{\eta}{2}\left((1 - b_t)\|\nabla L_i(\theta_{t-1}) - \mathbf{g}_t^a\|^2 + b_t\|\nabla L_i(\theta_{t-1}) - \mathbf{g}_t^b\|^2\right) \\
&\quad - \frac{\eta}{2}\left(\|\nabla L_i(\theta_{t-1})\|^2 + (1 - \eta L)\|\mathbf{g}_t\|^2\right) \\
&\overset{(d)}{\leq} \frac{\eta}{2}\left((1 - b_t)\|\nabla L_i(\theta_{t-1}) - \mathbf{g}_t^a\|^2 + 4b_tG^2\right) - \eta\mu L_i(\theta_{t-1}), \qquad (6.10)
\end{aligned}$$

where (a) is due to Assumption 2-(i); (b) follows the update of $\theta_t = \theta_{t-1} - \eta\mathbf{g}_t$; (c) is due to the definition of $\mathbf{g}_t$ and the convexity of $\|\cdot\|^2$; (d) follows the Assumption 2-(i),(ii), and $\eta L \leq 1$.

Let $\mathcal{F}_t^\tau := \mathcal{F}\left(I(X_j \in \mathcal{A}_t^\tau), j \in [n_t]\right)$. Then we have

$$\begin{aligned}
\mathrm{E}_*\left(\left\|\mathbf{g}_t^a - \nabla L_i(\theta_{t-1})\right\|^2 \middle| \mathcal{F}_t^\tau\right) &= \frac{1}{|\mathcal{A}_t^\tau|^2}\mathrm{E}_*\left(\left\|\sum_{X\in\mathcal{A}_t^\tau}\left(\nabla l(\theta_t; X) - \nabla L(\theta_t)\right)\right\|^2 \middle| \mathcal{F}_t^\tau\right) \\
&= \frac{1}{|\mathcal{A}_t^\tau|^2}\mathrm{E}_*\left(\sum_{X\in\mathcal{A}_t^\tau}\left\|\nabla l(\theta_t; X) - \nabla L(\theta_t)\right\|^2 \middle| \mathcal{F}_t^\tau\right) \\
&\leq \frac{4G^2}{|\mathcal{A}_t^\tau|},
\end{aligned}$$

where the last inequality holds due to Assumption 2-(i). For a given $\delta > 0$, by using Lemma 3 with $\lambda = 1/\delta$, we have

$$P_*\left(P_*\left(\left\|\mathbf{g}_t^a - \nabla L_i(\theta_{t-1})\right\|^2 \leq \frac{4G^2}{\delta|\mathcal{A}_t^\tau|}\middle| \mathcal{F}_t^\tau\right) \geq 1-\delta\right) = 1,$$

and hence

$$P_*\left(\left\|\mathbf{g}_t^a - \nabla L_i(\theta_{t-1})\right\|^2 \leq \frac{4G^2}{\delta|\mathcal{A}_t^\tau|}\right) \geq 1-\delta. \tag{6.11}$$

We combine (6.4) and (6.11) to achieve

$$\left\|\mathbf{g}_t^a - \nabla L_i(\theta_{t-1})\right\|^2 \leq \frac{4G^2}{\delta c_1 n_0}\gamma^{-(t-1)}, \tag{6.12}$$

with a probability of at least $1-5\delta$. Using the above bound in (6.12), we can further expand the bound in (6.10) as follows:

$$\begin{aligned}
L_i(\theta_t) - L_i(\theta_{t-1}) &\leq \frac{\eta}{2}\left((1-b_t)\frac{4G^2}{\delta c_1 n_0}\gamma^{-(t-1)} + 4b_t G^2\right) - \eta\mu L_i(\theta_{t-1}) \\
&\leq \frac{\eta}{2}\left(\frac{4G^2}{\delta c_1 n_0}\gamma^{-(t-1)} + 4G^2\frac{c_2}{c_1}\gamma^{-(t-1)}\right) - \eta\mu L_i(\theta_{t-1}) \\
&= \frac{2\eta G^2}{c_1}\left(\frac{1}{\delta n_0} + c_2\right)\gamma^{-(t-1)} - \eta\mu L_i(\theta_{t-1}).
\end{aligned}$$

Hence, we have

$$\begin{aligned} L_i(\theta_t) &\leq (1-\eta\mu)L_i(\theta_{t-1}) + \frac{2\eta G^2}{c_1}\left(\frac{1}{\delta n_0} + c_2\right)\gamma^{-(t-1)} \\ &\leq \gamma\left[(1-\eta\mu)a_2 + \frac{2\eta G^2}{c_1}\left(\frac{1}{\delta n_0} + c_2\right)\right]\gamma^{-t}, \end{aligned}$$

with a probability of at least $1-5\delta$. Let us select the learning rate η between the interval given as:

$$\frac{1}{\mu}\left(1-\frac{1}{2\gamma}\right) \leq \eta \leq \frac{a_2 c_1}{4\gamma G^2\left(\frac{1}{\delta n_0} + c_2\right)}.$$

Then, we have

$$\begin{aligned} L_i(\theta_t) &\leq \gamma\left[(1-\eta\mu)a_2 + \frac{2\eta G^2}{c_1}\left(\frac{1}{\delta n_0} + c_2\right)\right]\gamma^{-t} \\ &\leq \gamma\left(\frac{a_2}{2\gamma} + \frac{a_2}{2\gamma}\right)\gamma^{-t} \\ &= a_2\gamma^{-t}, \end{aligned}$$

and the proof is completed. □

B. Detailed experiment results

B.1 Data description

We evaluate a total of 46 tabular datasets, 6 image datasets, and 5 text datasets. These datasets are all obtained from a source known as ADBench. (Han et al., 2022b). Table B.1 provides a summary of the basic information for all datasets we analyze.

Number	Dataset Name	#Samples	#Features	#Anomaly	%Anomaly	Category
1	ALOI	49534	27	1508	3.04	Image
2	annthyroid	7200	6	534	7.42	Healthcare
3	backdoor	95329	196	2329	2.44	Network
4	breastw	683	9	239	34.99	Healthcare
5	campaign	41188	62	4640	11.27	Finance
6	cardio	1831	21	176	9.61	Healthcare
7	Cardiotocography	2114	21	466	22.04	Healthcare
8	celeba	202599	39	4547	2.24	Image
9	census	299285	500	18568	6.2	Sociology
10	cover	286048	10	2747	0.96	Botany
11	donors	619326	10	36710	5.93	Sociology
12	fault	1941	27	673	34.67	Physical
13	fraud	284807	29	492	0.17	Finance
14	glass	214	7	9	4.21	Forensic
15	Hepatitis	80	19	13	16.25	Healthcare
16	http	567498	3	2211	0.39	Web
17	InternetAds	1966	1555	368	18.72	Image
18	Ionosphere	351	32	126	35.9	Oryctognosy
19	landsat	6435	36	1333	20.71	Astronautics
20	letter	1600	32	100	6.25	Image
21	Lymphography	148	18	6	4.05	Healthcare
22	magic.gamma	19020	10	6688	35.16	Physical
23	mammography	11183	6	260	2.32	Healthcare
24	mnist	7603	100	700	9.21	Image
25	musk	3062	166	97	3.17	Chemistry
26	optdigits	5216	64	150	2.88	Image
27	PageBlocks	5393	10	510	9.46	Document
28	pendigits	6870	16	156	2.27	Image
29	Pima	768	8	268	34.9	Healthcare
30	satellite	6435	36	2036	31.64	Astronautics
31	satimage-2	5803	36	71	1.22	Astronautics
32	shuttle	49097	9	3511	7.15	Astronautics
33	skin	245057	3	50859	20.75	Image
34	smtp	95156	3	30	0.03	Web
35	SpamBase	4207	57	1679	39.91	Document
36	speech	3686	400	61	1.65	Linguistics
37	Stamps	340	9	31	9.12	Document
38	thyroid	3772	6	93	2.47	Healthcare
39	vertebral	240	6	30	12.5	Biology
40	vowels	1456	12	50	3.43	Linguistics
41	Waveform	3443	21	100	2.9	Physics
42	WBC	223	9	10	4.48	Healthcare
43	WDBC	367	30	10	2.72	Healthcare
44	Wilt	4819	5	257	5.33	Botany
45	wine	129	13	10	7.75	Chemistry
46	WPBC	198	33	47	23.74	Healthcare
47	yeast	1484	8	507	34.16	Biology
48	CIFAR10	5263	512	263	5	Image
49	FashionMNIST	6315	512	315	5	Image
50	MNIST-C	10000	512	500	5	Image
51	MVTec-AD	5354	512	1258	23.5	Image
52	SVHN	5208	512	260	5	Image
53	Agnews	10000	768	500	5	NLP
54	Amazon	10000	768	500	5	NLP
55	Imdb	10000	768	500	5	NLP
56	Yelp	10000	768	500	5	NLP
57	20news	11905	768	591	4.96	NLP

Table B.1: Description of `ADBench` datasets

B.2 Detailed AUC and PRAUC results over ADBench datasets

Table B.2.1- B.5.2 provide detailed results of averaged AUC and PRAUC for each method over the `ADBench` datasets in unsupervised and semi-supervised settings.

	CBLOF	COPOD	ECOD	FeatureBagging	HBOS	IForest	kNN	LODA	LOF	MCD	OCSVM	PCA
aloi	0.556	0.515	0.531	0.792	0.531	0.542	0.613	0.495	0.767	0.520	0.549	0.549
annthyroid	0.676	0.777	0.789	0.788	0.608	0.816	0.761	0.453	0.710	0.918	0.682	0.676
backdoor	0.897	0.500	0.500	0.790	0.740	0.725	0.826	0.515	0.764	0.848	0.889	0.888
breastw	0.961	0.994	0.990	0.408	0.984	0.983	0.980	0.970	0.446	0.985	0.935	0.946
campaign	0.738	0.783	0.769	0.594	0.768	0.704	0.750	0.493	0.614	0.775	0.737	0.734
cardio	0.832	0.921	0.935	0.579	0.839	0.922	0.830	0.856	0.551	0.815	0.934	0.949
cardiotocography	0.561	0.664	0.784	0.538	0.595	0.681	0.503	0.708	0.527	0.500	0.691	0.747
celeba	0.753	0.757	0.763	0.514	0.754	0.707	0.736	0.600	0.432	0.803	0.781	0.792
census	0.664	0.500	0.500	0.538	0.611	0.607	0.671	0.454	0.562	0.731	0.655	0.662
cover	0.922	0.882	0.919	0.571	0.707	0.873	0.866	0.922	0.568	0.696	0.952	0.934
donors	0.808	0.815	0.888	0.691	0.743	0.771	0.829	0.566	0.629	0.765	0.770	0.825
fault	0.665	0.455	0.468	0.591	0.506	0.544	0.715	0.478	0.579	0.505	0.537	0.480
fraud	0.954	0.943	0.949	0.616	0.945	0.950	0.955	0.856	0.548	0.911	0.954	0.952
glass	0.855	0.760	0.710	0.659	0.820	0.790	0.870	0.624	0.618	0.795	0.661	0.715
hepatitis	0.635	0.807	0.737	0.469	0.768	0.683	0.669	0.557	0.468	0.721	0.704	0.748
http	0.996	0.991	0.980	0.288	0.991	1.000	0.051	0.060	0.338	1.000	0.994	0.997
internetads	0.616	0.676	0.677	0.494	0.696	0.686	0.616	0.541	0.587	0.660	0.615	0.609
ionosphere	0.892	0.783	0.717	0.876	0.544	0.833	0.922	0.788	0.864	0.951	0.838	0.777
landsat	0.548	0.422	0.368	0.540	0.575	0.474	0.614	0.382	0.549	0.607	0.423	0.366
letter	0.763	0.560	0.573	0.886	0.589	0.616	0.812	0.537	0.878	0.804	0.598	0.524
lymphography	0.994	0.996	0.995	0.523	0.995	0.999	0.995	0.900	0.636	0.989	0.996	0.997
magic.gamma	0.725	0.681	0.638	0.700	0.709	0.721	0.795	0.655	0.678	0.699	0.673	0.667
mammography	0.795	0.905	0.906	0.726	0.838	0.860	0.852	0.867	0.702	0.690	0.871	0.888
musk	1.000	0.948	0.953	0.575	1.000	0.998	0.964	0.993	0.581	1.000	1.000	1.000
optdigits	0.785	0.500	0.500	0.539	0.868	0.696	0.395	0.493	0.538	0.413	0.507	0.518
pageblocks	0.893	0.875	0.914	0.758	0.779	0.897	0.919	0.712	0.703	0.923	0.914	0.907
pendigits	0.864	0.906	0.927	0.518	0.925	0.947	0.828	0.895	0.534	0.834	0.929	0.936
pima	0.655	0.662	0.604	0.573	0.704	0.674	0.723	0.595	0.563	0.686	0.631	0.651
satellite	0.742	0.633	0.583	0.545	0.762	0.695	0.721	0.614	0.550	0.804	0.662	0.601
satimage-2	0.999	0.975	0.965	0.526	0.976	0.993	0.992	0.981	0.539	0.995	0.997	0.977
shuttle	0.621	0.995	0.993	0.493	0.986	0.997	0.732	0.389	0.526	0.990	0.992	0.990
skin	0.675	0.471	0.490	0.534	0.588	0.670	0.720	0.442	0.550	0.892	0.547	0.447
smtp	0.863	0.912	0.882	0.794	0.809	0.905	0.933	0.819	0.899	0.948	0.845	0.856
spambase	0.541	0.688	0.656	0.424	0.664	0.637	0.566	0.480	0.453	0.446	0.534	0.548
speech	0.471	0.489	0.470	0.509	0.473	0.476	0.480	0.466	0.512	0.494	0.466	0.469
stamps	0.660	0.929	0.877	0.502	0.904	0.907	0.870	0.831	0.512	0.838	0.882	0.909
thyroid	0.909	0.939	0.977	0.707	0.948	0.979	0.965	0.819	0.657	0.986	0.958	0.955
vertebral	0.463	0.263	0.417	0.473	0.317	0.362	0.379	0.294	0.487	0.389	0.426	0.378
vowels	0.884	0.496	0.593	0.933	0.679	0.763	0.951	0.705	0.932	0.732	0.779	0.604
waveform	0.701	0.739	0.603	0.715	0.694	0.707	0.750	0.594	0.693	0.572	0.669	0.635
wbc	0.977	0.994	0.994	0.388	0.987	0.996	0.982	0.992	0.607	0.988	0.987	0.993
wdbc	0.990	0.993	0.971	0.867	0.989	0.988	0.980	0.980	0.849	0.969	0.984	0.988
wilt	0.396	0.345	0.394	0.666	0.348	0.451	0.511	0.313	0.678	0.859	0.317	0.239
wine	0.453	0.865	0.738	0.323	0.907	0.786	0.470	0.822	0.330	0.975	0.671	0.819
wpbc	0.487	0.519	0.489	0.436	0.548	0.516	0.512	0.501	0.447	0.534	0.485	0.486
yeast	0.461	0.380	0.443	0.465	0.402	0.394	0.396	0.461	0.453	0.406	0.420	0.418
CIFAR10	0.663	0.548	0.567	0.687	0.572	0.629	0.659	0.591	0.686	0.639	0.663	0.659
MNIST-C	0.757	0.500	0.500	0.702	0.689	0.733	0.786	0.591	0.699	0.739	0.751	0.741
MVTec-AD	0.754	0.500	0.500	0.745	0.732	0.747	0.763	0.644	0.742	0.618	0.735	0.724
SVHN	0.601	0.500	0.500	0.629	0.542	0.580	0.604	0.534	0.628	0.583	0.604	0.599
mnist	0.843	0.500	0.500	0.664	0.574	0.811	0.867	0.564	0.658	0.856	0.849	0.848
FashionMNIST	0.871	0.500	0.500	0.748	0.748	0.831	0.875	0.672	0.738	0.840	0.860	0.853
20news	0.564	0.533	0.544	0.610	0.537	0.550	0.567	0.539	0.610	0.583	0.559	0.545
agnews	0.619	0.551	0.552	0.715	0.554	0.584	0.647	0.568	0.714	0.665	0.601	0.566
amazon	0.579	0.571	0.541	0.572	0.563	0.558	0.603	0.526	0.571	0.597	0.565	0.550
imdb	0.496	0.512	0.471	0.499	0.499	0.489	0.494	0.466	0.500	0.504	0.484	0.478
yelp	0.635	0.605	0.578	0.661	0.600	0.602	0.670	0.581	0.661	0.655	0.621	0.592
Average	0.731	0.692	0.689	0.608	0.714	0.737	0.728	0.627	0.611	0.748	0.725	0.718

Table B.2.1: ROC AUC for the unsupervised setting on ADBench (1)

	DAGMM	DeepSVDD	DROCC	GOAD	ICL	PlanarFlow	DDPM	DTE-NP	DTE-IG	DTE-C	ODIM	ALTBI_G	ALTBI_I
aloi	0.517	0.514	0.500	0.497	0.548	0.520	0.532	0.645	0.541	0.525	0.527	0.545	0.534
amazon	0.548	0.739	0.631	0.453	0.599	0.966	0.814	0.781	0.923	0.964	0.603	0.864	0.642
annthyroid	0.752	0.735	0.500	0.587	0.936	0.787	0.892	0.806	0.753	0.875	0.886	0.881	0.873
backdoor	0.811	0.625	0.847	0.845	0.807	0.965	0.766	0.976	0.905	0.891	0.992	0.989	0.981
breastw	0.581	0.508	0.500	0.443	0.766	0.566	0.724	0.746	0.660	0.789	0.727	0.746	0.725
campaign	0.625	0.498	0.655	0.908	0.461	0.796	0.723	0.777	0.631	0.721	0.911	0.714	0.857
cardio	0.546	0.488	0.449	0.624	0.372	0.643	0.579	0.493	0.506	0.510	0.649	0.483	0.542
cardiotocography	0.627	0.491	0.726	0.432	0.684	0.703	0.796	0.699	0.700	0.812	0.839	0.724	0.803
celeba	0.491	0.527	0.443	0.488	0.668	0.604	0.659	0.672	0.629	0.646	0.665	0.702	0.662
census	0.742	0.580	0.747	0.124	0.681	0.417	0.808	0.838	0.635	0.697	0.901	0.924	0.901
cover	0.558	0.511	0.747	0.225	0.739	0.899	0.806	0.832	0.796	0.785	0.813	0.810	0.601
donors	0.495	0.522	0.668	0.546	0.661	0.469	0.562	0.726	0.577	0.590	0.544	0.703	0.664
fault	0.857	0.769	0.500	0.724	0.931	0.895	0.924	0.956	0.942	0.938	0.940	0.957	0.923
fraud	0.630	0.517	0.743	0.545	0.729	0.766	0.560	0.881	0.681	0.864	0.708	0.812	0.771
glass	0.600	0.361	0.582	0.637	0.616	0.654	0.461	0.631	0.451	0.577	0.781	0.674	0.804
hepatitis	0.838	0.249	0.500	0.996	0.921	0.994	0.998	0.051	0.973	0.995	0.995	0.994	0.996
http	0.515	0.583	0.500	0.614	0.592	0.608	0.614	0.634	0.635	0.656	0.618	0.690	0.720
imdb	0.641	0.514	0.766	0.829	0.629	0.884	0.758	0.924	0.697	0.911	0.768	0.922	0.857
internetads	0.533	0.631	0.626	0.506	0.649	0.464	0.496	0.602	0.473	0.544	0.457	0.571	0.538
ionosphere	0.503	0.517	0.780	0.598	0.737	0.689	0.847	0.850	0.676	0.781	0.628	0.906	0.744
landsat	0.840	0.681	0.878	0.995	0.884	0.940	0.958	0.989	0.852	0.834	1.000	0.987	0.991
letter	0.584	0.604	0.728	0.442	0.676	0.742	0.763	0.801	0.782	0.765	0.728	0.829	0.743
lymphography	0.719	0.451	0.779	0.414	0.658	0.782	0.749	0.849	0.799	0.810	0.835	0.865	0.806
magic.gamma	0.912	0.538	0.575	1.000	0.790	0.748	1.000	0.882	0.785	0.965	1.000	1.000	1.000
mammography	0.408	0.519	0.565	0.657	0.533	0.492	0.402	0.386	0.513	0.508	0.570	0.536	0.684
mnist	0.753	0.592	0.914	0.609	0.768	0.908	0.820	0.906	0.850	0.924	0.886	0.887	0.868
musk	0.548	0.383	0.520	0.592	0.650	0.780	0.700	0.786	0.624	0.713	0.945	0.892	0.954
optdigits	0.522	0.510	0.542	0.606	0.524	0.615	0.537	0.707	0.599	0.624	0.689	0.713	0.717
pageblocks	0.675	0.562	0.608	0.702	0.627	0.671	0.715	0.702	0.582	0.711	0.712	0.677	0.760
pendigits	0.911	0.551	0.579	0.996	0.898	0.970	0.996	0.980	0.858	0.946	0.998	0.998	0.998
pima	0.898	0.576	0.500	0.208	0.642	0.852	0.975	0.698	0.669	0.976	0.985	0.996	0.982
satellite	0.554	0.548	0.708	0.579	0.265	0.773	0.461	0.718	0.741	0.741	0.626	0.812	0.874
satimage-2	0.868	0.895	0.500	0.915	0.656	0.784	0.956	0.930	0.769	0.951	0.844	0.872	0.846
shuttle	0.488	0.584	0.490	0.496	0.459	0.528	0.510	0.545	0.509	0.515	0.555	0.655	0.503
skin	0.522	0.512	0.483	0.458	0.512	0.496	0.466	0.487	0.488	0.495	0.465	0.457	0.477
smtp	0.719	0.465	0.760	0.774	0.505	0.838	0.556	0.820	0.692	0.753	0.920	0.646	0.883
spambase	0.719	0.505	0.889	0.574	0.693	0.992	0.871	0.964	0.828	0.990	0.917	0.977	0.947
speech	0.470	0.394	0.425	0.468	0.449	0.409	0.563	0.400	0.451	0.458	0.316	0.327	0.402
stamps	0.464	0.514	0.738	0.791	0.784	0.888	0.903	0.964	0.705	0.914	0.843	0.955	0.868
thyroid	0.523	0.609	0.674	0.592	0.661	0.640	0.617	0.729	0.523	0.602	0.700	0.678	0.746
vertebral	0.821	0.503	0.821	0.949	0.853	0.934	0.948	0.979	0.894	0.871	1.000	0.983	0.995
vowels	0.715	0.602	0.347	0.983	0.738	0.985	0.965	0.975	0.566	0.835	0.965	0.962	0.979
waveform	0.432	0.465	0.400	0.555	0.649	0.794	0.659	0.552	0.834	0.851	0.322	0.644	0.357
wbc	0.513	0.507	0.621	0.734	0.455	0.390	0.374	0.425	0.310	0.557	0.904	0.825	0.932
wdbc	0.449	0.493	0.483	0.466	0.488	0.483	0.493	0.502	0.489	0.468	0.544	0.539	0.495
wilt	0.503	0.520	0.396	0.503	0.466	0.442	0.463	0.400	0.446	0.420	0.398	0.419	0.411
wine	0.530	0.555	0.503	0.659	0.557	0.621	0.663	0.660	0.595	0.629	0.928	0.913	0.888
wpbc	0.581	0.552	0.594	0.752	0.670	0.705	0.751	0.788	0.703	0.746	0.737	0.743	0.740
yeast	0.596	0.603	0.544	0.730	0.683	0.637	0.732	0.761	0.655	0.730	0.941	0.850	0.880
yelp	0.528	0.521	0.521	0.597	0.571	0.580	0.605	0.607	0.567	0.600	0.553	0.541	0.581
MNIST-C	0.631	0.605	0.615	0.698	0.691	0.645	0.816	0.853	0.756	0.819	0.859	0.545	0.866
FashionMNIST	0.664	0.647	0.564	0.860	0.758	0.819	0.861	0.873	0.767	0.841	0.907	0.631	0.879
CIFAR10	0.518	0.515	0.496	0.553	0.547	0.513	0.547	0.570	0.527	0.579	0.716	0.675	0.722
SVHN	0.508	0.494	0.500	0.592	0.591	0.497	0.571	0.652	0.545	0.627	0.788	0.728	0.868
MVTec-AD	0.501	0.464	0.500	0.560	0.528	0.495	0.551	0.603	0.535	0.556	0.520	0.520	0.546
20news	0.487	0.526	0.500	0.486	0.521	0.493	0.478	0.495	0.486	0.484	0.508	0.488	0.528
agnews	0.498	0.524	0.504	0.590	0.545	0.527	0.594	0.671	0.514	0.602	0.551	0.556	0.556
Average	0.614	0.543	0.600	0.627	0.643	0.696	0.700	0.722	0.660	0.728	0.748	0.753	0.762

Table B.2.2: ROC AUC for the unsupervised setting on ADBench (2)

	CBLOF	COPOD	ECOD	FeatureBagging	HBOS	IForest	kNN	LODA	LOF	MCD	OCSVM	PCA
aloi	0.037	0.031	0.033	0.104	0.034	0.034	0.048	0.033	0.097	0.032	0.039	0.037
annthyroid	0.169	0.174	0.272	0.206	0.228	0.312	0.224	0.098	0.163	0.503	0.188	0.196
backdoor	0.547	0.025	0.025	0.217	0.052	0.045	0.479	0.101	0.358	0.122	0.534	0.531
breastw	0.890	0.989	0.982	0.284	0.954	0.956	0.932	0.955	0.297	0.962	0.897	0.946
campaign	0.287	0.368	0.354	0.145	0.352	0.279	0.289	0.131	0.158	0.325	0.283	0.284
cardio	0.482	0.576	0.567	0.161	0.458	0.559	0.402	0.428	0.159	0.364	0.536	0.609
cardiotocography	0.335	0.403	0.502	0.276	0.361	0.436	0.324	0.463	0.272	0.311	0.408	0.462
celeba	0.069	0.093	0.095	0.024	0.090	0.063	0.061	0.047	0.018	0.092	0.103	0.112
census	0.088	0.062	0.062	0.061	0.073	0.073	0.088	0.065	0.069	0.153	0.085	0.087
cover	0.070	0.068	0.113	0.019	0.026	0.052	0.054	0.090	0.019	0.016	0.099	0.075
donors	0.148	0.209	0.265	0.120	0.135	0.124	0.182	0.255	0.109	0.141	0.139	0.166
fault	0.473	0.313	0.325	0.396	0.360	0.395	0.522	0.337	0.388	0.334	0.401	0.332
fraud	0.145	0.252	0.215	0.003	0.209	0.145	0.169	0.146	0.003	0.488	0.110	0.149
glass	0.144	0.111	0.183	0.151	0.161	0.144	0.167	0.090	0.144	0.113	0.130	0.112
hepatitis	0.304	0.389	0.295	0.225	0.328	0.243	0.252	0.275	0.214	0.363	0.277	0.339
http	0.464	0.280	0.145	0.047	0.302	0.886	0.010	0.004	0.050	0.865	0.356	0.500
internetads	0.297	0.505	0.505	0.182	0.523	0.486	0.296	0.242	0.232	0.344	0.291	0.276
ionosphere	0.881	0.663	0.633	0.821	0.353	0.779	0.911	0.741	0.807	0.947	0.829	0.721
landsat	0.212	0.176	0.164	0.246	0.231	0.194	0.258	0.183	0.250	0.253	0.175	0.163
letter	0.166	0.068	0.077	0.445	0.078	0.086	0.203	0.083	0.433	0.174	0.113	0.076
lymphography	0.915	0.907	0.894	0.090	0.919	0.972	0.894	0.491	0.135	0.767	0.885	0.935
magic.gamma	0.666	0.588	0.533	0.539	0.617	0.638	0.724	0.579	0.520	0.632	0.625	0.589
mammography	0.140	0.430	0.435	0.070	0.132	0.218	0.181	0.218	0.085	0.036	0.187	0.204
musk	1.000	0.369	0.475	0.140	0.999	0.945	0.708	0.842	0.118	0.992	1.000	1.000
optdigits	0.059	0.029	0.029	0.036	0.192	0.046	0.022	0.029	0.035	0.022	0.027	0.027
pageblocks	0.547	0.370	0.520	0.341	0.319	0.464	0.556	0.410	0.292	0.617	0.531	0.525
pendigits	0.192	0.177	0.270	0.048	0.247	0.260	0.100	0.186	0.040	0.069	0.226	0.219
pima	0.484	0.536	0.484	0.412	0.577	0.510	0.530	0.404	0.406	0.498	0.477	0.492
satellite	0.656	0.570	0.526	0.378	0.688	0.649	0.582	0.613	0.381	0.768	0.654	0.606
satimage-2	0.972	0.797	0.666	0.042	0.760	0.918	0.690	0.857	0.041	0.682	0.965	0.872
shuttle	0.184	0.962	0.905	0.081	0.965	0.976	0.193	0.168	0.109	0.841	0.907	0.913
skin	0.289	0.179	0.183	0.207	0.232	0.254	0.290	0.180	0.221	0.490	0.220	0.172
smtp	0.403	0.005	0.589	0.001	0.005	0.005	0.415	0.312	0.022	0.006	0.383	0.382
spambase	0.402	0.544	0.518	0.344	0.518	0.488	0.415	0.387	0.360	0.349	0.402	0.409
speech	0.019	0.019	0.020	0.022	0.023	0.021	0.019	0.016	0.022	0.019	0.019	0.018
stamps	0.211	0.398	0.324	0.143	0.332	0.347	0.317	0.280	0.153	0.257	0.318	0.364
thyroid	0.272	0.179	0.472	0.069	0.501	0.562	0.392	0.189	0.077	0.702	0.329	0.356
vertebral	0.123	0.085	0.110	0.124	0.091	0.097	0.095	0.089	0.130	0.101	0.107	0.099
vowels	0.166	0.034	0.083	0.314	0.078	0.162	0.443	0.127	0.326	0.085	0.196	0.069
waveform	0.122	0.057	0.040	0.078	0.048	0.056	0.133	0.040	0.071	0.040	0.052	0.044
wbc	0.691	0.883	0.882	0.037	0.728	0.948	0.743	0.898	0.077	0.839	0.813	0.913
wdbc	0.688	0.760	0.493	0.155	0.761	0.702	0.521	0.527	0.128	0.395	0.539	0.613
wilt	0.040	0.037	0.042	0.081	0.039	0.044	0.049	0.036	0.083	0.153	0.035	0.032
wine	0.170	0.364	0.195	0.061	0.412	0.207	0.081	0.250	0.064	0.737	0.135	0.264
wpbc	0.227	0.234	0.217	0.206	0.241	0.237	0.234	0.227	0.210	0.257	0.222	0.229
yeast	0.314	0.308	0.332	0.326	0.328	0.304	0.294	0.330	0.315	0.298	0.303	0.302
CIFAR10	0.103	0.065	0.067	0.115	0.075	0.089	0.102	0.086	0.115	0.084	0.102	0.101
MNIST-C	0.173	0.050	0.050	0.128	0.126	0.178	0.191	0.101	0.127	0.166	0.179	0.170
MVTec-AD	0.570	0.236	0.236	0.536	0.546	0.570	0.580	0.464	0.532	0.451	0.555	0.540
SVHN	0.079	0.050	0.050	0.084	0.064	0.073	0.079	0.064	0.083	0.068	0.078	0.078
mnist	0.386	0.092	0.092	0.241	0.109	0.290	0.409	0.170	0.233	0.308	0.385	0.381
FashionMNIST	0.329	0.050	0.050	0.194	0.269	0.320	0.346	0.180	0.188	0.245	0.329	0.319
20news	0.067	0.061	0.062	0.087	0.061	0.062	0.069	0.062	0.088	0.072	0.064	0.062
agnews	0.072	0.059	0.058	0.125	0.059	0.064	0.082	0.064	0.125	0.077	0.068	0.061
amazon	0.061	0.060	0.055	0.058	0.059	0.058	0.062	0.054	0.058	0.062	0.059	0.057
imdb	0.047	0.050	0.045	0.049	0.047	0.047	0.047	0.046	0.049	0.049	0.047	0.046
yelp	0.073	0.072	0.065	0.085	0.070	0.070	0.083	0.067	0.085	0.075	0.073	0.069
Average	0.318	0.288	0.296	0.179	0.308	0.336	0.308	0.260	0.181	0.337	0.324	0.328

Table B.3.1: PR AUC for the unsupervised setting on ADBench (1)

	DAGMM	DeepSVDD	DROCC	GOAD	ICL	PlanarFlow	DDPM	DTE-NP	DTE-IG	DTE-C	ODIM	ALTBI-G	ALTBI-I
aloi	0.033	0.034	0.030	0.033	0.046	0.032	0.036	0.056	0.040	0.033	0.040	0.037	0.036
annthyroid	0.109	0.192	0.186	0.131	0.123	0.654	0.297	0.228	0.380	0.670	0.166	0.350	0.156
backdoor	0.250	0.372	0.025	0.347	0.717	0.336	0.520	0.473	0.438	0.481	0.406	0.128	0.150
breastw	0.660	0.482	0.776	0.826	0.635	0.908	0.537	0.921	0.770	0.715	0.982	0.966	0.944
campaign	0.163	0.149	0.113	0.105	0.267	0.191	0.299	0.281	0.237	0.321	0.315	0.333	0.247
cardio	0.193	0.177	0.272	0.540	0.108	0.471	0.278	0.376	0.184	0.268	0.526	0.386	0.521
cardiotocography	0.271	0.252	0.258	0.403	0.188	0.348	0.338	0.312	0.250	0.276	0.422	0.337	0.419
celeba	0.044	0.031	0.047	0.021	0.045	0.066	0.093	0.052	0.058	0.077	0.123	0.061	0.092
census	0.062	0.075	0.058	0.072	0.095	0.074	0.086	0.090	0.083	0.081	0.089	0.098	0.089
cover	0.044	0.048	0.056	0.005	0.022	0.010	0.046	0.048	0.025	0.021	0.068	0.178	0.050
donors	0.086	0.112	0.123	0.040	0.119	0.241	0.143	0.188	0.164	0.140	0.136	0.140	0.072
fault	0.361	0.375	0.496	0.381	0.473	0.329	0.392	0.532	0.417	0.422	0.412	0.512	0.474
fraud	0.084	0.250	0.002	0.257	0.127	0.447	0.146	0.137	0.188	0.648	0.369	0.334	0.346
glass	0.111	0.090	0.159	0.076	0.122	0.113	0.073	0.206	0.135	0.168	0.161	0.111	0.095
hepatitis	0.253	0.170	0.221	0.291	0.231	0.317	0.165	0.238	0.215	0.257	0.319	0.259	0.430
http	0.368	0.093	0.004	0.441	0.091	0.363	0.642	0.024	0.295	0.440	0.259	0.222	0.294
internetads	0.207	0.252	0.197	0.288	0.237	0.262	0.295	0.290	0.275	0.302	0.281	0.500	0.354
ionosphere	0.473	0.392	0.728	0.781	0.472	0.824	0.633	0.920	0.610	0.880	0.700	0.911	0.814
landsat	0.230	0.362	0.272	0.198	0.329	0.187	0.200	0.255	0.203	0.223	0.181	0.241	0.210
letter	0.083	0.099	0.252	0.099	0.208	0.153	0.367	0.255	0.181	0.257	0.088	0.395	0.130
lymphography	0.454	0.254	0.463	0.897	0.264	0.417	0.731	0.805	0.388	0.381	1.000	0.705	0.780
magic.gamma	0.450	0.499	0.627	0.326	0.548	0.692	0.651	0.730	0.657	0.664	0.650	0.767	0.685
mammography	0.111	0.025	0.114	0.046	0.046	0.074	0.099	0.175	0.082	0.170	0.086	0.163	0.068
musk	0.500	0.107	0.196	1.000	0.128	0.391	0.984	0.434	0.137	0.553	1.000	1.000	1.000
optdigits	0.026	0.039	0.032	0.039	0.030	0.027	0.022	0.021	0.028	0.028	0.029	0.029	0.039
pageblocks	0.255	0.288	0.632	0.373	0.285	0.538	0.493	0.530	0.507	0.555	0.503	0.467	0.554
pendigits	0.056	0.022	0.027	0.075	0.045	0.060	0.056	0.089	0.044	0.044	0.236	0.116	0.188
pima	0.372	0.366	0.413	0.476	0.385	0.476	0.400	0.528	0.437	0.447	0.485	0.500	0.500
satellite	0.527	0.406	0.465	0.658	0.451	0.596	0.662	0.563	0.380	0.529	0.668	0.531	0.696
satimage-2	0.289	0.052	0.076	0.949	0.102	0.484	0.783	0.507	0.095	0.138	0.953	0.836	0.931
shuttle	0.438	0.149	0.072	0.136	0.135	0.346	0.779	0.187	0.247	0.626	0.948	0.953	0.958
skin	0.226	0.221	0.285	0.232	0.173	0.335	0.175	0.290	0.316	0.302	0.235	0.371	0.455
smtp	0.179	0.240	0.000	0.358	0.004	0.006	0.502	0.411	0.012	0.422	0.333	0.358	0.057
spambase	0.389	0.456	0.383	0.387	0.370	0.433	0.384	0.407	0.399	0.400	0.408	0.485	0.388
speech	0.022	0.018	0.020	0.019	0.020	0.018	0.020	0.019	0.019	0.020	0.020	0.018	0.017
stamps	0.198	0.099	0.241	0.285	0.117	0.284	0.143	0.273	0.235	0.226	0.355	0.196	0.317
thyroid	0.126	0.024	0.338	0.318	0.066	0.734	0.325	0.360	0.118	0.705	0.244	0.452	0.225
vertebral	0.134	0.107	0.118	0.124	0.115	0.111	0.150	0.098	0.133	0.119	0.153	0.090	0.103
vowels	0.041	0.037	0.178	0.154	0.219	0.295	0.311	0.504	0.166	0.417	0.255	0.521	0.210
waveform	0.032	0.061	0.150	0.042	0.063	0.150	0.050	0.109	0.037	0.043	0.060	0.061	0.101
wbc	0.327	0.069	0.358	0.736	0.211	0.431	0.758	0.722	0.348	0.194	1.000	0.679	0.902
wdbc	0.152	0.063	0.039	0.589	0.065	0.568	0.483	0.465	0.074	0.157	0.393	0.290	0.568
wilt	0.047	0.046	0.041	0.065	0.109	0.115	0.076	0.054	0.211	0.163	0.036	0.071	0.037
wine	0.120	0.116	0.126	0.229	0.087	0.086	0.075	0.074	0.064	0.103	0.335	0.219	0.385
wpbc	0.214	0.240	0.234	0.214	0.234	0.236	0.238	0.227	0.238	0.231	0.255	0.269	0.224
yeast	0.353	0.350	0.284	0.332	0.318	0.309	0.320	0.295	0.306	0.306	0.292	0.305	0.299
CIFAR10	0.062	0.073	0.060	0.102	0.070	0.085	0.102	0.104	0.078	0.092	0.557	0.499	0.531
MNIST-C	0.092	0.097	0.096	0.177	0.098	0.154	0.178	0.192	0.141	0.157	0.281	0.281	0.305
MVTec-AD	0.362	0.387	0.317	0.546	0.404	0.454	0.546	0.578	0.439	0.517	0.738	0.444	0.560
SVHN	0.059	0.063	0.060	0.078	0.068	0.074	0.078	0.080	0.069	0.077	0.060	0.058	0.065
mnist	0.215	0.253	0.237	0.297	0.232	0.259	0.374	0.400	0.276	0.368	0.980	0.238	0.581
FashionMNIST	0.138	0.181	0.106	0.328	0.158	0.297	0.325	0.339	0.213	0.267	0.987	0.321	0.648
20news	0.054	0.058	0.055	0.063	0.063	0.056	0.063	0.072	0.060	0.068	0.110	0.088	0.116
agnews	0.053	0.053	0.051	0.066	0.069	0.050	0.062	0.085	0.063	0.076	0.149	0.106	0.276
amazon	0.049	0.046	0.050	0.058	0.052	0.050	0.057	0.062	0.055	0.057	0.051	0.051	0.054
imdb	0.049	0.053	0.050	0.047	0.054	0.051	0.046	0.047	0.047	0.047	0.053	0.050	0.056
yelp	0.049	0.058	0.051	0.068	0.054	0.056	0.069	0.085	0.054	0.066	0.054	0.054	0.057
Average	0.198	0.170	0.199	0.285	0.185	0.283	0.301	0.295	0.216	0.288	0.368	0.336	0.348

Table B.3.2: PR AUC for the unsupervised setting on ADBench (2)

	CBLOF	COPOD	ECOD	FeatureBagging	HBOS	IForest	kNN	LODA	LOF	MCD	OCSVM	PCA
aloi	0.537	0.495	0.517	0.491	0.522	0.507	0.510	0.492	0.488	0.485	0.543	0.540
annthyroid	0.901	0.768	0.785	0.890	0.660	0.903	0.928	0.774	0.886	0.902	0.885	0.852
backdoor	0.697	0.500	0.500	0.948	0.708	0.749	0.938	0.476	0.953	0.851	0.625	0.646
breastw	0.991	0.995	0.991	0.591	0.992	0.995	0.991	0.981	0.889	0.987	0.994	0.992
campaign	0.771	0.782	0.769	0.691	0.771	0.736	0.785	0.589	0.706	0.785	0.777	0.771
cardio	0.935	0.932	0.950	0.921	0.807	0.933	0.920	0.913	0.922	0.828	0.956	0.965
cardiotocography	0.676	0.664	0.793	0.636	0.612	0.742	0.621	0.728	0.645	0.571	0.752	0.789
celeba	0.793	0.757	0.763	0.469	0.767	0.712	0.731	0.625	0.437	0.844	0.798	0.805
census	0.708	0.500	0.500	0.559	0.625	0.626	0.723	0.511	0.585	0.741	0.700	0.705
cover	0.940	0.882	0.919	0.992	0.711	0.863	0.975	0.949	0.992	0.700	0.962	0.944
donors	0.935	0.815	0.887	0.952	0.812	0.894	0.995	0.635	0.970	0.819	0.921	0.881
fault	0.590	0.491	0.504	0.483	0.531	0.559	0.587	0.503	0.474	0.594	0.572	0.559
fraud	0.949	0.943	0.949	0.948	0.950	0.947	0.954	0.891	0.944	0.911	0.956	0.954
glass	0.894	0.760	0.711	0.885	0.826	0.811	0.920	0.673	0.888	0.797	0.697	0.734
hepatitis	0.863	0.809	0.738	0.678	0.848	0.827	0.965	0.690	0.669	0.806	0.906	0.845
http	0.999	0.992	0.980	0.921	0.986	0.994	1.000	0.477	1.000	1.000	1.000	1.000
internetads	0.652	0.659	0.660	0.714	0.492	0.479	0.681	0.587	0.717	0.477	0.656	0.651
ionosphere	0.968	0.783	0.718	0.945	0.707	0.912	0.974	0.856	0.943	0.954	0.963	0.891
landsat	0.572	0.493	0.420	0.664	0.732	0.588	0.683	0.447	0.666	0.568	0.480	0.439
letter	0.332	0.365	0.454	0.448	0.359	0.320	0.354	0.302	0.448	0.315	0.322	0.303
lymphography	0.998	0.995	0.995	0.966	0.997	0.995	0.999	0.670	0.982	0.989	1.000	0.999
magic.gamma	0.758	0.680	0.636	0.842	0.745	0.771	0.833	0.705	0.834	0.737	0.743	0.706
mammography	0.847	0.906	0.907	0.863	0.850	0.880	0.876	0.896	0.855	0.729	0.886	0.899
musk	1.000	0.997	0.999	1.000	1.000	0.906	1.000	0.997	1.000	0.939	1.000	1.000
optdigits	0.835	0.500	0.500	0.963	0.899	0.811	0.937	0.328	0.967	0.649	0.634	0.582
pageblocks	0.912	0.809	0.880	0.911	0.656	0.826	0.897	0.836	0.913	0.871	0.886	0.861
pendigits	0.967	0.907	0.930	0.995	0.936	0.972	0.999	0.921	0.991	0.837	0.964	0.944
pima	0.729	0.666	0.606	0.719	0.748	0.743	0.769	0.627	0.705	0.736	0.715	0.723
satellite	0.732	0.683	0.622	0.801	0.855	0.775	0.822	0.697	0.803	0.728	0.739	0.666
satimage-2	0.994	0.979	0.971	0.995	0.980	0.991	0.997	0.987	0.994	0.999	0.996	0.982
shuttle	0.997	0.995	0.993	0.869	0.986	0.997	0.999	0.717	1.000	0.990	0.996	0.994
skin	0.918	0.472	0.491	0.784	0.769	0.894	0.995	0.755	0.863	0.884	0.903	0.597
smtp	0.873	0.912	0.883	0.848	0.828	0.904	0.924	0.730	0.934	0.949	0.847	0.818
spambase	0.815	0.721	0.688	0.696	0.779	0.852	0.834	0.724	0.732	0.807	0.817	0.814
speech	0.359	0.370	0.360	0.375	0.367	0.377	0.364	0.380	0.375	0.388	0.366	0.364
stamps	0.934	0.931	0.876	0.942	0.918	0.935	0.959	0.919	0.937	0.849	0.937	0.927
thyroid	0.985	0.938	0.976	0.932	0.987	0.990	0.987	0.961	0.927	0.985	0.986	0.986
vertebral	0.544	0.263	0.420	0.641	0.401	0.456	0.577	0.317	0.643	0.471	0.505	0.421
vowels	0.787	0.528	0.615	0.853	0.533	0.618	0.822	0.555	0.863	0.277	0.759	0.523
waveform	0.729	0.724	0.594	0.770	0.693	0.723	0.752	0.610	0.760	0.584	0.704	0.647
wbc	0.983	0.994	0.994	0.581	0.990	0.994	0.991	0.979	0.805	0.989	0.996	0.994
wdbc	0.987	0.992	0.967	0.996	0.986	0.987	0.991	0.970	0.996	0.970	0.993	0.991
wilt	0.429	0.321	0.375	0.734	0.391	0.480	0.637	0.411	0.688	0.817	0.348	0.261
wine	0.978	0.864	0.739	0.979	0.956	0.939	0.992	0.909	0.984	0.973	0.978	0.938
wpbc	0.596	0.523	0.495	0.568	0.609	0.563	0.637	0.513	0.574	0.634	0.534	0.525
yeast	0.504	0.389	0.446	0.464	0.429	0.418	0.447	0.465	0.458	0.431	0.448	0.432
CIFAR10	0.679	0.550	0.569	0.703	0.579	0.640	0.675	0.616	0.703	0.652	0.678	0.674
MNIST-C	0.811	0.500	0.500	0.873	0.704	0.768	0.841	0.694	0.872	0.753	0.796	0.784
MVTec-AD	0.800	0.500	0.500	0.805	0.760	0.774	0.815	0.723	0.804	0.868	0.774	0.764
SVHN	0.610	0.500	0.500	0.639	0.547	0.590	0.617	0.545	0.638	0.589	0.613	0.608
mnist	0.911	0.500	0.500	0.926	0.623	0.866	0.939	0.647	0.929	0.883	0.906	0.902
FashionMNIST	0.891	0.500	0.500	0.917	0.754	0.842	0.899	0.793	0.916	0.844	0.882	0.876
20news	0.571	0.529	0.541	0.603	0.536	0.549	0.574	0.535	0.602	0.629	0.563	0.544
agnews	0.628	0.551	0.551	0.746	0.557	0.584	0.671	0.570	0.746	0.680	0.606	0.569
amazon	0.582	0.568	0.538	0.579	0.563	0.564	0.606	0.522	0.579	0.604	0.565	0.549
imdb	0.499	0.511	0.469	0.495	0.499	0.495	0.501	0.472	0.496	0.512	0.487	0.480
yelp	0.638	0.602	0.574	0.671	0.600	0.611	0.681	0.563	0.672	0.662	0.621	0.592
Average	0.781	0.689	0.688	0.770	0.727	0.757	0.809	0.673	0.785	0.751	0.765	0.740

Table B.4.1: ROC AUC for the semi-supervised setting on ADBench (1)

	DAGMM	DeepSVDD	DROCC	GOAD	ICL	PlanarFlow	DDPM	DTE-NP	DTE-IG	DTE-C	ALTBI
aloi	0.508	0.509	0.500	0.480	0.475	0.485	0.499	0.512	0.509	0.504	0.539
annthyroid	0.722	0.550	0.889	0.810	0.811	0.932	0.888	0.929	0.876	0.975	0.655
backdoor	0.544	0.911	0.943	0.529	0.936	0.760	0.809	0.933	0.940	0.917	0.927
breastw	0.895	0.970	0.473	0.989	0.983	0.979	0.987	0.993	0.787	0.928	0.982
campaign	0.615	0.622	0.500	0.479	0.809	0.698	0.745	0.788	0.748	0.780	0.749
cardio	0.779	0.654	0.621	0.960	0.800	0.889	0.869	0.918	0.738	0.873	0.869
cardiotocography	0.671	0.478	0.460	0.761	0.542	0.699	0.545	0.638	0.524	0.601	0.686
celeba	0.638	0.562	0.689	0.438	0.722	0.716	0.786	0.704	0.745	0.822	0.701
census	0.522	0.542	0.554	0.352	0.706	0.593	0.702	0.721	0.618	0.696	0.704
cover	0.759	0.491	0.958	0.138	0.893	0.475	0.984	0.977	0.958	0.978	0.947
donors	0.622	0.730	0.742	0.336	0.999	0.916	0.825	0.993	0.993	0.982	0.960
fault	0.529	0.543	0.557	0.589	0.606	0.575	0.611	0.586	0.594	0.595	0.734
fraud	0.853	0.831	0.500	0.698	0.928	0.907	0.937	0.956	0.908	0.935	0.954
glass	0.653	0.837	0.649	0.590	0.994	0.853	0.667	0.896	0.985	0.924	0.741
hepatitis	0.702	0.996	0.518	0.845	0.999	0.958	0.977	0.932	0.999	0.988	0.593
http	0.918	0.613	0.500	0.997	0.982	0.994	1.000	1.000	0.807	0.995	0.994
internetads	0.495	0.730	0.534	0.657	0.722	0.709	0.658	0.700	0.715	0.776	0.909
ionosphere	0.740	0.972	0.611	0.915	0.990	0.969	0.946	0.978	0.952	0.954	0.934
landsat	0.563	0.594	0.539	0.405	0.651	0.509	0.514	0.682	0.447	0.528	0.634
letter	0.390	0.364	0.553	0.311	0.427	0.387	0.381	0.344	0.399	0.367	0.829
lymphography	0.949	0.997	0.324	0.999	1.000	0.996	0.999	0.999	1.000	0.990	0.994
magic.gamma	0.592	0.630	0.788	0.695	0.756	0.741	0.860	0.836	0.865	0.875	0.836
mammography	0.760	0.715	0.818	0.699	0.719	0.789	0.810	0.876	0.846	0.864	0.857
musk	0.950	1.000	0.330	1.000	0.994	0.767	1.000	1.000	0.942	1.000	1.000
optdigits	0.400	0.395	0.853	0.675	0.972	0.341	0.908	0.943	0.798	0.824	0.871
pageblocks	0.828	0.784	0.923	0.881	0.884	0.849	0.869	0.893	0.857	0.899	0.926
pendigits	0.565	0.463	0.759	0.900	0.967	0.835	0.981	0.996	0.970	0.978	0.961
pima	0.545	0.580	0.475	0.623	0.797	0.722	0.703	0.815	0.686	0.699	0.727
satellite	0.728	0.762	0.734	0.688	0.852	0.723	0.777	0.821	0.765	0.786	0.797
satimage-2	0.918	0.929	0.992	0.990	0.995	0.967	0.996	0.997	0.953	0.993	0.998
shuttle	0.846	0.998	0.500	0.704	0.999	0.865	0.999	0.999	0.999	0.998	0.886
skin	0.679	0.600	0.895	0.650	0.066	0.913	0.887	0.989	0.987	0.918	0.925
smtp	0.871	0.852	0.571	0.788	0.744	0.842	0.954	0.930	0.816	0.953	0.916
spambase	0.694	0.702	0.754	0.818	0.835	0.823	0.645	0.837	0.775	0.830	0.663
speech	0.507	0.489	0.490	0.366	0.489	0.486	0.370	0.414	0.396	0.382	0.421
stamps	0.801	0.711	0.502	0.815	0.967	0.873	0.918	0.979	0.934	0.916	0.955
thyroid	0.911	0.888	0.950	0.952	0.954	0.984	0.980	0.986	0.894	0.987	0.947
vertebral	0.506	0.448	0.438	0.467	0.792	0.498	0.707	0.543	0.746	0.664	0.285
vowels	0.426	0.557	0.547	0.685	0.851	0.546	0.864	0.814	0.857	0.869	0.963
waveform	0.519	0.599	0.677	0.650	0.687	0.648	0.622	0.745	0.737	0.652	0.667
wbc	0.868	0.914	0.442	0.991	0.997	0.960	0.992	0.995	0.910	0.805	0.980
wdbc	0.738	0.993	0.401	0.990	0.998	0.989	0.993	0.995	0.996	0.985	0.995
wilt	0.418	0.344	0.495	0.514	0.764	0.746	0.717	0.629	0.938	0.851	0.378
wine	0.662	0.922	0.438	0.941	0.999	0.954	0.996	0.994	1.000	1.000	0.724
wpbc	0.470	0.827	0.438	0.514	0.966	0.575	0.665	0.832	0.707	0.689	0.491
yeast	0.510	0.476	0.484	0.525	0.490	0.451	0.491	0.446	0.486	0.471	0.468
CIFAR10	0.540	0.561	0.496	0.675	0.636	0.628	0.679	0.678	0.624	0.685	0.948
MNIST-C	0.637	0.647	0.572	0.793	0.856	0.712	0.801	0.847	0.799	0.861	0.858
MVTec-AD	0.647	0.896	0.605	0.771	0.948	0.728	0.780	0.897	0.859	0.894	0.929
SVHN	0.534	0.539	0.500	0.608	0.617	0.589	0.614	0.621	0.592	0.629	0.604
mnist	0.722	0.664	0.831	0.901	0.901	0.819	0.873	0.940	0.808	0.874	0.927
FashionMNIST	0.708	0.755	0.516	0.880	0.906	0.822	0.885	0.901	0.843	0.902	0.916
20news	0.515	0.556	0.528	0.549	0.613	0.516	0.549	0.600	0.583	0.643	0.807
agnews	0.510	0.498	0.497	0.599	0.626	0.502	0.578	0.680	0.566	0.682	0.889
amazon	0.505	0.512	0.500	0.561	0.542	0.499	0.551	0.608	0.519	0.567	0.585
imdb	0.486	0.500	0.514	0.485	0.523	0.492	0.479	0.504	0.510	0.481	0.575
yelp	0.499	0.499	0.507	0.611	0.558	0.536	0.593	0.687	0.573	0.599	0.564
Average	0.651	0.679	0.603	0.688	0.794	0.732	0.779	0.815	0.779	0.804	0.794

Table B.4.2: ROC AUC for the semi-supervised setting on ADBench (2)

	CBLOF	COPOD	ECOD	FeatureBagging	HBOS	IForest	kNN	LODA	LOF	MCD	OCSVM	PCA
aloi	0.064	0.057	0.061	0.068	0.064	0.058	0.060	0.059	0.065	0.056	0.065	0.065
annthyroid	0.636	0.296	0.400	0.485	0.390	0.590	0.681	0.490	0.535	0.597	0.601	0.566
backdoor	0.091	0.048	0.048	0.495	0.086	0.094	0.465	0.060	0.535	0.222	0.077	0.079
breastw	0.991	0.994	0.992	0.524	0.991	0.995	0.989	0.968	0.800	0.983	0.994	0.992
campaign	0.486	0.511	0.495	0.333	0.497	0.457	0.490	0.298	0.402	0.479	0.494	0.488
cardio	0.809	0.749	0.786	0.716	0.589	0.786	0.772	0.725	0.702	0.671	0.836	0.862
cardiotocography	0.617	0.561	0.690	0.570	0.507	0.629	0.574	0.606	0.573	0.528	0.662	0.697
celeba	0.185	0.165	0.169	0.039	0.168	0.117	0.119	0.095	0.036	0.190	0.203	0.210
census	0.203	0.117	0.117	0.120	0.140	0.142	0.217	0.134	0.137	0.290	0.203	0.200
cover	0.160	0.123	0.192	0.781	0.054	0.087	0.558	0.226	0.829	0.031	0.223	0.162
donors	0.465	0.335	0.413	0.653	0.363	0.405	0.891	0.254	0.634	0.312	0.427	0.352
fault	0.613	0.532	0.517	0.508	0.539	0.592	0.620	0.545	0.504	0.634	0.611	0.604
fraud	0.278	0.384	0.332	0.631	0.323	0.182	0.387	0.366	0.551	0.601	0.296	0.269
glass	0.317	0.201	0.250	0.361	0.276	0.214	0.423	0.156	0.381	0.203	0.268	0.210
hepatitis	0.634	0.561	0.458	0.446	0.635	0.554	0.903	0.502	0.437	0.568	0.776	0.649
http	0.903	0.463	0.252	0.082	0.390	0.534	1.000	0.075	0.971	0.922	0.999	0.917
internetads	0.470	0.617	0.619	0.493	0.308	0.292	0.492	0.393	0.504	0.344	0.482	0.470
ionosphere	0.973	0.785	0.756	0.949	0.646	0.917	0.980	0.852	0.946	0.967	0.975	0.909
landsat	0.369	0.338	0.311	0.615	0.601	0.473	0.549	0.357	0.614	0.397	0.370	0.327
letter	0.083	0.089	0.107	0.117	0.087	0.082	0.087	0.080	0.113	0.081	0.083	0.080
lymphography	0.983	0.939	0.944	0.727	0.966	0.944	0.992	0.241	0.842	0.868	1.000	0.985
magic.gamma	0.802	0.722	0.679	0.869	0.772	0.803	0.859	0.758	0.864	0.772	0.792	0.752
mammography	0.411	0.546	0.552	0.293	0.213	0.379	0.413	0.432	0.341	0.080	0.405	0.417
musk	1.000	0.961	0.982	1.000	1.000	0.404	1.000	0.908	1.000	0.663	1.000	1.000
optdigits	0.140	0.056	0.056	0.412	0.424	0.154	0.291	0.039	0.436	0.071	0.069	0.060
pageblocks	0.706	0.415	0.585	0.702	0.225	0.434	0.676	0.486	0.711	0.632	0.643	0.594
pendigits	0.512	0.309	0.415	0.857	0.423	0.588	0.970	0.372	0.786	0.132	0.518	0.386
pima	0.721	0.691	0.648	0.695	0.759	0.737	0.754	0.594	0.684	0.686	0.720	0.712
satellite	0.773	0.733	0.696	0.858	0.865	0.824	0.860	0.798	0.859	0.799	0.809	0.778
satimage-2	0.968	0.853	0.797	0.907	0.877	0.945	0.967	0.937	0.885	0.983	0.969	0.919
shuttle	0.968	0.981	0.952	0.464	0.975	0.986	0.979	0.557	0.998	0.909	0.977	0.963
skin	0.695	0.297	0.305	0.492	0.534	0.646	0.982	0.530	0.617	0.624	0.663	0.364
smtp	0.497	0.010	0.680	0.004	0.012	0.011	0.505	0.082	0.481	0.012	0.645	0.495
spambase	0.820	0.736	0.713	0.684	0.784	0.883	0.833	0.802	0.727	0.818	0.822	0.818
speech	0.027	0.028	0.029	0.030	0.032	0.033	0.028	0.030	0.032	0.028	0.028	0.028
stamps	0.622	0.564	0.490	0.656	0.523	0.588	0.717	0.572	0.648	0.417	0.649	0.588
thyroid	0.815	0.302	0.640	0.365	0.770	0.797	0.809	0.643	0.606	0.801	0.789	0.813
vertebral	0.252	0.155	0.199	0.329	0.189	0.208	0.261	0.167	0.339	0.210	0.222	0.193
vowels	0.239	0.071	0.177	0.327	0.079	0.120	0.302	0.104	0.331	0.044	0.274	0.105
waveform	0.225	0.099	0.074	0.287	0.090	0.105	0.270	0.078	0.307	0.078	0.109	0.084
wbc	0.868	0.932	0.931	0.127	0.877	0.942	0.920	0.757	0.249	0.902	0.972	0.943
wdbc	0.757	0.838	0.610	0.937	0.778	0.720	0.820	0.548	0.936	0.553	0.874	0.821
wilt	0.081	0.069	0.077	0.192	0.079	0.088	0.123	0.080	0.157	0.215	0.071	0.064
wine	0.868	0.523	0.326	0.887	0.777	0.671	0.951	0.579	0.900	0.831	0.887	0.692
wpbc	0.448	0.382	0.358	0.410	0.426	0.407	0.461	0.383	0.412	0.452	0.409	0.400
yeast	0.507	0.468	0.494	0.499	0.498	0.468	0.483	0.490	0.489	0.457	0.480	0.468
CIFAR10	0.197	0.121	0.126	0.222	0.140	0.165	0.196	0.169	0.222	0.159	0.194	0.192
MNIST-C	0.425	0.095	0.095	0.522	0.216	0.328	0.462	0.326	0.519	0.258	0.416	0.403
MVTec-AD	0.749	0.378	0.378	0.758	0.676	0.700	0.758	0.657	0.758	0.805	0.730	0.721
SVHN	0.151	0.095	0.095	0.161	0.120	0.139	0.153	0.127	0.160	0.128	0.150	0.149
mnist	0.665	0.169	0.169	0.693	0.222	0.542	0.727	0.341	0.710	0.558	0.662	0.650
FashionMNIST	0.578	0.095	0.095	0.639	0.349	0.447	0.592	0.469	0.636	0.374	0.565	0.562
20news	0.126	0.111	0.113	0.150	0.111	0.116	0.135	0.112	0.150	0.155	0.118	0.113
agnews	0.138	0.111	0.109	0.259	0.112	0.119	0.167	0.121	0.259	0.146	0.128	0.116
amazon	0.115	0.112	0.104	0.111	0.111	0.111	0.117	0.102	0.110	0.117	0.111	0.107
imdb	0.090	0.093	0.085	0.090	0.090	0.090	0.089	0.087	0.090	0.095	0.089	0.087
yelp	0.137	0.133	0.119	0.161	0.130	0.132	0.160	0.118	0.161	0.138	0.134	0.128
Average	0.499	0.388	0.401	0.469	0.419	0.438	0.562	0.383	0.521	0.439	0.504	0.470

Table B.5.1: PR AUC for the semi-supervised setting on ADBench (1)

	DAGMM	DeepSVDD	DROCC	GOAD	ICL	PlanarFlow	DDPM	DTE-NP	DTE-IG	DTE-C	ALTBI
aloi	0.061	0.062	0.059	0.057	0.055	0.055	0.060	0.061	0.060	0.058	0.040
annthyroid	0.480	0.278	0.637	0.587	0.458	0.652	0.629	0.682	0.499	0.829	0.207
backdoor	0.075	0.848	0.846	0.063	0.892	0.322	0.142	0.457	0.820	0.624	0.815
breastw	0.910	0.960	0.632	0.988	0.968	0.975	0.986	0.992	0.814	0.883	0.960
campaign	0.324	0.370	0.203	0.231	0.489	0.428	0.489	0.500	0.462	0.469	0.340
cardio	0.559	0.389	0.512	0.848	0.479	0.689	0.693	0.774	0.411	0.693	0.530
cardiotocography	0.597	0.458	0.439	0.675	0.487	0.593	0.513	0.587	0.396	0.533	0.490
celeba	0.090	0.071	0.077	0.040	0.097	0.129	0.180	0.107	0.134	0.142	0.042
census	0.132	0.154	0.143	0.087	0.212	0.147	0.197	0.211	0.163	0.179	0.106
cover	0.098	0.027	0.313	0.011	0.345	0.020	0.733	0.600	0.804	0.637	0.227
donors	0.195	0.428	0.302	0.090	0.984	0.493	0.267	0.856	0.958	0.713	0.433
fault	0.568	0.555	0.578	0.621	0.632	0.604	0.648	0.622	0.638	0.639	0.560
fraud	0.156	0.483	0.003	0.294	0.539	0.628	0.692	0.421	0.511	0.621	0.354
glass	0.186	0.524	0.231	0.183	0.924	0.309	0.312	0.374	0.806	0.415	0.362
hepatitis	0.544	0.987	0.349	0.658	0.998	0.896	0.951	0.823	0.998	0.958	0.239
http	0.575	0.361	0.007	0.684	0.708	0.522	1.000	0.971	0.788	0.555	0.233
internetads	0.318	0.516	0.431	0.474	0.600	0.476	0.477	0.513	0.587	0.552	0.820
ionosphere	0.775	0.981	0.717	0.932	0.991	0.976	0.964	0.982	0.969	0.968	0.910
landsat	0.403	0.494	0.376	0.312	0.531	0.342	0.348	0.545	0.327	0.368	0.276
letter	0.104	0.089	0.157	0.081	0.128	0.092	0.095	0.086	0.102	0.090	0.200
lymphography	0.735	0.968	0.309	0.988	1.000	0.962	0.993	0.993	0.998	0.868	0.917
magic.gamma	0.645	0.695	0.832	0.761	0.813	0.785	0.880	0.862	0.887	0.897	0.771
mammography	0.220	0.275	0.272	0.278	0.171	0.185	0.199	0.421	0.334	0.398	0.242
musk	0.706	0.999	0.157	1.000	0.922	0.327	1.000	1.000	0.889	1.000	1.000
optdigits	0.050	0.045	0.192	0.078	0.509	0.039	0.256	0.318	0.221	0.153	0.119
pageblocks	0.603	0.521	0.735	0.635	0.681	0.583	0.621	0.675	0.575	0.664	0.671
pendigits	0.117	0.093	0.146	0.334	0.664	0.145	0.611	0.919	0.592	0.484	0.283
pima	0.565	0.598	0.534	0.652	0.786	0.712	0.712	0.797	0.696	0.680	0.563
satellite	0.760	0.811	0.775	0.790	0.876	0.779	0.851	0.858	0.817	0.848	0.755
satimage-2	0.475	0.763	0.793	0.959	0.947	0.625	0.881	0.962	0.833	0.682	0.969
shuttle	0.660	0.980	0.134	0.602	0.997	0.517	0.979	0.981	0.994	0.940	0.244
skin	0.504	0.430	0.656	0.422	0.325	0.747	0.764	0.948	0.969	0.691	0.571
smtp	0.209	0.307	0.087	0.324	0.038	0.008	0.408	0.502	0.336	0.504	0.353
spambase	0.742	0.753	0.791	0.821	0.868	0.854	0.729	0.837	0.810	0.838	0.543
speech	0.040	0.034	0.036	0.028	0.034	0.033	0.030	0.032	0.029	0.029	0.017
stamps	0.465	0.426	0.285	0.496	0.795	0.524	0.647	0.825	0.728	0.577	0.613
thyroid	0.631	0.691	0.744	0.801	0.515	0.758	0.822	0.810	0.457	0.817	0.398
vertebral	0.251	0.234	0.234	0.214	0.588	0.230	0.358	0.252	0.515	0.351	0.090
vowels	0.073	0.169	0.132	0.209	0.274	0.097	0.427	0.316	0.336	0.381	0.499
waveform	0.061	0.115	0.201	0.089	0.186	0.251	0.093	0.279	0.196	0.100	0.059
wbc	0.568	0.565	0.240	0.920	0.951	0.710	0.938	0.961	0.724	0.300	0.841
wdbc	0.309	0.843	0.122	0.788	0.956	0.775	0.843	0.905	0.921	0.688	0.804
wilt	0.084	0.071	0.096	0.109	0.289	0.171	0.172	0.122	0.521	0.254	0.039
wine	0.509	0.786	0.185	0.701	0.983	0.789	0.977	0.968	0.997	0.999	0.153
wpbc	0.372	0.749	0.360	0.389	0.893	0.455	0.546	0.690	0.658	0.604	0.244
yeast	0.518	0.492	0.498	0.508	0.496	0.470	0.511	0.481	0.511	0.497	0.323
CIFAR10	0.120	0.140	0.124	0.194	0.174	0.159	0.196	0.199	0.167	0.197	0.693
MNIST-C	0.234	0.314	0.269	0.412	0.515	0.341	0.418	0.474	0.441	0.472	0.506
MVTec-AD	0.581	0.838	0.593	0.726	0.895	0.679	0.737	0.829	0.829	0.851	0.806
SVHN	0.114	0.124	0.112	0.149	0.156	0.142	0.151	0.155	0.142	0.155	0.077
mnist	0.461	0.460	0.597	0.651	0.685	0.552	0.624	0.737	0.561	0.563	0.991
FashionMNIST	0.297	0.451	0.296	0.566	0.631	0.468	0.571	0.598	0.537	0.550	0.989
20news	0.102	0.129	0.120	0.115	0.146	0.106	0.115	0.156	0.141	0.173	0.241
agnews	0.102	0.102	0.097	0.124	0.154	0.097	0.119	0.174	0.128	0.192	0.367
amazon	0.095	0.102	0.095	0.109	0.102	0.096	0.108	0.117	0.102	0.112	0.061
imdb	0.092	0.097	0.099	0.088	0.102	0.095	0.087	0.090	0.101	0.089	0.060
yelp	0.093	0.100	0.101	0.131	0.104	0.107	0.128	0.164	0.123	0.130	0.057
Average	0.356	0.444	0.334	0.440	0.557	0.434	0.524	0.571	0.545	0.520	0.440

Table B.5.2: PR AUC for the semi-supervised setting on ADBench (2)

B.3 Ablation studies

The optimal loss usage percentage We examine the optimal amount of loss truncation required when implementing ALTBI. The table shows the averaged AUC on `ADBench` datasets for various loss usage percentages. We can identify that using 92 percent of loss values performs best in our method.

ρ	**0.90**	**0.92**	**0.94**	**0.96**	**0.98**	**1.00**
AUC	0.759	**0.762**	0.759	0.758	0.756	0.755

Table B.6: Averaged results of training AUC scores with various values of ρ.

The increase in mini-batch sizes We investigate how much to increase the mini-batch size at each update for optimal performance. There was no significant difference in scores with varying γ values, but the highest score was achieved with a γ of 1.03. Therefore, we chose this value for our experiments.

γ	**1.00**	**1.01**	**1.02**	**1.03**	**1.04**	**1.05**	**1.07**	**1.1**
AUC	0.761	0.760	0.761	**0.762**	0.761	0.762	0.761	0.761

Table B.7: Averaged results of training AUC scores with various values of γ

Number of samples used in the IWAE Table B.8 shows the averaged AUC values on 57 `ADBench` datasets with various numbers of samples in the IWAE ranging from 1 to 100. It should be noted that K=1 is equivalent to the original VAE. We observe that as the value of K increases, the scores decline and then stabilize. Therefore, we choose K=2 in our experiments.

Learning Schedule We examine the performance of ALTBI with respect to the learning rate. We use the Adam optimizer with various learning rates ranging from 1e-4 to 1e-1, and

K	1	2	5	10	20	50	70	100
AUC	0.760	**0.762**	0.755	0.760	0.759	0.758	0.753	0.753

Table B.8: Averaged results of training AUC scores with various values of K

the table B.9 compares the averaged AUC on `ADBench` datasets. We observe that when the learning rate is larger than 1e-03, the performance of the model deteriorates and stabilizes. For this reason, we set the learning rate to 1e-03 in our experiments.

Learning rate	1e-04	2.5e-04	5e-04	1e-03	2.5e-03	5e-03	1e-02	2.5e-02	5e-02	1e-01
AUC	0.712	0.743	0.756	**0.762**	0.744	0.738	0.735	0.737	0.739	0.738

Table B.9: Averaged results of training AUC scores with various values of learning rate

The initial mini-batch size We examine the performance of ALTBI with respect to the initial mini-batch size. As shown in B.10, increasing n_0 generally improves performance, peaking at 128, and gradually declines and saturates when n_0 exceeds 192. We observe that our method is not significantly sensitive to the choice of n_0.

n_0	32	64	96	128	160	192	224	256
AUC	0.753	0.758	0.759	**0.762**	0.756	0.759	0.757	0.756

Table B.10: Averaged results of training AUC scores with various values of n_0

Iteration for ensembling within a single model We evaluate the best T_1 and T_2 for ensembling within a single model. We compare three values of T_1 and four values of $T_2 - T_1$, and the table B.10 shows the averaged results on `ADBench` datasets. We can see that low values of T_1 and T_2 result in lower scores, so we set T_1 to 70 and T_2 to 90.

ALTBI for addressing SSOD We compare ALTBI with other methods in SSOD tasks. First, we split inliers into two sets with ratios of $7:3$ and use the first partition as a training dataset, and regard the rest inliers and outliers as the test dataset.

	T_1		
$T_2 - T_2$	**50**	**60**	**70**
5	0.757	0.760	0.760
10	0.758	0.761	0.760
20	0.759	0.760	**0.762**
30	0.760	0.762	0.760

Table B.10: Averaged results of training AUC scores with various values of iteration for ensembling

Table B.4.1 - B.5.2, and Figure B.1 show the AUC and PRAUC results of the test datasets. We can observe that ALTBI is still competitive in addressing SSOD tasks.

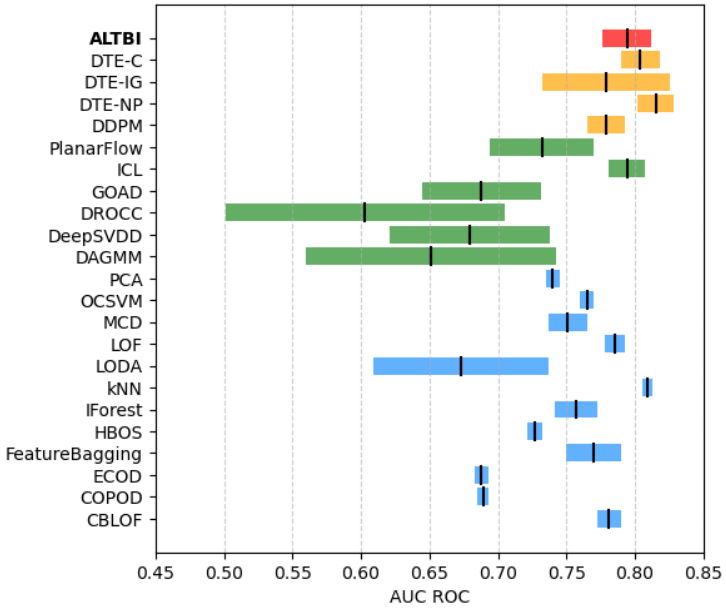

Figure B.1: Test AUC ROC means and standard deviation on the 57 datasets from ADBench over five different seeds in semi-supervised setting. Color scheme: red (IM-based), orange (diffusion-based), green (deep-learning-based), blue (machine-learning-based).

B.4 Further discussions: Robustness of ALTBI in DP

DP-SGD (Abadi et al., 2016) is a variant of SGD used for updating parameters while ensuring differential privacy (DP, Dwork (2006)) for the model. For each sample $\boldsymbol{x}$, with its corresponding per-sample loss $\tilde{l}(\theta; \boldsymbol{x})$, we compute the gradient vector $\nabla_\theta l(\theta; \boldsymbol{x})$. This gradient is then clipped using a specified positive constant $C > 0$ and combined with Gaussian noise drawn from $\mathcal{N}(0, \sigma^2 C^2 I)$ to produce a modified gradient:

$$\nabla_\theta^{\mathrm{DP}} \tilde{l}(\theta; \boldsymbol{x}) = \frac{\nabla_\theta \tilde{l}(\theta; \boldsymbol{x})}{\max\left(1, \frac{\|\nabla_\theta \tilde{l}(\theta; \boldsymbol{x})\|_2}{C}\right)} + \mathcal{N}(0, \sigma^2 C^2 I),$$

where $\sigma > 0$ is another pre-specified constant. In practice, we set $(C, \sigma) = (10, 0.7)$ to carry out DP experiments. The parameters θ is then updated using this modified gradient $\tilde{l}$ with the conventional SGD method or its variants such as Adam (Kingma and Ba, 2014).

For a given loss function to be compatible with DP-SGD, the loss function must be separable across mini-batch samples. However, ALTBI uses a threshold that is calculated based on the current per-sample loss values. Due to this, DP-SGD cannot be directly applied to ALTBI, as the truncated loss function is not separable. This issue can be easily addressed by calculating the threshold using the per-sample loss values from the previous update, i.e., using τ_{t-1} instead of τ_t.

We consider two versions of ALTBI: one that applies mini-batch increment and truncated loss, i.e., $(\gamma, \rho) = (1.03, 0.92)$, and one that does not,i.e., $(\gamma, \rho) = (1.0, 1.0)$. As a measure of DP, we adopt (ϵ, δ)-DP, which is the standard measure to assess differential privacy. With a fixed $\delta = 1 \times 10^{-5}$, we train the models using a DP-SGD algorithm until the privacy budget ϵ does not exceed 10. To implement DP-SGD, we use the `Opacus` library (Yousefpour et al., 2021) in Python.

Similar to the SSOD analysis, we split the entire dataset into two parts with a 7:3 ratio, treating the larger portion as training data and the remaining portion as test data. The test

AUC results for 20 tabular datasets are presented in Table B.11. We observe that using mini-batch increment and the truncated loss function results in less degradation in averaged AUC performance (from 0.759 to 0.750) compared to not using these techniques (from 0.729 to 0.651). This indicates that our method provides robustness in performance when applying DP.

	(γ, ρ) = (1.03,0.92)		**(γ, ρ) = (1.0,1.0)**	
Data	**w/o DP**	**w/ DP**	**w/o DP**	**w/ DP**
breastw	0.725	0.981	0.981	0.969
cardio	0.542	0.927	0.802	0.820
cardiotocography	0.803	0.669	0.566	0.623
celeba	0.662	0.809	0.739	0.711
census	0.901	0.569	0.665	0.464
fault	0.923	0.588	0.684	0.557
fraud	0.771	0.928	0.943	0.658
landsat	0.744	0.368	0.519	0.296
magic.gamma	0.806	0.780	0.794	0.722
musk	0.684	0.439	1.000	0.393
optdigits	0.868	0.552	0.629	0.520
pageblocks	0.954	0.790	0.882	0.776
pima	0.760	0.716	0.715	0.683
satimage-2	0.982	0.952	0.998	0.865
skin	0.846	0.889	0.804	0.859
spambase	0.477	0.722	0.545	0.437
speech	0.883	0.581	0.474	0.487
stamps	0.947	0.849	0.876	0.805
wine	0.495	0.914	0.886	0.907
wpbc	0.411	0.559	0.503	0.478
Average	0.759	0.729	0.750	0.651

Table B.11: The average test AUC results for 20 datasets with and without DP applied under the conditions of $\gamma = 1.03, \rho = 0.92$ and $\gamma = 1.0, \rho = 1.0$

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, may 2000a. ISSN 0163-5808. doi: 10.1145/335191.335388.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000b.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics*, 3(1):79–127, 2006.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-35908-1.

Cecile Fauconnier and Gentiane Haesbroeck. Outliers detection with the minimum covariance determinant estimator in practice. *Statistical Methodology*, 6(4):363–379, 2009.

Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1):267–305, 2016.

Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. *arXiv preprint arXiv:1805.10917*, 2018.

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

Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. DROCC: deep robust one-class classification. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3711–3721. PMLR, 2020.

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

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Dongha Kim, Jaesung Hwang, and Yongdai Kim. On casting importance weighted autoencoder to an em algorithm to learn deep generative models. In *International Conference on Artificial Intelligence and Statistics*, pages 2153–2163. PMLR, 2020.

Dongha Kim, Jaesung Hwang, Jongjin Lee, Kunwoong Kim, and Yongdai Kim. ODIM: an efficient method to detect outliers via inlier-memorization effect of deep generative models. *CoRR*, abs/2301.04257, 2023. doi: 10.48550/ARXIV.2301.04257. URL `https://doi.org/10.48550/arXiv.2301.04257`.

Dongha Kim, Jaesung Hwang, Jongjin Lee, Kunwoong Kim, and Yongdai Kim. Odim: Outlier detection via likelihood of under-fitted generative models, 2024. URL `https://arxiv.org/abs/2301.04257`.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10236–10245, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Charline Le Lan and Laurent Dinh. Perfect density models cannot guarantee anomaly detection. *Entropy*, 23(12):1690, 2021. doi: 10.3390/E23121690.

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

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

Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Victor Livernoche, Vineet Jain, Yashar Hezaveh, and Siamak Ravanbakhsh. On diffusion modeling for anomaly detection. *CoRR*, abs/2305.18593, 2023. doi: 10.48550/ARXIV.2305.18593.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality, 2019a.

Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b.

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

Boris T Polyak. Gradient methods for solving equations and inequalities. *USSR Computational Mathematics and Mathematical Physics*, 4(6):17–32, 1964.

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

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018.

Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.

Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander Smola, and Robert Williamson. Estimating support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 07 2001. doi: 10.1162/089976601750264965.

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

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE foundations and new directions of data mining workshop*, pages 172–179. IEEE Press, 2003.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differen-

tial equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11839–11852. Curran Associates, Inc., 2020.

David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54:45–66, 2004.

Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yufeng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11525–11536. PMLR, 2021. URL http://proceedings.mlr.press/v139/xu21e.html.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

Zhuoning Yuan, Yan Yan, Rong Jin, and Tianbao Yang. Stagewise training accelerates convergence of testing error over SGD. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019,*

NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 2604–2614, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fcdf25d6e191893e705819b177cddea0-Abstract.html.

Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International conference on machine learning*, pages 1100–1109. PMLR, 2016.

Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 665–674. ACM, 2017. doi: 10.1145/3097983.3098052. URL https://doi.org/10.1145/3097983.3098052.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

국문 초록

이상치 탐지를 위한 맞춤형 임계 기법을 이용한 심층 생성 방법론 개선에 대한 연구

성신여자대학교
대학원
통계학과
조서영

이상치 탐지는 주어진 데이터 또는 향후 데이터에서 정상적인 관측치의 고유한 패턴을 학습하여 비정상적인 관측치를 식별하는 작업입니다. 최근 한 연구에서는 생성 모델의 새로운 관찰을 바탕으로 한 강력한 비지도 이상 탐지 방법론을 도입했으며, 이를 정상치 기억 효과(IM 효과)라고 부릅니다. 이 효과는 생성 모델이 학습 초기 단계에서 이상치보다 정상치를 먼저 기억한다고 제안합니다. 이번 연구에서는 IM 효과를 최대한 활용하여 비지도 이상치 탐지 작업을 해결하기 위한 이론적으로 정립된 방법을 개발하는 것을 목표로 합니다. 우리는 주어진 훈련 데이터에 이상치가 적게 포함될수록 IM 효과가 더 명확하게 관찰된다는 점을 시작으로 합니다. 이 발견은 손실 함수를 설계할 때 미니 배치에서 이상치를 효과적으로 제외할 수 있다면 비지도 이상치 탐지 체제에서 IM 효과를 향상시킬 수 있는 가능성을 시사합니다. 이를 위해 두 가지 주요 기술을 도입합니다: 1) 모델 훈련이 진행됨에 따라 미니 배치 크기를 증가시키고, 2) 손실 함수의 일부만을 잘라내어 사용하기 위해 적응형 임계값을 사용하는 것입니다. 우리는 이 두 가지 기술이 절단된 손실 함수에서 이상치를 효과적으로 필터링하여 IM 효과를 최대한 활용할 수 있게 함을 이론적으로 보여줍니다. 추가적인 앙상블 기법과 결합하여 우리는 이 방법을 제안하며, 이를 ALTBI 라고 명명합니다. 우리는 ALTBI가 다른 최신 방법들과 비교했을 때, 훨씬 낮은 계산 비용으로도 이상치를 식별하는 데 있어 최첨단 성능을 달성함을 광범위한 실험

결과를 통해 입증합니다. 또한, 우리 방법이 프라이버시를 보호하는 알고리즘과 결합되었을 때도 견고한 성능을 발휘함을 보여줍니다.

핵심용어 : 이상치탐지, 생성 모델, IM 효과