Meta-learning for Robust Anomaly Detection

Atsutoshi Kumagai
NTT

Tomoharu Iwata
NTT

Hiroshi Takahashi
NTT Docomo

Yasuhiro Fujiwara
NTT

Abstract

We propose a meta-learning method to improve the anomaly detection performance on unseen target tasks that have only unlabeled data. Existing meta-learning methods for anomaly detection have shown remarkable performance but require labeled data in target tasks. Although they can treat unlabeled data as normal assuming anomalies in the unlabeled data are negligible, this assumption is often violated in practice. As a result, the methods have low performance. Our method meta-learns with related tasks that have labeled and unlabeled data such that the expected test anomaly detection performance is directly improved when the anomaly detector is adapted to given unlabeled data. Our method is based on autoencoders (AEs), which are widely used neural network-based anomaly detectors. We model anomalous attributes for each unlabeled instance in the reconstruction loss of the AE, which are used to prevent the anomalies from being reconstructed; they can remove the effect of the anomalies. We formulate adaptation to the unlabeled data as a learning problem of the last layer of the AE and the anomalous attributes. This formulation enables the optimum solution to be obtained with a closed-form alternate update formula, which is preferable to efficiently maximize the expected test anomaly detection performance. The effectiveness of our method is experimentally shown with four real-world datasets.

1 INTRODUCTION

Anomaly detection is an important problem in machine learning, which attempts to detect anomalies or outliers that do not conform to the expected normal pattern (Chandola et al., 2009; Ruff et al., 2021; Salehi et al., 2021). Anomaly detection has been successfully used in various applications such as intrusion detection (Dokas et al., 2002), fraud detection (Kou et al., 2004), medical diagnosis (Fernando et al., 2020), and failure detection (Idé et al., 2017).

Many unsupervised anomaly detection methods have been proposed such as autoencoders (AEs) (Sakurada and Yairi, 2014), isolation forests (IFs) (Liu et al., 2008), and one-class support vector machines (OSVMs) (Schölkopf et al., 2001). Since they use only unlabeled data, which are easy to prepare, they are widely used in practice. To learn the anomaly detectors from unlabeled data, they assume that most unlabeled instances are normal. However, this assumption is often violated in real-word applications. For example, in cyber security, normal behavior-based detection systems collect unlabeled data of each user to model the user's normal pattern. If the user is infected with malware, the dataset is contaminated by a large amount of anomalous activities. Moreover, even if unlabeled data contain few anomalies, these methods can be affected by the anomalies, often resulting in low performance (Chalapathy et al., 2017; Beggel et al., 2019).

When the label (normal or anomalies) information of each instance is available, supervised or semi-supervised anomaly detection methods can improve the performance (Ruff et al., 2019; Yamanaka et al., 2019; Akcay et al., 2018). However, labeled data are often difficult to collect since annotation requires domain expertise and is time-consuming. Especially, when there are many target tasks (e.g., there are many new users appearing one after another), the data are more difficult to collect for every task.

Although labeled data are difficult to collect in a target task, they might be available in related tasks, called source tasks. In the above example, anomalous and normal data might be obtained from other users who have existed for a long time. Several meta-learning methods have been recently proposed that use labeled data in multiple source tasks for anomaly detection on unseen target tasks (Frikha et al., 2021; Iwata and Kumagai, 2021; Kumagai et al., 2019; Kruspe, 2019; Ding et al., 2021). Meta-learning is formulated as a bilevel optimization problem. In the inner optimization problem, task-specific models are adapted to the given task-specific instances, called a support set. In the

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

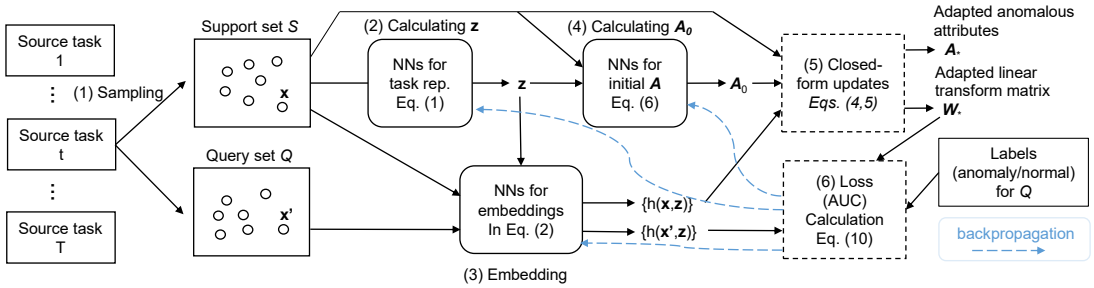


Figure 1: Overview of our meta-learning procedure. (1) For each training iteration, we randomly sample unlabeled instances (support set) and labeled instances (query set) from a randomly selected source task. (2) We calculate task representation $\mathbf{z}$ from the support set by neural networks. (3) Each instance is embedded to a task-specific embedded space by neural networks with $\mathbf{z}$. (4) Initial anomalous attributes $\mathbf{A}_0$ are calculated by neural networks with the support set. (5) Linear transformation matrix of the AE $\mathbf{W}$ is adapted by minimizing the reconstruction error of the support set while adapting anomalous attributes $\mathbf{A}$. (6) AUC is calculated on the query set and is backpropagated to update all neural networks.

outer optimization problem, common parameters shared across all tasks are meta-learned to improve the expected test performance when the task-specific model adapted in the inner problem is used. Although these methods are useful, they require normal data (Frikha et al., 2021; Kumagai et al., 2019; Kruspe, 2019) or both normal and anomalous data (Iwata and Kumagai, 2021; Ding et al., 2021) in target tasks. Thus, they are inappropriate when only unlabeled target data are available. Further, simply treating unlabeled data as normal data significantly degrades their performance, which will be demonstrated in our experiments.

In this paper, we propose a method to learn anomaly detectors appropriate for unseen target tasks that have only unlabeled data by meta-learning with multiple source tasks. We assume that each source task has both labeled (anomalous and normal) and unlabeled data. The anomaly score of each instance is calculated based on the reconstruction error of AE, which is modeled by neural networks, and has been used in various anomaly detection problems (Sakurada and Yairi, 2014; Kumagai et al., 2019; Chen et al., 2017).

With the proposed method, the inner problem corresponds to adapting the AE to the given unlabeled support set. To effectively adapt to various tasks, task-specific AEs need to be constructed. To this end, we calculate a vector representation of each task by permutation-invariant neural networks (Zaheer et al., 2017) that take the support set as input. This task representation contains information of the support set. With the task representation, the AE first nonlinearly encodes each instance to a task-specific embedding space appropriate for the task. Then, the AE decodes each embedded instance into the original space by a linear transformation. By minimizing the reconstruction errors between instances and reconstructed ones, the AE can learn the normal pattern when data are all normal. However, since the unlabeled support set may contain anomalies in our setting, minimizing the reconstruction errors of all unlabeled data causes performance degradation. To tackle this problem, we introduce anomalous attributes for each unlabeled instance in the reconstruction loss of the AE, which are used to prevent the anomalies from being reconstructed, as in recent studies (Chalapathy et al., 2017; Zhou and Paffenroth, 2017). By estimating both the anomalous attributes and the linear transformation matrix of the AE with the support set, we can effectively learn the normal pattern while removing the effects of the anomalies. Our formulation enables the inner problem to be solved by a closed-form alternative update formula, which is preferable for the efficient meta-learning. Moreover, the proposed method has an advantage in terms of interpretability since we can identify the anomalies in the support set by investigating the anomalous attributes. We further model the initial anomalous attributes of each support instance by neural networks that take both the instance and the task representation as input. By this modeling, we can obtain good initial anomalous attributes for each task and can efficiently adapt to the task even with a few updates.

In the outer problem, we meta-learn all the neural networks such that the anomaly detection performance improves when the AE is adapted to the support set. All parameters of the neural networks are common parameters shared across all tasks. Since the solution of the inner problem is differentiable and can be easily calculated due to the closed-form update formula, we can solve the bilevel optimization problem efficiently by a gradient descent method. We use the differential approximation of the area under the ROC curve (AUC) as the objective function in the outer problem, which is often used in previous anomaly detection studies since the AUC can precisely evaluate the performance in class-imbalanced problems (Bekkar et al., 2013; Iwata and Yamanaka, 2019; Kumagai et al., 2019). By maximizing the expected test AUC, we can learn accurate anomaly detectors for unseen target tasks from unlabeled target data. Figure 1 shows the overview of our meta-learning procedure.

2 RELATED WORK

Many unsupervised anomaly detection methods have been proposed such as AE-based methods (Sakurada and Yairi, 2014; Somepalli et al., 2021; Perera et al., 2019; Gong et al., 2019; Yoon et al., 2021; Zong et al., 2018), classification-based methods (Ruff et al., 2018; Golan and El-Yaniv, 2018; Bergman and Hoshen, 2019), and density estimation-based methods (An and Cho, 2015; Zong et al., 2018; Ren et al., 2019; Phan and Idé, 2019). These methods can treat unlabeled data as normal data to learn the normal pattern (anomaly detectors). However, they do not have an explicit mechanism to exclude anomalies in unlabeled data. To robustly learn anomaly detectors from unlabeled data, several techniques has been proposed such as using soft margins (Schölkopf et al., 2001), ensemble learning (Liu et al., 2008; Aryal et al., 2014), noise robust probabilistic modeling (Eduardo et al., 2020), using gradient properties (Wang et al., 2019), and using robust statistics (Rousseeuw and Hubert, 2011). Among them, an approach to model anomalous attributes of training data and exclude them from a detector's training is widely used due to its performance and interpretability (Xiong et al., 2011; Huang et al., 2009). Especially, robust AEs, which model anomalous attributes in the AE framework, work well by using the high expressive capability of neural networks (Chalapathy et al., 2017; Zhou and Paffenroth, 2017). We incorporate this approach in our meta-learning framework since its formulation enables us to solve the inner problem efficiently and effectively as described later. Although these methods are effective, they often greatly deteriorate the performance when the effect of the anomalies is not small (Zhang et al., 2021; Zhou and Paffenroth, 2017). Semi-supervised or supervised approaches use labeled data to improve the performance (Ruff et al., 2019; Akcay et al., 2018; Pang et al., 2019; Zhang et al., 2021). Although they are useful, they cannot be applied to our problem where there are no labeled data in target tasks.

Transfer learning methods for anomaly detection, which use source data and unlabeled target data, have been proposed (Vincent et al., 2020; Michau and Fink, 2021; Fan et al., 2021). They usually treat only two tasks (source and target tasks) and require target data in the training phase. In contrast, the proposed method treats multiple tasks and does not require any target data in the (meta-)training phase. After meta-learning with source tasks, the proposed method can quickly and effectively adapt to target tasks given unlabeled target data in the test phase, which is especially preferable when many target tasks appear one after another.

Many meta-learning methods have recently been proposed that aim to efficiently and effectively adapt to new tasks by using multiple tasks (Hospedales et al., 2020). Although most are designed for few-shot classification, there are several methods for anomaly detection (Frikha et al., 2021; Iwata and Kumagai, 2021; Kumagai et al., 2019; Kruspe, 2019; Ding et al., 2021; Dahia and Pamplona Segundo, 2021; Huang et al., 2022). Meta-learning methods for anomaly detection based on model-agnostic meta-learning (MAML) (Finn et al., 2017) have been proposed (Frikha et al., 2021; Wu et al., 2021; Lu et al., 2020). In the inner problem, the MAML-based methods adapt the whole parameters of neural networks to the support set by gradient descent methods. Since they require the second-order derivative of the whole parameters for training, they have considerable computation and memory burdens (Rajeswaran et al., 2019; Bertinetto et al., 2018). In contrast, the proposed method adapts two parameter matrices (linear transformation and anomalous attribute matrices), and the adapted parameters are obtained by simple calculation with the closed-from update formula, which achieves more efficient adaptation. Some methods are based on encoder-decoder-based meta-learning methods such as neural processes (NPs) (Garnelo et al., 2018a,b) that perform efficient adaptation by forwarding the support set to neural networks (Kumagai et al., 2019; Oladosu et al., 2020). They use only neural networks for the adaptation and do not directly minimize the loss of the support set when adapting to target tasks. Thus, they might have difficulty performing effective adaptation. In contrast, the proposed method explicitly minimizes reconstruction errors of support instances, which leads to more effective adaptation. All these methods assume normal data or both normal and anomalous data in target tasks. Therefore, they will not work well when there are only unlabeled target data. Zhao et al. (2021) proposed a recommendation method for unsupervised anomaly detection algorithms such as OSVMs and IFs using labeled data in source tasks. This method only recommends an algorithm and cannot learn its detector.

Several methods use the meta-learning techniques such as MAML for learning from noisy labeled data (Zheng et al., 2021; Ren et al., 2018; Shu et al., 2019; Wang et al., 2020; Zhang et al., 2019). These methods assume a single task that has clean labeled and noisy labeled data. Unlike the proposed method, they cannot handle multiple tasks and are not methods for anomaly detection.

Some methods have been proposed for robust few-shot classification (Zhu et al., 2020; Killamsetty and Li, 2022; Chen et al., 2020; Liang et al., 2022). They learn classifiers from noisy labeled data by meta-learning in source tasks. Although the proposed method requires unlabeled data in target tasks, they require noisy labeled data in target tasks. In addition, they are also not designed for anomaly detection. To the best of our knowledge, there are no meta-learning methods for robust anomaly detection on target tasks containing only unlabeled data.

3 PROPOSED METHOD

3.1 Problem Setup

Let $X_t := \{\mathbf{x}_{tn}\}_{n=1}^{N_t}$ be a set of unlabeled instances in the t-th task, where $\mathbf{x}_{tn} \in \mathbb{R}^D$ is the D-dimensional feature vector of the n-th unlabeled instance of the t-th task and N_t is the number of the unlabeled instances in the t-th task. Similarly, let $X_t^{\mathrm{A}} := \{\mathbf{x}_{tn}^{\mathrm{A}}\}_{n=1}^{N_t^{\mathrm{A}}}$ and $X_t^{\mathrm{N}} := \{\mathbf{x}_{tn}^{\mathrm{N}}\}_{n=1}^{N_t^{\mathrm{N}}}$ be anomalous and normal instances of the t-th task, respectively. We assume that feature vector size D is the same across all tasks, but the joint distribution of instances and labels can differ across tasks. In the training phase, T source tasks $\mathcal{D} := \{X_t \cup X_t^{\mathrm{A}} \cup X_t^{\mathrm{N}}\}_{t=1}^{T}$ are given. Although we assume that each source task has anomalous instances in the main paper, our method can also handle source tasks that do not contain anomalous instances, as described in Section A. In the test phase, we are given unlabeled instances (support set) $\mathcal{S} := \{\mathbf{x}_n\}_{n=1}^{N_\mathcal{S}}$ in a target task, which is different from source tasks. Our aim is to identify whether any instance $\mathbf{x}$ in the target task is anomalous or not.

3.2 Model

We present our model that outputs an anomaly score $s(\mathbf{x}; \mathcal{S})$ of instance $\mathbf{x}$ given unlabeled support set $\mathcal{S}$. First, our model calculates a task representation from $\mathcal{S}$ by using permutation-invariant neural networks (Zaheer et al., 2017):

$$\mathbf{z} := g\left(\frac{1}{N_\mathcal{S}} \sum_{n=1}^{N_\mathcal{S}} f(\mathbf{x}_n)\right) \in \mathbb{R}^K, \tag{1}$$

where f and g are any feed-forward neural networks. Since summation is permutation-invariant, the neural network in Eq. (1) outputs the same vector even when the order of support instances varies. In addition, this neural network can handle different numbers of instances. Thus, the neural network in Eq. (1) is well defined as functions for set inputs. Task representation $\mathbf{z}$ contains information of the empirical distribution of instances in the support set. The proposed method can use any other permutation-invariant function such as summation (Zaheer et al., 2017) and set transformer (Lee et al., 2019a) for the task representations.

With the proposed method, anomaly score $s(\mathbf{x}; \mathcal{S})$ is defined by the reconstruction error of AE:

$$s(\mathbf{x}; \mathcal{S}) := \| \mathbf{x} - \mathbf{W} h([\mathbf{x}, \mathbf{z}]) \|_2^2, \tag{2}$$

where $[\cdot, \cdot]$ is concatenation, $h : \mathbb{R}^{D+K} \to \mathbb{R}^J$ is a feed-forward neural network as the encoder, $\mathbf{W} \in \mathbb{R}^{D \times J}$ is a linear weight matrix as the decoder, and $\| \cdot \|_2$ is ℓ_2 norm. Since this score depends on $\mathbf{z}$, it can change its properties to fit the task in accordance with $\mathbf{z}$. Linear decoder $\mathbf{W}$ is important for deriving the analytical update formula, as discussed later. For anomaly detection, AE is usually trained with normal data, and thus the reconstruction errors of normal instances become low. In contrast, the reconstruction errors of anomalous instances can be expected to be high since anomalies are not learned. However, the support set may contain anomalies in our setting, and thus minimizing the reconstruction errors of all instances would cause performance degradation.

To deal with this problem, we consider the following convex loss function to be minimized:

$$\mathcal{L}(\mathbf{W}, \mathbf{A}; \mathcal{S}) := \frac{1}{N_\mathcal{S}} \sum_{n=1}^{N_\mathcal{S}} \| \mathbf{x}_n + \mathbf{a}_n - \mathbf{W} h([\mathbf{x}_n, \mathbf{z}]) \|_2^2 + \frac{\lambda}{N_\mathcal{S}} \sum_{n=1}^{N_\mathcal{S}} \| \mathbf{a}_n \|_1 + \mu \| \mathbf{W} \|_{\mathrm{F}}^2, \tag{3}$$

where $\mathbf{a}_n \in \mathbb{R}^D$ represents anomalous attributes for the n-th instance $\mathbf{x}_n$, $\mathbf{A} := (\mathbf{a}_1, \ldots, \mathbf{a}_{N_\mathcal{S}})^\top \in \mathbb{R}^{N_\mathcal{S} \times D}$, $\| \cdot \|_1$ is ℓ_1 norm, $\| \cdot \|_{\mathrm{F}}$ is Frobenius norm, and λ and μ are positive real numbers, respectively. Standard AEs (i.e., the case of $\mathbf{A} = \mathbf{0}$ in Eq. (3)) force the reconstruction error to be smaller even if anomalies in the support set are more difficult to reconstruct than the other support instances. In contrast, the proposed method can avoid forcing such erroneous reconstruction by estimating non-zero anomalous attributes $\mathbf{a}_n$ for such instances. In other words, by minimizing Eq. (3), we can expect that linear weight matrix $\mathbf{W}$ can be learned without being affected by anomalies. By adapting only $\mathbf{W}$ with fixed neural network h, we can obtain the optimal solution of Eq. (3). Several studies have shown the adaptation of only the last layer performs quite well (Bertinetto et al., 2018; Lee et al., 2019b; Kumagai et al., 2021). By the ℓ_1 regularizer, many attributes of $\mathbf{A}$ become zeros, which is natural since the number of anomalies is usually not large. We note that if there is no regularization ($\lambda = 0$), the optimal solution becomes trivial ($\mathbf{A} = -\mathbf{X}_\mathcal{S}$ and $\mathbf{W} = \mathbf{0}$.)

The loss in Eq. (3) can be minimized by alternatively updating $\mathbf{W}$ and $\mathbf{A}$ with the following update rules:

$$\mathbf{W} = \frac{1}{N_\mathcal{S}} (\mathbf{X}_\mathcal{S} + \mathbf{A})^\top \mathbf{H}_\mathcal{S} \left(\frac{1}{N_\mathcal{S}} \mathbf{H}_\mathcal{S}^\top \mathbf{H}_\mathcal{S} + \mu \mathbf{I}_J\right)^{-1}, \tag{4}$$

$$\mathbf{A} = \mathrm{sgn}(\mathbf{H}_\mathcal{S} \mathbf{W}^\top - \mathbf{X}_\mathcal{S}) \left[|\mathbf{H}_\mathcal{S} \mathbf{W}^\top - \mathbf{X}_\mathcal{S}| - \frac{\lambda}{2} \mathbf{1}_{N_\mathcal{S}} \mathbf{1}_D^\top \right]_+, \tag{5}$$

where $\mathbf{X}_\mathcal{S} := (\mathbf{x}_1, \ldots, \mathbf{x}_{N_\mathcal{S}})^\top \in \mathbb{R}^{N_\mathcal{S} \times D}$, $\mathbf{H}_\mathcal{S} := (h([\mathbf{x}_1, \mathbf{z}]), \ldots, h([\mathbf{x}_{N_\mathcal{S}}, \mathbf{z}]))^\top \in \mathbb{R}^{N_\mathcal{S} \times J}$, $\mathbf{I}_J$ is a J-dimensional identity matrix, $\mathbf{1}_c$ is a c-dimensional vector whose elements are all one, sgn is a sign function, and $[\cdot]_+ = \mathrm{ReLU}(\cdot)$. $\mathrm{sgn}(\cdot)$, $[\cdot]_+$, and $|\cdot|$ are applied to each component of the matrix. The first step Eq. (4) is

derived from the condition of stationary points $\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \mathbf{0}$ for fixed $\mathbf{A}$. Similarly, the second step Eq. (5) is derived by minimizing $\mathcal{L}$ w.r.t. $\mathbf{A}$ for fixed $\mathbf{W}$, which is equivalent to using the soft thresholding operator (Bach et al., 2011). In Eq. (4), $\mu > 0$ makes matrix $(\frac{1}{N_\mathcal{S}}\mathbf{H}_\mathcal{S}^\top \mathbf{H}_\mathcal{S} + \mu \mathbf{I}_\mathcal{S})$ positive-definite. By repeating a large number of iterations of Eqs. (4) and (5), we can obtain the optimum solution of Eq. (3). However, many iterations can be problematic in the meta-learning since they significantly increase the computation cost (Rajeswaran et al., 2019; Bertinetto et al., 2018). To alleviate this problem, good initial parameters for each task need to be estimated. To this end, the proposed method models the initial parameters of $\mathbf{A}$, $\mathbf{A}_0 := (\mathbf{a}_{01}, \dots, \mathbf{a}_{0N_\mathcal{S}})^\top \in \mathbb{R}^{N_\mathcal{S} \times D}$, by the following neural networks:

$$\mathbf{a}_{0n} := v([\mathbf{x}_n, \mathbf{z}]) \in \mathbb{R}^D, \tag{6}$$

where v is any feed-forward neural networks. Since initial anomalous attributes $\mathbf{a}_{0n}$ for n-th instance $\mathbf{x}_n$ depend on $\mathbf{z}$, $\mathbf{a}_{0n}$ can take appropriate values in accordance with $\mathbf{z}$. This neural network is meta-learned such that it outputs good initial anomalous attributes as described in Section 3.3. By using an adapted linear weight matrix $\mathbf{W}_*$ that is obtained after I iterations of Eqs. (4) and (5), the anomaly score is calculated by

$$s_*(\mathbf{x}; \mathcal{S}) = \| \mathbf{x} - \mathbf{W}_* h([\mathbf{x}, \mathbf{z}]) \|_2^2 . \tag{7}$$

We note that our formulation is not restricted to AEs. For example, it can be easily applied to DeepSVDD (Ruff et al., 2018), which is a widely used anomaly detection method. We describe the details in Section B.

3.3 Training

We explain the training procedure for our model. In this subsection, we use notation $\mathcal{S}$ as a support set in source tasks. The common parameters to be meta-learned Θ are parameters of neural networks f, g, h, and v, and regularization parameters λ and μ. In the outer problem, we want to improve expected test AUC when the AE adapted in the inner problem with support set $\mathcal{S}$ is used:

$$\max_{\Theta} \mathbb{E}_{t \sim \{1,\dots,T\}} \mathbb{E}_{(\mathcal{S},\mathcal{Q}) \sim \mathcal{D}_t} \left[\mathrm{AUC}(\mathcal{Q}; \mathcal{S}, \Theta) \right], \tag{8}$$

where $\mathbb{E}$ is expectation, $\mathcal{Q} = \{\mathbf{x}_n^\mathrm{A}\}_{n=1}^{N_\mathrm{A}} \cup \{\mathbf{x}_m^\mathrm{N}\}_{m=1}^{N_\mathrm{N}}$ is labeled data drawn from the same task as support set $\mathcal{S}$, called a query set, and

$$\mathrm{AUC}(\mathcal{Q}; \mathcal{S}, \Theta) = \frac{1}{N_\mathrm{A} N_\mathrm{N}} \sum_{n=1}^{N_\mathrm{A}} \sum_{m=1}^{N_\mathrm{N}} I(s_*(\mathbf{x}_n^\mathrm{A}; \mathcal{S}, \Theta) > s_*(\mathbf{x}_m^\mathrm{N}; \mathcal{S}, \Theta)), \tag{9}$$

where I is the indicator function, i.e., $I(U) = 1$ when U is true, $I(U) = 0$ otherwise, and $s_*(\mathbf{x}; \mathcal{S}, \Theta)$ is an anomaly score of $\mathbf{x}$ with the AE adapted to support set $\mathcal{S}$ in Eq. (7). Here, we explicitly describe the dependency of common parameter Θ for clarity. Since the AUC takes a high value when anomalous instances take higher anomaly scores than normal ones, it is the successfully used metric in class-imbalanced problems such as anomaly detection (Bekkar et al., 2013; Iwata and Yamanaka, 2019). To make the AUC differentiable, we use the following smoothed approximation of the AUC:

$$\widetilde{\mathrm{AUC}}(\mathcal{Q}; \mathcal{S}, \Theta) = \frac{1}{N_\mathrm{A} N_\mathrm{N}} \sum_{n=1}^{N_\mathrm{A}} \sum_{m=1}^{N_\mathrm{N}} \sigma(s_*(\mathbf{x}_n^\mathrm{A}; \mathcal{S}, \Theta) - s_*(\mathbf{x}_m^\mathrm{N}; \mathcal{S}, \Theta)), \tag{10}$$

where $\sigma(u) = \frac{1}{1+\exp(-u)}$ is the sigmoid function used for the smoothed approximation of the indicator function (Iwata and Yamanaka, 2019). Since the anomaly score $s_*(\mathbf{x}; \mathcal{S}, \Theta)$ is easily obtained by the closed-form update formula in Eqs. (4) and (5), the outer problem in Eq. (8) is efficiently constructed. In addition, the outer problem is differentiable since anomaly score $s_*(\mathbf{x}; \mathcal{S}, \Theta)$ is differentiable. Thus, we can solve it by a stochastic gradient descent method. Algorithm 1 shows the pseudocode for our training procedure. For each iteration, we randomly sample task t from source tasks (Line 2). From unlabeled data X_t^U, we randomly sample support set $\mathcal{S}$ (Lines 3). From labeled data $X_t^\mathrm{A} \cup X_t^\mathrm{N}$, we randomly sample query set $\mathcal{Q}$ (Lines 4). We note that even when there are only labeled data in a task, we can sample unlabeled data from labeled data without using label information. We calculate task representation $\mathbf{z}$ (Line 5) and initial anomalous attributes $\mathbf{A}_0$ from $\mathcal{S}$

Algorithm 1 Training procedure of our model.

Require: Datasets in source tasks $\mathcal{D}$, support set size $N_\mathcal{S}$, query set size $N_\mathcal{Q}$, and the number of iterations for solving the inner problem I
Ensure: Common parameters of our model Θ

1: **repeat**
2: Randomly sample task t from $\{1, \dots, T\}$
3: Randomly sample support set $\mathcal{S}$ with size $N_\mathcal{S}$ from X_t^U
4: Randomly sample query set $\mathcal{Q}$ with size $N_\mathcal{Q}$ from $X_t^\mathrm{A} \cup X_t^\mathrm{N}$
5: Calculate task representation $\mathbf{z}$ from $\mathcal{S}$ by Eq. (1)
6: Calculate initial anomalous attributes $\mathbf{A}_0$ from $\mathcal{S}$ by Eq. (6)
7: **for** $l := 1$ to I **do**
8: Update linear weights $\mathbf{W}$ and anomalous attributes $\mathbf{A}$ with $\mathcal{S}$ by Eqs. (4) and (5)
9: **end for**
10: Calculate the smoothed AUC on $\mathcal{Q}$ by Eq. (10)
11: Update common parameters Θ with the gradients of the smoothed AUC
12: **until** End condition is satisfied;

(Line 6). We iteratively update linear weight matrix $\mathbf{W}$ and anomalous attributes $\mathbf{A}$ with $\mathcal{S}$ by Eqs. (4) and (5) (Lines 7 – 9). Using the adapted linear weight matrix $\mathbf{W}_*$, we calculate the smoothed AUC on query set $\mathcal{Q}$ (Line 10). Lastly, the common parameters Θ are updated with the gradient of the smoothed AUC (Line 11).

4 EXPERIMENTS

4.1 Data

We used four real-world datasets: Omniglot, Mnist-r, Isolet, and IoT. These datasets have been commonly used in meta-learning for anomaly detection studies (Kruspe, 2019; Frikha et al., 2021; Kumagai et al., 2019, 2021) [1]. Omniglot consists of hand-written images of 964 characters (classes) from 50 alphabets (Lake et al., 2015). Each class has 20 images and its feature dimension is 784. Mnist-r is created from the Mnist dataset by rotating the images (Ghifary et al., 2015). This dataset has six domains (six rotating angles) with 10 class (digit) labels. Each class of each domain has 100 instances and its feature dimension is 256. Isolet consists of 26 letters (classes) spoken by 150 speakers, and speakers are grouped into five groups (domains) by speaking similarity (Fanty and Cole, 1990). Each instance is represented as a 617-dimensional vector. IoT is real network traffic data for cyber security, which are generated from nine IoT devices (domains) infected by malware (Meidan et al., 2018). Each instance is represented by a 115-dimensional vector. For each domain, we randomly used 500 normal and 500 malicious (anomalous) instances.

For Omniglot, we first randomly split all 964 classes into three groups: 764, 100, and 100 classes. Then, following previous anomaly detection studies (Frikha et al., 2021; Kruspe, 2019; Ruff et al., 2018, 2019), we created anomaly detection tasks by regarding one class in a group as normal and the others in the same group as anomalous. By changing normal classes, we created 764 source, 100 validation, and 100 target anomaly detection tasks. Note that creating tasks in the same dataset is a standard procedure in meta-learning (for anomaly detection) studies (Frikha et al., 2021; Kruspe, 2019; Kumagai et al., 2019, 2021; Snell et al., 2017; Finn et al., 2017), and data do not overlap in the source, validation, and target tasks. For Mnist-r, we first split all six domains into four, one, and one. Then, by using the same procedure for Omniglot, we created 40 source, 10 validation, and 10 target anomaly detection tasks. Similarly, for Isolet, we split all five domains into three, one, and one, and then created 78 source, 26 validation, and 26 target anomaly detection tasks. IoT has a natural multiple task structure, so we directly split all nine domains into six source, two validation, and one target anomaly detection tasks. For each dataset, we randomly created 10 different source/validation/target anomaly detection task splits.

For each source/validation task, we set the number of labeled data to be l_r times the number of all normal data in the task. We investigated the cases of $l_r = 0.1, 0.2,$ and 0.3. Within the labeled data, 10% of instances are anomalous and 90% are normal. We used the remaining normal instances in each task as part of the unlabeled data. We injected the remaining anomalous instances into the unlabeled data so that its anomaly ratio became a_r. For each task, a_r was uniformly randomly selected from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Thus, unlabeled data in each task had different numbers of anomalies. We note that labeled and unlabeled data in each task have different anomaly ratios in our setting, which is a challenging setting since the anomaly ratio in unlabeled data is not easy to estimate from labeled data (Saerens et al., 2002; Du Plessis and Sugiyama, 2014). For each target task, we set the number of unlabeled data (support set) $N_{\mathcal{S}}$ to half of all normal data in the task. We evaluated mean test AUCs on the target tasks by changing anomaly ratios in the target support set within $\{0.1, 0.2, 0.3, 0.4, 0.5\}$.

4.2 Comparison Methods

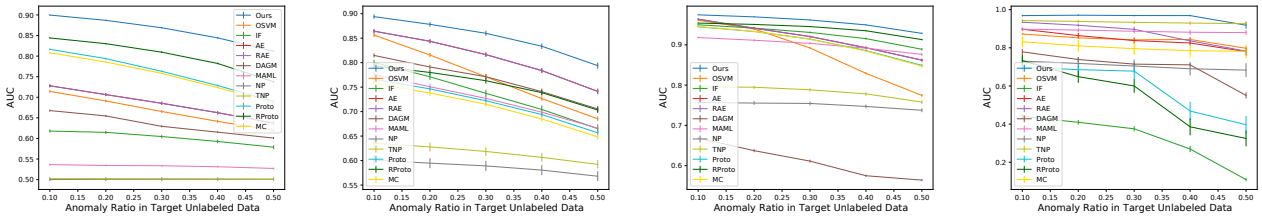
We compared the proposed method with 11 existing methods: OSVM (Schölkopf et al., 2001), IF (Liu et al., 2008), AE (Sakurada and Yairi, 2014), robust AE (RAE) (Chalapathy et al., 2017), deep autoencoding Gaussian mixture model (DAGM) (Zong et al., 2018), MAML-based methods (MAML) (Frikha et al., 2021), NP-based methods (NP and transfer anomaly detection with neural processes (TNP) (Kumagai et al., 2019)), prototypical network (Snell et al., 2017) -based methods (Proto (Dahia and Pamplona Segundo, 2021) and robust Proto (RProto) (Zhu et al., 2020)), and metacleaner-based method (MC) (Zhang et al., 2019).

OSVM, IF, AE, and DAGM are widely used unsupervised anomaly detection methods. RAE is an extension of AE made robust to outliers by modeling anomalous attributes of data. These methods use only the target unlabeled support set for training. We evaluated these methods to investigate the effectiveness of using source tasks. MAML, NP, TNP, and Proto are meta-learning methods for one-class anomaly detection. These methods use unlabeled support sets assuming that anomalies in the unlabeled data are negligible. In the inner problem, MAML adapts binary classifiers to the unlabeled support set with cross-entropy loss to meta-learn the initial classifier parameters by a gradient-based method. NP and TNP are encoder-decoder-based meta-learning methods. They infer task-specific AEs from the unlabeled support set by permutation-invariant neural

[1] We did not use other datasets, such as Fashion-MNIST (Xiao et al., 2017) and MVTec AD (Bergmann et al., 2021), since they do not have multiple domains or tasks.

Table 1: Average test AUCs [%] over different label ratios in source tasks within $\{0.1, 0.2, 0.3\}$ and anomaly ratios in target unlabeled support set within $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Boldface denotes the best and comparable methods according to the paired t-test and the significance level of 5 %.

Data	Ours	OSVM	IF	AE	RAE	DAGM	NP	TNP	MAML	Proto	RProto	MC
Omniglot	**86.23**	66.63	60.17	68.39	68.39	63.36	50.07	50.18	53.26	75.83	80.07	75.22
Mnist-r	**85.19**	77.10	73.57	80.99	80.99	76.47	58.69	61.62	72.37	71.71	75.57	70.88
Isolet	**95.69**	87.93	92.54	91.61	91.63	61.11	75.03	78.31	90.03	90.54	93.97	90.50
IoT	**95.94**	84.16	32.09	84.09	87.38	69.87	70.53	93.42	88.79	58.53	53.85	80.09



(a) Omniglot (b) Mnist-r (c) Isolet (d) IoT

Figure 2: Average and standard errors of test AUCs over different label ratios in source tasks when changing anomaly ratios in target unlabeled support set.

networks. Proto is a representative method of metric-based meta-learning methods. This method first embeds each unlabeled support instance by neural networks and calculates a prototypical vector by averaging embedded support data. The anomaly score of an instance is calculated on the basis of the distance between the embedded instance and the prototypical vector. We note that Proto can be regarded as a meta-learning extension of DeepSVDD (Ruff et al., 2018) since the prototypical vector is the support set dependent center vector of DeepSVDD. In the outer problem, MAML, NP, and Proto maximize the AUC as in the proposed method. TNP maximizes the sum of the AUC and negative reconstruction error of normal query data. RProto and MC are meta-learning methods from noisy labeled data. In the inner problem, RProto calculates the representative vector while removing anomalous embedded support instances that are far from the other embedded data. MC uses a weighted prototype vector in Proto to remove outliers, where the weight is outputted by a neural network from an input instance. We condition the neural network for weights by task representation vectors used in the proposed method to deal with the task's diversity. In the outer problem, RProto and MC maximize the AUC. We evaluated the comparison methods with the cross-entropy loss in the outer problem in Section F.3.

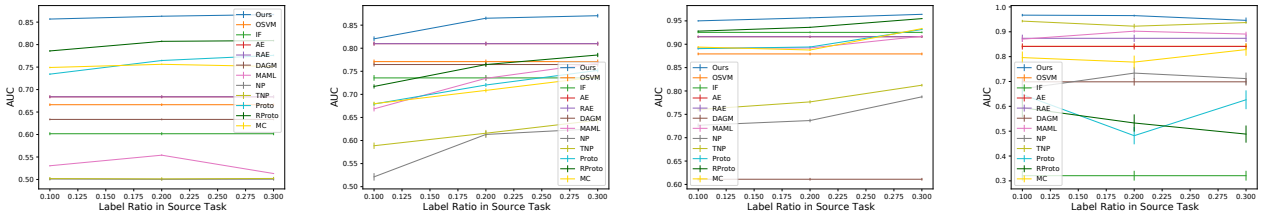
4.3 Hyperparameter Settings

For OSVM, IF, AE, RAE, and DAGM, we reported the best test AUCs by changing their hyperparameters. For other meta-learning methods including the proposed method, we reported test AUCs when hyperparameters were used that were determined on the basis of mean AUC on validation tasks. For all neural network-based methods, we used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-3}. The validation AUC was also used for early stopping to avoid over-fitting. Query set size $N_{\mathcal{Q}}$ was set to the number of labeled data on each task. The neural network architectures and hyperparameter candidates in our experiments are described in Sections D and E.

4.4 Results

Table 1 shows the average test AUCs over different label ratios in source tasks and anomaly ratios in the target unlabeled support set. The proposed method outperformed the other methods with all datasets by a large margin. Unsupervised anomaly detection methods (OSVM, IF, AE, RAE, and DAGM) performed worse than the proposed method, which indicates the effectiveness of using information in related tasks. Among the meta-learning methods, NP and TNP did not work particularly well with most datasets. This is because both methods are encoder-decoder-based methods and thus have difficulty performing effective adaptation by only neural networks. MAML outperformed NP and TNP since it adapts to the support set by target-specific training. RProto outperformed Proto since it is an outlier-robust method. The proposed method clearly outperformed these all meta-learning methods by properly incorporating the robust anomaly detection mechanism in the meta-learning framework.

Figure 2 shows the average and standard errors of test AUCs when changing anomaly ratios in the target unlabeled support set. As the anomaly ratio increases, the per-



(a) Omniglot (b) Mnist-r (c) Isolet (d) IoT

Figure 3: Average and standard errors of test AUCs over different anomaly ratios in target unlabeled support set when changing label ratios in source tasks.

Table 2: Ablation study: average test AUCs [%] over different label ratios in source tasks and anomaly ratios in target unlabeled support set. Boldface denotes the best and comparable methods according to the paired t-test and the significance level of 5 %.

Data	Ours	w/o A	w/o Initial	w/o Iter	w/o hTask	w/o ATask	w/o Task
Omniglot	**86.23**	82.36	83.73	84.16	**86.13**	85.77	**86.02**
Mnist-r	**85.19**	80.97	81.56	82.57	81.73	**85.26**	81.19
Isolet	**95.69**	92.83	94.03	93.84	**95.61**	**95.61**	95.34
IoT	**95.94**	**95.32**	**95.32**	80.11	**95.22**	93.68	**96.28**

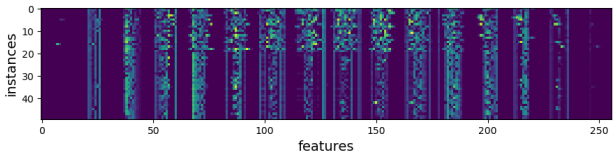


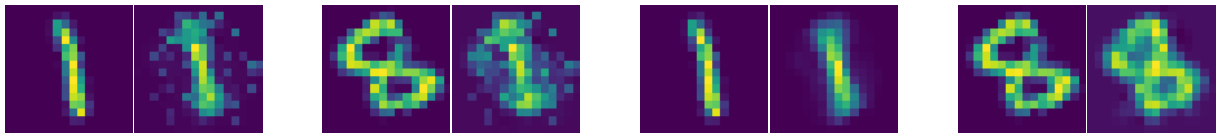
Figure 4: An example of estimated anomaly attributes of the target unlabeled support set $\mathbf{A}$ when anomaly rate in the support set is 0.4 in Mnist-r. We plotted the absolute values of the attributes of $\mathbf{A}$. The lighter color indicates higher absolute values (the dark purple represents zeros). The first 20 instances are anomalous, and the other 30 instances are normal. The proposed method captured anomalous attributes in the first 20 instances as expected.

formance of all methods tended to decrease. This is because the normal pattern is difficult to learn from unlabeled data when the anomaly ratio is large. However, the proposed method outperformed the other methods over almost all anomaly ratios with all datasets. By properly incorporating a mechanism to eliminate anomalies in the adaptation, our method worked well even with large anomaly ratios.

Figure 3 shows the average and standard errors of test AUCs when changing label ratios in source tasks. The performance of all meta-learning methods tended to increase as the label ratio increases because they can precisely evaluate the models adapted in the inner problems when there are many labels. The proposed method outperformed the other methods over all label ratios with all datasets.

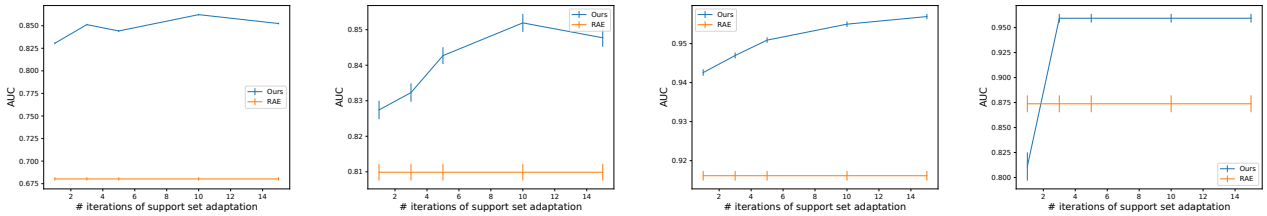
Table 2 shows an ablation study of our model. We evaluated six variants of our model: w/o A, w/o Initial, w/o Iter, w/o hTask, w/o ATask, and w/o Task. w/o A is our model without anomalous attributes $\mathbf{A}$ in Eq. (3). w/o Initial is our model without initial anomalous attributes $\mathbf{A}_0$ estimated by neural networks in Eq. (6). This model set $\mathbf{A}_0 = \mathbf{0}$ as in the previous studies (Zhou and Paffenroth, 2017). w/o Iter is our model without iterative adaptation with Eqs. (4) and (5). This model directly uses $\mathbf{A}_0$ in Eq. (6) as estimated anomalous attributes $\mathbf{A}$ and then $\mathbf{W}$ is determined by Eq. (4). w/o hTask is our model without task representation $\mathbf{z}$ in the encoder neural network h of Eq. (3). w/o ATask is our model without $\mathbf{z}$ in initial anomalous attributes $\mathbf{A}_0$ in Eq. (6). w/o Task is our model without $\mathbf{z}$ in both h and $\mathbf{A}_0$. The proposed method showed the best or comparable results with all datasets. Especially, it clearly outperformed w/o A with most datasets, which indicates the effectiveness of estimating $\mathbf{A}$ to remove harmful effects of the anomalies. w/o Initial and w/o Iter did not work well with most datasets. This result indicates that both estimating $\mathbf{A}_0$ and explicit support set adaptation are essential in our framework. w/o hTask, w/o ATask, and w/o Task outperformed the other variants. However, by using task representation $\mathbf{z}$ in both h and $\mathbf{A}_0$, the proposed method tended to outperform these three methods. Overall, these results show the effectiveness of our model design.

Figure 4 shows one example of estimated anomalous attributes in the target unlabeled support set by the proposed method. We confirmed that the anomalous instances have more non-zero anomalous attributes than the normal ones do. Since the proposed method precisely captured the anomalous attributes in the unlabeled data as expected, it



(a) Ours (normal) (b) Ours (anomaly) (c) w/o A (normal) (d) w/o A (anomaly)

Figure 5: Examples of the reconstructed instances in target unlabeled support set in Mnist-r. In each paired image, left and right images represent the original and reconstructed images, respectively. w/o A reconstructed not only the normal image (digit '1') but also the anomalous one (digit '8'). The proposed method correctly reconstructed the normal image (digit '1') but fails to reconstruct the anomalous data (digit '8') as expected.



(a) Omniglot (b) Mnist-r (c) Isolet (d) IoT

Figure 6: Average and standard errors of test AUCs over different anomaly ratios in target unlabeled support set and label ratios in source tasks when changing the number of iterations I for solving the inner problem of the proposed method.

Table 3: Training and testing computational time in seconds of meta-learning methods on Mnist-r.

	Ours	NP	TNP	MAML	Proto	RProto	MC
Train	222.4	49.0	49.4	348.7	42.9	47.0	61.2
Test	0.059	0.029	0.028	0.116	0.023	0.022	0.027

worked well by removing the harmful effects of the anomalies. The ability to identify anomalous attributes in the support set is an important practical property as the interpretability.

Figure 5 shows examples of the reconstructed normal and anomalous instances in the target unlabeled support set. Both the proposed method and w/o A, which is our model without anomalous attributes $\mathbf{A}$, precisely reconstructed the normal instance (digit '1'). However, w/o A also reconstructed the anomalous instance (digit '8') since it minimizes the reconstruction errors of all unlabeled instances. In contrast, the proposed method did not reconstruct the anomaly instance and thus worked well.

Figure 6 shows average and standard errors of test AUCs when changing the number of iterations I for solving the inner problem of the proposed method. For all datasets, the performance of the proposed method tended to improve as the number of iterations I increased. This is because large I can accurately solve our inner problems. However, the proposed method performed well even with few iterations ($I = 10$) with all datasets. This is because the proposed method was meta-learned such that few iterations lead to good solutions for anomaly detection.

Table 3 shows the training and testing time in seconds of meta-learning methods on Mnist-r. We used a computer with a 2.20 GHz CPU. The computation time of MAML was much longer than those of the other methods because it requires iterative optimization for all neural network parameters in the inner problem. Although the proposed method had a much longer computation time than NP, TNP, Proto, RProto, and MC, which do not require iterative adaptation, it outperformed the other methods in terms of anomaly detection by a large margin as described in Table 1.

5 CONCLUSION

We proposed a meta-learning method for robust anomaly detection that can learn accurate anomaly detectors from unlabeled data in unseen target tasks while removing the harmful effects of anomalies in the unlabeled data. Our experiments demonstrated that the proposed method outperformed various existing anomaly detection methods with various anomaly-noise ratios in four real-world datasets. For future work, we plan to extend our framework to robust anomaly detection for structured data such as time-series and graphs.

References

Akcay, S., Atapour-Abarghouei, A., and Breckon, T. P. (2018). Ganomaly: semi-supervised anomaly detection via adversarial training. In *ACCV*.

An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18.

Aryal, S., Ting, K. M., Wells, J. R., and Washio, T. (2014). Improving iforest with relative mass. In *PAKDD*.

Bach, F., Jenatton, R., Mairal, J., Obozinski, G., et al. (2011). Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 5:19–53.

Beggel, L., Pfeiffer, M., and Bischl, B. (2019). Robust anomaly detection in images using adversarial autoencoders.

Bekkar, M., Djemaa, H. K., and Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced data sets. *J Inf Eng Appl*, 3(10).

Bergman, L. and Hoshen, Y. (2019). Classification-based anomaly detection for general data. In *ICLR*.

Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., and Steger, C. (2021). The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059.

Bertinetto, L., Henriques, J. F., Torr, P. H., and Vedaldi, A. (2018). Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*.

Chalapathy, R., Menon, A. K., and Chawla, S. (2017). Robust, deep and inductive anomaly detection. In *ECML PKDD*.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: a survey. *ACM computing surveys (CSUR)*, 41(3):15.

Chen, D., Wu, L., Tang, S., Xu, F., Li, J., Zong, C., Tan, C., and Zhuang, Y. (2020). Robust meta-learning with noise via eigen-reptile.

Chen, J., Sathe, S., Aggarwal, C., and Turaga, D. (2017). Outlier detection with autoencoder ensembles. In *ICDM*.

Dahia, G. and Pamplona Segundo, M. (2021). Meta learning for few-shot one-class classification. *AI*, 2(2):195–208.

Ding, K., Zhou, Q., Tong, H., and Liu, H. (2021). Few-shot network anomaly detection via cross-network meta-learning. In *TheWebConf*.

Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., and Tan, P.-N. (2002). Data mining for network intrusion detection. In *Proc. NSF Workshop on Next Generation Data Mining*, pages 21–30.

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

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

Fan, C., Zhang, F., Liu, P., Sun, X., Li, H., Xiao, T., Zhao, W., and Tang, X. (2021). Importance weighted adversarial discriminative transfer for anomaly detection. *arXiv preprint arXiv:2105.06649*.

Fanty, M. and Cole, R. (1990). Spoken letter recognition. In *NeurIPS*.

Fernando, T., Gammulle, H., Denman, S., Sridharan, S., and Fookes, C. (2020). Deep learning for medical anomaly detection–a survey. *arXiv preprint arXiv:2012.02364*.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Frikha, A., Krompaß, D., Köpken, H.-G., and Tresp, V. (2021). Few-shot one-class classification via meta-learning. In *AAAI*.

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., and Eslami, S. A. (2018a). Conditional neural processes. In *ICML*.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S., and Teh, Y. W. (2018b). Neural processes. *arXiv preprint arXiv:1807.01622*.

Ghifary, M., Bastiaan Kleijn, W., Zhang, M., and Balduzzi, D. (2015). Domain generalization for object recognition with multi-task autoencoders. In *ICCV*.

Golan, I. and El-Yaniv, R. (2018). Deep anomaly detection using geometric transformations. In *NeuriPS*.

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., and Hengel, A. v. d. (2019). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *ICCV*.

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2020). Meta-learning in neural networks: a survey. *arXiv preprint arXiv:2004.05439*.

Huang, C., Guan, H., Jiang, A., Zhang, Y., Spratling, M., and Wang, Y.-F. (2022). Registration based few-shot anomaly detection.

Huang, S.-Y., Yeh, Y.-R., and Eguchi, S. (2009). Robust kernel principal component analysis. *Neural computation*, 21(11):3179–3213.

Idé, T., Phan, D. T., and Kalagnanam, J. (2017). Multi-task multi-modal models for collective anomaly detection. In *ICDM*.

Iwata, T. and Kumagai, A. (2021). Meta-learning one-class classifiers with eigenvalue solvers for supervised anomaly detection. *arXiv preprint arXiv:2103.00684*.

Iwata, T. and Yamanaka, Y. (2019). Supervised anomaly detection based on deep autoregressive density estimators. *arXiv preprint arXiv:1904.06034*.

Killamsetty, K. and Li, C. (2022). A nested bi-level optimization framework for robust few shot learning. In *AAAI*.

Kingma, D. P. and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kou, Y., Lu, C.-T., Sirwongwattana, S., and Huang, Y.-P. (2004). Survey of fraud detection techniques. In *ICNSC*.

Kruspe, A. (2019). One-way prototypical networks. *arXiv preprint arXiv:1906.00820*.

Kumagai, A., Iwata, T., and Fujiwara, Y. (2019). Transfer anomaly detection by inferring latent domain representations. In *NeurIPS*.

Kumagai, A., Iwata, T., and Fujiwara, Y. (2021). Meta-learning for relative density-ratio estimation. In *NeurIPS*.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. (2019a). Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. (2019b). Meta-learning with differentiable convex optimization. In *CVPR*.

Liang, K. J., Rangrej, S. B., Petrovic, V., and Hassner, T. (2022). Few-shot learning with noisy labels. In *CVPR*.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *ICDM*.

Lu, Y., Yu, F., Reddy, M. K. K., and Wang, Y. (2020). Few-shot scene-adaptive anomaly detection. In *ECCV*.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., and Elovici, Y. (2018). N-baiotnetwork-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22.

Michau, G. and Fink, O. (2021). Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer. *Knowledge-Based Systems*, 216:106816.

Oladosu, A., Xu, T., Ekfeldt, P., Kelly, B. A., Cranmer, M., Ho, S., Price-Whelan, A. M., and Contardo, G. (2020). Meta-learning one-class classification with deepsets: application in the milky way. *arXiv e-prints*, pages arXiv–2007.

Pang, G., Shen, C., and van den Hengel, A. (2019). Deep anomaly detection with deviation networks. In *KDD*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Perera, P., Nallapati, R., and Xiang, B. (2019). Ocgan: one-class novelty detection using gans with constrained latent representations. In *CVPR*.

Phan, D. T. and Idé, T. (2019). l0-regularized sparsity for probabilistic mixture models. In *SDM*.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. (2019). Meta-learning with implicit gradients. In *NeurIPS*.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., and Lakshminarayanan, B. (2019). Likelihood ratios for out-of-distribution detection. In *NeurIPS*.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. (2018). Learning to reweight examples for robust deep learning. In *ICML*.

Rousseeuw, P. J. and Hubert, M. (2011). Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(1):73–79.

Ruff, L., Görnitz, N., Deecke, L., Siddiqui, S. A., Vandermeulen, R., Binder, A., Müller, E., and Kloft, M. (2018). Deep one-class classification. In *ICML*.

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

Ruff, L., Vandermeulen, R. A., Görnitz, N., Binder, A., Müller, E., Müller, K.-R., and Kloft, M. (2019). Deep semi-supervised anomaly detection. In *ICLR*.

Saerens, M., Latinne, P., and Decaestecker, C. (2002). Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41.

Sakurada, M. and Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, page 4. ACM.

Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M. H., and Sabokrou, M. (2021). A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: solutions and future challenges. *arXiv preprint arXiv:2110.14051*.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support

of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. (2019). Meta-weight-net: kearning an explicit mapping for sample weighting. *NeurIPS*.

Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical networks for few-shot learning. In *NeurIPS*.

Somepalli, G., Wu, Y., Balaji, Y., Vinzamuri, B., and Feizi, S. (2021). Unsupervised anomaly detection with adversarial mirrored autoencoders. In *UAI*.

Vincent, V., Wannes, M., and Jesse, D. (2020). Transfer learning for anomaly detection through localized and unsupervised instance selection. In *AAAI*.

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

Wang, Z., Hu, G., and Hu, Q. (2020). Training noise-robust deep neural networks via meta-learning. In *CVPR*.

Wu, J.-C., Chen, D.-J., Fuh, C.-S., and Liu, T.-L. (2021). Learning unsupervised metaformer for anomaly detection. In *ICCV*.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xiong, L., Chen, X., and Schneider, J. (2011). Direct robust matrix factorizatoin for anomaly detection. In *ICDM*.

Yamanaka, Y., Iwata, T., Takahashi, H., Yamada, M., and Kanai, S. (2019). Autoencoding binary classifiers for supervised anomaly detection. In *PRICAI*.

Yao, H., Zhang, L., and Finn, C. (2021). Meta-learning with fewer tasks through task interpolation. *arXiv preprint arXiv:2106.02695*.

Yoon, S., Noh, Y.-K., and Park, F. (2021). Autoencoding under normalization constraints. In *ICML*.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. In *NeurIPS*.

Zhang, H., Cao, L., VanNostrand, P., Madden, S., and Rundensteiner, E. A. (2021). Elite: Robust deep anomaly detection with meta gradient. In *KDD*.

Zhang, W., Wang, Y., and Qiao, Y. (2019). Metacleaner: learning to hallucinate clean representations for noisy-labeled visual recognition. In *CVPR*.

Zhao, Y., Rossi, R., and Akoglu, L. (2021). Automatic unsupervised outlier model selection. In *NeurIPS*.

Zheng, G., Awadallah, A. H., and Dumais, S. (2021). Meta label correction for noisy label learning. In *AAAI*.

Zhou, C. and Paffenroth, R. C. (2017). Anomaly detection with robust deep autoencoders. In *KDD*.

Zhu, J., Yi, X., Guan, N., and Cheng, H. (2020). Robust re-weighting prototypical networks for few-shot classification. In *ICRAI*.

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 *ICLR*.

A OBJECTIVE FUNCTION IN OUTER OPTIMIZATION PROBLEMS WITHOUT ANOMALOUS DATA

The proposed method can treat source tasks that do not contain labeled anomalous data. For these source tasks, we maximize the negative reconstruction error on labeled normal data in the outer problem instead of the AUC in Eq. (9):

$$-\frac{1}{N_{\mathrm{N}}}\sum_{m=1}^{N_{\mathrm{N}}} \parallel \mathbf{x}_m^{\mathrm{N}} - \mathbf{W}_* h([\mathbf{x}_m^{\mathrm{N}}, \mathbf{z}]) \parallel_2^2 .$$

By maximizing the negative reconstruction error of normal data, the proposed method can learn the normal pattern of data, which is useful for anomaly detection.

B OUR FORMULATION WITH SVDD

When we use DeppSVDD, an anomaly score of instance $\mathbf{x}$ given support set $\mathcal{S}$, $s(\mathbf{x};\mathcal{S})$, is calculated by

$$s(\mathbf{x};\mathcal{S}) := \parallel \mathbf{c} - \mathbf{W}h([\mathbf{x}_n, \mathbf{z}]) \parallel_2^2,$$

where $\mathbf{c} \in \mathbb{R}^C$ is the predefined fixed center vector, $h : \mathbb{R}^{D+K} \to \mathbb{R}^J$ is a feed-forward neural network, and $\mathbf{W} \in \mathbb{R}^{C\times J}$ is a linear weight matrix. The loss function in the inner optimization problem Eq. (3) becomes

$$\mathcal{L}(\mathbf{W}, \mathbf{A}; \mathcal{S}) := \frac{1}{N_{\mathcal{S}}}\sum_{n=1}^{N_{\mathcal{S}}} \parallel \mathbf{c} + \mathbf{a}_n - \mathbf{W}h([\mathbf{x}_n, \mathbf{z}]) \parallel_2^2 + \frac{\lambda}{N_{\mathcal{S}}}\sum_{n=1}^{N_{\mathcal{S}}} \parallel \mathbf{a}_n \parallel_1 + \mu \parallel \mathbf{W} \parallel_{\mathrm{F}}^2 .$$

When $\lambda = \infty$, this loss function is equivalent to that of DeepSVDD since $\mathbf{A} = \mathbf{0}$. By minimizing this loss, the proposed method learns $\mathbf{W}$ while avoiding anomalies approaching center vector $\mathbf{c}$.

C DOWNLOAD LINKS OF REAL-WORLD DATASETS

We used four real-world datasets for our experiments: Omniglot[2], Mnist-r[3], Isolet[4], and IoT[5].

D NETWORK ARCHITECTURES

For the proposed method, a three(two)-layered feed-forward neural network was used for $f(g)$ in Eq. (1). For f, the number of hidden and output nodes was 32. For h in Eq. (2), a three-layered feed-forward neural network with 32 hidden and 32 output nodes was used ($J = 32$). For v in Eq. (6), a four-layered feed-forward neural network with 32 hidden nodes was used. For AE, RAE, DAGM, NP, and TNP, a four-layered feed-forward neural network with 32 hidden nodes was used for AE networks. For estimation networks in DAGM, a three-layered feed-forward neural network with 32 hidden nodes was used. For MAML, a four-layered feed-forward neural network with 32 hidden nodes was used for binary classifiers. For Proto, RProto, and MC, a four-layered feed-forward neural network with 32 hidden was used for instance embedding networks. For the task representation vectors in NP, TNP, and MC, the same neural network architectures as the proposed method (f and g) were used. For weight networks in MC, a four-layered feed-forward neural network with 32 hidden nodes was used. All neural networks used the ReLU function as their activation functions. We implemented all neural network-based methods on the basis of PyTorch (Paszke et al., 2017). All experiments were conducted on a Linux server with an Intel Xeon CPU and a NVIDIA GeForce GTX 1080 GPU.

E HYPERPARAMETERS

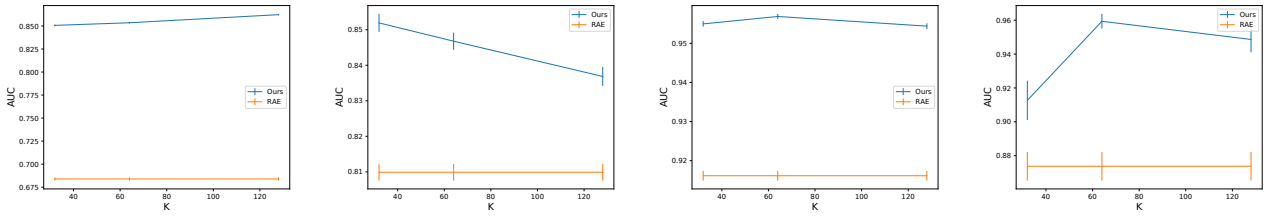
For OSVM, the RBF kernel was used and outlier fraction parameter ν was selected from $\{0.1, 0.3, 0.5\}$. For IF, the number of base estimators was chosen from $\{50, 100, 200\}$. For AE, RAE, and DAGM, the number of training iterations

[2]https://github.com/shashankhalo7/Omniglot_meta_learning

[3]https://github.com/ghif/mtae

[4]http://archive.ics.uci.edu/ml/datasets/ISOLET

[5]https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT



(a) Omniglot (b) Mnist-r (c) Isolet (d) IoT

Figure 7: Average and standard errors of test AUCs over different anomaly ratios in target unlabeled support set and label ratios in source tasks when changing the dimension of task representations K of the proposed method.

Table 4: Detailed analysis of effects of modeling initial anomalous attributes: average test AUCs [%] and training computational time in seconds on Mnist-r.

Metric	Ours ($I = 10$)	w/o Initial ($I = 10$)	w/o Initial ($I = 20$)	w/o Initial ($I = 30$)	w/o Initial ($I = 50$)
Train Time	222.4	190.7	317.9	467.9	808.9
Test AUC	85.19	81.56	81.96	81.90	81.78

was selected from $\{100, 500, 1000\}$. For RAE, ℓ_1-regularization parameter λ was chosen from $\{10^{-2}, 10^{-1}, \ldots, 10^{2}\}$. For DAGM, the number of mixture components was chosen from $\{2, 4\}$ and regularization parameters were set to the recommended values in the original paper: $\lambda_1 = 0.1$ and $\lambda_2 = 0.005$. For these methods, we reported the best test AUCs. For all methods except for OSVM, IF, AE, RAE, and DAGM, hyperparameters were determined on the basis of mean AUC on validation tasks. For the proposed method, NP, TNP, and MC, the dimension of task representation was selected from $\{32, 64, 128\}$. For the proposed method, the initial value for ℓ_1- regularization trainable parameter λ was selected from $\{10^{-1}, 1, 10\}$. We set the initial value of ℓ_2- regularization trainable parameter μ to 10^{-1}. The iteration number in the inner problem I was set to 10. For TNP, regularization parameter λ was selected from $\{1, 10, 10^2, 10^3\}$. For MAML, the step size and the iteration number in the inner problem were chosen from $\{10^{-1}, 10^{-2}\}$ and $\{5, 10\}$, respectively. For Proto, RProto, and MC, the dimension of output nodes was chosen from $\{32, 64, 128\}$. For all neural network-based methods, we used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-3}. The validation AUC was also used for early stopping to avoid over-fitting. The maximum number of training iterations was set to $10,000$ except for MAML. For MAML, it was chosen from $\{10,000, 30,000\}$ to improve the performance. Query set size $N_{\mathcal{Q}}$ was set to the number of labeled data on each task.

F ADDITIONAL EXPERIMENTAL RESULTS

F.1 Dependency of Dimension of Task Representations

Figure 7 shows average and standard errors of test AUCs when changing the dimension of task representation vectors K. The proposed method performed better than RAE over all K with all datasets. Therefore, the proposed method is relatively robust against K values.

F.2 Detailed Analysis of Modeling Initial Anomalous Attributes $\mathbf{A}_0$

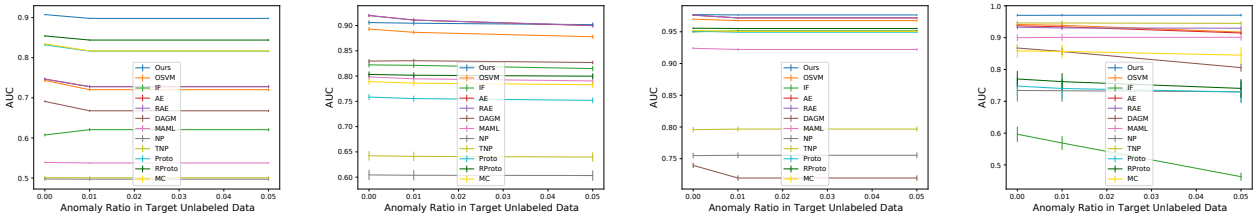
We showed the effectiveness of modeling initial anomalous attributes with neural networks in the ablation study (Table 2). Here, we further analyzed its effect in detail. Specifically, we investigated whether w/o Initial, which is our method with fixed initial anomalous attributes $\mathbf{A}_0 = \mathbf{0}$, can perform well when the iterations for solving the inner problem I are increased. Table 4 shows the mean test AUCs and the training time of the proposed method and w/o Initial. As the iteration number I was increased, the training time of w/o Initial significantly increased. However, the performance of w/o Initial did not improve much. This would be because many iterations expand the computation graph of neural networks, and this makes the optimization difficult. In contrast, the proposed method performed well even if the number of iterations I was small ($I = 10$) by modeling the initial anomalous attributes with neural networks. This result supports the validity of modeling initial anomalous attributes.

Table 5: Comparison with meta-learning methods using the cross-entropy loss: average test AUCs [%] over different label ratios in source tasks within $\{0.1, 0.2, 0.3\}$ and anomaly ratios in target unlabeled support set within $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. The first letter 'C' stands for the cross-entropy loss. Boldface denotes the best and comparable methods according to the paired t-test and the significance level of 5 %.

Data	Ours	C-NP	C-MAML	C-Proto	C-RProto	C-MC
Omniglot	**86.23**	49.80	52.00	80.46	85.19	80.02
Mnist-r	**85.19**	50.29	71.48	67.61	73.85	69.32
Isolet	**95.69**	50.20	89.95	88.49	94.00	91.73
IoT	**95.94**	90.78	92.09	77.44	81.35	84.06

Table 6: Average test AUCs [%] with the anomaly ratio in labeled source data is 0.01 over different label ratios in source tasks within $\{0.1, 0.2, 0.3\}$ and anomaly ratios in target unlabeled support set within $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Boldface denotes the best and comparable methods according to the paired t-test and the significance level of 5 %.

Data	Ours	OSVM	IF	AE	RAE	DAGM	NP	TNP	MAML	Proto	RProto	MC
Omniglot	**86.23**	66.63	60.17	68.39	68.39	63.36	50.07	50.18	53.26	75.83	80.07	75.22
Mnist-r	**82.77**	77.10	73.57	80.99	80.99	76,47	52.92	58.03	66.45	68.75	72.08	68.41
Isolet	**95.58**	87.93	92.54	91.61	91.63	61.11	72.03	78.37	89.28	89.25	93.47	89.63
IoT	**93.59**	84.16	32.09	84.09	87.38	69.87	60.46	90.65	84.04	59.35	51.55	85.97



(a) Omniglot (b) Mnist-r (c) Isolet (d) IoT

Figure 8: Average and standard errors of test AUCs over different label ratios in source tasks when changing anomaly ratios in target unlabeled support set within $\{0.0, 0.01, 0.05\}$.

F.3 Comparison Methods with Cross-entropy Loss

In the main paper, the comparison methods use the AUC in the outer problem as in the proposed method for fair comparisons. We additionally evaluate the comparison methods (NP, MAML, Proto, RProto, and MC) with the cross-entropy loss. To use the cross-entropy loss, we transformed the anomaly score of each method $s(\mathbf{x}) \in \mathbb{R}_{\geq 0}$ by function $s_{\mathrm{new}}(\mathbf{x}) = \frac{\epsilon + s(\mathbf{x})}{1 + s(\mathbf{x})} \in [\epsilon, 1)$, where ϵ is 10^{-8}. Table 5 shows the average test AUCs for each dataset. The proposed method clearly outperformed these methods with all datasets.

F.4 Results with Small Anomaly Ratios in Labeled Source Data

We investigated the performance of the proposed method when the anomaly ratio in labeled source data is small (0.01) since many anomalous labeled data might be difficult to collect in real-world applications. Table 6 shows the results. Although all methods tended to perform worse than the methods with the anomaly ratio of 0.1 in the main paper, the proposed method clearly outperformed the other methods. These results show that the proposed method works robustly even when labeled data contain few anomalies.

F.5 Results with Small Anomaly Ratios in Target Unlabeled Support Sets

We investigated the performance of the proposed method when the anomaly ratio in target unlabeled support sets is small since unlabeled data might contain few anomalies in real-world applications. Figures 8 show the results when changing anomaly ratios in target unlabeled support sets within $\{0.0, 0.01, 0.05\}$. The proposed method outperformed the other

Table 7: Average test AUCs [%] with different numbers of source tasks T on Omniglot.

	$T = 10$	$T = 30$	$T = 100$	$T = 250$	$T = 500$	$T = 764$
Ours	72.30	72.06	77.74	83.13	85.29	86.23
RAE	68.39	68.39	68.39	68.39	68.39	68.39

Table 8: Average test AUCs [%] when 10% of source tasks have labeled anomalous data. Boldface denotes the best and comparable methods according to the paired t-test and the significance level of 5 %.

Data	Ours	OSVM	IF	AE	RAE	DAGM	NP	TNP	Proto	RProto	MC
Omniglot	**72.66**	66.63	60.17	68.39	68.39	63.36	50.06	50.19	67.74	69.91	68.37
Mnist-r	**81.04**	77.10	73.57	**80.99**	**80.99**	76.47	51.11	52.16	60.31	61.80	62.54
Isolet	**92.22**	87.93	**92.54**	91.61	91.63	61.11	50.22	64.73	77.33	82.53	77.56
IoT	**91.38**	84.16	32.09	84.09	87.38	69.87	90.86	88.88	77.74	86.94	87.82

methods with all datasets except for Mnist-r. Especially, the proposed method worked well even when the unlabeled data did not contain anomalies (the anomaly ratio was 0.0). For Mnist-r, although AE and RAE worked well due to the small anomaly ratios, the proposed method still performed comparably.

F.6 Results with Different Numbers of Source Tasks

The meta-learning methods generally require many source tasks to learn how to learn. Therefore, it is interesting to evaluate the proposed method's performance with different numbers of source tasks. Table 7 shows the average test AUCs with different numbers of source tasks on Omniglot. Here, we used the Omniglot since it has many tasks. As the number of source tasks increased, the performance of the proposed method tended to increase. The results suggest that it is essential for the proposed method to be meta-learned on various source tasks. However, even when the number of source tasks was small (e.g., $T = 10$), the proposed method outperformed RAE, which only learns with target unlabeled data. This result shows that the proposed method is effective even when many source tasks cannot be prepared, which might often be in practice. Note that this claim is also supported by results with IoT, which has a small number of source tasks ($T = 6$), in Table 1.

F.7 Results without Anomalous Data in Some Source Tasks

In the main paper, we evaluated the proposed method when all source tasks have labeled anomalous data. However, in practice, it may be difficult to prepare such source tasks since anomalous data are rare. Therefore, we investigated the performance of the proposed method when there are two types of source tasks: tasks with and without labeled anomalous data. Table 8 shows the average test AUCs when 10% of source tasks have labeled anomalous data (90% of source tasks have only labeled normal data and unlabeled data) in each dataset. For source tasks without labeled anomalous data, AE-based meta-learning methods (the proposed method, NP, and TNP) used the reconstruction error on normal data as the objective function of outer problems as described Section A. Other meta-learning methods (Proto, RProto, and MC) used the distance between embedded normal data and the prototype vector as the objective function as in DeepSVDD (Ruff et al., 2018). We did not evaluate MAML because it is not trivial to properly handle only normal data in outer problems. The proposed method outperformed the other methods. These results suggest that the proposed method is effective even when most of the source tasks have no anomalous data.

G LIMITATIONS

The proposed method uses multiple source tasks to improve anomaly detection performance on unseen target tasks. However, when source and target tasks are significantly different, the performance on the target tasks risks degrading. This is known as "negative transfer," and overcoming it is one of the important problems in transfer/meta-learning studies. Developing methods to automatically remove negative effects of such tasks is a promising research direction.

The proposed method also assumes that the feature space is the same across all tasks, which may hinder its practicality in some applications. However, this assumption is not unique to the proposed method but is common to almost all meta-learning methods (for anomaly detection) (Snell et al., 2017; Finn et al., 2017; Bertinetto et al., 2018; Rajeswaran et al.,

2019; Kumagai et al., 2021; Frikha et al., 2021; Kruspe, 2019; Kumagai et al., 2019). Even if the feature space is the same, there are many practical applications. For example, as mentioned in Section 1, imagine building user-specific anomaly detectors based on user-behavior logs in a service that has multiple users. Within the same service, each users log is usually written in the same format (features). For anomaly product detection from images in factories (visual inspection), images from each factory can be rescaled identically. Developing meta-learning methods for anomaly detection that can handle different feature spaces is also a promising research direction.

The proposed method can improve anomaly detection performance as the number of source tasks increases. However, when the number of source tasks is small, the performance improvement might be not large. To deal with this problem, one possible solution is to use task-augmentation methods (Yao et al., 2021). Combining them with our problem would be one of the most critical research topics.

H NEGATIVE SOCIAL IMPACTS

The proposed method has some potential risks to be addressed when it is deployed to real-world applications. First, although we experimentally demonstrated that the proposed method outperformed existing methods, it is not perfect; that is, it may cause false positives or false negatives, which may lead to wrong decision making in some cases. To mitigate this, people can use the proposed method as a support tool for their detailed analysis. Second, the proposed method needs to access datasets obtained from multiple tasks. When each dataset is provided from different owners such as companies, sensitive information in the dataset risks being stolen and abused by malicious people that use the proposed method. To evade this risk, we suggest promoting research for developing transfer/meta-learning without accessing raw datasets.