ALTBI: Constructing Improved Outlier Detection Models via Optimization of Inlier-Memorization Effect

Seoyoung Cho[1], Jaesung Hwang[2], Kwan-Young Bak[13], Dongha Kim[13]*

[1]Department of Statistics, Sungshin Women's University
[2]SK Telecom
[3]Data Science Center, Sungshin Women's University
{katesycho, postechiminuru}@gmail.com, {kybak, dongha0718}@sungshin.ac.kr

Abstract

Outlier detection (OD) is the task of identifying unusual observations (or outliers) from a given or upcoming data by learning unique patterns of normal observations (or inliers). Recently, a study introduced a powerful unsupervised OD (UOD) solver based on a new observation of deep generative models, called *inlier-memorization (IM) effect*, which suggests that generative models memorize inliers before outliers in early learning stages. In this study, we aim to develop a theoretically principled method to address UOD tasks by *maximally utilizing the IM effect*. We begin by observing that the IM effect is observed more clearly when the given training data contain fewer outliers. This finding indicates a potential for enhancing the IM effect in UOD regimes if we can effectively exclude outliers from mini-batches when designing the loss function. To this end, we introduce two main techniques: 1) increasing the mini-batch size as the model training proceeds and 2) using an adaptive threshold to calculate the truncated loss function. We theoretically show that these two techniques effectively filter out outliers from the truncated loss function, allowing us to utilize the IM effect to the fullest. Coupled with an additional ensemble technique, we propose our method and term it *Adaptive Loss Truncation with Batch Increment (ALTBI)*. We provide extensive experimental results to demonstrate that ALTBI achieves state-of-the-art performance in identifying outliers compared to other recent methods, even with lower computation costs. Additionally, we show that our method yields robust performances when combined with privacy-preserving algorithms.

Introduction

Outlier detection: Outlier detection (OD) is an important task in various fields, aiming to identify unusual observations, or outliers. This process involves learning the distinct patterns of normal observations, known as inliers, and developing efficient scores to distinguish between inliers and outliers. The ability to accurately detect outliers is essential for ensuring data quality and reliability in applications such as fraud detection, network security, and fault diagnosis.

OD tasks can be divided into three cases depending on availability of anomalousness information of given training data. Supervised OD (SOD) uses labeled data to classify each sample as either outlier or not. Semi-supervised OD (SSOD), also known as out-of-distribution detection, assumes that all training data are normal and builds models based only on these inliers. Unsupervised OD (UOD) works with data that may contain outliers but has no labels to identify them. In UOD tasks, the primary goal is to accurately identify outliers in a given training dataset. In general, many real-world anomaly detection tasks belong to UOD because outliers in large datasets are usually unknown beforehand. Hence, we focus on addressing UOD problems in this study.

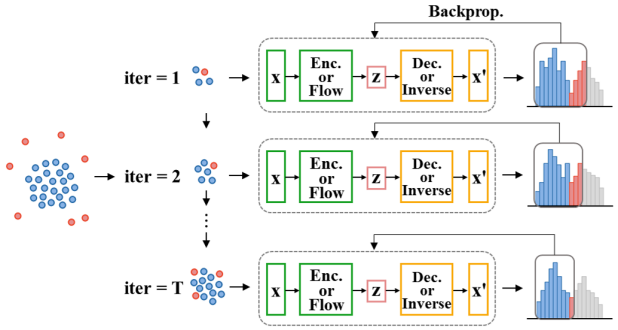

Figure 1: An illustration of ALTBI.

Recent advancements in machine learning domains have introduced powerful UOD methods. In particular, many studies have leveraged deep generative models (DGM) to develop unique scores to identify outliers. Surprisingly, conventional likelihood was not utilized, as it has been widely recognized that it often confuses outliers with inliers when the models are fully trained (Nalisnick et al. 2019b,a; Lan and Dinh 2021).

Recently, a notable study has highlighted the potential of the likelihood values of DGMs in UOD tasks, based on the observation of the *inlier-memorization (IM) effect* (Kim et al. 2024). This effect suggests that when a DGM is trained, the loss values, i.e., the negative log-likelihoods, of inliers decrease before those of outliers in early training stages. This implies that the likelihood value itself can be a favorable measure to identify outliers with *under-fitted DGMs*. Leveraging this phenomenon, Kim et al. (2024) developed a UOD solver called ODIM, which has demonstrated powerful yet computationally efficient performance in identifying

*Corresponding author.

inliers within a dataset.

Improvement of IM effect: Inspired by ODIM, this study aims to develop an enhanced UOD method by maximally exploiting the IM effect. We begin by observing that the clarity of the IM effect becomes more pronounced as the training data contain fewer outliers. We visually illustrate this simple but important finding in Figure 2. This observation suggests that boosting the IM effect could be achieved if we can effectively separate outliers from inliers and exclude them when constructing loss functions during the early training stages.

To this end, we introduce two key techniques: 1) gradually increasing the mini-batch size and 2) adopting an adaptive threshold to truncate the loss function. These techniques are designed to maximize the utility of the IM effect, ensuring more accurate identification of outliers. We provide theoretical results showing that these two techniques result in the ratio of outliers included in the truncated loss function decreasing toward zero as training proceeds.

Additionally, we incorporate an ensemble strategy of loss values within a single DGM with various updates to further enhance our method's performance and stability. We demonstrate that this simple technique significantly improves the outlier detecting performance without additional computational or resource costs.

Combining all the above elements, we develop a powerful framework for addressing UOD tasks and term it *Adaptive Loss Truncation with Batch Increment (ALTBI)*. We present Figure 1 to visualize our method. Our method has several notable advantages over other existing UOD solvers. First, our method consistently achieves superior results in detecting outliers from a given dataset. Through extensive experiments analyzing 57 datasets, we demonstrate that ALTBI achieves state-of-the-art performance in outlier detection.

Additionally, ALTBI only requires a simple and underfitted likelihood-based DGM, training with variational autoencoders (VAE, Kingma and Welling (2013)) or normalizing flows (NF, Dinh, Sohl-Dickstein, and Bengio (2017)) for up to hundreds of mini-batch updates. This makes ALTBI highly efficient, requiring significantly reduced computational costs compared to other recent methods. Our findings indicate that ALTBI is a promising approach for efficient and effective UOD solution in 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. 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.

Related Works

We first review studies dealing with SSOD problems. SVDD (Tax and Duin 2004) uses kernel functions to construct a boundary around a dataset for outlier detection, while DeepSVDD (Ruff et al. 2018) extends SVDD by using a deep autoencoder to obtain a feature space where normal data lies within a boundary, while anomalies fall outside. DeepSAD (Ruff et al. 2020) generalizes DeepSVDD by considering an extended scenarios where a small amount of labeled outliers are also available.

Self-supervised learning has been widely applied to address SSOD tasks (Tack et al. 2020; Golan and El-Yaniv 2018). In particular, SimCLR (Chen et al. 2020) leverages the contrastive learning to obtain high-quality feature representations of inliers, and ICL (Shenkar and Wolf 2022) maximizes mutual information between masked and unmasked parts of data to successfully detect outliers.

Numerous traditional approaches have been proposed to address UOD problems. LOF (Breunig et al. 2000a) detects local outliers in a dataset based on density. This idea has been extended to CBLOF (He, Xu, and Deng 2003), which identifies outliers based on the distance from the nearest cluster and the size of the cluster it belongs to, measuring the significance of an outlier. MCD (Fauconnier and Haesbroeck 2009) identifies outliers by finding a subset of the data with the smallest covariance determinant, providing powerful outlier scores using estimates of location and scatter. IF (Liu, Ting, and Zhou 2008) detects anomalies by isolating data points using tree structures.

There are also various techniques to solve UOD problems using deep learning models. RDA (Zhou and Paffenroth 2017) extends deep autoencoder by incorporating robustness against outliers, and DSEBM (Zhai et al. 2016) generates an energy function as the output of a deterministic deep neural network. Additionally, ODIM (Kim et al. 2024) utilizes the inlier-memorization effect observed in the early training updates of deep generative models to identify outliers. DTE (Livernoche et al. 2023) estimates the distribution over diffusion time for a given input and uses the mode or mean of this distribution as the anomaly score.

Detailed Description of ALTBI

Preliminaries

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, Grosse, and Salakhutdinov 2016; Kim, Hwang, and Kim 2020), or NF-based ones (Dinh, Krueger, and Bengio 2015; Dinh, Sohl-Dickstein, and Bengio 2017; Kingma and Dhariwal 2018), $l(\theta; \boldsymbol{x})$ 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 $L_i(\theta_*) = 0$.

Brief review of ODIM ODIM (Kim et al. 2024) is a UOD solver that utilizes the IM effect for the first time. The IM effect refers to a phenomenon where, when training a DGM with a given dataset that may contain outliers, the model eventually learns all the patterns of the data, but there is a gap in the memorization order between inliers and outliers. That is, the model memorizes inliers first during the early updates. Inliers are more prevalent and densely distributed than outliers, thus, reducing the per-sample loss values of inliers first is a more beneficial direction to minimize the overall (averaged) loss function in the early training stages, which might be an intuitive explanation of the IM effect.

ODIM trains likelihood-based DGMs, such as VAE (Kingma and Welling 2013), for a certain number of updates and uses the per-sample loss values as the outlier score. To find the optimal number of updates where the IM effect is most clearly observed, ODIM examines the degree of bi-modality in the per-sample loss distribution at each update.

To this end, at each update, ODIM fits a 2-cluster Gaussian mixture model to the per-sample loss distribution and calculates the dissimilarity between the two clusters using measures such as the Wasserstein distance. The bi-modality measures are monitored for all updates, and the update with the maximum measure is chosen as the optimal point.

To enhance outlier identification performance, an ensemble technique is also adopted. Multiple under-fitted DGMs are independently trained, and the final score of a given sample $\boldsymbol{x}$ is calculated as the average per-sample loss value, given as:

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

where $\theta^{(b)}, b = 1, \ldots, B$ are the B estimated parameters from independent initializations. And the sample $\boldsymbol{x}$ is regarded as an outlier if its score is large, and vice versa.

Relationship Between IM Effect and Outlier Ratio

The IM effect implies that distinguishing inliers from outliers is viable by using the per-sample loss values with an under-fitted DGM. In this section, we claim that the clarity of the IM effect increases as the proportion of outliers in the training dataset decreases. To demonstrate this, we conduct a simple experiment analyzing the `PageBlocks` dataset with various outlier ratios ranging from 1% to 9%.

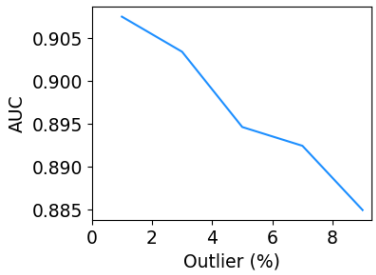

Figure 2: Relationship between the outlier ratio in training data and IM effect.

In each setting, we train a VAE for 100 mini-batch updates with a mini-batch size of 128 and evaluate the AUC values of the training data for identifying outliers using per-sample loss values.

Figure 2 illustrates that the IM effect is observed more clearly as the training data contain fewer outliers, which strongly validates our claim. This observation leads to a new belief: if we can effectively filter out outliers when constructing loss values using a mini-batch to update parameters, we can strengthen the IM effect, thereby enhancing outlier detection performance. This is the main motivation of our study.

Proposed Method

Mini-batch increment and adaptive threshold We again note that the goal is to maximize the utility of the IM effect in early training updates. To this end, we introduce two simple but powerful 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, we first train a DGM with a conventional loss function using mini-batches with a fixed size of n_0 for T_0 updates. We use this non-truncated loss function to obtain an estimated parameter where the IM effect starts to appear.

After that, as the second phase, we apply the mini-batch increment and loss truncation strategies. At each update iteration t, we access a mini-batch $\mathcal{D}_t \subset \mathcal{D}^{\text{tr}}$ whose size is *an exponential function of the iteration* t, i.e., $|\mathcal{D}_t| = n_0 \gamma^{t-1} (=: n_t)$ for a constant $\gamma > 1$. And instead of using all the samples included in $\mathcal{D}_t$ to calculate the loss function, we exploit the *truncated loss function* which is formularized as:

$$\widehat{L}(\theta, \tau_t; \mathcal{D}_t) = \frac{\sum_{\boldsymbol{x} \in \mathcal{D}_t} l(\theta; \boldsymbol{x}) \cdot I(l(\theta; \boldsymbol{x}) \leq \tau_t)}{\sum_{\boldsymbol{x}' \in \mathcal{D}_t} I(l(\theta; \boldsymbol{x}') \leq \tau_t)}, \qquad (1)$$

where $\tau_t > 0$ is an adaptive threshold.

Theoretically, we set τ_t to be the inlier risk, i.e., $\tau_t = L_i(\theta_{t-1})$, where θ_{t-1} is the estimated parameter at $(t-1)$-th update with an SGD-based optimizer. However, the computation of $\tau_t = L_i(\theta_{t-1})$ is infeasible in practice since we do not know the anomaly information of the training samples. Instead, we introduce the quantile as the value of τ_t. Specifically, for a pre-specified value $0 < \rho < 1$, we filter out the

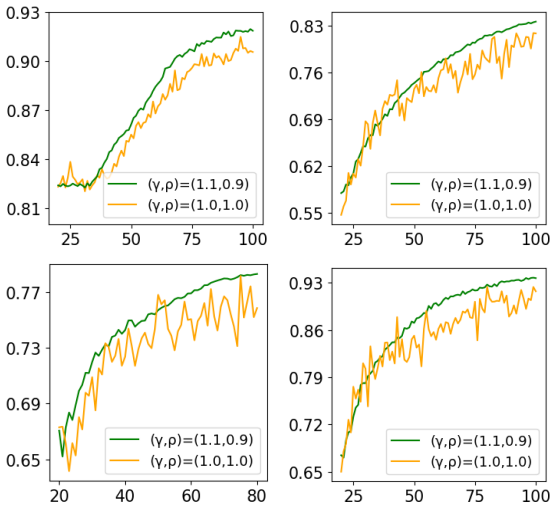

Figure 3: Outlier detection AUC values for DGMs with and without applying mini-batch increment and adaptive threshold, coloured as green and orange, respectively. (**Upper left to clockwise**) We analyze `Ionosphere`, `Letter`, `Vowels`, and `MagicGamma` datasets.

$100 \times (1-\rho)\%$ of the samples in the mini-batch that have the largest per-sample loss values.

We conduct a simple experiment to empirically validate the effect of use of mini-batch increment and adaptive threshold. Two VAEs are trained in two scenarios—one with and the other without applying the two strategies. For the first scenario, we set $(T_0, n_0, \gamma, \rho) = (10, 128, 1.1, 0.9)$.

The outlier detection AUC results of the training data over four datasets are illustrated in Figure 3. We can clearly observe that the IM effect is more pronounced when using the two strategies, leading to superior performance in outlier detection. Additionally, using an increasing mini-batch size reduces fluctuation over updates, resulting in a more stable trained model. The theoretical properties of using adaptive mini-batch size and threshold will be discussed in the subsequent 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.*

Ensemble within a single model Recall that ODIM measures the degree of bi-modality in the per-sample loss distribution to find the optimal update. We empirically find that this heuristic approach often fails to identify the optimal update where the IM effect is maximized, sometimes even selecting an update where the IM effect does not appear. Additionally, ODIM employs an ensemble method to enhance performance using multiple under-fitted models, which increases computation time and resources.

To address this issue, we neither measure bi-modality nor use multiple models. Instead, we adopt the ensemble approach *within a single model*. For given two integers T_1, T_2 with $T_1 < T_2$, we take the average of per-sample loss values from $T_1 + 1$ to T_2 updates. That is, for a given input $\boldsymbol{x}$, we compute its outlier score as

$$s^{\text{ALTBI}}(\boldsymbol{x}) = \frac{1}{T_2 - T_1} \sum_{t=T_1+1}^{T_2} l(\theta_t; \boldsymbol{x}), \tag{2}$$

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

It is obvious that using an ensemble approach with a single model for various updates leads to greater computational efficiency compared to considering multiple models. 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.

Choice of DGM framework There are numerous DGMs involved in likelihood maximization, such as VAE-based (Kingma and Welling 2013; Burda, Grosse, and Salakhutdinov 2016), NF-based (Dinh, Sohl-Dickstein, and Bengio 2017; Kingma and Dhariwal 2018), and score-based models (Ho, Jain, and Abbeel 2020; Song et al. 2021). Among these, we decide to use two approaches: IWAE (Burda, Grosse, and Salakhutdinov 2016) and GLOW (Kingma and Dhariwal 2018), which are widely used DGMs based on VAE and NF, respectively. Accordingly, their loss functions, $l(\boldsymbol{x}; \theta)$, would be the ELBO-like upper bound and exact log-likelihood, respectively. Given that one of our method's key properties is computational efficiency, we do not consider score-based DGMs due to their large model sizes.

Theoretical Analysis

In this section, we provide theoretical explanations to show that using the mini-batch size increment and truncated loss actually boosts the IM effect. We first state the rigorous definition of the IM effect below.

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, when a given DGM is trained for a while, there is a gap in risk values between inliers and outliers. 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:*

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

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

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:
  $\widehat{L}(\theta_0; \mathcal{D}_t) = \frac{1}{n_0} \sum_{j=1}^{n_0} l(\theta_0; \boldsymbol{x}_j^{\text{mb}})$.
  Update θ_0:
  $\theta_0 \leftarrow \theta_0 - \eta \cdot \nabla_\theta \widehat{L}(\theta_0; \mathcal{D}_t)$.
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 $\widehat{L}(\theta_{t-1}, \tau_t)$ as (1)
  Update θ_t:
  $\theta_t \leftarrow \theta_{t-1} - \eta \cdot \nabla_\theta \widehat{L}(\theta_{t-1}, \tau_t)$.
  if $(t > T_1)$ **then**
    Incorporate the per-sample loss values to the final ALTBI scores as (2)
  end if
end for

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

2) $L_i(\theta)$ is smooth with a L-Lipschitz continuous gradient, i.e., there exists a constant $L > 0$ such that

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

3) 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:*

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

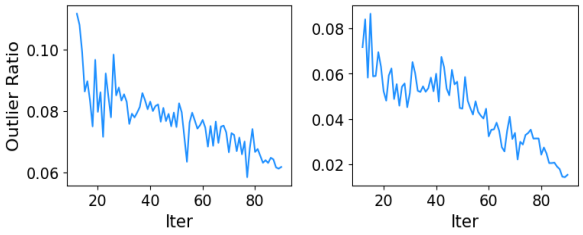

Figure 4: Trace plot of outlier ratio in truncated samples over various iterations. We visualize two datasets: (**Left**) `Cardio` and (**Right**) `Shuttle`.

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 is included in the truncated loss as $\mathcal{A}_t^\tau$. Similarly, we can define $\mathcal{B}_t^\tau$ for outliers. Then, under 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 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 $\lfloor(\log(a_1/a_2))/(\log(1/\gamma))\rfloor$ consecutive updates.

Experiments

We validate the superiority of our proposed method by analyzing an extensive set of 57 datasets, including image, text, and tabular data. We prove that ALTBI is the state-of-the-art solution for various types of data with its high computational efficiency compared to other recent methods. 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.

Dataset description We analyze all 57 outlier detection benchmark datasets from `ADBench` (Han et al. 2022), including tabular, image, and text data. And as done in Kim et al. (2024), we conduct the min-max scaling to preprocess each dataset. We first consider 46 widely used tabular datasets that cover various application domains, including healthcare, finance, and astronautics. Additionally, we include five benchmark datasets commonly used in the field of natural language processing (NLP): `20news`, `Agnews`, `Amazon`, `IMDB`, and `Yelp`. For these datasets, we utilize embedding features of these datasets via BERT (Devlin et al. 2019) or RoBERTa (Liu et al. 2019), both publicly accessible 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).

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, Rastogi, and Shim 2000), LOF (Breunig et al. 2000b), OCSVM (Schölkopf et al. 2001), CBLOF (He, Xu, and Deng 2003), PCA (Shyu et al. 2003), FeatureBagging (Lazarevic and Kumar 2005), IForest (Liu, Ting, and Zhou 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).

Implementation details As mentioned previously, we use two likelihood-based DGM frameworks: 1) IWAE (Burda, Grosse, and Salakhutdinov 2016), an ELBO-based model and 2) GLOW (Nalisnick et al. 2019b), an NF-based model.

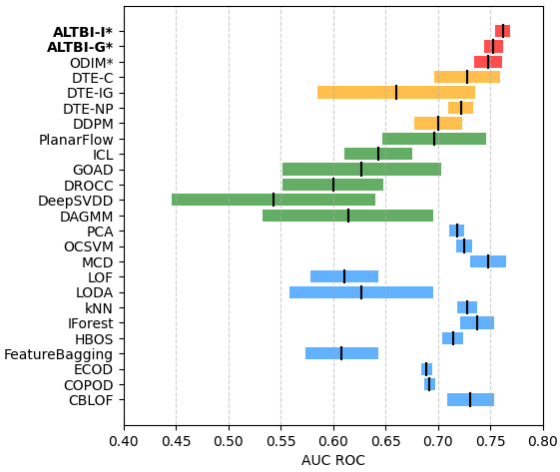

Figure 5: Averaged AUC results, including means and standard deviations, across 57 datasets from ADBench over three different implementations. We mark an asterisk (*) next to methods for our own implementations. Color scheme: red (IM-based), orange (diffusion-based), green (deep-learning-based), blue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

IWAE uses multiple latent vectors to make the objective function tighter than the standard ELBO. We use the same DNN architecture for IWAE as in Kim et al. (2024), but set the number of the latent samples K to two. Detailed descriptions of the architectures and loss functions are presented in Appendix B. And GLOW (Kingma and Dhariwal 2018) introduces invertible 1×1 convolution filters to create normalizing flows with high complexity. 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–$(n_0, \gamma, \rho, T_0, T_1, T_2)$–to (128,1.03,0.92,10,60,80), unless stated otherwise. Performance results for other hyperparameter values are provided in the ablation studies.

Performance Results

We evaluate the outlier detection performance of ALTBI with IWAE (ALTBI-I) and GLOW (ALTBI-G) in comparison with other baselines. For each dataset, the mean and standard deviation of outlier detection AUC and PRAUC over three different implementations are measured. We report the averaged means and standard deviations of AUC across all datasets in Figure 5. Detailed results for each dataset, including PRAUC, are summarized in Appendix B. We acknowledge that we implemented ALTBI and ODIM ourselves, while all other baseline results are referenced from the Appendix in Livernoche et al. (2023).

Figure 5 shows that the scores of ALTBI-I achieves the best performance, followed by ALTBI-G, both outperforming all other baselines. Considering the computational efficiency of IWAE compared to GLOW, ALTBI-I could be a

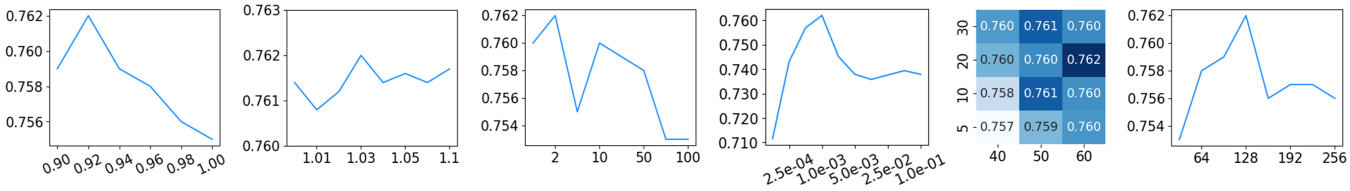

Figure 6: (**From left to right**) 1) AUC scores with various values of ρ. 2) AUC scores with various values of γ. 3) AUC scores with various values of K in IWAE. 4) AUC scores with various values of learning rate. 5) Heatmap of AUC scores for ensembling with various values of (x-axis) T_1 and (y-axis)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

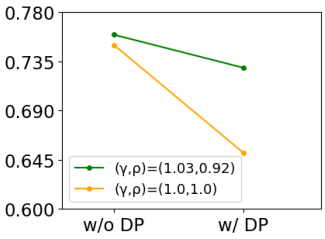

Figure 7: The impact of mini-batch increment and loss truncation when applying an DP-SGD algorithm.

more favorable method as a UOD solver. Additionally, our method showcases smaller standard deviations compared to other baselines. These results indicate that ALTBI has superior and stable performance across various data types in empirical experiments, again highlighting ALTBI as a off-the-shelf method for UOD.

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 The performance improves as the truncation percentage increases up to 8%, but it starts to decline afterward.

② Increasing the mini-batch size by a factor of $\gamma = 1.03$ at each update leads to optimal performance.

③ Performance is best when $K = 2$ in IWAE, and it tends to decline as K increases further.

④ The learning rate increases up to $1e-3$, resulting in continuous performance improvement, but after that, performance decreases and stabilizes.

⑤ 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.

⑥ Increasing n_0 enhances ALTBI's performance up to a value of 128, after which the performance begins to decline and stabilize.

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.

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 averaged outlier detection AUC values in Figure 7. 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 yields more robust performance when applying the DP algorithm, indicating that combining ALTBI and DP has a synergistic effect.

Concluding Remarks

In this study, we developed a novel UOD method called ALTBI, which maximally exploits the IM effect. By introducing two key techniques–gradually increasing the mini-batch size and adopting an adaptive threshold to truncate the loss function–ALTBI demonstrated superior and stable outlier detection performance across various datasets while maintaining computational efficiency. Extensive experiments validated ALTBI's state-of-the-art results, making it a robust and effective solution for UOD. 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.

Acknowledgments

DK was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2022R1G1A1010894 and RS2023-00218231). GB was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00342014 and RS-2022-00165581).

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 308–318.

Bergman, L.; and Hoshen, Y. 2020. Classification-Based Anomaly Detection for General Data. In *International Conference on Learning Representations*.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000a. LOF: Identifying Density-Based Local Outliers. *SIGMOD Rec.*, 29(2): 93–104.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000b. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 93–104.

Burda, Y.; Grosse, R. B.; and Salakhutdinov, R. 2016. Importance Weighted Autoencoders. In *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 1597–1607. PMLR.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. 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)*, 4171–4186. Association for Computational Linguistics.

Dinh, L.; Krueger, D.; and Bengio, Y. 2015. NICE: Nonlinear Independent Components Estimation. In *3rd International Conference on Learning Representations, ICLR 2015, Workshop Track Proceedings*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using Real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*. OpenReview.net.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net.

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

Golan, I.; and El-Yaniv, R. 2018. Deep anomaly detection using geometric transformations. *arXiv preprint arXiv:1805.10917*.

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

Goyal, S.; Raghunathan, A.; Jain, M.; Simhadri, H. V.; and Jain, P. 2020. DROCC: Deep Robust One-Class Classification. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, 3711–3721. PMLR.

Han, S.; Hu, X.; Huang, H.; Jiang, M.; and Zhao, Y. 2022. ADBench: Anomaly Detection Benchmark. In *Neural Information Processing Systems (NeurIPS)*.

He, Z.; Xu, X.; and Deng, S. 2003. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10): 1641–1650.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Kim, D.; Hwang, J.; and Kim, Y. 2020. On casting importance weighted autoencoder to an EM algorithm to learn deep generative models. In *International Conference on Artificial Intelligence and Statistics*, 2153–2163. PMLR.

Kim, D.; Hwang, J.; Lee, J.; Kim, K.; and Kim, Y. 2024. ODIM: Outlier Detection via Likelihood of Under-Fitted Generative Models. arXiv:2301.04257.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, 10236–10245.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lan, C. L.; and Dinh, L. 2021. Perfect Density Models Cannot Guarantee Anomaly Detection. *Entropy*, 23(12): 1690.

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

Li, Z.; Zhao, Y.; Botta, N.; Ionescu, C.; and Hu, X. 2020. COPOD: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, 1118–1123. IEEE.

Li, Z.; Zhao, Y.; Hu, X.; Botta, N.; Ionescu, C.; and Chen, G. 2022. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*, 413–422. IEEE.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Livernoche, V.; Jain, V.; Hezaveh, Y.; and Ravanbakhsh, S. 2023. On Diffusion Modeling for Anomaly Detection. *CoRR*, abs/2305.18593.

Nalisnick, E.; Matsukawa, A.; Teh, Y. W.; and Lakshminarayanan, B. 2019a. Detecting Out-of-Distribution Inputs to Deep Generative Models Using Typicality.

Nalisnick, E. T.; Matsukawa, A.; Teh, Y. W.; Görür, D.; and Lakshminarayanan, B. 2019b. Do Deep Generative Models Know What They Don't Know? In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

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

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

Ramaswamy, S.; Rastogi, R.; and Shim, K. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 427–438.

Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S. A.; Binder, A.; Müller, E.; and Kloft, M. 2018. Deep One-Class Classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4393–4402. PMLR.

Ruff, L.; Vandermeulen, R. A.; Görnitz, N.; Binder, A.; Müller, E.; Müller, K.-R.; and Kloft, M. 2020. Deep Semi-Supervised Anomaly Detection. In *International Conference on Learning Representations*.

Schölkopf, B.; Platt, J.; Shawe-Taylor, J.; Smola, A.; and Williamson, R. 2001. Estimating Support of a High-Dimensional Distribution. *Neural Computation*, 13: 1443–1471.

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

Shyu, M.-L.; Chen, S.-C.; Sarinnapakorn, K.; and Chang, L. 2003. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE foundations and new directions of data mining workshop*, 172–179. IEEE Press.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net.

Tack, J.; Mo, S.; Jeong, J.; and Shin, J. 2020. CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances. In *Advances in Neural Information Processing Systems*, volume 33, 11839–11852. Curran Associates, Inc.

Tax, D. M.; and Duin, R. P. 2004. Support vector data description. *Machine learning*, 54: 45–66.

Xu, Y.; Shang, L.; Ye, J.; Qian, Q.; Li, Y.; Sun, B.; Li, H.; and Jin, R. 2021. Dash: Semi-Supervised Learning with Dynamic Thresholding. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, 11525–11536. PMLR.

Yuan, Z.; Yan, Y.; Jin, R.; and Yang, T. 2019. Stagewise Training Accelerates Convergence of Testing Error Over SGD. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, 2604–2614.

Zhai, S.; Cheng, Y.; Lu, W.; and Zhang, Z. 2016. Deep structured energy based models for anomaly detection. In *International conference on machine learning*, 1100–1109. PMLR.

Zhou, C.; and Paffenroth, R. C. 2017. Anomaly Detection with Robust Deep Autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS*, 665–674. ACM.

Zong, B.; Song, Q.; Min, M. R.; Cheng, W.; Lumezanu, C.; Cho, D.; and Chen, H. 2018. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In *International Conference on Learning Representations*.