Prototype-oriented unsupervised anomaly detection for multivariate time series

Yuxin Li [1] **Wenchao Chen** [1] **Bo Chen** [1] **Dongsheng Wang** [1] **Long Tian** [2] **Mingyuan Zhou** [3]

Abstract

Unsupervised anomaly detection (UAD) of multivariate time series (MTS) aims to learn robust representations of normal multivariate temporal patterns. Existing UAD methods try to learn a fixed set of mappings for each MTS, entailing expensive computation and limited model adaptation. To address this pivotal issue, we propose a prototype-oriented UAD (PUAD) method under a probabilistic framework. Specifically, instead of learning the mappings for each MTS, the proposed PUAD views multiple MTSs as the distribution over a group of prototypes, which are extracted to represent a diverse set of normal patterns. To learn and regulate the prototypes, PUAD introduces a reconstruction-based unsupervised anomaly detection approach, which incorporates a prototype-oriented optimal transport method into a Transformer-powered probabilistic dynamical generative framework. Leveraging meta-learned transferable prototypes, PUAD can achieve high model adaptation capacity for new MTSs. Experiments on five public MTS datasets all verify the effectiveness of the proposed UAD method.

1. Introduction

Modern information technology operations generate an enormous amount of high-dimensional sensor data that need to be continuously monitored. Two typical examples are content delivery network (CDN) systems (Dai et al., 2021) and server machines in large data centers (Su et al., 2021; 2019; Sun et al., 2021). Given the monitoring data of large-scale systems represented as multivariate time series (MTS), one way to discover system malfunctions is to detect the abnormal time points within the MTS, which is quite meaningful for ensuring security and service quality and mitigating financial losses (Xu et al., 2021). The goal of anomaly detection is to identify abnormal input data that does not conform to the description of usual data (Cao et al., 2022). Machine learning-based anomaly or outlier detection methods can be broadly categorized into either supervised anomaly detection (AD) (Liu et al., 2015; Shon & Moon, 2007; Yamada et al., 2013), or unsupervised AD (UAD) (Su et al., 2021; 2019; Zhang et al., 2019; Li et al., 2019; Xu et al., 2018a; Audibert et al., 2020; Malhotra et al., 2016; Hundman et al., 2018a). As anomalies are usually rare and buried within a vast amount of normal points, not only the labeling of anomalies is hard and expensive, but also the performance of a supervisedly-learned AD classifier is often sub-optimal due to severe class imbalance (Chalapathy & Chawla, 2019). For this reason, we focus on UAD for MTS which requires no labels for anomalies.

The basic idea of UAD for MTS is to detect anomalies by comparing an MTS against multivariate temporal patterns, which are extracted from previous MTSs that are deemed normal. Over the years, many reconstruction-based UAD methods have been developed. They first learn normal MTS patterns and then use the reconstruction errors under these patterns as anomaly scores. Several lines of work have been proposed, including those based on probabilistic dynamical models, such as GmVRNN (Dai et al., 2022) and VGCRN (Chen et al., 2022) that have achieved superior detection performance by considering the temporal dependence and variability within the MTS. As demonstrated in Transformer (Vaswani et al., 2017), the attention mechanisms have a remarkable ability to capture long-range dependencies. Originally developed for modeling discrete sequences, Transformer has also been modified to detect anomalies in MTS, where Anomaly Transformer (Xu et al., 2021) and TranAD (Tuli et al., 2022) are two representative examples.

A key challenge for UAD in an MTS coming from a large-scale system is that each device contained in it has its own distinct normal mode. For example, there is a clear difference between the distribution of the server used for video websites and the server used for shopping websites. It is thus challenging to capture the multiple diverse normal patterns via a fixed set of mappings. In order to better learn normal patterns, previous methods typically model one pattern of

[1]National Key Laboratory of Radar Signal Processing, Xidian University, Xi'an, 710071, China. [2]Software Engineering Institute, Xidian University, Xi'an, 710071, China. [3]McCombs School of Business, The University of Texas at Austin, Austin, TX 78712. Correspondence to: Wenchao Chen <wcchen_xidian@163.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

MTS with an individual set of parameters, which demands not only a massive number of parameters, when the number of MTSs is large, but also a large amount of data to train these parameters when adapting to new MTSs. GmVRNN (Dai et al., 2022) tries to model diverse MTSs with a single group of parameters by assigning latent states into the Gaussian mixture distribution. While achieving promising performance, it ignores the fact that a single group of parameters couldn't fit all the patterns and the model needs to be adjusted for different MTSs to achieve the best performance.

Addressing the limitations of previous works in capturing diverse temporal dependencies within multiple MTSs and adapting to new MTSs with few observations, we develop a prototype-oriented UAD (PUAD) method for MTS under a probabilistic framework, where optimal transport (OT) (Peyré et al., 2019) is leveraged to learn the prototypes. We show the use of prototypes in PUAD helps identify diverse normal patterns in MTSs and adapt to new MTSs given a few examples. Different from most previous UAD methods that model different MTSs with different group of parameters, PUAD considers the diverse normal dynamic patterns within multiple MTSs as a group of global prototypes and learns these prototype memories with the proposed novel prototype-oriented OT module, inspired by (Tanwisuth et al., 2021; Wang et al., 2022; Guo et al., 2022). Since each prototype in PUAD is encouraged to capture the statistical temporal dependency shared by multiple MTSs, which is similar to the transfer patterns useful for all related tasks in meta-learning (Guo et al., 2022; Vilalta & Drissi, 2002; Zhen et al., 2020; Du et al., 2021), thus to enable PUAD to achieve high model adaption capacity. Moreover, we also introduce the local prototypes for newly arrived MTSs to enhance the adaption capacity of PUAD. Finally, PUAD incorporates the prototype-oriented OT module into a powerful probabilistic dynamical generative framework for a reconstruction-based unsupervised anomaly detection approach for MTS.

The main contributions of our work are summarized as follows:

- We realize the learning of the diverse normal patterns within multiple MTSs by extracting a group of prototypes and propose a probabilistic framework named PUAD for the anomaly detection.
- We develop a prototype-oriented OT module that leverages the OT distance between the distributions to guide the learning of the prototypes.
- We define the global and local prototypes to enhance the capacity of PUAD in adapting the to new MTSs with the limited observations.
- We provide extensive experimental results and comparisons on five datasets to demonstrate that our method achieves the overall SOTA performance on both traditional and meta anomaly detection tasks.

2. Background

2.1. Prototype-based Methods

The motivation for the prototype-based models comes from the cluster assumption (Grandvalet & Bengio, 2004), which states that decision boundaries should not cross high-density regions of the data. These models attempt to learn a prototype representation for different classes of patterns. The prototype-based model has received a lot of attention in various fields, such as metric-based few-shot learning (Snell et al., 2017), unsupervised domain adaptation (Tanwisuth et al., 2021), or representation learning (Guo et al., 2021). Recently, prototype-based methods have also been introduced in computer vision anomaly detection. For example, prototype-guided discriminative latent embedding (Lai et al., 2021) is proposed for video anomaly detection (VAD), which tries to learn a deep autoencoder to describe normal event patterns with small reconstruction errors. Prototype selection-based method for industrial machine anomaly detection was introduced in Grandvalet et al. (de Paula Monteiro et al., 2022), where the model input is spectrograms. Moreover, Snell et al. (Liu et al., 2021) showed a dual prototypes autoencoder for industrial surface inspection anomaly detection. However, there is still no prototype-based methods for MTS modeling, although they have a unique advantage in unsupervised anomaly detection of MTS due to their ability to represent multiple patterns and adopt to new patterns.

2.2. Unsupervised Anomaly Detection

The procedures of unsupervised anomaly detection can be summarized as three steps. Firstly, the pre-processing of the original MTS data is needed so that they can be used by the learning model for training. Specifically, the normalization and sliding time window approaches (Dai et al., 2021) are adopted in this work. Secondly, anomaly detection model is trained unsupervisedly with the processed data. Then, anomaly scores for the testing data are obtained with the trained model and achieve detection by selecting threshold with a defined metric. We consider two settings of unsupervised anomaly detection, including: *"one-for-all"* (Dai et al., 2022): the models are trained and performed on all MTSs; *"one-for-one"* (Dai et al., 2021): the models are trained and performed on each MTS individually.

2.3. Optimal Transport

Optimal Transport (OT) is a wildly used tool for quantifying the difference between two distributions (Peyré et al., 2019). Specifically, considering two discrete distributions as $p = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $q = \sum_{j=1}^{m} b_j \delta_{y_j}$, where $x_i, y_j \in \mathbb{R}^d$ and δ_x is the Dirac function that places a unit point mass at x.

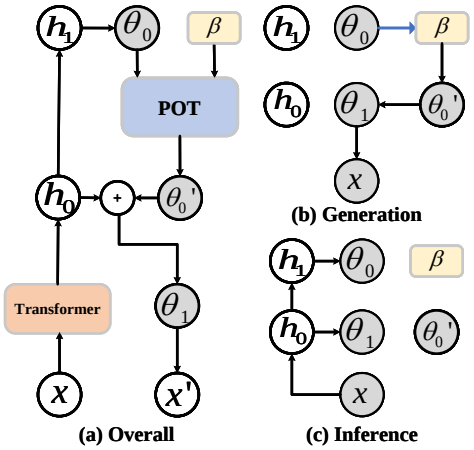

Figure 1. Graphical illustration of each operation of the PUAD: (a) overall operations of PUAD; (b) generation process of $\boldsymbol{x}$; (c) inference of the variational distribution of $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$. Note that gray circles indicate hidden variables, white circles indicate input data and features, and yellow squares indicate prototypes. Arrows represent the flow of data and the generation relationship.

The OT distance between p and q can be expressed as:

$$\mathrm{OT}(p,q) = \min_{\mathbf{T}\in\Pi(p,q)} \langle \mathbf{T}, \mathbf{C} \rangle \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius dot-product, $\mathbf{C} \in \mathbb{R}_{\geq 0}^{n\times m}$ is the transport cost matrix, and $C_{ij} = C(x_i, y_j)$. $\mathbf{T} \in \mathbb{R}_{>0}^{n\times m}$ refers to the doubly stochastic transport probability matrix that $\Pi(p,q) := \{\mathbf{T} \mid \sum_{i=1}^{n} T_{ij} = b_j, \sum_{j=1}^{m} T_{ij} = a_i\}$, which can be learned by minimizing $\mathrm{OT}(p,q)$. As the optimization of Eq. (1) often demands a high computational cost, the Sinkhorn algorithm for discrete OT, which is achieved by introducing the entropic regulation $H = -\sum_{ij} T_{ij} \ln T_{ij}$, is commonly used in practice to reduce the computation (Peyré et al., 2019).

2.4. Meta Anomaly Detection

Meta-learning, also known as learning to learn, is referred to the techniques that focus on helping deep models quickly adapt to new environments (Cao et al., 2022; Wu et al., 2021). In our paper, we focus on the meta anomaly detection for MTS, which is defined as the task that enables anomaly detection methods to quickly adapt to the new arrived MTSs, such as the new machines, new websites and so on, with limited observations, and we aim to design a model-based method being able to perform meta anomaly detection.

3. Methodology

In this section, we first define the anomaly detection problem solved in this paper. Then we present PUAD, which consists of a prototype-oriented optimal transport (POT) module to learn the prototypes for MTSs and improve it by an OT algorithm (Peyré et al., 2019), a deep probabilistic generative module that is guided by the prototypes which consider the diverse temporal dependencies within multiple MTSs for robust representations learning, and an inference module (Zhang et al., 2018; Duan et al., 2021) based on Transformer (Vaswani et al., 2017) to approximate the intractable posterior distribution in the generative module. Finally, we introduce the training process of PUAD.

3.1. Problem Definition

Defining an MTS as $\boldsymbol{x} = (x_1, x_2, ..., x_T) \in \mathbb{R}^{V\times T}$, where T is the duration of $\boldsymbol{x}$ and $x_t \in \mathbb{R}^V$ denotes the V-dimensional observation at time t. Anomaly detection on MTS is defined as a problem that determines whether an observation collected at a certain time x_t is anomalous or not. To solve this problem efficiently in an unsupervised manner, we need a powerful method for learning the robust representations of the input data.

3.2. Prototype-oriented Unsupervised Anomaly Detection

As shown in Fig. 1, PUAD constructs a transformer-powered probabilistic dynamical generative framework with a prototype-oriented optimal transport method. PUAD has a latent space that consists of three parts: $\boldsymbol{\theta}_0$, $\boldsymbol{\theta}_1$ and the prototypes $\boldsymbol{\beta}$. Inspired by hierarchical VAEs (Vahdat & Kautz, 2020; Duan et al., 2021), $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are proposed to generate multiple MSTs hierarchically, while a group of prototypes $\boldsymbol{\beta}$ are designed to capture the diverse normal dynamic patterns within multiple MTSs. Given an embedding sampled form $\boldsymbol{\theta}_0$, OT is used to index the related information represented by prototypes to guide the generation of MTSs. There are two kinds of prototypes included in PUAD: the global prototypes contain the shared information (Dai et al., 2022) summarized from all the history MTSs, while the local prototypes consider the specific information within the new MTS in meta anomaly detection defined below. PUAD is a reconstruction-based anomaly detection approach, where the generation process can be written as

$$p(\boldsymbol{x}) = \int_{\theta_0} \int_{\theta_1} p(\boldsymbol{x} \mid \boldsymbol{\theta}_1) p(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_0, \boldsymbol{\beta}) p(\boldsymbol{\theta}_0) d\theta_0 d\theta_1$$

3.2.1. Learning Prototypes with Optimal Transport

Most existing works (Dai et al., 2022; Xu et al., 2021; Su et al., 2019) focus on modeling one pattern of MTS with an individual set of parameters. By contrast, we propose to characterize the normal patterns of multiple MTSs with a group of prototypes and refactor different MTSs by combining these prototypes. This change has several advantages. First, the prototypes summarize various normal dynamic patterns for different MTSs, which enables PUAD to cover multiple MTSs with diverse characteristics by the group of prototypes. Second, each global prototype is encouraged to capture the statistical information shared by multiple MTSs,

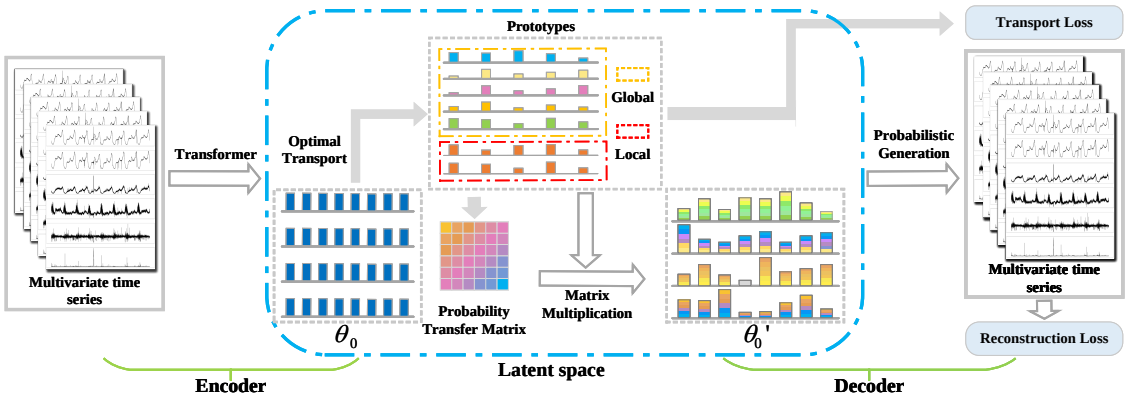

Figure 2. An overview of the framework of PUAD, which consists of a transformer based encoder, probabilistic generative model based decoder, and a POT module in latent space. $\boldsymbol{\theta}_0$ are features extracted from MTSs with the encoder, the probability transfer matrix is acquired using the OT arithmetic from $\boldsymbol{\theta}_0$ to prototypes, $\boldsymbol{\theta}_0'$ are latent representations obtained by multiplying the probability transfer matrix and the prototypes.

which is similar to the transferable patterns useful for all related tasks extracted by meta-learning (Guo et al., 2022). So we can adapt PUAD to meta-learning problems that adapt to the anomaly detection of new MTSs with limited observations efficiently. Last but not least, we define a few local prototypes independently to capture specific information for new MTSs. And when we apply the model to a new MTS, just a few local prototypes need to be optimized, which doesn't affect the summarized global prototypes and significantly saves deployment costs. As shown in Fig. 2, we introduce the POT module to capture different patterns and improve the prototypes by optimization the OT loss inspired by (Tanwisuth et al., 2021; Guo et al., 2022). POT could be regarded as a selector to index the prototypes, both global and local, which helps the model to handle the extreme conditions that if global prototypes couldn't fit current MTS or local prototype didn't work for limited data.

Given a group of global prototypes $\boldsymbol{\beta_g} = [b_g^1, b_g^2, ..., b_g^{K_g}] \in \mathbb{R}^{K_g \times d_f}$ and the local prototypes $\boldsymbol{\beta_l} = [b_l^1, b_l^2, ..., b_l^{K_l}] \in \mathbb{R}^{K_l \times d_f}$, where the dimension of each prototype, d_f , is the same as the hidden dimension after the feature encoder, K_g and K_l are the number of the global prototypes and the local prototypes. The prototypes be defined as $\boldsymbol{\beta} = [\boldsymbol{\beta_g}; \boldsymbol{\beta_l}] \in \mathbb{R}^{(K_g+K_l)\times d_f}$. To summarize the information shared between multiple MTSs, we can represent N_j samples on the MTS set as an empirical distribution over N_j like (Guo et al., 2022):

$$P_{\theta_0} = \sum_{i=1}^{N_j} \frac{1}{N_j} \delta_{\boldsymbol{\theta}_0^i},\ \boldsymbol{\theta}_0^i \in \mathbb{R}^{d_f} \tag{2}$$

where $\boldsymbol{\theta}_0$ is the embedding sample from hidden variables. The prototypes are proposed to represent the different patterns over multiple MTSs, so the importance between prototypes is equal when we try to index the suitable prototypes to represent one concrete MTS. Hence, the distribution over global prototypes could be defined as an empirical distribution:

$$P_{\boldsymbol{\beta_g}} = \sum_{i=1}^{K_g} \frac{1}{K_g} \delta_{\boldsymbol{b_g^i}},\ \boldsymbol{b_g^i} \in \mathbb{R}^{d_f} \tag{3}$$

where $\boldsymbol{\beta_g}$ is the global prototypes. In this way, we can acquiring the transport probability matrix $\boldsymbol{M} \in \mathbb{R}_{\geq 0}^{N_j \times K_g}$ from P_{θ_0} to $P_{\boldsymbol{\beta_g}}$ by Sinkhorn algorithm (Cuturi, 2013):

$$\boldsymbol{M}^* = \mathbf{OT}(P_{\theta_0}, P_{\boldsymbol{\beta_g}}) = \min_{\boldsymbol{M}} \langle \boldsymbol{M}, \boldsymbol{C} \rangle \overset{\text{def.}}{=} \sum_i^{N_j} \sum_j^{K_l} M_{ij} C_{ij}$$

$\mathbf{C} \in \mathbb{R}_{\geq 0}^{N_j \times K_g}$ is the transport cost matrix, we use the Euclidean distance between embedding $\boldsymbol{\theta}_0$ and the prototype $\boldsymbol{\beta_g}$, $C_{ij} = \sqrt{(\theta_0^i - \beta_g^j)^2}$. The transport probability matrix $\boldsymbol{M}$ should satisfy $\Pi(\boldsymbol{g}, \boldsymbol{h}) := \left\{ \boldsymbol{M} \mid \boldsymbol{M}\mathbf{1}_{K_g} = \boldsymbol{g}, \boldsymbol{M}^\top \mathbf{1}_{N_j} = \boldsymbol{h} \right\}$, where $\boldsymbol{g} = [\frac{1}{K_g}]$ and $\boldsymbol{h} = [\frac{1}{N_j}]$ are two probability vectors defend in Eq. 2 and Eq. 3. OT gives us an optimal transport plan from embedding P_{θ_0} to prototype P_{β_g} based on the cost matrix $\mathbf{C}$, and we could reconstruct $\boldsymbol{\theta}_0$ by the transport probability $\boldsymbol{M}$ and prototypes $\boldsymbol{\beta_g}$:

$$\boldsymbol{\theta}_0' = \boldsymbol{M} \times \boldsymbol{\beta_g}, \boldsymbol{\theta}_0' \in \mathbb{R}^{N_j \times d_f} \tag{4}$$

Compared with the original $\boldsymbol{\theta}_0$, $\boldsymbol{\theta}_0'$ contains the diversity dynamic information transmitted from the prototypes, thus to cover various patterns of MTSs. Inspired by the existing OT based prototype-oriented method (Guo et al., 2022; Tanwisuth et al., 2021), to learn the prototypes $\boldsymbol{\beta}$, we adopt the entropic constraint (Cuturi, 2013) and define the average

OT loss for all training sets as:

$$\begin{aligned}\mathcal{L}_{OT} &= \min_{\boldsymbol{\beta_g}} \mathbb{E}_{\theta\sim F_\phi(D_x)}\big[\sum_{i}^{N_j}\sum_{j}^{K_g} M_{ij}C_{ij} \\ &\quad + \sum_{i}^{N_j}\sum_{j}^{K_g} M_{ij} ln(M_{ij})\big] \\ &= \min_{\boldsymbol{\beta_g}} \mathbb{E}_{\theta\sim F_\phi(D_x)}\big[\mathbf{OT}(P_{\theta_0}, P_{\boldsymbol{\beta_g}})\big]\end{aligned} \tag{5}$$

$F_\phi(\cdot)$ is the inference module that will be introduced below, whose parameters are denoted by $\boldsymbol{\phi}$, D_x is the training set which consists of normal MTS data.

So far, the global prototypes can represent the diverse normal patterns within MTSs, and they are encouraged to capture the statistical temporal dependency shared by multiple MTSs, which is similar as the transfer patterns in meta learning, thus to possess powerful capacity in adapting to the new MTSs. But the specific information for new MTSs may be ignored by the global prototypes. Therefore, we further introduce the local prototypes $\boldsymbol{\beta_l}$ to replenish the ignored information and enhance the capacity of model adaptation. $\boldsymbol{\beta_l}$ is summarized from a few samples from the new MTSs first, then $\boldsymbol{\beta_g}$ and $\boldsymbol{\beta_l}$ are indexed together by OT when detecting the anomaly on the new MTSs.

3.2.2. Prototype guided probabilistic generative model

With the prototypes of POT module, we formulate a hierarchical probabilistic generative model for reconstruction-based unsupervised anomaly detection, as shown in Fig. 1 (a). Unlike existing hierarchical VAEs (Vahdat & Kautz, 2020; Su et al., 2019), they store the information in the neural network between the random variables. We introduce a hierarchical probabilistic generative model that generates data with the direction of the related information stored in the prototypes. Formally, the generative process can be expressed as:

$$\begin{aligned}\boldsymbol{\theta}_0 &\sim \mathcal{N}(0,1) \\ \boldsymbol{\theta}_1 &\sim \mathcal{N}\left(\mathscr{F}_1^\mu(\boldsymbol{\theta}_0'), \mathscr{F}_1^\sigma(\boldsymbol{\theta}_0')\right) \\ \boldsymbol{x}' &\sim \mathcal{N}\left(\mathscr{F}_2^\mu(\boldsymbol{\theta}_1), \mathscr{F}_2^\sigma(\boldsymbol{\theta}_1)\right) \\ \boldsymbol{\theta}_0' &= \boldsymbol{M}\times\boldsymbol{\beta_g}, \boldsymbol{M} = \mathrm{POT}(\boldsymbol{\theta}_0, \boldsymbol{\beta_g})\end{aligned} \tag{6}$$

Where $\boldsymbol{x}' \in \mathbb{R}^{T\times V}$ denotes the generated MTS vector for the current time step, $\mathcal{N}(\cdot,\cdot)$ is the Gaussian distribution where the values in the parenthesis are the distribution coefficient. Specifically, after sampling the latent representation $\boldsymbol{\theta}_0$, we incorporate the learned prototypes into $\boldsymbol{\theta}_0$ to get $\boldsymbol{\theta}_0'$ by the POT module. Since the information indexed from the prototypes represents the various normal dynamic patterns, which improves its generation capacity for normal MTSs, thus bringing smooth anomaly score for normal MTSs and higher anomaly score for anomaly MTS when OT couldn't find the suitable prototypes, as shown in Fig 1 (b). Finally, four nonlinearity functions $\mathscr{F}_1^\mu, \mathscr{F}_2^\mu, \mathscr{F}_1^\sigma, \mathscr{F}_2^\sigma$ in Eq. 7 are introduced to generate the distribution coefficient for the $\boldsymbol{\theta}_1$ and $\boldsymbol{x}'$, and they are defined as:

$$\begin{aligned}\mathscr{F}_1^\mu(\boldsymbol{\theta}_0') &= f(\boldsymbol{W}_1^\mu\boldsymbol{\theta_0'} + \boldsymbol{b}_1^\mu), \mathscr{F}_1^\sigma(\boldsymbol{\theta}_0') = f(\boldsymbol{W}_1^\sigma\boldsymbol{\theta_0'} + \boldsymbol{b}_1^\sigma) \\ \mathscr{F}_2^\mu(\boldsymbol{\theta}_1) &= f(\boldsymbol{W}_2^\mu\boldsymbol{\theta_1} + \boldsymbol{b}_2^\mu), \mathscr{F}_2^\sigma(\boldsymbol{\theta}_1) = f(\boldsymbol{W}_2^\sigma\boldsymbol{\theta_1} + \boldsymbol{b}_2^\sigma)\end{aligned}$$

We use the fully connected network as the nonlinearity function, where the weights $\boldsymbol{W}_1^\mu, \boldsymbol{W}_1^\sigma, \boldsymbol{W}_2^\mu, \boldsymbol{W}_2^\sigma \in \mathbb{R}^{d_f\times d_f}$ and the bias $\boldsymbol{b}_1^\mu, \boldsymbol{b}_1^\sigma, \boldsymbol{b}_2^\mu, \boldsymbol{b}_2^\sigma \in \mathbb{R}^{d_f}$ are learnable parameters, and $f(\cdot)$ is a deterministic non-linear transition function.

3.2.3. Transformer-structured Inference Model

Considering the long-term and complex temporal dependencies within MTS and focusing on learning the expressive representations, we introduce a transformer-structured inference model to approximate the true posterior distribution for $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$, which is always intractable. The variational distribution could be defined as:

$$q(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1 \mid \boldsymbol{x}, \boldsymbol{\theta}_0') = q(\boldsymbol{\theta}_0 \mid \boldsymbol{x}) q(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_0', \boldsymbol{x}) \tag{7}$$

As shown in Fig. 1 (a) and (c), a transformer (Vaswani et al., 2017), which is widely used to process time sequence data, is deployed to encode temporal relationships between different time steps. For upward information transition, we define the feature extracted from transformer as $\boldsymbol{h}_0$ and apply a fully connected network to get the feature $\boldsymbol{h}_1$:

$$\begin{aligned}\boldsymbol{h_0} &= \mathrm{Transformer}(\boldsymbol{x}) \\ \boldsymbol{h_1} &= f(\boldsymbol{W}_{h_0h_1}\boldsymbol{h_0} + \boldsymbol{b}_{h_0h_1})\end{aligned} \tag{8}$$

Then, given the features extracted by transformer, the inference process can be described by:

$$\begin{aligned}q(\boldsymbol{\theta}_0 \mid \boldsymbol{x}) &\sim \mathcal{N}\left(\boldsymbol{\theta_0} \mid \tilde{\boldsymbol{\mu}}_{\theta_0}, \mathrm{diag}\left(\tilde{\boldsymbol{\sigma}}_{\theta_0}^2\right)\right) \\ q(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_0', \boldsymbol{x}) &\sim \mathcal{N}\left(\boldsymbol{\theta_1} \mid \tilde{\boldsymbol{\mu}}_{\theta_1}, \mathrm{diag}\left(\tilde{\boldsymbol{\sigma}}_{\theta_1}^2\right)\right) \\ \tilde{\boldsymbol{\mu}}_{\theta_0} &= \Phi(\boldsymbol{h_0}) = f(\tilde{\boldsymbol{V}}_{\theta_0}^\mu\boldsymbol{h_0} + \tilde{\boldsymbol{b}}_{\theta_0}^\mu) \\ \tilde{\boldsymbol{\sigma}}_{\theta_0} &= \Phi(\boldsymbol{h_0}) = f(\tilde{\boldsymbol{V}}_{\theta_0}^\sigma\boldsymbol{h_0} + \tilde{\boldsymbol{b}}_{\theta_0}^\sigma) \\ \tilde{\boldsymbol{\mu}}_{\theta_1} &= \Psi(\boldsymbol{\theta}_0', \boldsymbol{h_1}) = f(\tilde{\boldsymbol{W}}_{\theta_1}^\mu\boldsymbol{\theta}_0' + \tilde{\boldsymbol{V}}_{\theta_1}^\mu\boldsymbol{h_1} + \tilde{\boldsymbol{b}}_{\theta_1}^\mu) \\ \tilde{\boldsymbol{\sigma}}_{\theta_1} &= \Psi(\boldsymbol{\theta}_0', \boldsymbol{h_1}) = f(\tilde{\boldsymbol{W}}_{\theta_1}^\sigma\boldsymbol{\theta}_0' + \tilde{\boldsymbol{V}}_{\theta_1}^\sigma\boldsymbol{h_1} + \tilde{\boldsymbol{b}}_{\theta_1}^\sigma)\end{aligned} \tag{9}$$

Where Φ and Ψ are two nonlinearity function that include the learnable parameters $\{\tilde{\boldsymbol{V}}_{\theta_0}^\mu, \tilde{\boldsymbol{V}}_{\theta_0}^\sigma, \tilde{\boldsymbol{V}}_{\theta_1}^\mu, \tilde{\boldsymbol{V}}_{\theta_1}^\sigma, \tilde{\boldsymbol{W}}_{\theta_1}^\mu, \tilde{\boldsymbol{W}}_{\theta_1}^\sigma\} \in \mathbb{R}^{d_f\times d_f}$ and $\{\tilde{\boldsymbol{b}}_{\theta_0}^\mu, \tilde{\boldsymbol{b}}_{\theta_0}^\sigma, \tilde{\boldsymbol{b}}_{\theta_1}^\mu, \tilde{\boldsymbol{b}}_{\theta_1}^\sigma\} \in \mathbb{R}^{d_f}$. In this way, the latent variable $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are inferred by combining the bottom-up likelihood information ($\boldsymbol{h_1}$) and prior information ($\boldsymbol{\theta}_0'$) from the generative distribution using the inference network (Rezende et al., 2014).

Algorithm 1 Upward-Downward Autoencoding Variational Inference for PUAD

Input: The pre-processed MTS training dataset $\mathcal{D}_x$. The new MTS dataset $\mathcal{D}_{new}$.
Parameter: The encoder parameters $\eta = \{ \tilde{\boldsymbol{V}}^{\mu}_{\theta_0}, \tilde{\boldsymbol{V}}^{\sigma}_{\theta_0}, \tilde{\boldsymbol{V}}^{\mu}_{\theta_1}, \tilde{\boldsymbol{V}}^{\sigma}_{\theta_1}, \tilde{\boldsymbol{W}}^{\mu}_{\theta_1}, \tilde{\boldsymbol{W}}^{\sigma}_{\theta_1}, \tilde{\boldsymbol{b}}^{\mu}_{\theta_0}, \tilde{\boldsymbol{b}}^{\sigma}_{\theta_0}, \tilde{\boldsymbol{b}}^{\mu}_{\theta_1}, \tilde{\boldsymbol{b}}^{\sigma}_{\theta_1}, \boldsymbol{\phi}\}$; The decoder parameters $\gamma = \{ \boldsymbol{W}^{\mu}_1, \boldsymbol{W}^{\sigma}_1, \boldsymbol{W}^{\mu}_2, \boldsymbol{W}^{\sigma}_2, \boldsymbol{b}^{\mu}_1, \boldsymbol{b}^{\sigma}_1, \boldsymbol{b}^{\mu}_2, \boldsymbol{b}^{\sigma}_2\}$; The global prototypes $\boldsymbol{\beta_g}$ and local prototypes $\boldsymbol{\beta_l}$, $\boldsymbol{\beta} = [\boldsymbol{\beta_g}; \boldsymbol{\beta_l}]$.

while epoch **do**
  Randomly select a mini-batch $\{x^i_{1:T}\}^N_{i=1}$ in $\mathcal{D}_x(x_{1:T})$;
  Inference the posterior $q(\boldsymbol{\theta}_0)$ and $q(\boldsymbol{\theta}_1)$ through Eq. 9, reconstruct $\boldsymbol{\theta}'_0$ by $\boldsymbol{\beta}_g$ in Eq. 4;
  Update the parameters(η, γ, $\boldsymbol{\beta}_g$) by Eq. 11;
end while
while epoch **do**
  Randomly select a mini-batch $\{x^i_{1:T}\}^N_{i=1}$ in $\mathcal{D}_{new}(x_{1:T})$;
  Inference the posterior $q(\boldsymbol{\theta}_0)$ and $q(\boldsymbol{\theta}_1)$ through Eq. 9, reconstruct $\boldsymbol{\theta}'_0$ by $\boldsymbol{\beta}$ in Eq. 4;
  Update the parameters($\boldsymbol{\beta}_l$) by Eq. 11;
end while
return Parameters after training η, γ, $\boldsymbol{\beta}$.

3.3. Model Training

As the definition described in Eq. 2, the optimization objective of PUAD can be achieved by maximizing the evidence lower bound (ELBO) of the log marginal likelihood (see the Appendix for details), which can be written as:

$$\mathcal{L} = \mathbf{E}_{q(\boldsymbol{\theta_0}),q(\boldsymbol{\theta_1})}\left[\log p(\boldsymbol{x} \mid \boldsymbol{\theta_1})\right] - \mathbf{D}_{KL}\left(q(\boldsymbol{\theta}_0 \mid \boldsymbol{x}) \| p(\boldsymbol{\theta}_0)\right) - \mathbf{D}_{KL}\left(q(\boldsymbol{\theta}_1 \mid \boldsymbol{x}, \boldsymbol{\theta}'_0) \mid \boldsymbol{x}\right) \| p(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}'_0)) \tag{10}$$

Inspired by beta-VAE (Higgins et al., 2016), we introduce three hyperparameters $\rho_1 > 0$, $\rho_2 > 0$ and $\rho_3 > 0$, then adding the KL loss gradually with $\rho 1$, ρ_2 and ρ_3 increasing from 0 to 1 during the first K training epochs. Finally, the network parameters and the global prototypes are learned by jointly optimizing the ELBO and OT loss in Eq. 5:

$$\begin{aligned}\mathcal{L} = &\mathbf{E}_{q(\boldsymbol{\theta_0}),q(\boldsymbol{\theta_1})}\left[\log p(\boldsymbol{x} \mid \boldsymbol{\theta_1})\right] \\ &- \rho_1 \mathbf{D}_{KL}\left(q(\boldsymbol{\theta}_1 \mid \boldsymbol{x}, \boldsymbol{\theta}'_0) \mid \boldsymbol{x}\right) \| p(\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}'_0)) \\ &- \rho_2 \mathbf{D}_{KL}\left(q(\boldsymbol{\theta}_0 \mid \boldsymbol{x}) \| p(\boldsymbol{\theta}_0)\right) - \rho_3 \mathcal{L}_{OT}\end{aligned} \tag{11}$$

In summary, optimizing the OT loss defined by the prototype distribution $P_{\boldsymbol{\beta}}$ and the embedding distribution P_{θ_0} provides a principled and unsupervised way to encourage the prototypes to capture the diverse normal patterns within multiple MTSs. As shown in Algorithm 1, the model parameters are optimized by stochastic gradient descent through an end-to-end way.

3.4. Anomaly Detection

Since the model is trained to learn normal patterns of MTSs, the more an observation follows normal patterns, the more likely it can be reconstructed well with higher confidence. Hence, we apply the reconstruction probability of $\boldsymbol{x}$ as the anomaly score to determine whether an observed variable is anomalous or not (An & Cho, 2015; Su et al., 2019; 2021; Xu et al., 2018b), and it is computed as:

$$S_t = \log p(\boldsymbol{x} \mid \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \tag{12}$$

Observation $\boldsymbol{x}$ will be classified as anomalous if S_t is below a specific threshold. From a practical point of view, we use the Peaks-Over-Threshold (Siffer et al., 2017) approach to help select threshold.

Dataset	SMD	MSL	PSM	SMAP	DND
OC-SVM	56.19	70.82	70.67	56.34	69.28
IsolationForest	53.64	66.45	83.48	55.53	71.85
LOF	46.68	61.18	70.61	57.60	72.38
Deep-SVDD	79.10	83.58	90.73	69.04	75.94
DAGMM	57.30	74.62	80.08	68.51	75.11
MMPCACD	75.02	69.95	77.29	81.73	73.57
VAR	74.08	77.90	87.13	64.83	75.24
LSTM	81.78	83.95	82.80	83.39	76.47
CL-MPPCA	79.09	80.44	71.80	72.88	75.81
ITAD	79.48	76.07	68.13	73.85	74.64
LSTM-VAE	82.30	82.62	80.96	78.10	77.02
BeatGAN	78.10	87.53	92.04	69.61	78.37
Men-SkipAE	79.77	88.61	92.57	87.67	81.03
MemAE	78.41	87.48	92.17	75.42	79.90
TSMAE	85.35	87.87	80.91	86.06	81.38
OmniAnomaly	85.22	87.67	80.83	86.92	80.90
InterFusion	86.22	86.62	83.52	89.14	81.62
THOC	84.99	89.69	89.54	90.68	82.19
PGDLE	85.03	90.39	90.37	92.02	83.11
SummerNet	92.69	92.08	94.49	92.11	81.90
GmVRNN	93.56	91.41	96.97	95.51	85.58
Anomaly Transformer	92.33	93.59	97.89	96.69	84.16
TranAD	96.05	94.94	96.97	89.15	84.54
PUAD	**96.16**	**95.04**	**98.14**	**96.72**	**86.62**

Table 1. F1-score results for different methods on five public datasets and one real-world datasets. F1-score is the harmonic mean of precision and recall. For this metric, a higher value indicates a better performance, see the Appendix for more metrics.

4. Experimental Evaluation

4.1. Experiment Setup

Dataset: Five datasets are used in our experiments, including four public datasets: SMD (Su et al., 2019), MSL and SMAP from NASA (Hundman et al., 2018b), PSM (Abdulaal et al., 2021), and one real world dataset: DND (Chen et al., 2022). See the Appendix for details.

Implementation details: In our experiment, a three-layer transformer with 512 dimensions of the hidden states is implemented as the encoder. The dimension of MPL that mapping the feature $\boldsymbol{h_0}, \boldsymbol{h_1}$ is 256. POT consists of 10 (K_g) global prototypes and 2 (K_l) local prototypes with dimension 256 for all datasets. We set latent states $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ has the same dimension with 512. Hyper-parameters ρ_1 , ρ_2 and ρ_3 are set to 0.01 to balance the reconstruction and KL parts for all datasets. The Adam optimizer is employed

Our implementation is publicly available at https://github.com/BoChenGroup/PUAD

Methods	Dataset	SMD					MSL					SMAP				
	Data Number	1	5	10	20	200	1	5	10	20	200	1	5	10	20	200
	Metric	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1
Random Initialization	DAGMM	43.11	44.16	44.87	44.02	63.38	50.66	50.52	51.36	51.45	65.15	53.38	53.26	54.89	60.16	65.68
	MMPCACD	54.40	55.49	55.63	56.21	67.05	51.31	51.94	52.57	52.75	65.08	52.90	52.76	56.01	61.05	65.87
	VAR	54.40	54.29	54.65	55.75	66.39	53.99	53.02	53.60	53.82	66.31	55.29	55.18	56.36	62.90	66.00
	LSTM	56.17	56.68	56.87	56.89	67.76	60.58	60.62	61.29	61.37	69.59	56.73	56.57	57.58	61.98	67.74
	CL-MPPCA	57.00	57.05	57.17	57.87	68.14	59.67	59.34	60.30	60.69	70.08	58.76	58.43	58.28	65.55	68.36
	ITAD	56.36	56.08	56.51	56.90	66.93	62.98	62.71	63.17	63.35	69.50	62.74	62.71	63.85	66.95	68.85
	LSTM-VAE	58.38	58.74	58.08	59.48	69.15	61.89	61.10	61.90	62.16	71.05	61.61	61.23	62.19	67.41	69.16
	BeatGAN	61.30	61.18	61.04	62.02	68.73	63.09	63.57	64.23	65.88	74.30	61.49	61.18	61.72	66.89	71.10
	OmniAnomaly	64.29	65.15	63.53	62.05	68.31	64.36	64.43	65.14	68.77	73.80	63.13	65.55	65.61	68.72	70.27
	InterFusion	63.68	62.66	62.37	62.28	69.83	63.60	63.56	64.32	65.83	62.16	63.50	63.34	63.95	68.79	72.44
	THOC	64.41	63.47	63.12	63.94	71.01	64.33	64.13	64.22	66.00	74.65	62.24	62.46	63.92	70.66	75.22
	GmVRNN	91.03	89.51	90.01	90.78	90.34	81.68	81.15	82.12	81.59	81.22	93.47	93.41	93.01	92.21	94.12
	Anomaly Transformer	64.24	65.09	71.85	76.98	81.11	65.16	64.99	68.52	70.46	77.02	63.82	66.72	67.34	71.45	78.88
Pretraining with History MTSs	OmniAnomaly	85.40	85.42	85.78	86.19	85.83	81.12	80.58	82.13	81.86	81.64	84.28	84.71	84.61	84.97	85.49
	InterFusion	84.60	84.73	84.74	84.85	85.09	82.09	81.87	82.28	82.46	83.00	86.70	86.80	87.26	87.27	87.12
	THOC	86.01	86.05	86.24	86.62	86.54	81.56	81.55	81.38	82.73	82.86	88.95	89.28	88.92	88.54	88.95
	GmVRNN	91.03	89.51	90.01	90.78	90.34	81.68	81.15	82.12	81.59	81.22	93.47	93.41	93.01	92.21	94.12
	Anomaly Transformer	91.58	91.92	91.46	91.78	91.69	80.65	82.87	82.15	83.55	84.01	93.32	93.89	93.51	94.01	94.11
Prototypes	PUAD	**95.68**	**95.42**	**95.01**	**95.50**	**95.51**	**91.72**	**93.71**	**93.57**	**93.30**	**93.64**	**95.20**	**95.67**	**95.97**	**96.06**	**96.12**

Table 2. Quantitative results of different methods for meta anomaly detection on three public datasets.

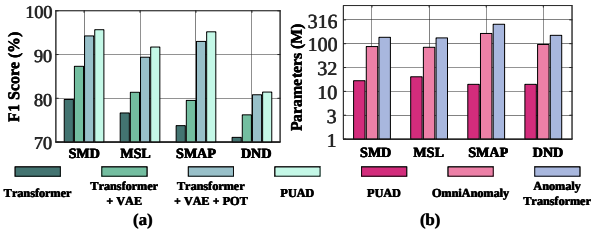

Figure 3. (a) Ablation study of PUAD on four datasets. (b) Comparison of model parameters.

with a learning rate of 0.00002, and the batch size is set to 256. The probability p associated with the initial threshold used in Peaks-Over-Threshold is set to 0.01 empirically. We adopt a sliding window to obtain a set of sub-series (Shen et al., 2020), the sliding window is with a fixed size of 20 for all datasets. All the experiments are implemented in Pytorch (Paszke et al., 2019) with NVIDIA RTX 3090 24GB GPU.

4.2. Main Result

4.2.1. Baselines

We extensively compare our model with 24 baselines, including the reconstructionbased models: InterFusion (Li et al., 2021), BeatGAN (Zhou et al., 2019), OmniAnomaly (Su et al., 2019), LSTM-VAE (Park et al., 2018); the density-estimation models: DAGMM (Zong et al., 2018), MPPCACD (Yairi et al., 2017), LOF (Breunig et al., 2000); the clusteringbased methods: ITAD (Shin et al., 2020), THOC (Shen et al., 2020), Deep-SVDD (Ruff et al., 2018); the autoregression-based models: CL-MPPCA (Tariq et al., 2019), LSTM (Hundman et al., 2018b), VAR (Clements & Mizon, 1991); the classic methods: OC-SVM (Tax & Duin, 2004), IsolationForest (Liu et al., 2008); the Mem-autoencoder-based models: Men-SkipAE (Yan et al., 2023), MemAE (Gong et al., 2019) and TSMAE (Gao et al., 2022); the prototype-based models: PGDLE (Lai et al., 2021) and SummerNet (Guo et al., 2021). Anomaly Transformer (Xu et al., 2021)and InterFusion (Li et al., 2021) are the state-of-the-art deep models. GmVRNN (Dai et al., 2022) is a state-of-the-art deep probability model for anomaly detection. TranAD (Tuli et al., 2022) is the latest method of anomaly detection in our cognition. We list more descriptions in the Appendix.

4.2.2. Quantitative Comparison

Anomay detection: To evaluate the performance of PUAD in the typical anomaly detection setting, we consider our model on five datasets with multiple competitive baselines. F1-score (Dai et al., 2022) is employed as the performance indicators. We note that GmVRNN and PUAD are *one-for-all* models, while the others are *one-for-one* models. As shown in Table 1, by considering stochasticity and diversity within MTS, GmSVRNN achieves better performance than other non-transformer methods. Transformer-based methods outperform other methods for the powerful capacity to modeling complex dynamics. PUAD achieves the best F1-score among all the methods on all test datasets, showing the effectiveness of considering diverse dynamic patterns within multiple MTSs with prototypes and formulating transformer-powered probabilistic generative model. We list the precision and recall performance in the Appendix.

Number of Parameters: To demonstrate the computational efficiency of PUDA, We compare the number of model parameters and list the results in Fig. 3 (b). As we can see, as a *one-for-all* method, the number of parameters of PUAD is much smaller than *one-for-one* methods, while it can achieve higher F1-score as shown in Table 2. We also test the time efficiency and list in the Appendix.

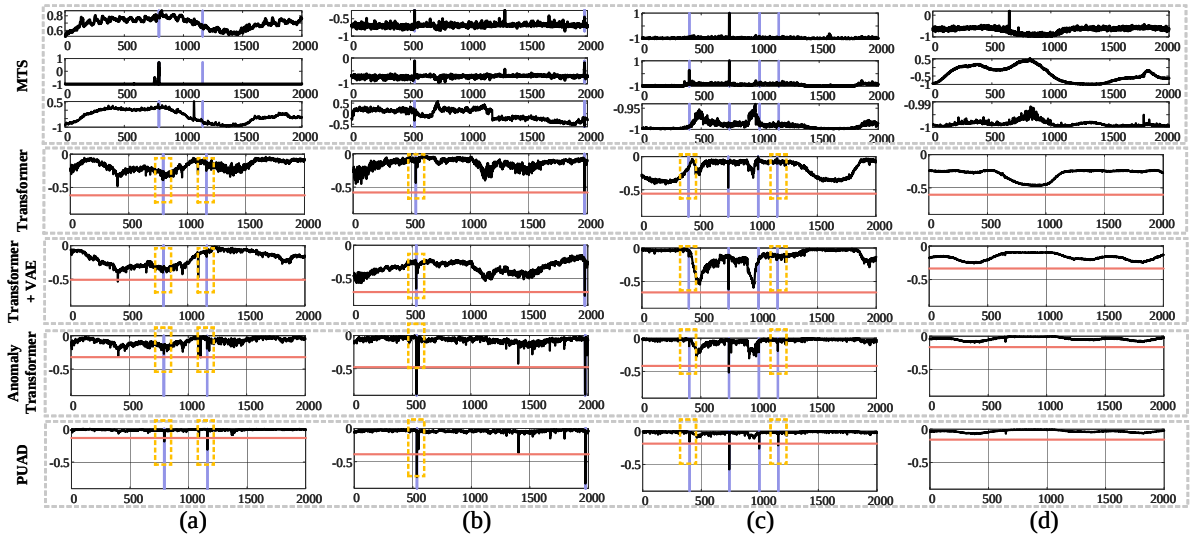

Figure 4. Case study of anomaly score on SMD dataset. Regions highlighted in purple represent the groundtruth anomaly segments, read lines refer to the threshold. (a), (b), (c), and (d) are the data pieces at different moments, respectively.

Meta anomaly detection: In real word applications with MTS data, there is a potential need to adapt the detection model to new MTSs, such as new machines or new websites, with limited observed data. In this way, we define a novel experiment setting named meta anomaly detection to evaluate the performance of the models in this practical application scenario. For all datasets in our experiment, we used 80% of the MTSs as the history data and others as new data. In real word applications, if a *one-for-one* anomaly detection model is deployed on new equipment, we may only have a few data samples to train the model. For the above reasons, we train *one-for-one* models on i ($i \in \{1, 5, 10, 20, 200\}$) samples in every new MTS and test the model on the rest of the data. For the *one-for-all* models, the parameters have been updated with the history MTSs. The detection performance of different methods is listed in Table 2. As we can see, the *one-for-one* models couldn't learn normal patterns correctly with the limited observations, leading to performance degradation. Oppositely, benefiting from the stored information from the history MTSs, the *one-for-all* models, including GmVRNN and PUAD, perform much better than the *one-for-one* models. Moreover, with the aid of the transferable global prototypes and meta-learned local prototypes, PUAD achieves the best meta anomaly detection performance. In addition, to further prove the advantages of our prototypes designed for the *one-for-all* setting, we pre-training the *one-for-one* models with historical data and finetune on the first i data in new MTSs. As shown in Table 2, *one-for-one* models can perform better because of the history MTSs involved, but their performance is still much worse than PUAD. We list more results in the Appendix.

4.2.3. Ablation Study

As shown in Fig. 3 (a), we further investigate the effect of each part in PUAD. Four experiments are performed in our ablation study, *Transformer* is a baseline model which only has three layers of transformer, *Transformer+VAE* incorporates the transformer-structure into the encoder of VAE, *Transformer+VAE+POT* further introduces the proposed POT module and *PUAD* is the integrated version that brings the OT cost as the regular terms in the loss. We can see that PUAD performs 4.52% better than *Transformer+VAE+POT* averaged over four datasets in Fig. 3 (a), showing the effectiveness of proposed OT based regular terms in the loss function. Comparing *Transformer+VAE+POT* and *Transformer+VAE*, we find that the proposed POT model improves the F1 score by 7.11% on average, which is a benefit from the prototype-oriented framework. In short, all components we incorporate into our model can bring improvement in performance, illustrating the effectiveness of each.

4.2.4. Qualitative Analysis

Anomaly score: Firstly, we compare the anomaly scores between PUAD, Anomaly Transformer, *Transformer + VAE*, and a simple Transformer. The results are visualized in Fig. 4. As the deterministic methods, Transformer gets more rough anomaly scores since they ignore the stochastic of MTS. For probabilistic methods, the anomaly score of *Transformer + VAE* is smoother due to more elaboration probabilistic model design. However, it is still difficult to deal with some difficult situations with a simple Gaussian hidden variable model. Fig. 4 (a) - (c) represents several cases where PUAD can be detected but other methods cannot. In (a) and (c), the model easily ignores some inconspicuous anomalies as noise, and PUAD can detect them but others cannot provide a high enough anomaly score due to insufficient robustness. In Fig. 4 (b), the more practical model design makes Anomaly Transformer and PUAD have higher anomaly scores and thus successfully detect the anomaly. As shown in Fig. 4 (d), thanks to the diverse

Methods	Average training times per epoch (sec)					Testing times per sample (sec × 10^{-5})				
	SMD	MSL	SMAP	PSM	DND	SMD	MSL	SMAP	PSM	DND
GmVRNN	25.20	2.67	5.04	37.96	35.26	7.02	13.54	8.17	8.26	8.89
TranAD	43.76	5.57	10.55	63.58	60.30	9.17	17.63	10.49	10.66	11.71
Anomaly Transformer	18.76	1.47	3.37	28.27	26.29	8.29	12.59	8.41	9.73	10.51
Ours	19.15	1.75	3.51	28.79	26.81	**3.30**	**6.34**	**3.87**	**3.87**	**4.11**

Table 3. Training and testing time of different methods.

global information provided by POT, the anomaly score of PUAD is much lower than other models in the non-anomaly state. The process of POT to find relevant information in prototypes can be equivalent to comparing current MTS with a large amount of historical data. This allows PUAD to have a smoother anomaly score in non-anomaly cases and obtain a higher anomaly score in anomaly cases.

Transport Probability Matrix: In this paper, a significant contribution is to propose a prototype-oriented UAD model, where we want to capture diverse temporal information within multiple MTSs by prototypes to enhance the capturing of the normal pattern. Recalling the appropriate information for new MTSs from well-learned prototypes is essential to our model. To verify this ability, we observed the transport probability matrix weights of POT in Fig. 5. We randomly select five prototypes to visualize the transport probability weights between selected prototypes and $\boldsymbol{\theta}_0$ for observation. The top of the figure shows the MTS data, which contains four different dynamic patterns indicated by the red boxes. The bar chart below represents the mean of the probability weights in the red box. As shown in Fig. 5, the corresponding prototypes can be generated for different MTSs by adjusting the transport probability weights.

4.2.5. Time Efficiency

Similar to previous works (Dai et al., 2022), we test the time efficiency of different methods, including our proposed PUAD, transformer-based methods TranAD, RNN-based methods GmVRNN, in terms of their training and testing time, and list the results in Table 3. As we can see, being accelerated with Graphical Process Units (GPUs), the training and testting time of our proposed method are much lower than other methods. Meanwhile, all models can perform anomaly detection for a sample within one-tenth second versus the data collecting interval of 60 seconds, which illustrates that these methods can be employed for online detection.

5. Conclusion

In this paper, we propose a novel prototype-oriented probabilistic meta anomaly detection method, named PUAD, to improve the limitation of the existing unsupervised anomaly detection methods for MTS in modeling the diverse normal patterns and adapting to new data. PUAD considers

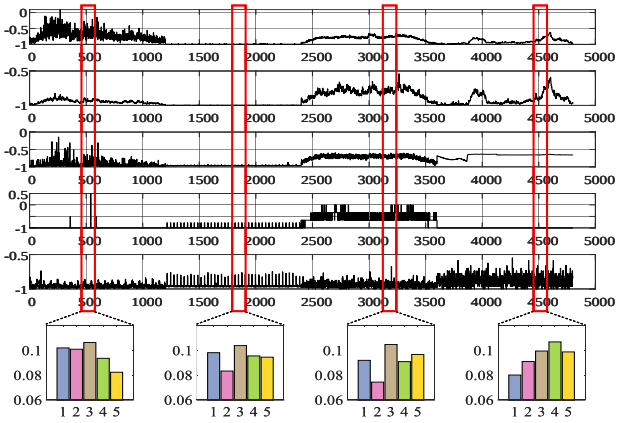

Figure 5. Illustration of transport probability matrix weights. The top half of the figure shows part of the MTS channels, with four different patterns intercepted, and the bottom half height=5.0cm,width=9.0cmshows the matrix weights of the corresponding positions in red boxes.

the various dynamics within multiple MTSs by defining a group of prototypes, and views each MTS as the distribution over these prototypes. A novel POT is developed to optimize the prototypes, and then PUAD formulates a Transformer-based powerful probabilistic dynamical generative framework for a reconstruction-based unsupervised anomaly detection approach. PUAD can not only make use of global prototypes to capture the diverse normal patterns for multiple MTSs, but also leverages meta-learned transferable prototypes to achieve high model adaption capacity for new MTSs. Extensive experiments on five datasets show that PUAD achieves SOTA performance on both regular and meta anomaly detection tasks for MTS.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant U21B2006; in part by Shaanxi Youth Innovation Team Project; in part by the Fundamental Research Funds for the Central Universities QTZX23037 and QTZX22160; in part by the 111 Project under Grant B18039; The work of Wenchao Chen acknowledges the support of the stabilization support of National Radar Signal Processing Laboratory under Grant (JKW202X0X) and National Natural Science Foundation of China (NSFC) (6220010437).

References

Abdulaal, A., Liu, Z., and Lancewicki, T. Practical approach to asynchronous multivariate time series anomaly detection and localization. *In ACM SIGKDD*, pp. 2485–2494, 2021.

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

Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. USAD: unsupervised anomaly detection on multivariate time series. *In ACM SIGKDD*, pp. 3395–3404, 2020.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof: identifying density-based local outliers. *In ACM SIGMOD*, pp. 93–104, 2000.

Cao, H., Guo, X., and Wang, G. Meta-learning with gans for anomaly detection, with deployment in high-speed rail inspection system. *arXiv preprint arXiv:2202.05795*, 2022.

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

Chen, W., Tian, L., Chen, B., Dai, L., Duan, Z., and Zhou, M. Deep variational graph convolutional recurrent network for multivariate time series anomaly detection. In *International Conference on Machine Learning*, pp. 3621–3633. PMLR, 2022.

Clements, M. P. and Mizon, G. E. Empirical analysis of macroeconomic time series: Var and structural models. *In European Economic Review*, 35(4):887–917, 1991.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Dai, L., Lin, T., Liu, C., Jiang, B., Liu, Y., Xu, Z., and Zhang, Z.-L. Sdfvae: Static and dynamic factorized vae for anomaly detection of multivariate cdn kpis. In *Proceedings of the Web Conference 2021*, pp. 3076–3086, 2021.

Dai, L., Chen, W., Liu, Y., Argyriou, A., Liu, C., Lin, T., Wang, P., Xu, Z., and Chen, B. Switching gaussian mixture variational rnn for anomaly detection of diverse cdn websites. *In INFOCOM 2022*, pp. 300–309, 2022.

de Paula Monteiro, R., Lozada, M. C., Mendieta, D. R. C., Loja, R. V. S., and Bastos Filho, C. J. A. A hybrid prototype selection-based deep learning approach for anomaly detection in industrial machines. *Expert Systems with Applications*, 204:117528, 2022.

Du, Y., Zhen, X., Shao, L., and Snoek, C. G. Hierarchical variational memory for few-shot learning across domains. *arXiv preprint arXiv:2112.08181*, 2021.

Duan, Z., Wang, D., Chen, B., Wang, C., Chen, W., Li, Y., Ren, J., and Zhou, M. Sawtooth factorial topic embeddings guided gamma belief network. In *International Conference on Machine Learning*, pp. 2903–2913. PMLR, 2021.

Gao, H., Qiu, B., Barroso, R. J. D., Hussain, W., Xu, Y., and Wang, X. Tsmae: a novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder. *IEEE Transactions on network science and engineering*, 2022.

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., and Hengel, A. v. d. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. pp. 1705–1714, 2019.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

Guo, D., Tian, L., Zhang, M., Zhou, M., and Zha, H. Learning prototype-oriented set representations for meta-learning. *arXiv preprint arXiv:2110.09140*, 2021.

Guo, D., Tian, L., Zhang, M., Zhou, M., and Zha, H. Learning prototype-oriented set representations for meta-learning. In *International Conference on Learning Representations*, 2022.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Söderström, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. *In ACM SIGKDD*, pp. 387–395, 2018a.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. *In ACM SIGKDD*, pp. 387–395, 2018b.

Lai, Y., Han, Y., and Wang, Y. Anomaly detection with prototype-guided discriminative latent embeddings. pp. 300–309, 2021.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., and Ng, S. MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks. *In ICANN*, pp. 703–716, 2019.

Li, Z., Zhao, Y., Han, J., Su, Y., Jiao, R., Wen, X., and Pei, D. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. *In ACM SIGKDD*, pp. 3220–3230, 2021.

Liu, D., Zhao, Y., Xu, H., Sun, Y., Pei, D., Luo, J., Jing, X., and Feng, M. Opprentice: Towards practical and automatic anomaly detection through machine learning. *In ACM IMC*, pp. 211–224, 2015.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. *In IEEE ICDM*, pp. 413–422, 2008.

Liu, J., Song, K., Feng, M., Yan, Y., Tu, Z., and Zhu, L. Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. *Optics and Lasers in Engineering*, 136:106324, 2021.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G. Lstm-based encoder-decoder for multi-sensor anomaly detection. *In ICML*, 2016.

Park, D., Hoshi, Y., and Kemp, C. C. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *In IEEE Robot*, 3(3):1544–1551, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *In NeurIPS*, 32, 2019.

Peyré, G., Cuturi, M., et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and variational inference in deep latent gaussian models. In *International conference on machine learning*, volume 2, pp. 2, 2014.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. Deep one-class classification. *In ICML*, pp. 4393–4402, 2018.

Shen, L., Li, Z., and Kwok, J. Timeseries anomaly detection using temporal hierarchical one-class network. *In NeurIPS*, 33:13016–13026, 2020.

Shin, Y., Lee, S., Tariq, S., Lee, M. S., Jung, O., Chung, D., and Woo, S. S. Itad: integrative tensor-based anomaly detection system for reducing false positives of satellite systems. *In 29th CIKM*, pp. 2733–2740, 2020.

Shon, T. and Moon, J. A hybrid machine learning approach to network anomaly detection. *Inf. Sci.*, 177(18):3799–3821, 2007.

Siffer, A., Fouque, P.-A., Termier, A., and Largouet, C. Anomaly detection in streams with extreme value theory. *IN ACM SIGKDD*, pp. 1067–1075, 2017.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. *In ACM SIGKDD*, pp. 2828–2837, 2019.

Su, Y., Zhao, Y., Sun, M., Zhang, S., Wen, X., Zhang, Y., Liu, X., Liu, X., Tang, J., Wu, W., et al. Detecting outlier machine instances through gaussian mixture variational autoencoder with one dimensional cnn. *IEEE Transactions on Computers*, 71(4):892–905, 2021.

Sun, M., Su, Y., Zhang, S., Cao, Y., Liu, Y., Pei, D., Wu, W., Zhang, Y., Liu, X., and Tang, J. Ctf: Anomaly detection in high-dimensional time series with coarse-to-fine model transfer. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2021.

Tanwisuth, K., Fan, X., Zheng, H., Zhang, S., Zhang, H., Chen, B., and Zhou, M. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:17194–17208, 2021.

Tariq, S., Lee, S., Shin, Y., Lee, M. S., Jung, O., Chung, D., and Woo, S. S. Detecting anomalies in space using multivariate convolutional lstm with mixtures of probabilistic pca. *In ACM SIGKDD*, pp. 2123–2133, 2019.

Tax, D. M. and Duin, R. P. Support vector data description. *In Machine learning*, 54(1):45–66, 2004.

Tuli, S., Casale, G., and Jennings, N. R. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*, 2022.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vilalta, R. and Drissi, Y. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

Wang, D., Guo, D., Zhao, H., Zheng, H., Tanwisuth, K., Chen, B., and Zhou, M. Representing mixtures of word embeddings with mixtures of topic embeddings. In *International Conference on Learning Representations*, 2022.

Wu, J., Chen, D., Fuh, C., and Liu, T. Learning unsupervised metaformer for anomaly detection. *ICCV*, pp. 4349–4358, 2021.

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., Chen, J., Wang, Z., and Qiao, H. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. *In WWW 18*, pp. 187–196, 2018a.

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. *In WWW*, pp. 187–196, 2018b.

Xu, J., Wu, H., Wang, J., and Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

Yairi, T., Takeishi, N., Oda, T., Nakajima, Y., Nishimura, N., and Takata, N. A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction. *In IEEE T AERO ELEC SYS*, 53(3):1384–1401, 2017.

Yamada, M., Kimura, A., Naya, F., and Sawada, H. Change-point detection with feature selection in high-dimensional time-series data. *In IJCAI*, pp. 1827–1833, 2013.

Yan, H., Liu, Z., Chen, J., Feng, Y., and Wang, J. Memory-augmented skip-connected autoencoder for unsupervised anomaly detection of rocket engines with multi-source fusion. *ISA transactions*, 133:53–65, 2023.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., and Chawla, N. V. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *In AAAI*, pp. 1409–1416, 2019.

Zhang, H., Chen, B., Guo, D., and Zhou, M. Whai: Weibull hybrid autoencoding inference for deep topic modeling. *arXiv preprint arXiv:1803.01328*, 2018.

Zhen, X., Du, Y., Xiong, H., Qiu, Q., Snoek, C., and Shao, L. Learning to learn variational semantic memory. *Advances in Neural Information Processing Systems*, 33: 9122–9134, 2020.

Zhou, B., Liu, S., Hooi, B., Cheng, X., and Ye, J. Beatgan: Anomalous rhythm detection using adversarially generated time series. *IJCAI*, pp. 4433–4439, 2019.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. *In ICLR*, 2018.

A. Datasets

Five datasets are used in our experiments, including (1) SMD (Su et al., 2019) is a 5-week-long dataset that is collected from a large Internet company with 38 dimensions. (2) Both MSL (Mars Science Laboratory rover) and SMAP (Soil Moisture Active Passive satellite) are public datasets from NASA (Hundman et al., 2018b) with 55 dimensions, which contain the telemetry anomaly data derived from the Incident Surprise Anomaly (ISA) reports of spacecraft monitoring systems. (3) PSM (Abdulaal et al., 2021) is collected internally from multiple application server nodes at eBay with 26 dimensions. (4) The DND, multivariate KPIs dataset, is the real-world dataset that collected from a large internet company in China. It contains 12 websites monitored with 36 KPIs individually. These websites are different from each other in types of services, e.g., Video on Demand (VoD) or live streaming video, etc. Besides, for each website, KPIs span about one and a half months and are collected every 60 seconds. In our experiments, for each website, the first half of the KPIs are used for training, while the last half are used for testing. Note that ground-truth anomalies at test time of the DND have been confirmed by human operators. The basic statistical information of datasets is reported in Table 4.

Dataset	SMD	MSL	SMAP	PSM	DND
Dimension	38	55	55	25	32
Window	20	20	20	20	20
Training	708405	58317	135181	105984	344843
Test (labeled)	708420	73729	427617	87841	344843
Anomaly ratio (%)	4.1	10.7	13.13	27.8	3.44

Table 4. Basic statistics of datasets.

B. Why the Baselines Is Selected

In reconstruction-based models, we select multiple reconstruction methods including hierarchical VAE (InterFusion), LSTM-based VAE (LSTM-VAE and OmniAnomaly), and GAN (BeatGAN). The density-estimation models include a Gaussian mixture AE (DAGMM), a probabilistic dimensionality reduction method (MPPCACD), and a classic density-estimation model (LOF). Three clustering-based methods are involved in our experiments: a tensor-based decomposition method (ITAD), a hierarchical one-class network, and a deep learning-based Support Vector Data Description method (Deep-SVDD). The autoregression-based models include a convolutional-LSTM based method (CL-MPPCA), a LSTM-based method (LSTM), and a widely used VAR method (VAR). We also introduce the classic methods, including OC-SVM and IsolationForest, to show the gains from deep learning. The Mem-autoencoder-based models (Men-SkipAE, MemAE and TSMAE) and prototype-based models(PGDLE and SummerNet) is selected to compare the advantage of PUAD. Then, a recently proposed one-for-all model GmVRNN is used as a baseline to compare the adapt-to-new performance with PUAD. Finally, InterFusion, Anomaly Transformer, and TranAD are recently proposed methods and achieve SOTA performance, we introduce them as the baselines to illustrate the overall SOTA performance of PUAD. Generally speaking, the selected baselines aim to cover all detection path and SOTA methods.

C. Evaluation Metrics

Four metrics is used to evaluate the proposed model. The P, R, F1 and AUC represent the precision, recall, F1-score (as %) and area under the ROC curve respectively. Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. F1-score is the harmonic mean of precision and recall. For these three metrics, a higher value indicates a better performance. AUC is a metric that is widely used to anomaly detection.

D. More Experiments for Meta Anomaly Detection (Random Initialization)

In real word applications with MTS data, such as CDN system, there is a potential need that adapting detection model to new MTSs, such as new machines or new websites, with limited observed data. In this way, we proposed a novel experiment setting named meta anomaly detection to evaluate the performance of the models in this practical application scenario. For all datasets in our experiment, we used 80% of the MTSs as the history data and others as new data. In real word applications, if a *one-for-one* anomaly detection model is deployed on new equipment, we may have ten or twenty data samples to train

Data Number	1				5				10				20				200			
Metric	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
DAGMM	40.91	45.58	43.11	51.15	43.26	45.08	44.16	51.54	44.55	45.22	44.87	51.49	42.80	45.30	44.02	51.25	59.57	67.70	63.38	77.32
MMPCACD	57.94	51.26	54.40	60.40	56.43	54.59	55.49	60.97	57.13	54.21	55.63	60.96	57.89	54.62	56.21	60.40	68.49	65.68	67.05	79.01
VAR	49.55	60.30	54.40	57.91	53.99	54.61	54.29	57.25	54.45	54.84	54.65	57.51	57.58	54.02	55.75	57.01	67.72	65.12	66.39	78.38
LSTM	51.51	61.75	56.17	62.78	56.96	56.40	56.68	62.95	57.32	56.42	56.87	62.73	56.89	56.88	56.89	62.19	68.18	67.33	67.76	79.03
CL-MPPCA	66.60	49.81	57.00	63.19	57.03	57.07	57.05	63.46	56.89	57.47	57.17	63.01	58.56	57.21	57.87	63.23	67.36	68.94	68.14	80.51
ITAD	54.57	58.28	56.36	58.37	54.11	58.20	56.08	58.16	54.57	58.60	56.51	58.95	54.45	59.56	56.90	58.36	63.70	70.53	66.93	78.21
LSTM-VAE	52.13	66.32	58.38	58.71	52.39	66.85	58.74	58.18	51.58	66.47	58.08	58.55	53.64	66.78	59.48	58.03	63.13	76.45	69.15	79.02
BeatGAN	62.34	60.29	61.30	70.47	61.53	60.82	61.18	70.16	60.79	61.30	61.04	70.64	58.44	66.06	62.02	70.98	69.97	67.55	68.73	79.12
OmniAnomaly	64.40	64.19	64.29	78.73	65.40	64.91	65.15	78.84	62.34	64.77	63.53	78.15	59.78	64.49	62.05	78.63	62.08	75.93	68.31	79.91
InterFusion	62.96	64.42	63.68	71.86	60.72	64.74	62.66	71.69	60.38	64.51	62.37	71.70	60.64	64.01	62.28	71.27	63.97	76.90	69.83	81.71
THOC	63.63	65.23	64.41	76.14	61.50	65.58	63.47	76.97	61.00	65.41	63.12	76.62	62.22	65.74	63.94	76.50	74.73	67.66	71.01	83.79
GmVRNN	91.57	90.50	91.03	98.81	88.39	90.68	89.51	98.32	89.05	90.98	90.01	98.84	91.37	90.21	90.78	98.06	90.47	90.19	90.34	98.41
Anomaly Transformer	62.15	66.47	64.24	87.17	63.90	66.32	65.09	87.89	78.68	66.11	71.85	87.84	77.60	76.39	76.98	87.55	82.74	79.56	81.11	97.97
Ours	92.33	99.29	**95.68**	**99.60**	91.84	99.29	**95.42**	**99.60**	90.91	99.51	**95.01**	**99.60**	92.00	99.27	**95.50**	**99.56**	92.00	99.29	**95.51**	**99.58**

Table 5. Result of SMD

Data Number	1				5				10				20				200			
Metric	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
DAGMM	49.45	51.91	50.66	90.25	49.45	51.64	50.52	90.76	51.08	51.63	51.36	90.54	51.56	51.34	51.45	90.56	63.66	66.70	65.15	92.45
MMPCACD	51.32	51.29	51.31	91.07	52.06	51.84	51.94	91.91	54.02	51.18	52.57	91.44	53.68	51.87	52.75	91.17	64.52	65.65	65.08	92.53
VAR	53.30	54.71	53.99	91.73	51.89	54.18	53.02	91.17	52.56	54.70	53.60	91.36	53.33	54.32	53.82	91.53	64.01	68.79	66.31	93.89
LSTM	61.91	59.31	60.58	93.33	61.41	59.85	60.62	93.31	63.27	59.43	61.29	93.74	63.27	59.56	61.37	93.93	67.53	71.77	69.59	95.12
CL-MPPCA	58.75	60.63	59.67	92.13	58.06	60.67	59.34	92.56	59.64	60.98	60.30	92.61	61.22	60.15	60.69	92.55	69.30	70.88	70.08	94.85
ITAD	63.01	62.97	62.98	93.59	62.89	62.54	62.71	93.77	63.47	62.87	63.17	93.40	63.82	62.88	63.35	93.16	68.77	70.23	69.50	95.18
LSTM-VAE	62.15	61.64	61.89	94.27	60.26	61.95	61.10	94.05	62.22	61.57	61.90	94.16	63.32	61.05	62.16	94.29	68.39	73.95	71.05	95.80
BeatGAN	61.72	64.52	63.09	95.19	63.15	64.00	63.57	95.95	63.85	64.60	64.23	95.89	66.86	64.93	65.88	95.36	73.41	75.20	74.30	96.90
OmniAnomaly	65.25	63.48	64.36	97.83	65.86	63.07	64.43	97.32	66.36	63.97	65.14	97.59	75.38	63.22	68.77	97.27	73.41	74.21	73.80	97.82
InterFusion	62.92	64.31	63.60	97.96	63.13	64.00	63.56	97.59	63.92	64.71	64.32	97.80	67.53	64.22	65.83	97.84	58.71	66.06	62.16	98.77
THOC	63.13	65.59	64.33	97.53	62.96	65.32	64.13	97.25	63.39	65.07	64.22	97.43	66.05	65.95	66.00	97.58	73.96	75.35	74.65	98.45
GmVRNN	79.68	83.80	81.68	98.63	78.72	83.75	81.15	98.38	80.45	83.86	82.12	98.12	79.49	83.79	81.59	98.96	79.37	83.17	81.22	98.89
Anomaly Transformer	66.64	63.74	65.16	97.27	66.95	63.15	64.99	97.04	67.48	69.59	68.52	97.64	71.74	69.22	70.46	97.68	75.23	78.89	77.02	98.84
Ours	94.13	89.44	**91.72**	**99.48**	93.44	93.98	**93.71**	**99.71**	93.16	93.98	**93.57**	**99.70**	92.63	93.98	**93.30**	**99.67**	93.30	93.98	**93.64**	**99.70**

Table 6. Result of MSL

Data Number	1				5				10				20				200			
Metric	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
DAGMM	52.13	54.68	53.38	90.42	51.94	54.67	53.26	90.63	55.38	54.41	54.89	90.10	67.77	54.10	60.16	90.23	64.85	66.54	65.68	94.51
MMPCACD	51.36	54.54	52.90	91.96	50.86	54.81	52.76	91.64	57.08	54.99	56.01	91.77	69.58	54.39	61.05	91.11	64.97	66.80	65.87	94.29
VAR	55.07	55.52	55.29	91.82	54.40	55.98	55.18	91.56	56.93	55.79	56.36	91.85	72.65	55.47	62.90	91.66	63.51	68.68	66.00	93.71
LSTM	56.19	57.30	56.73	90.65	55.43	57.76	56.57	90.85	57.85	57.32	57.58	90.86	67.84	57.04	61.98	90.32	67.29	68.18	67.74	93.38
CL-MPPCA	58.51	59.02	58.76	92.01	57.08	59.86	58.43	92.33	57.03	59.59	58.28	92.02	72.79	59.63	65.55	92.29	66.67	70.13	68.36	94.50
ITAD	63.82	61.69	62.74	93.56	63.78	61.67	62.71	93.66	66.26	61.60	63.85	93.03	73.72	61.32	66.95	93.15	66.52	71.35	68.85	96.58
LSTM-VAE	59.76	63.59	61.61	93.18	58.78	63.88	61.23	93.84	60.72	63.75	62.19	93.80	71.40	63.85	67.41	93.62	67.68	70.70	69.16	95.06
BeatGAN	60.57	62.46	61.49	93.63	59.97	62.41	61.18	93.23	61.00	62.46	61.72	93.34	72.44	62.12	66.89	93.92	70.01	72.21	71.10	96.24
OmniAnomaly	62.20	64.10	63.13	94.82	66.33	64.79	65.55	94.90	66.45	64.80	65.61	94.90	73.49	64.54	68.72	94.45	70.04	70.49	70.27	96.64
InterFusion	61.84	65.23	63.50	94.69	61.10	65.76	63.34	94.26	62.51	65.48	63.95	94.20	72.44	65.50	68.79	94.78	70.47	74.52	72.44	96.45
THOC	59.87	64.82	62.24	93.39	60.48	64.58	62.46	93.27	63.13	64.72	63.92	93.10	78.56	64.20	70.66	93.06	73.92	76.58	75.22	97.14
GmVRNN	92.76	94.20	93.47	99.42	92.14	94.71	93.41	99.36	91.49	94.59	93.01	99.33	89.89	94.65	92.21	99.16	90.83	97.68	94.12	99.15
Anomaly Transformer	62.92	64.75	63.82	94.19	68.99	64.59	66.72	94.04	64.66	70.23	67.34	94.10	73.25	69.72	71.45	94.17	77.17	80.68	78.88	97.16
Ours	90.84	99.99	**95.20**	**99.45**	91.71	99.99	**95.67**	**99.51**	92.26	99.99	**95.97**	**99.54**	92.42	99.99	**96.06**	**99.55**	92.54	99.99	**96.12**	**99.56**

Table 7. Result of SMAP

the model (acquiring one data sample need 60s; usually, we want to take about 20 minutes to get the equipment up). For the above reasons, we train *one-for-one* models by first i ($i \in \{1, 5, 10, 20, 200\}$) samples in every new MTS and test the model on the rest of the data. For the *one-for-all* models (GmVRNN and PUAD), the parameters have been updates with the history MTSs first. Table 5 6 7 shown the results of proposed PUAD achieves competitive performance on three databases.

E. More Experiments for Meta Anomaly Detection (Pretraining with History MTSs)

To further prove the advantages of our methods designed for the *one-for-all* setting, we pre-training the *one-for-one* models with historical data and finetune on the first i data in new MTSs. As shown in Table 8, *one-for-one* models can also perform

Dataset	SMD					MSL					SMAP				
Data Number	1	5	10	20	200	1	5	10	20	200	1	5	10	20	200
Metric	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1
DAGMM	72.08	72.55	72.82	73.04	73.39	68.47	67.71	68.28	68.68	69.17	70.81	71.04	71.66	71.94	72.07
MMPCACD	73.15	72.90	73.90	73.31	73.11	69.99	69.68	70.34	70.57	70.04	72.95	73.01	73.05	73.25	73.44
VAR	74.54	75.01	74.11	74.78	74.10	70.88	71.14	71.09	71.67	71.96	71.87	72.16	72.67	72.38	72.56
LSTM	73.77	73.61	73.78	73.92	73.70	71.79	71.32	72.01	72.17	71.94	74.07	74.57	74.44	73.76	74.81
CL-MPPCA	75.23	75.18	74.92	75.31	75.87	73.57	72.99	73.51	73.36	73.12	76.50	75.96	76.15	76.74	76.64
ITAD	77.03	76.85	77.76	77.67	76.99	77.08	77.00	77.49	77.18	77.64	78.80	78.89	79.31	78.82	79.34
LSTM-VAE	80.65	80.73	80.13	81.68	81.51	76.19	76.31	76.47	76.20	77.02	77.92	77.83	77.96	78.19	78.50
BeatGAN	82.90	83.02	83.08	82.93	82.57	79.67	79.66	80.56	80.21	80.84	81.66	81.61	82.12	82.50	82.36
Men-SkipAE	83.26	83.32	83.85	83.23	83.59	81.29	80.99	81.71	81.78	81.85	84.12	84.43	84.01	84.80	84.54
MemAE	83.31	83.22	83.34	83.40	83.54	80.32	80.55	80.47	80.19	81.10	82.32	82.27	82.52	82.75	82.78
TSMAE	84.94	85.60	85.41	85.71	85.24	82.06	82.12	82.40	82.65	82.72	84.62	84.11	84.58	84.19	84.22
OmniAnomaly	85.40	85.42	85.78	86.19	85.83	81.12	80.58	82.13	81.86	81.64	84.28	84.71	84.61	84.97	85.49
InterFusion	84.60	84.73	84.74	84.85	85.09	82.09	81.87	82.28	82.46	83.00	86.70	86.80	87.26	87.27	87.12
THOC	86.01	86.05	86.24	86.62	86.54	81.56	81.55	81.38	82.73	82.86	88.95	89.28	88.92	88.54	88.95
PGDLE	86.28	86.46	86.66	86.88	86.18	81.73	81.48	81.96	81.56	82.38	89.38	89.24	89.28	89.04	89.34
SummerNet	90.22	89.90	90.16	90.71	90.55	82.79	82.70	82.43	82.80	82.74	91.10	91.29	91.18	91.81	91.55
GmVRNN	91.03	89.51	90.01	90.78	90.34	81.68	81.15	82.12	81.59	81.22	93.47	93.41	93.01	92.21	94.12
Anomaly Transformer	91.58	91.92	91.46	91.78	91.69	80.65	82.87	82.15	83.55	84.01	93.32	93.89	93.51	94.01	94.11
Ours	**95.68**	**95.42**	**95.01**	**95.50**	**95.51**	**91.72**	**93.71**	**93.57**	**93.30**	**93.64**	**95.20**	**95.67**	**95.97**	**96.06**	**96.12**

Table 8. Quantitative results of different methods for *adapting to new MTSs* on three public datasets. Models are pre-trained on history MTSs first, then test on the new MTSs.

better because of the history MTSs involved, but their performance is still much worse than PUAD in this setting.

F. Quantitative Comparison

Dataset	SMD			MSL			PSM			SMAP			DND		
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
OC-SVM	44.34	76.72	56.19	59.78	86.87	70.82	62.75	80.89	70.67	53.85	59.07	56.34	68.23	70.36	69.28
IsolationForest	42.31	73.29	53.64	53.94	86.54	66.45	76.09	92.45	83.48	52.39	59.07	55.53	69.63	74.21	71.85
LOF	56.34	39.86	46.68	47.72	85.25	61.18	57.89	90.49	70.61	58.93	56.33	57.60	70.24	74.66	72.38
Deep-SVDD	78.54	79.67	79.10	91.92	76.63	83.58	95.41	86.49	90.73	89.93	56.02	69.04	73.09	79.02	75.94
DAGMM	67.30	49.89	57.30	89.60	63.93	74.62	93.49	70.03	80.08	86.45	56.73	68.51	72.72	77.67	75.11
MMPCACD	71.20	79.28	75.02	81.42	61.31	69.95	76.26	78.35	77.29	88.61	75.84	81.73	71.52	75.75	73.57
VAR	78.35	70.26	74.08	74.68	81.42	77.90	90.71	83.82	87.13	81.38	53.88	64.83	72.92	77.72	75.24
LSTM	78.55	85.28	81.78	85.45	82.50	83.95	76.93	89.64	82.80	89.41	78.13	83.39	73.52	79.67	76.47
CL-MPPCA	82.36	76.07	79.09	73.71	88.54	80.44	56.02	99.93	71.80	86.13	63.16	72.88	73.12	78.71	75.81
ITAD	86.22	73.71	79.48	69.44	84.09	76.07	72.80	64.02	68.13	82.42	66.89	73.85	73.87	75.43	74.64
LSTM-VAE	75.76	90.08	82.30	85.49	79.94	82.62	73.62	89.92	80.96	92.20	67.75	78.10	75.10	79.05	77.02
BeatGAN	72.90	84.09	78.10	89.75	85.42	87.53	90.30	93.84	92.04	92.38	55.85	69.61	76.64	80.18	78.37
Men-SkipAE	75.77	84.22	79.77	90.64	86.68	88.61	92.05	93.10	92.57	85.77	89.66	87.67	78.72	83.50	81.03
MemAE	74.09	83.27	78.41	90.16	84.97	87.48	91.86	92.49	92.17	85.63	67.39	75.42	77.32	82.67	79.90
TSMAE	83.47	87.32	85.35	88.93	86.84	87.87	82.73	79.17	80.91	91.29	81.41	86.06	80.33	82.46	81.38
OmniAnomaly	83.68	86.82	85.22	89.02	86.37	87.67	88.39	74.46	80.83	92.49	81.99	86.92	79.47	82.37	80.90
InterFusion	87.02	85.43	86.22	81.28	92.70	86.62	83.61	83.45	83.52	89.77	88.52	89.14	78.21	85.34	81.62
THOC	79.76	90.95	84.99	88.45	90.97	89.69	88.14	90.99	89.54	92.06	89.34	90.68	80.97	83.45	82.19
PGDLE	80.95	89.56	85.03	89.48	91.32	90.39	89.84	90.91	90.37	93.20	90.86	92.02	81.58	84.71	83.11
SummerNet	93.23	92.17	92.69	91.61	92.56	92.08	95.59	93.43	94.49	92.53	91.70	92.11	82.21	81.60	81.90
GmVRNN	96.07	91.23	93.56	90.81	92.10	91.41	95.62	98.36	96.97	96.51	94.54	95.51	83.80	87.67	85.58
Anomaly Transformer	89.40	95.45	92.33	92.09	95.15	93.59	96.91	98.90	97.89	94.13	99.40	96.69	82.13	86.31	84.16
TranAD	92.62	99.74	96.05	90.38	99.99	94.94	95.36	98.65	96.97	80.43	99.99	89.15	82.59	86.60	84.54
Ours	93.75	98.71	**96.16**	91.91	98.39	**95.04**	97.87	98.42	**98.14**	94.49	99.07	**96.72**	84.77	88.57	**86.62**

Table 9. Quantitative results for different methods on five public datasets and one real-world datasets. The P, R and F1 represent the precision, recall and F1-score (as %) respectively. F1-score is the harmonic mean of precision and recall. For these three metrics, a higher value indicates a better performance.

To evaluate the performance of PUAD in the typical anomaly detection setting, we consider our model on five datasets with multiple competitive baselines. Three metrics is used to evaluate the proposed model. The P, R and F1 represent the precision, recall and F1-score (as %) respectively. As show in Table 9, our proposed model goes beyond the reconstruction models such as OmniAnomaly, the density-estimation methods such as MPPCACD, and the well performance deep learning model Anomaly Transformer. The results in Table 9 are persuasive for the advantage of association learning in time series anomaly detection. PUAD has the highest performance on all four datasets.

G. Evidence Lower Bound (ELBO) of Log Likelihood

Use the Bayes' theorem for $p(x)$, we can add the collocation terms $q(\theta_1 \mid \theta_0', x)$, $p(\theta_1 \mid \theta_0')$, $q(\theta_0 \mid x)$ and $p(\theta_0)$ as

$$\log p(x) = \mathbf{E}\left[\log \frac{p(x \mid \theta_1)p(\theta_1)}{p(\theta_1 \mid x)} \frac{q(\theta_1 \mid \theta_0', x)}{q(\theta_1 \mid \theta_0', x)} \frac{p(\theta_1 \mid \theta_0')}{p(\theta_1 \mid \theta_0')} \frac{p(\theta_0)}{p(\theta_0)} \frac{q(\theta_0 \mid x)}{q(\theta_0 \mid x)}\right] \tag{1}$$

Organizing Eq. 1 can conclude:

$$\begin{aligned} \log p(x) = & \mathbf{E}\left[\log p(x \mid \theta_1)\right] - \mathbf{E}\left[\log \frac{q(\theta_1 \mid \theta_0', x)}{p(\theta_1 \mid \theta_0')}\right] \\ & - \mathbf{E}\left[\log \frac{q(\theta_0 \mid x)}{p(\theta_0)}\right] + \mathbf{E}\left[\log \frac{q(\theta_0 \mid x)}{p(\theta_0)}\right] \\ & + \mathbf{E}\left[\log \frac{q(\theta_1 \mid \theta_0', x)}{p(\theta_1 \mid \theta_0')}\right] + \mathbf{E}\left[\log \frac{p(\theta_1)}{p(x \mid \theta_1)}\right] \end{aligned} \tag{2}$$

Then, with the KillbackLeibler (KL) Divergence equation $\mathbf{D}_{KL}\left(p \| q\right) = \mathbf{E}[\log(p/q)]$, Eq. 2 can be further reexpresed as:

$$\begin{aligned} \log p(x) = & \mathbf{E}\left[\log p(x \mid \theta_1)\right] - \mathbf{D}_{KL}\left(q(\theta_1 \mid \theta_0', x) \| p(\theta_1 \mid \theta_0')\right) \\ & - \mathbf{D}_{KL}\left(q(\theta_0 \mid x) \| p(\theta_0)\right) + \mathbf{D}_{KL}\left(q(\theta_0 \mid x) \| p(\theta_0)\right) \\ & + \mathbf{D}_{KL}\left(q(\theta_1 \mid \theta_0', x) \| p(\theta_1 \mid \theta_0')\right) + \mathbf{D}_{KL}\left(p(\theta_1) \| p(x \mid \theta_1)\right) \end{aligned} \tag{3}$$

Using Jensen's inequality, ELBO can obtain afterward:

$$\begin{aligned} \log p(x) \geq & \mathbf{E}[\log p(x \mid \theta_1)] - \mathbf{D}_{KL}\left(q(\theta_0 \mid x) \| p(\theta_0)\right) \\ & - \mathbf{D}_{KL}\left(q(\theta_1 \mid \theta_0', x) \mid x) \| p(\theta_1 \mid \theta_0')\right) \end{aligned} \tag{4}$$

Inspired by beta-VAE (Higgins et al., 2016), we introduce three hyperparameters $\rho_1 > 0$, $\rho_2 > 0$ and $\rho_3 > 0$, then adding the KL loss gradually with $\rho 1$, ρ_2 and ρ_3 increasing from 0 to 1 during the first N training epochs.

$$\begin{aligned} \mathcal{L} = & \mathbf{E}_{q(\theta_0), q(\theta_1)}\left[\log p(x \mid \theta_1)\right] \\ & - \rho_1 \mathbf{D}_{KL}\left(q(\theta_0 \mid x) \| p(\theta_0)\right) \\ & - \rho_2 \mathbf{D}_{KL}\left(q(\theta_1 \mid \theta_0', x) \mid x) \| p(\theta_1 \mid \theta_0')\right) \end{aligned} \tag{5}$$

H. More In-Depth Discussion on PUAD

H.1. The advantages of the proposed method over the existing anomaly detection methods

Leveraging meta-learned transferable prototypes, PUAD can achieve high model adaptation capacity for new MTS, which is the biggest advantage of our model has over existing anomaly detection methods. This adaptation is achieved by defining global and local prototypes. Specifically, the global prototypes in PUAD is encouraged to capture the statistical temporal dependency shared by multiple MTS, which is similar to the transfer patterns useful for all related tasks in meta-learning, thus to enable PUAD to achieve high model adaption capacity. In addition, we also introduce the local prototypes for newly arrived MTSs to capture specific information for new MTSs to enhance the adaption capacity. We note that the existing memory-augmented deep Autoencoder (Yan et al., 2023; Gong et al., 2019; Gao et al., 2022) have difficulty in modeling new MTS.

In addition to model adaptation capacity, modeling different MTSs with a single group of parameters is another advantage of PUAD. Specifically, previous methods have difficulty in modeling multiple MTSs for the diverse statistic characteristics within them, so they always characterize different MTS by training different specific groups of parameters, which consumes huge computing and storage resources. PUAD can model different MTSs with a single group of parameters for considering the diverse normal patterns within multiple MTSs as a group of prototypes, and then incorporating the prototypes into the developed powerful probabilistic dynamical generative module, enhancing its generative capacity for multiple MTSs with diverse statistic characteristics, thus to achieve superior unsupervised anomaly detection based on reconstruction. As shown in Table 2 and Fig. 3 in our manuscript, PUAD can outperform SOTA baselines with much less model parameters.

Our proposed PUDA can be regarded as a mem-autoencoder-based approach, while a group of prototypes are similar as the memories. Compared with previous mem-autoencoder-based approaches (Yan et al., 2023; Gong et al., 2019; Gao et al., 2022), our defined prototypes can capture the statistical temporal dependency of normal patterns centers within multiple MTSs, enhancing its capacity in model adaptation and diversity consideration. In addition, different from previous methods, we introduce a prototype-oriented OT module in PUAD, which uses OT distance between distributions to guide the learning process of prototypes, while the existing methods always use cosine similarity and softmax operation to learn memory vectors. There are two main advantages of using OT distance in our model. Firstly, introducing the OT lose can ensure the representative of prototypes as in (Guo et al., 2021). Secondly, using a simple cosine similarity (Yan et al., 2023; Gong et al., 2019; Gao et al., 2022) to balance the global and local information stored in prototypes is hard. The distance between query vectors and memory vectors is not enough to achieve this balance. The OT distance provides a transport plan for how to transport query vectors to both global and local prototypes, thus to balance the importance of two kinds of prototypes.

To better illustrate the efficiency of PUAD, we add more baselines about Mem-autoencoder-based approaches (Yan et al., 2023; Gong et al., 2019; Gao et al., 2022) , and perform experiment to compare them with PUAD (as show in Table 8 and Table 9). On both traditional and meta anomaly detection tasks, PUAD significantly outperforms existing memory-based methods, illustrating its advantage over the existing well-established anomaly detection methods for MTS.

H.2. The primary advantage of PUAD in actual engineering requirements

Modern information technology (IT) operations generate an enormous amount of high-dimensional sensor data that must be continuously monitored. Anomaly detection for MTS is a fundamental scenario in IT operations, as it is critical for managing service quality of industry devices or internet infrastructures. With the development of the IT field, a large number of anomaly detection models for MTS need to be deployed on servers for different purposes every day. Unlike anomaly detection in computer vision, a key challenge of UAD for MTS coming from a large-scale system is that each device has distinct normal mode. For example, there is a clear distributed difference between the MTSs from the server used for video websites and shopping websites. To address this problem, previous methods always train different models for different MTS. However, according to the statistics, global server shipments of about 13.539 million units in 2021, and these servers will be deployed to multiple domains. Training new models on such a large server scale can be very challenging and wasteful. The meta anomaly detection and *"one-for-all"* problems are both formulated towards this real world scenario. Compared with previous methods, the proposed PUAD has the following advantages for practical application scenarios:

PUAD is a *"one-for-all"* model that considers the diverse normal dynamic patterns within multiple MTSs as a group of global prototypes. It learns these prototype memories with the proposed novel prototype-oriented OT module. This eliminates the potential need to train individual model parameters for different MTSs, significantly reducing deployment costs.

In real-word applications, when we deploy anomaly detection model to new scenarios, it is difficult to get enough data to train the model at limited time. We formulate meta anomaly detection task for such common situations, and enhance the capacity of PUAD to perform well on this task.

H.3. The detailed description of the PUAD framework

Anomaly detection on multivariate time series is defined as a problem that determines whether an observation from a certain task and at a certain time is anomalous or not. The complete framework for unsupervised anomaly detection for multivariate time series contains three key modules. The first module pre-processes the original multivariate time series data so that they can be used by the learning model for training. Specifically, the normalization and sliding time window approaches are adopted in this work. In the representation module, we propose a PUAD to learn the complex structural and dynamic characteristics within multivariate time series. Finally, we apply the reconstruction probability and as the anomaly score to

determine whether an observed variable is anomalous or not. An observation will be classified as anomalous if anomaly score is below a specific threshold. From a practical point of view, we use the Peaks-Over-Threshold approach to help select threshold. In our case, the lower anomaly scores are more likely considered to be extreme values since the lower anomaly score.