POND: Multi-Source Time Series Domain Adaptation with Information-Aware Prompt Tuning

Junxiang Wang[*]
NEC Labs America
Princeton, New Jersey, USA

Guangji Bai[*]
Emory University
Atlanta, Georgia, USA

Wei Cheng
NEC Labs America
Princeton, New Jersey, USA

Zhengzhang Chen
NEC Labs America
Princeton, New Jersey, USA

Liang Zhao
Emory University
Atlanta, Georgia, USA

Haifeng Chen
NEC Labs America
Princeton, New Jersey, USA

ABSTRACT

Time series domain adaptation stands as a pivotal and intricate challenge with diverse applications, including but not limited to human activity recognition, sleep stage classification, and machine fault diagnosis. Despite the numerous domain adaptation techniques proposed to tackle this complex problem, they primarily focus on domain adaptation from a single source domain. Yet, it is more crucial to investigate domain adaptation from multiple domains due to the potential for greater improvements. To address this, three important challenges need to be overcome: 1). The lack of exploration to utilize domain-specific information for domain adaptation, 2). The difficulty to learn domain-specific information that changes over time, and 3). The difficulty to evaluate learned domain-specific information. In order to tackle these challenges simultaneously, in this paper, we introduce PrOmpt-based domaiN Discrimination (POND), the first framework to utilize prompts for time series domain adaptation. Specifically, to address Challenge 1, we extend the idea of prompt tuning to time series analysis and learn prompts to capture common and domain-specific information from all source domains. To handle Challenge 2, we introduce a conditional module for each source domain to generate prompts from time series input data. For Challenge 3, we propose two criteria to select good prompts, which are used to choose the most suitable source domain for domain adaptation. The efficacy and robustness of our proposed POND model are extensively validated through experiments across 50 scenarios encompassing four datasets. Experimental results demonstrate that our proposed POND model outperforms all state-of-the-art comparison methods by up to 66% on the F1-score.

CCS CONCEPTS

• **Mathematics of computing** → **Time series analysis**; • **Computing methodologies** → *Transfer learning*; Neural networks; Supervised learning.

KEYWORDS

Time Series; Domain Adaptation; Prompt Tuning; Information Bottleneck

ACM Reference Format:
Junxiang Wang, Guangji Bai, Wei Cheng, Zhengzhang Chen, Liang Zhao, and Haifeng Chen. 2024. POND: Multi-Source Time Series Domain Adaptation with Information-Aware Prompt Tuning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, Barcelona, Spain, 12 pages. https://doi.org/10.1145/3637528.3671721

1 INTRODUCTION

Due to the prevalence of time series sensor data, time series analysis has found applications in various real-world scenarios, including human activity recognition [1], sleep stage classification [48], and machine fault diagnosis [18, 38, 39]. In these applications, time series data are measured under different subjects, operating conditions, or sensor configurations (*i.e.*, domains). In other words, time series analysis should be conducted across different domains. Unfortunately, the labels of time series data are difficult to collect due to the expensive costs of the labeling process [42]. To mitigate labeling costs, researchers aim to leverage labeled data from some domains (*i.e.*, source domains) to infer labels for unlabeled data in other domains (*i.e.*, target domains) [40], which is defined as a time series domain adaptation problem. For example, the goal of the transponder fault diagnosis problem is to detect the working statuses of transponders (*i.e.*, normal or abnormal) based on fiber-optic signals. In this problem, the model is trained under certain working modes (*e.g.*, single mode) using labeled time series data, and then this trained model is applied to other working modes (*e.g.*, multimode).

However, the time series domain adaptation problem is highly challenging due to complex dynamic time series patterns, distribution shift (*i.e.*, different distributions of inputs among different domains), and possible label shift (*i.e.*, different distributions of labels among different domains) [2, 5, 12]. These challenges have been extensively investigated by researchers, leading to the proposal of various methods to address the domain gap, such as kernel matching [22], context information alignment [17], and temporal-spectral fusion [45]. Most existing methods, however, primarily focus on domain adaptation from a single source domain. Yet, it is more crucial to investigate it from multiple sources. This is because the more source domains are utilized, the greater potential improvements it can achieve. For instance, the collection of labeled

[*] Both authors contributed equally to this research.

KDD '24, August 25–29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671721

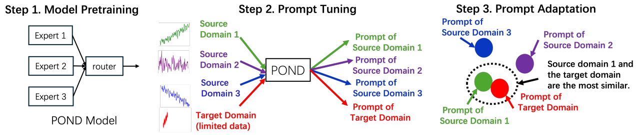

Step 1. Model Pretraining
Expert 1
Expert 2
Expert 3
router
POND Model
Step 2. Prompt Tuning
Source Domain 1
Source Domain 2
Source Domain 3
Target Domain (limited data)
POND
Prompt of Source Domain 1
Prompt of Source Domain 2
Prompt of Source Domain 3
Prompt of Target Domain
Step 3. Prompt Adaptation
Prompt of Source Domain 3
Prompt of Source Domain 2
Source domain 1 and the target domain are the most similar.
Prompt of Target Domain
Prompt of Source Domain 1

Figure 1: Pipeline of our proposed POND model: Step 1 pretrains the proposed POND model; Step 2 learns prompts of all source domains and the target domain; Step 3 utilizes learned prompts to select the most similar source domain to the target domain for domain adaptation.

signal data from more modes facilitates a better understanding of transponder statuses. Despite the importance of the multi-source domain adaptation problem, it is rarely explored in previous literature and requires attention and extensive investigations from researchers.

In order to effectively handle the multi-source time series domain adaptation problem, three important challenges need to be overcome: **1. The lack of exploration to utilize domain-specific information for domain adaptation.** Existing domain adaptation methods primarily focus on learning a common feature extractor to encode time series inputs from different source domains into domain-invariant representations, and then apply this feature extractor to the target domain [15, 20, 23, 27, 42]. While this strategy has its rationale, it often overlooks domain-specific information (*i.e.*, information unique to a specific time series domain), such as global trends, local trends, and temporal patterns. Such domain-specific information is valuable to evaluate which source domains are more suitable for adaptation to the target domain. **2. The difficulty to learn domain-specific information that changes over time.** While it is important to capture domain-specific information for better domain adaptation, such information can be dynamically changing, which is extremely difficult to capture. In the example of the transponder fault diagnosis problem, different domains generate different distributions of fiber-optic signals, which are important domain-specific information to capture. However, such distributions can be shifted drastically when the transponder suddenly suffers from a failure. **3. The difficulty to evaluate learned domain-specific information.** Not only is learning domain-specific information difficult, but it is also challenging to evaluate learned domain-specific information. In other words, it is unclear whether learned domain-specific information accurately reflects the true one. This ambiguity arises because domain-specific information is often associated with unique but inexplicable underlying patterns. Unlike images and languages with human-recognizable features, such time series patterns are difficult for humans to understand [24]. Consequently, it becomes challenging, if not impossible, for humans to evaluate whether learned domain-specific information matches such time series patterns.

In order to tackle these three challenges simultaneously, we propose PrOmpt-based domaiN Discrimination (POND), the first framework to utilize prompts for time series domain adaptation to our knowledge. Its pipeline is shown in Figure 1, which consists of three steps: model pertaining, prompt tuning, and prompt adaptation. Specifically, to address Challenge 1, we extend the idea of prompt tuning to time series analysis and learn prompts to capture common and domain-specific information. To handle Challenge 2, we introduce a conditional module for each source domain to generate prompts from time series input data. For Challenge 3, we propose two criteria to choose good prompts, which are used to select the most suitable source domain for domain adaptation (*i.e.*, prompt adaptation). Our contributions can be summarized as follows:

- **Propose a flexible prompt generator to learn domain-specific information.** We extend the idea of prompt tuning to time series analysis to capture information specific to source domains. However, traditional prompts have limited flexibility in learning domain-specific information that evolves over time. To address this limitation, we introduce a conditional module that generates prompts parameterized by a neural network to capture domain-specific information. Theoretical analysis also demonstrates the superiority of our proposed prompt generator over traditional prompt tuning.
- **Develop two criteria for selecting good prompts.** We propose two criteria, fidelity and distinction, to ensure that prompts accurately capture domain-specific information from all source domains. Fidelity is achieved by maximizing the mutual information between prompts and labels, while distinction is achieved by minimizing the mutual information between prompts from different source domains. Theoretical guarantees establish that our generated prompts maintain fidelity and introduce new information.
- **Present an efficient algorithm with a robust architecture.** We introduce a simple yet effective optimization algorithm based on meta-learning to efficiently learn the objective. Additionally, we leverage the Mixture of Experts (MoE) technique to enhance the robustness of our proposed POND model.
- **Conduct comprehensive experiments on multiple benchmark datasets.** Extensive experiments across 50 scenarios on four benchmark datasets demonstrate the effectiveness and robustness of our proposed POND model. Experimental results indicate that our proposed POND model outperforms all state-of-the-art comparison methods by up to 66% on the F1-score.

2 RELATED WORK

Previous research related to this study can be categorized into two main areas: time series domain adaptation and Large Language Models (LLMs) for time series.

Time Series Domain Adaptation: Works in this domain can be classified into Unsupervised Domain Adaptation (UDA) and supervised methods.

UDA is a common approach, particularly beneficial as it does not rely on labels in the target domain. For example, Liu and Xue introduced the Adversarial Spectral Kernel Matching (AdvSKM) approach, employing a specialized hybrid spectral kernel network to redefine the Maximum Mean Discrepancy (MMD) metric [22]. Lai et al. aligned context information between different time series domains using a Markov decision process formulation and employed deep reinforcement learning for anomaly detection [17]. He et al. addressed feature and label shifts between the source and target domains using temporal and frequency features [12]. Other notable approaches include autoregressive models [30], sparse associative structure alignment [4], variational methods [20, 29], contrastive learning [47], and temporal-spectral fusion [45].

In addition to UDA, other methods transfer time series knowledge in a supervised manner. For instance, Jin et al. proposed an attention-based shared module to learn common latent features, incorporating a domain discriminator retaining domain-specific features across multiple domains [15]. Wilson et al. leveraged target-domain label distributions to enhance model performance with benefits from multi-source time series data [42]. However, to our knowledge, all existing time series domain adaptation methods neglect domain-specific information such as unique temporal patterns, which could potentially be utilized for better domain adaptation.

LLMs for Time Series: Large Language Models (LLMs) have shown excellent performance in various Natural Language Processing (NLP) tasks such as natural language inference, question answering, and named entity recognition [50]. Recent research has extended LLMs to address time series problems, generally falling into two classes: prompt tuning and fine-tuning.

In prompt tuning methods, pretrained LLMs use prompts (*i.e.*, a sequence of tokens prepended to the time series input) to learn specific downstream tasks. For example, Xue and Salim proposed PromptCast, a novel approach that transforms numerical input and output into prompts and frames the time series forecasting task in a sentence-to-sentence manner [44]. Cao et al. presented the TEMPO framework, which decomposed complex interactions between trend, seasonal, and residual components, introducing selection-based prompts to facilitate distribution adaptation in non-stationary time series [6]. Jin et al. proposed the TIME-LLM framework, reprogramming the input time series with text prototypes before feeding it into a frozen LLM to align the two modalities, with Prompt-as-Prefix (PaP) introduced to enrich the input context and guide the transformation of the reprogrammed input [13]. LLMTime highlighted the efficacy of LLMs as zero-shot learners by encoding numbers into texts as prompts and sampling possible extrapolations as prompt completions [11]. Sun et al. proposed the TEST model, training an encoder to embed time series tokens with contrastive learning and aligning text prototypes with time series, utilizing prompts to adapt LLMs to different time series tasks [36].

In contrast, fine-tuning is the other type of method to adapt LLMs to time series, adjusting some components while keeping others frozen. For example, Zhou et al. presented the OFA framework, where only the embedding and normalization layers of LLMs were fine-tuned, while self-attention and feed-forward layers remained frozen [51]. Chang et al. proposed the Llm4ts framework, fine-tuning in two stages: first, supervised fine-tuning to orient the LLM towards time series data, followed by task-specific downstream fine-tuning [7]. For more information, please refer to the recent survey paper by Jin et al. [14]. While these methods transfer knowledge from LLMs to the time series domain, they do not address the time series domain adaptation problem, where knowledge from the source time series domain, rather than text, is transferred to the target domain.

3 PROBLEM SETUP

In this section, we mathematically formulate the multi-source time series domain adaptation problem. Important notations are shown in Table 1. Given M source time series domains $S_i (i = 1, \cdots, M)$ and a target domain T, their j-th time series inputs are denoted as $X_j^{(S_i)} \sim p(X|Y_j^{(S_i)})$ and $X_j^{(T)} \sim p(X|Y_j^{(T)})$, respectively, where $Y_j^{(S_i)}$ and $Y_j^{(T)}$ are corresponding labels of $X_j^{(S_i)}$ and $X_j^{(T)}$, respectively. Here, $X_j^{(S_i)}, X_j^{(T)} \in \mathbb{R}^{n \times L}$, where n is the number of channels and L is the sequence length. The labels $Y_j^{(S_i)}, Y_j^{(T)} \in C = \{c_1, c_2, \cdots, c_K\}$, where $c_i (i = 1, \cdots, |C|)$ represents a label class, and the number of classes is $|C|$. $Y^{(S_i)} = \{Y_j^{(S_i)}\}$ and $Y^{(T)} = \{Y_j^{(T)}\}$ are the label sets for the source domain S_i and the target domain T, respectively. Sets $X^{(S_i)} = \{X_j^{(S_i)}\}$ and $X^{(T)} = \{X_j^{(T)}\}$ represent the input sets for the source domain S_i and the target domain T, respectively. We assume that the labeled time series of all source domains $S_i (i = 1, \cdots, M)$ are abundant, but the labeled time series are limited in the target domain T. Then the multi-source time series domain adaptation problem is formulated as follows:

Table 1: Important notations and Descriptions.

Notations	Descriptions
S_i	The i-th source domain
T	Target domain
C	Class set
$(X_j^{(S_i)}, Y_j^{(S_i)})$	The j-th time series pair for S_i
$(X_j^{(T)}, Y_j^{(T)})$	The j-th time series pair for T
$Y^{(S_i)}, Y^{(T)}$	Label sets for S_i and T
P	Common prompt
$\Delta P^{(S_i)}$	Domain-level prompt for S_i
$\Delta P_j^{(S_i)}$	Instance-level prompt generated by $X_j^{(S_i)}$ for S_i

Problem Formulation: Given the time series input sets $X^{(S_i)}$ and label sets $Y^{(S_i)} (i = 1, 2, \ldots, M)$ of M source domains, and the time series input set $X^{(T)}$ of the target domain T, the goal of the problem is to predict the label set $Y^{(T)}$ by learning the mapping F:

$$F : X_i^{(T)} \rightarrow Y_i^{(T)}$$

Our problem formulation is very flexible: the time series input can be either univariate (*i.e.*, $N = 1$) or multivariate (*i.e.*, $N > 1$); the time series domain adaptation can be from a single source (*i.e.*, $M = 1$) or multiple sources (*i.e.*, $M > 1$); the classification problem can be either binary (*i.e.*, $K = 2$) or multi-class (*i.e.*, $K > 2$).

4 PROMPT-BASED DOMAIN DISCRIMINATION

In this section, we present our POND model to address the multi-source time series domain adaptation problem.

4.1 The Flexible Prompt Generator

The goal of this section is to explore methods for learning information that changes over time from different source domains for domain adaptation (*i.e.*, tackling Challenges 1 and 2). Most existing papers propose various strategies to extract domain-invariant representations from all source domains by making different domains indistinguishable [15, 20, 27, 42, 49]. However, this idea may discard domain-specific information from multiple source domains, which indicates which source domain is most similar to the target domain. To address this, a natural solution is to directly learn domain-specific information from the labeled time series pair $(X_j^{(S_i)}, Y_j^{(S_i)})$. This motivates us to utilize prompt tuning to learn domain-specific information, which was first introduced by the NLP community and demonstrated impressive success in many NLP tasks [3, 19, 21]. Compared with other domain adaptation techniques, prompt tuning has three advantages: firstly, prompts are adjusted via gradients by labeled data from multiple source domains, which offer domain-specific information; secondly, prompt tuning leverages small amounts of labeled data effectively for adaptation, which is suitable for the target domain with limited labeled data [19]; thirdly, prompts can be utilized as a heuristic to select the most similar source domain to the target domain for adaptation.

The prompt, which is extended from NLP to time series, is defined as a learnable vector that prepends to the time series input to learn domain-specific information by the labeled pair $(X_j^{(S_i)}, Y_j^{(S_i)})$. Mathematically, let $P^{(S_i)} \in \mathbb{R}^{n\times m}$ be the prompt of the source domain S_i, where m is the prompt length. Then, for the j-th time series input $X_j^{(S_i)}$, any time series model takes $[P^{(S_i)}, X_j^{(S_i)}]$ (*i.e.*, the concatenation of $P^{(S_i)}$ and $X_j^{(S_i)}$) as its model input. We decompose $P^{(S_i)}$ into two components:

$$P^{(S_i)} = P + \Delta P^{(S_i)}$$

where $P \in \mathbb{R}^{n\times m}$ is a common prompt to learn the common characteristics of all source domains, which can also be directly applied to the target domain T, and $\Delta P^{(S_i)} \in \mathbb{R}^{n\times m}$ is a prompt to learn domain-specific information (*i.e.*, information unique to the source domain S_i), which will be utilized to select the most similar source domain to the target domain T.

While the domain-specific prompt $\Delta P^{(S_i)}$ is potentially effective to learn domain-specific information about the source domain S_i (*i.e.*, address Challenge 1), it cannot directly address Challenge 2. This is because $\Delta P^{(S_i)}$ is time-independent and has little freedom to capture time-dependent domain-specific information (*e.g.*, distribution shifts of fiber-optic signals). To tackle this, instead of using a fixed prompt, we learn such domain-specific information by prompts generated from the time series input. This is because the time series input usually contains rich time-dependent information (*e.g.*, time series distributions and trends). Specifically, we introduce a conditional module $g^{(S_i)}$, parameterized by a neural network, to generate instance-level prompts based on time series instances:

$$\Delta P_j^{(S_i)} = g^{(S_i)}(X_j^{(S_i)}; \zeta) \in \mathbb{R}^{m\times n}$$

where $\Delta P_j^{(S_i)}$ is the instance-level prompt generated by the time series input $X_j^{(S_i)}$ and a random variable ζ, and the domain-level prompt $\Delta P^{(S_i)}$ is the aggregation of all instance-level prompts $\Delta P_j^{(S_i)}$ (*e.g.*, $\Delta P^{(S_i)} = \frac{1}{|S_i|}\sum_{j=1}^{|S_i|} \Delta P_j^{(S_i)}$). For any time series input $X_j^{(S_i)}$, its corresponding prompt is formulated as $P + \Delta P_j^{(S_i)}$.

Our proposed prompt generator $g^{(S_i)}$ conditionally generates instance-level prompts for specific time series inputs, which intuitively has more freedom of expression to learn domain-specific information than the traditional prompt tuning. More theoretical investigations are provided to illustrate the power of the common prompt P and the prompt generator $g^{(S_i)}$ in Section 4.4.

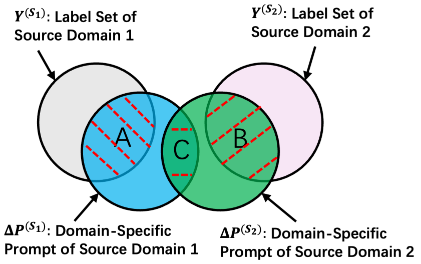

$Y^{(S_1)}$: Label Set of Source Domain 1
$Y^{(S_2)}$: Label Set of Source Domain 2
A
C
B
$\Delta P^{(S_1)}$: Domain-Specific Prompt of Source Domain 1
$\Delta P^{(S_2)}$: Domain-Specific Prompt of Source Domain 2

Figure 2: Illustration of two criteria: high fidelity and high distinction. Fidelity and distinction are represented as areas of $A + B$ and C, respectively.

4.2 Two Important Criteria for Good Prompts

In the previous section, we extended prompt tuning to capture information on specific time series domains. While prompts are easy to recognize in computer vision and natural language fields, the learned prompts of time series data are not recognizable to humans, making it hard, if not impossible, to evaluate whether prompts are good enough to learn information for time series data. For example, a hard prompt consists of natural language that clearly describes the task at hand, explicitly asks the model for some result or action, and makes it easy to understand why the prompt elicited such behavior from the model [19]. In contrast, the learned prompts of specific time series domains are visualized as extra time segments, which are difficult to understand by humans. Moreover, there is a lack of exploration on what constitutes a good prompt that captures domain-specific information without human-engineering priors. From our perspective, ideal prompts to capture domain-specific information should maintain high fidelity and high distinction, as illustrated in Figure 2: high fidelity suggests large overlaps between the learned domain-specific prompts and label information (*i.e.*, large $A + B$ in Figure 2), and high distinction implies small overlaps among domain-specific prompts of different source domains (*i.e.*, small C in Figure 2). They are introduced in details as follows:

High Fidelity. One important criterion for the prompt generator $g^{(S_i)}$ is fidelity (*i.e.*, the generated prompt $\Delta P_j^{(S_i)}$ preserves the domain-specific information of the source domain S_i). Motivated by the theory of information bottleneck [37], high fidelity is defined as the large mutual information between $\Delta P_j^{(S_i)}$ and $Y_j^{(S_i)}$, which should be maximized:

$$\max \sum_{i=1}^{M}\sum_{j=1}^{|S_i|} MI(\Delta P_j^{(S_i)}, Y_j^{(S_i)}), \qquad (1)$$

where $MI(\bullet, \bullet)$ denotes the operator of mutual information. Based on the definition of mutual information, we have:

$$MI(\Delta P_j^{(S_i)}, Y_j^{(S_i)}) = H(Y_j^{(S_i)}) - H(Y_j^{(S_i)}|\Delta P_j^{(S_i)}),$$

where $H(Y_j^{(S_i)})$ represents the entropy of $Y_j^{(S_i)}$ and $H(Y_j^{(S_i)}|\Delta P_j^{(S_i)})$ is the entropy of $Y_j^{(S_i)}$ conditioned on $\Delta P_j^{(S_i)}$. Since $H(Y_j^{(S_i)})$ is constant, Equation (1) is equivalent to minimizing the conditional entropy $H(Y_j^{(S_i)}|\Delta P_j^{(S_i)})$, which can be expressed as:

$$\min \sum_{i=1}^{M} \sum_{j=1}^{|S_i|} H(Y_j^{(S_i)}|\Delta P_j^{(S_i)}).$$

Due to the computational complexity of the conditional entropy $H(Y_j^{(S_i)}|\Delta P_j^{(S_i)})$, it can be approximated by the cross-entropy between $f([\Delta P_j^{(S_i)}, X_j^{(S_i)}])$ and $Y_j^{(S_i)}$ [24, 46], where $f([\Delta P_j^{(S_i)}, X_j^{(S_i)}])$ is the prediction obtained by concatenating $\Delta P_j^{(S_i)}$ and $X_j^{(S_i)}$ as an input to our proposed POND model, which will be illustrated in Section 4.4. The fidelity loss is then expressed as:

$$\ell_F = \sum_{i=1}^{M} \sum_{j=1}^{|S_i|} Y_j^{(S_i)} \log f([\Delta P_j^{(S_i)}, X_j^{(S_i)}]). \tag{2}$$

Now, we theoretically show that the learned prompt $\Delta P_j^{(S_i)}$, which minimizes the fidelity loss (*i.e.*, Equation (2)), possesses the following properties:

Property 1 (Preserving Fidelity). If $\Delta P_j^{(S_i)}$ minimizes Equation (2), the mutual information between $\Delta P_j^{(S_i)}$ and the label $Y_j^{(S_i)}$ is equivalent to that between the time series input $X_j^{(S_i)}$ and the label $Y_j^{(S_i)}$, *i.e.*, $MI(\Delta P_j^{(S_i)}, Y_j^{(S_i)}) = MI(X_j^{(S_i)}, Y_j^{(S_i)})$.

Property 2 (Adding New Information). By minimizing Equation (2), the generated prompt $\Delta P_j^{(S_i)}$ contains new information compared to the time series input $X_j^{(S_i)}$, *i.e.*, $H(\Delta P_j^{(S_i)}) \geq H(X_j^{(S_i)})$.

Detailed proofs are provided in Section A.1 in the Appendix. These properties demonstrate that minimizing Equation (2) ensures that the generated prompts will not decrease fidelity and may add new information to the time series input.

High Distinction. In addition to high fidelity, it is essential that the generated domain-specific prompt $\Delta P^{(S_i)}$ distinguishes the unique information of the source domain S_i from other source domains. This unique information not only aids in understanding the differences between multiple time series source domains but also provides valuable insights for selecting suitable sources for domain adaptation. To achieve this, from the perspective of information theory, we define the objective to maintain high distinction as minimizing the mutual information of domain-specific prompts between different source domains, which should be minimized as follows:

$$\min \sum_{i_1 \neq i_2} MI(\Delta P^{(S_{i_1})}, \Delta P^{(S_{i_2})}), \tag{3}$$

where $\Delta P^{(S_{i_1})}$ and $\Delta P^{(S_{i_2})}$ represent the domain-specific prompts of any two source domains S_{i_1} and S_{i_2}. Equation (3) is computationally infeasible to minimize directly, but it can be achieved by minimizing the leave-one-out upper bound [24, 28]. Other mutual information upper bounds, such as the contrastive log-ratio bound [8], can also conveniently be incorporated into our framework. Therefore, the objective to encourage high distinction is formulated as minimizing the leave-one-out bound (*i.e.*, discrimination loss):

$$\ell_D = \sum_{i_1 \neq i_2} \mathbb{E} \log \frac{\exp(\mathrm{sim}(\Delta P^{(S_{i_1})}, \Delta P^{(S_{i_2})}))}{\sum_{i \neq i_1, i \neq i_2} \exp(\mathrm{sim}(\Delta P^{(S_{i_1})}, \Delta P^{(S_i)}))}, \tag{4}$$

where $\mathrm{sim}(\Delta P^{(S_{i_1})}, \Delta P^{(S_{i_2})}) = tr((\Delta P^{(S_{i_1})})^T \Delta P^{(S_{i_2})})$ denotes the inner product of the two domain-specific prompts $\Delta P^{(S_{i_1})}$ and $\Delta P^{(S_{i_2})}$, and $tr(A)$ represents the trace of any matrix A.

4.3 The Learning Objective

After introducing two criteria for selecting good prompts, we present our learning objective in this section.

Combining the fidelity loss ℓ_F in Equation (2) and the discrimination loss ℓ_D in Equation (4), our learning objective is expressed as follows:

$$\min_{P, g^{(S_i)}} G(P, g^{(S_i)}) = \ell_R + \lambda_1 \ell_D + \lambda_2 \ell_F, \tag{5}$$

where $\ell_R = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|S_i|} \sum_{j=1}^{|S_i|} R(f([P + \Delta P_j^{(S_i)}, X_j^{(S_i)}]), Y_j^{(S_i)})$ is the training loss that measures the performance of prompt tuning. Here, $R(\cdot, \cdot)$ is the loss function, and $[P + \Delta P_j^{(S_i)}, X_j^{(S_i)}]$ is the concatenation of the overall prompt $P + \Delta P_j^{(S_i)}$ and the time series input $X_j^{(S_i)}$. Two tuning parameters $\lambda_1, \lambda_2 > 0$ control the trade-off among the training loss, the fidelity loss, and the discrimination loss.

To optimize Equation (5), we need to enumerate all source domains, which may be inefficient and unscalable [24]. To address this, we propose a simple yet effective learning algorithm based on the classic Reptile meta-learning framework [25], which randomly picks a source domain each time and conducts standard steps of gradient descent without the need for calculating second derivatives. The learning process is outlined in Algorithm 1. Specifically, Line 3 updates the prompt generator $g^{(S_\tau)}$, and Lines 4-5 update the common prompt P through extrapolation. Here, the local learning rate η performs the gradient descent step, and the global learning rate δ performs the extrapolation step.

Algorithm 1 Reptile-based meta-learning for Prompt Tuning

```
Require: $(X_j^{(S_i)}, Y_j^{(S_i)})$, the global learning rate $\delta \in (0, 1]$, the local
    learning rate $\eta > 0$, the number of global steps $N$.
Ensure: the common prompt $P$, the prompt generator $g^{(S_i)}$.
1: for $i = 1$ to $N$ do
2:     Randomly pick a source time series domain $S_\tau$.
3:     $g^{(S_\tau)} \leftarrow g^{(S_\tau)} - \eta \nabla_{g^{(S_\tau)}} G$.
4:     $Q \leftarrow P - \eta \nabla_P \ell_T$.
5:     $P \leftarrow P + \delta(Q - P)$.
6: end for
```

After learning the common prompt P and the prompt generator $g^{(S_i)}$, they can be utilized for target domain transfer. Specifically, the prompt generator $g^{(T)}$ is optimized by the labeled time series pairs $(X_i^{(T)}, Y_i^{(T)})$ in the target domain T as follows:

$$\min_{g^{(T)}} \frac{1}{|T|} \sum_{i=1}^{|T|} R(f([P + \Delta P_i^{(T)}, X_i^{(T)}]), Y_i^{(T)}), \tag{6}$$

where $\Delta P_i^{(T)} = g^{(T)}(X_i^{(T)}) \in \mathbb{R}^{m\times n}$ is the instance-level domain-specific prompt of the time series input $X_i^{(T)}$, and the domain-level domain-specific prompt of the target domain T is $\Delta P^{(T)} = \frac{1}{|T|}\sum_{j=1}^{|T|} \Delta P_j^{(T)}$. However, $g^{(T)}$ may not be reliable for prediction due to the limited labeled data involved. To handle this, $\Delta P^{(T)}$ is utilized as a heuristic to find the most similar source domain by the simple nearest neighbor rule (*i.e.*, prompt adaptation):

$$S_i = \arg\max_{S_i} \mathrm{sim}(\Delta P^{(S_i)}, \Delta P^{(T)}), \tag{7}$$

where $\mathrm{sim}(\Delta P^{(S_i)}, \Delta P^{(T)})$ is a similarity function (*e.g.*, cosine similarity) between the domain-specific prompts $\Delta P^{(S_i)}$ and $\Delta P^{(T)}$. Then, we utilize the prompt generator $g^{(S_i)}$ for prediction in the target domain T: $f([P + g^{(S_i)}(X_j^{(T)}), X_j^{(T)}])$.

4.4 Discussion

In this section, we discuss the model architecture and implementation, the theoretical aspects of our proposed POND model, and its comparison with previous papers.

4.4.1 Model Architecture and Implementation. For the model architecture of our proposed POND model, we employ the popular Mixture of Expert (MoE) technique to enhance performance [9]: each expert makes an independent prediction, and the router is responsible for learning probability distributions over all predictions. The overall output of our POND model is a linear combination of all predictions.

For the architecture of a single expert, the time series input is fed into "a patching layer" (i.e., splitting a timeseries input into subseries-level patches [26]), a projection layer, a position embedding layer, a transformer layer, and a linear head sequentially.

The model implementation is illustrated in the following steps:

(1) **Model Pretraining:** All experts of our POND model are pretrained by combining some labeled data from all source domains (e.g. 60%), and the router, which aggregates outputs from all experts to make final predictions, is pretrained using the same labeled data.

(2) **Prompt Tuning:** Given the pretrained POND model, other labeled time series data from all source domains (e.g. 40%) are utilized to learn the common prompt P and the prompt generator $g^{(S_i)}$ by Equation (5) (*i.e.*, Algorithm 1), and the prompt generator of the target domain $g^{(T)}$ is optimized by Equation (6).

(3) **Prompt Adaptation:** The most similar source domain is selected by Equation (7), whose prompt generator will be used in the target domain for prediction.

4.4.2 Theoretical Analysis. We demonstrate the commonality and differences of our proposed POND model compared with traditional prompt-tuning from the theoretical perspective. Specifically, we prove that our proposed POND model shares the universal approximation with prompt tuning, and then we illustrate that our proposed POND model overcomes the limitation of prompt tuning. Without loss of generality, we assume that only one expert model is available, and ζ is removed (*i.e.*, $\Delta P_j^{(S_i)} = g^{(S_i)}(X_j^{(S_i)};\zeta) = g^{(S_i)}(X_j^{(S_i)})$). Proofs of all theorems below are shown in Section A.2 in the Appendix due to space limitations.

One recent paper theoretically proves the universality of prompt tuning [41], and it can be extended to our proposed POND model. Specifically, for any $\mathcal{L}$-Lipschitz function $\mathcal{F} : [0,1]^{n\times L} \rightarrow [0,1]^{|C|}$ under norm q, it satisfies the following: $\forall x_1, x_2 \in [0,1]^{n\times L}$, $\|\mathcal{F}(x_1) - \mathcal{F}(x_2)\|_q \leq \mathcal{L}\|x_1 - x_2\|_q$. The approximation error under q norm is defined as $d_q(\mathcal{F}_1, \mathcal{F}_2) = (\int \|\mathcal{F}_1(x) - \mathcal{F}_2(x)\|_q^q dx)^{\frac{1}{q}}$. Then Theorem 1 states that our proposed POND model can approximate any time series classifier, which are trained from specific source domains.

Theorem 1 (Universality of our POND Model). *Let $1 \leq q < \infty$ and $\varepsilon > 0$, and $\mathcal{F}^{(S_i)} : [0,1]^{n\times L} \rightarrow [0,1]^{|C|}$ is a time series classifer, which is trained from source domain S_i and is $\mathcal{L}$-Lipschitz, there exist a prompt length m and a POND model f such that for any $\mathcal{F}^{(S_i)}$, we can find a domain-specific prompt generator $g^{(S_i)} : [0,1]^{n\times L} \rightarrow \mathbb{R}^{n\times m}$ from source domain S_i with $d_q(f([P+g^{(S_i)}(\cdot), \cdot]), \mathcal{F}^{(S_i)}) < \varepsilon$ for all $S_i (i = 1, 2, \cdots M)$.*

Not only our proposed POND model shares the universality, it also overcomes the limitations of prompt tuning. The following theorem states that while prompt tuning may not be flexible enough to learn some labeled time series pairs, our proposed POND model can overcome this limitation.

Theorem 2 (Flexibility of of our POND Model). *Consider two labeled time series pairs $(X_1^{(S_1)} = [\mathcal{X}_1, \mathcal{X}_0], Y_1^{(S_1)})$ and $(X_1^{(S_2)} = [\mathcal{X}_2, \mathcal{X}_0], Y_1^{(S_2)})$ from two source domains S_1 and S_2, respectively, where $Y_1^{(S_1)} \neq Y_2^{(S_1)}$. For some proposed POND model f:*
(a).[The limitation of prompt tuning] There exists no prompt P such that $f([P, X_1^{(S_i)}]) = Y_1^{(S_i)} (i = 1, 2)$.
(b).[Our POND model handles this limitation] There exist the common prompt P and the prompt generators $g^{(S_i)} (i = 1, 2)$ such that
$f([P + g^{(S_i)}(X_1^{(S_i)}), X_1^{(S_i)}]) = Y_1^{(S_i)} (i = 1, 2)$.

4.4.3 Comparison and Relation with Previous Methods. Finally, we compare our proposed POND model with existing multi-source domain adaptation approaches, which have the following drawbacks:

(1). Neglection of domain-specific information. The common goal of existing methods is to make different domains indistinguishable [49]. However, domain-specific information may be eliminated, which is important to select which source is the most similar to the target for adaptation. Our proposed POND model can address this by prompt tuning: the value of a prompt is updated by the gradient based on labeled data, which provides domain-specific information.

(2). Inability to capture time-dependent information. Most existing methods are designed to address domain adaptation problems in the fields of computer vision and NLP whose information is static, and they are not able to capture time-dependent information such as trends and distribution shifts of the time series. Our proposed POND model can address this by our proposed novel conditional module: it is learned to generate a prompt for each time series input, which is flexible to learn time-dependent information.

we show that several classic methods are special cases of our proposed POND model.

1. Generalization of Prompt Tuning. Let $\lambda_1 = \lambda_2 = 0$, and $g^{(S_i)} = 0$, then our proposed POND model is reduced to the classic prompt tuning [19].

Table 2: Statistics of four datasets.

Dataset	# Domain	# Channel	# Class	Seq Len	# Train	# Test
HAR	30	9	6	128	2300	990
WISDM	36	3	6	128	1350	720
HHAR	9	3	6	128	12716	5218
SSC	20	1	5	3000	14280	6130

2. Generalization of Information Bottleneck. Let $\lambda_1 = 0$ and $P = 0$, then our proposed POND is reduced to the famous information bottleneck [37].

3. Generalization of IDPG. Let $\lambda_1 = \lambda_2 = 0$, and $P = 0$. Then our proposed POND model is reduced to Instance-Dependent Prompt Generation (IDPG) [43].

5 EXPERIMENTS

In this section, we employ four benchmark datasets to evaluate our proposed POND model in comparison with six state-of-the-art methods. All experiments were conducted on a Linux server equipped with an Intel(R) Xeon(R) Silver 4214 CPU and an NVIDIA GPU running version 510. More experiments are included in the supplementary materials [1] due to space limitations.

5.1 Experimental Settings

Benchmark Dataset: We evaluated the performance of all methods on four benchmark datasets, HAR, WISDM, HHAR and SSC [31]. The statistics of all benchmark datasets are shown in Table 2, which are introduced as follows:

1. HAR [1]: The Human Activity Recognition (HAR) dataset incorporates data collected from three sensors—accelerometer, gyroscope, and body sensors—deployed on 30 subjects (*i.e.*, domains) engaged in six distinct activities.

2. WISDM [16]: The WIreless Sensor Data Mining (WISDM) dataset, using accelerometer sensors, involves 36 subjects participating in activities similar to the HAR dataset, with additional challenges due to class distribution imbalances among different subjects.

3. HHAR [34]: The Heterogeneity Human Activity Recognition (HHAR) dataset was collected from 9 subjects using sensor readings from smartphones and smartwatches.

4. SSC [10]: The Sleep Stage Classification (SSC) problem aims to categorize electroencephalography (EEG) signals into five stages. We utilize the Sleep-EDF dataset [10], including EEG recordings from 20 healthy subjects.

Comparison Methods: We compared our proposed POND method with six state-of-the-art time series domain adaptation approaches: Raincoat [12], CoDATs [42], Deep Coral [35], MMDA [32], DIRT-T [33] and DSAN [52]. All comparison methods are introduced as follows:

1. Raincoat [12]: it is an unsupervised domain adaptation method addressing both feature and label shifts.

2. CoDATs [42]: it is the first method to handle multi-source domain adaptation through adversarial training with weak supervision.

3. Deep Coral [35]: it minimizes domain shift by aligning second-order statistics of source and target distributions.

4. MMDA [32]: it integrates Maximum Mean Discrepancy (MMD) and CORrelation ALignment (CORAL) along with conditional entropy minimization to address domain shift.

5. DIRT-T [33]: it utilizes adversarial training, conditional entropy, and a teacher model to align source and target domains.

6. DSAN [52]: it minimizes the discrepancy between source and target domains via a Local Maximum Mean Discrepancy (LMMD) that aligns relevant subdomain distributions.

Metrics: Two performance metrics were employed: Macro-F1 score and Accuracy. Macro-F1 is the unweighted mean of per-class F1 scores, treating all classes equally. Accuracy is the ratio of accurately predicted samples to all samples.

Hyperparameter Settings: We adapted the setting of supervised domain adaptation, where ten samples in the target domain were used for domain transfer. All source-target scenarios were selected randomly to ensure the fairness of the performance evaluation. Single-source domain adaptation methods (e.g. Raincoat) were trained by combining all source domains. For the training set of all time series source domains, 60% was used for pretraining our POND model, 20% for prompt tuning, and 20% for validation sets. The batch size was set to 16. The number of global steps N, global learning rate δ and the local learning rate η were set to 50, 0.01 and 0.001, respectively. The number of experts was set to three. The prompt generator is a two-layer Multi-Layer Perceptron (MLP) with Tanh activation. For the transformer model, the numbers of encoder layers, decoder layers, and heads in the multi-head attention were set to 2, 1, and 4, respectively. The dimensions of the multi-head attention and the feed-forward layer were set to 16 and 128, respectively. The hyperparameters λ_1 and λ_2 were chosen based on performance on the validation set. λ_1 and λ_2, along with other hyperparameters such as the number of epochs, are provided in Table 3. All methods were averaged by ten times.

Table 3: Hyperparameters of all datasets.

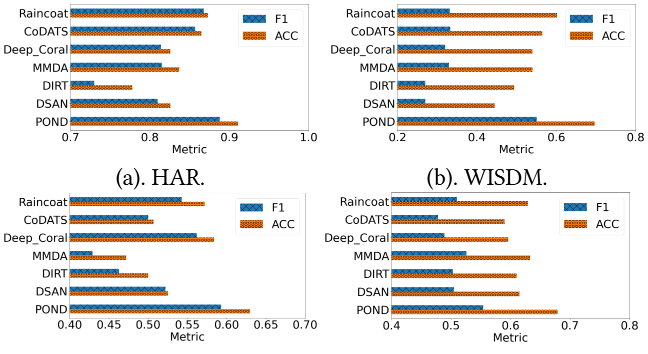
Dataset	#Epochs	Prompt Length	λ_1	λ_2
HAR	50	5	1	1
WISDM	200	3	1	1
HHAR	200	5	1	1
SSC	100	10	0.1	0.1

5.2 Experimental Results

Performance Evaluation: We conducted a comprehensive performance evaluation to test all methods across approximately 50 scenarios on four datasets. Figure 3 displays the F1-score and accuracy of all methods on these datasets. Our proposed POND method consistently outperforms others across all four datasets. Specifically, on the HAR dataset, the F1-score of POND is approximately 0.9, only 2% lower than the top-performing comparison method, Raincoat. The F1-score gaps on the HHAR, and SSC datasets are 5% and 4.4%, respectively. The largest gap is observed in the WISDM dataset, where the F1-score and accuracy of POND hover around 0.6 and 0.7, while all comparison methods score below 0.35 and 0.6, respectively. Considering the inherent difficulty of training on the WISDM dataset due to class imbalance, this highlights the effectiveness of our proposed POND, especially on challenging datasets.

Among the comparison methods, Raincoat emerges as the best overall. In terms of F1-score, Raincoat outperforms MMDA by 5%

[1] Link of supplementary materials: https://github.com/xianggebenben/Junxiang_Wang.github.io/blob/master/supplementary_material/KDD2024/supp.pdf.



(a). HAR. (b). WISDM.

(c). HHAR. (d). SSC.

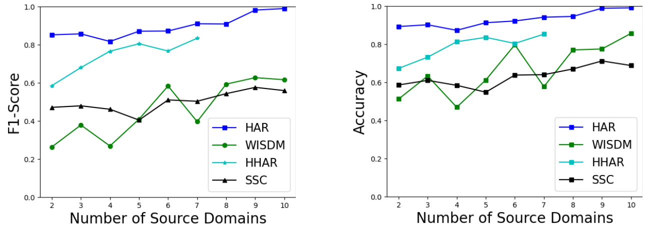
Figure 3: The F1-score and accuracy of all methods on four benchmark datasets: the proposed POND outperforms comparison methods consistently.

on the HAR dataset and shows an 8% superiority over CoDATs on the HHAR dataset. For accuracy, Raincoat performs 7% better than DIRT on the HHAR dataset and surpasses Deep Coral by 3% on the SSC dataset. CoDATs and Deep Coral also demonstrate competitive performance, achieving around 55% accuracy on the WISDM dataset, while DSAN lags behind at 45%. On the other hand, MMDA, DIRT, and DSAN exhibit varying performance across datasets. For instance, DSAN performs comparably to Raincoat on the SSC dataset but ranks the lowest on the WISDM dataset.

Table 4 presents the performance of all methods across various scenarios in four datasets, including the upper bound achieved by training and testing on the target domain. The reported values include means and standard deviations from ten implementations, with the best results highlighted in bold. The complete performance evaluation is available in the supplementary materials [1]. Overall, our proposed POND model consistently outperforms all methods, aligning with the observations in Figure 3. Notably, POND exhibits superior performance on the challenging WISDM dataset, as indicated by Figure 3. For instance, POND outperforms all comparison methods by at least 23% when transferring from domains 0-17 to domain 18. While POND excels overall, there are instances where comparison methods outperform it. For example, Deep Coral performs better than POND by 2% when transferring domains 1-15 to domain 28 on the HAR dataset, and MMDA marginally outperforms POND when transferring domains 1-15 to domain 21 on the HAR dataset.

In addition to superior performance, our proposed POND model demonstrates greater stability compared to all comparison methods, as indicated by lower standard deviations. For instance, the standard deviation of POND is 0.006 when transferring domains 0-9 to domain 17 on the SSC dataset, while the standard deviations of all comparison methods range between 0.024 and 0.118, being at least 3 times larger than that of POND. Importantly, POND achieves results close to the upper bound in many scenarios, such as "HAR 1-15 → 16", "SSC 0-9 → 18", and "HHAR 0-6 → 7".

Ablation Study: Next, we demonstrate the ablation study of the proposed POND method, whose goal is to identify whether all components of our proposed POND model contribute to the performance. Specifically, we explore the necessity of the MoE technique, common prompt, and prompt generator. The challenging WISDM dataset was utilized to test the performance. Table 5 illustrates the performance of different scenarios, all of which were averaged by 10 times. The first two rows show the performance with the common prompt, and the prompt generator available only, respectively. The fourth to sixth rows demonstrate the performance without the MoE, common prompt, and prompt generator, respectively, and the last row shows the performance of the complete POND model. Overall, our proposed POND model performs best when the MoE, common prompt, and prompt generator are all available, which suggests that all components are necessary for the outstanding performance of our proposed POND model. For example, in the scenario of "18-23→ 6", the best performance without any component only achieves a performance no more than 0.58, whereas that of the complete POND model is 5% better. The gap is widened to 7% for the scenario "0-17→ 25".



(a). F1-score. (b). Accuracy.

Figure 4: The F1-score and accuracy of the proposed POND model with different source domains: the performance grows with the increase of source domains. (The HHAR dataset has less than 10 domains.)

Sensitivity Analysis: In this section, we explore how source domains influence performance on the target domain. Figure 4 illustrates the relationship between performance metrics (F1-score and accuracy) and the number of source domains, averaged over 10 implementations. Generally, our proposed POND model demonstrates improved performance with an increasing number of source domains. For instance, POND achieves 50% accuracy with two source domains for training, but this figure rises by 30% when an additional 8 source domains are included. Similarly, the F1-score of POND increases by 20% when the number of source domains changes from 2 to 6. However, some exceptions exist. For example, there is a notable 25% drop in F1-score when increasing the number of source domains from 6 to 7 on the WISDM dataset. Another instance involves a 5% performance drop when increasing the source domains from 4 to 5 on the SSC dataset.

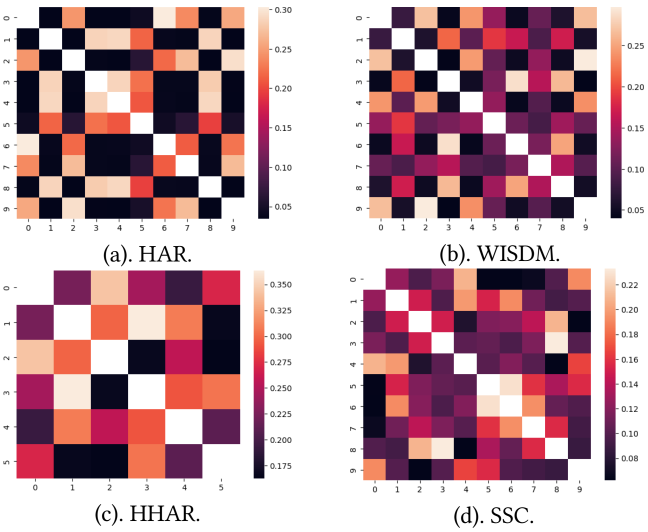
Visualization of Discrimination Loss: Finally, we present a visualization of the discrimination loss ℓ_D for pairwise source domains. Figure 5 illustrates the exponents of discrimination losses for all pairs of source domains across four datasets. Both the X-axis and Y-axis represent the indexes of source domains. Darker colors indicate smaller discrimination losses, reflecting better domain discrimination. The diagonals are left blank. Overall, our proposed POND model effectively discriminates most source domains, as evidenced by the predominance of dark squares. For instance, domains 3-5 and domains 6-7 exhibit clear discrimination with losses below 0.05. Similar effective discrimination is observed for domain pairs 6 and 0 on the WISDM dataset, domain pairs 1 and 5 on the HHAR dataset, and domains 5-7 and 0 in the SSC dataset. However,

Table 4: F1-score on different scenarios of four datasets: the proposed POND model outperforms all comparison methods.

Scenario	Raincoat	CoDATs	Deep_Coral	MMDA	DIRT	DSAN	POND	Target Only
HAR 1-15 → 16	0.823 ± 0.094	0.767 ± 0.093	0.773 ± 0.082	0.679 ± 0.084	0.612 ± 0.135	0.738 ± 0.095	**0.849 ± 0.021**	0.856 ± 0.027
HAR 1-15 → 20	0.872 ± 0.142	0.932 ± 0.025	0.923 ± 0.023	0.921 ± 0.034	0.848 ± 0.101	0.929 ± 0.033	**0.968 ± 0.021**	0.983 ± 0.018
HAR 1-15 → 21	0.867 ± 0.141	0.903 ± 0.070	0.882 ± 0.028	**0.974 ± 0.039**	0.921 ± 0.090	0.909 ± 0.110	0.972 ± 0.021	1.000 ± 0.000
HAR 1-15 → 28	0.766 ± 0.107	0.775 ± 0.166	**0.852 ± 0.044**	0.778 ± 0.085	0.671 ± 0.175	0.783 ± 0.046	0.829 ± 0.018	0.853 ± 0.019
HAR 16-20 → 1	0.792 ± 0.072	0.744 ± 0.053	0.667 ± 0.077	0.654 ± 0.074	0.546 ± 0.060	0.698 ± 0.037	**0.883 ± 0.017**	0.986 ± 0.010
HAR 16-20 → 2	0.825 ± 0.048	0.821 ± 0.151	0.796 ± 0.055	0.651 ± 0.045	0.509 ± 0.050	0.652 ± 0.057	**0.936 ± 0.017**	0.943 ± 0.024
HAR 16-20 → 3	0.814 ± 0.028	0.746 ± 0.078	0.741 ± 0.058	0.657 ± 0.033	0.605 ± 0.056	0.565 ± 0.043	**0.878 ± 0.018**	0.978 ±0.013
HAR 16-20 → 4	0.679 ± 0.084	0.605 ± 0.082	0.479 ± 0.110	0.513 ± 0.058	0.336 ± 0.110	0.436 ± 0.032	**0.754 ± 0.033**	0.921 ± 0.018
WISDM 0-17 → 18	0.379 ± 0.061	0.384 ± 0.049	0.346 ± 0.023	0.297 ± 0.016	0.300 ± 0.041	0.287 ± 0.045	**0.606 ± 0.020**	0.705 ± 0.046
WISDM 0-17 → 20	0.354 ± 0.040	0.368 ± 0.039	0.376 ± 0.031	0.452 ± 0.098	0.347 ± 0.071	0.269 ± 0.064	**0.570 ± 0.023**	0.704 ± 0.051
WISDM 0-17 → 21	0.355 ± 0.057	0.310 ± 0.088	0.259 ± 0.018	0.250 ± 0.000	0.276 ± 0.055	0.245 ± 0.046	**0.450 ± 0.026**	0.636 ± 0.095
WISDM 0-17 → 23	0.306 ± 0.015	0.327 ± 0.075	0.318 ± 0.031	0.327 ± 0.023	0.271 ± 0.016	0.277 ± 0.044	**0.482 ± 0.017**	0.538 ± 0.034
WISDM 0-17 → 25	0.365 ± 0.030	0.540 ± 0.125	0.435 ± 0.043	0.436 ± 0.094	0.314 ± 0.107	0.353 ± 0.120	**0.559 ± 0.050**	0.672 ± 0.039
WISDM 0-17 → 28	0.399 ± 0.028	0.431 ± 0.033	0.418 ± 0.032	0.454 ± 0.064	0.304 ± 0.044	0.339 ± 0.030	**0.656 ± 0.046**	0.689 ± 0.048
WISDM 0-17 → 30	0.314 ± 0.020	0.305 ± 0.028	0.298 ± 0.023	0.359 ± 0.072	0.266 ± 0.035	0.246 ± 0.076	**0.670 ± 0.039**	0.791 ± 0.028
WISDM 18-23 → 5	0.648 ± 0.001	0.558 ± 0.129	0.534 ± 0.102	0.510 ± 0.020	0.549 ± 0.097	0.484 ± 0.055	**0.652 ± 0.035**	0.734 ± 0.095
WISDM 18-23 → 6	0.544 ± 0.074	0.565 ± 0.143	0.437 ± 0.078	0.543 ± 0.160	0.405 ± 0.089	0.454 ± 0.112	**0.628 ± 0.033**	0.872 ± 0.049
WISDM 18-23 → 7	0.588 ± 0.070	0.404 ± 0.117	0.530 ± 0.094	0.477 ± 0.060	0.518 ± 0.120	0.476 ± 0.127	**0.672 ± 0.029**	0.888 ± 0.035
HHAR 0-6 → 7	0.765 ± 0.142	0.652 ± 0.108	0.815 ± 0.105	0.641 ± 0.050	0.649 ± 0.005	0.730 ± 0.164	**0.834 ± 0.014**	0.861 ± 0.016
HHAR 5-8 → 2	0.321 ± 0.023	0.347 ± 0.082	0.309 ± 0.032	0.216 ± 0.032	0.276 ± 0.021	0.314 ± 0.095	**0.352 ± 0.014**	0.881 ± 0.018
SSC 0-9 → 16	0.578 ± 0.028	0.510 ± 0.044	0.537 ± 0.024	0.559 ± 0.027	0.523 ± 0.019	0.515 ± 0.044	**0.568 ± 0.012**	0.601 ± 0.018
SSC 0-9 → 17	0.511 ± 0.024	0.413 ± 0.118	0.452 ± 0.077	0.504 ± 0.060	0.530 ± 0.053	0.463 ± 0.081	**0.559 ± 0.006**	0.602 ± 0.014
SSC 0-9 → 18	**0.605 ± 0.016**	0.548 ± 0.037	0.544 ± 0.046	0.597 ± 0.032	0.574 ± 0.021	0.569 ± 0.046	0.604 ± 0.014	0.602 ± 0.013
SSC 0-9 → 19	0.562 ± 0.024	0.540 ± 0.052	0.531 ± 0.055	**0.570 ± 0.044**	0.565 ± 0.028	0.568 ± 0.080	**0.570 ± 0.010**	0.613 ± 0.019
SSC 10-12 → 8	0.294 ± 0.028	0.380 ± 0.066	0.379 ± 0.076	0.398 ± 0.060	0.322 ± 0.048	0.411 ± 0.046	**0.470 ± 0.010**	0.531 ± 0.019

Table 5: Ablation study on the WISDM dataset: all components of our proposed POND model contribute to the outstanding performance.

MoE	Common Prompt	Prompt Generator	0-17→ 22	0-17→ 23	0-17→ 24	0-17→ 25	18-23→ 5	18-23→ 6	Overall
✗	✓	✗	0.622±0.057	0.415±0.015	0.510±0.030	0.581±0.036	0.623±0.058	0.516±0.038	0.545±0.039
✗	✗	✓	0.646±0.064	0.396±0.048	0.527±0.030	0.573±0.034	0.628±0.051	0.512±0.057	0.547±0.047
✗	✓	✓	0.632±0.069	0.384±0.041	0.498±0.032	0.572±0.045	0.611±0.055	0.514±0.025	0.535±0.045
✓	✗	✓	0.575±0.043	0.349±0.029	0.517±0.032	0.584±0.030	0.621±0.056	0.578±0.035	0.537±0.038
✓	✓	✗	0.719±0.062	0.405±0.052	0.529±0.042	0.588±0.034	0.616±0.050	0.565±0.049	0.570±0.048
✓	✓	✓	**0.725±0.031**	**0.482±0.017**	**0.559±0.050**	**0.695±0.035**	**0.652±0.035**	**0.628±0.033**	**0.624±0.034**



(a). HAR.

(b). WISDM.

(c). HHAR.

(d). SSC.

Figure 5: The visualization of the exponent of discrimination loss: most pairs of source domains are well discriminated.

discrimination losses for some domain pairs are larger than others. For instance, on the HAR dataset, the discrimination loss between domains 0 and 6 is the largest, approximately 0.30, but still within an acceptable range. It's worth noting that domain discrimination may not adhere to the transitive property. For example, domains 3 and 9, as well as domains 4 and 9, are well-discriminated, but domains 3 and 4 are relatively poor-discriminated.

6 CONCLUSION

Time series domain adaptation is an important problem with wide-ranging applications. Existing techniques primarily address single-source domain adaptation, yet exploring adaptation from multiple domains holds promise for greater improvements. In this paper, we introduce POND, the first framework to utilize prompts for time series domain adaptation. We extend prompt tuning to time series analysis to capture common and domain-specific information from all source domains, introduce conditional modules for prompt generation, and propose criteria for selecting effective prompts. Through extensive experiments across 50 scenarios on four datasets, we demonstrate the efficacy and robustness of POND, outperforming all state-of-the-art methods by up to 66% on the F1-score.

REFERENCES

[1] Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J. L., et al. A public domain dataset for human activity recognition using smartphones. In *Esann* (2013), vol. 3, p. 3.

[2] Bai, G., Ling, C., and Zhao, L. Temporal domain generalization with drift-aware dynamic neural networks. In *The Eleventh International Conference on Learning Representations* (2022).

[3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems 33* (2020), 1877–1901.

[4] Cai, R., Chen, J., Li, Z., Chen, W., Zhang, K., Ye, J., Li, Z., Yang, X., and Zhang, Z. Time series domain adaptation via sparse associative structure alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 6859–6867.

[5] Cai, Z., Bai, G., Jiang, R., Song, X., and Zhao, L. Continuous temporal domain generalization. *arXiv preprint arXiv:2405.16075* (2024).

[6] Cao, D., Jia, F., Arik, S. O., Pfister, T., Zheng, Y., Ye, W., and Liu, Y. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948* (2023).

[7] Chang, C., Peng, W.-C., and Chen, T.-F. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469* (2023).

[8] Cheng, P., Hao, W., Dai, S., Liu, J., Gan, Z., and Carin, L. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning* (2020), PMLR, pp. 1779–1788.

[9] Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research 23*, 1 (2022), 5232–5270.

[10] Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation 101*, 23 (2000), e215–e220.

[11] Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters. *arXiv preprint arXiv:2310.07820* (2023).

[12] He, H., Queen, O., Koker, T., Cuevas, C., Tsiligkaridis, T., and Zitnik, M. Domain adaptation for time series under feature and label shifts. In *Proceedings of the 40th International Conference on Machine Learning* (23–29 Jul 2023), A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202 of *Proceedings of Machine Learning Research*, PMLR, pp. 12746–12774.

[13] Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728* (2023).

[14] Jin, M., Wen, Q., Liang, Y., Zhang, C., Xue, S., Wang, X., Zhang, J., Wang, Y., Chen, H., Li, X., Pan, S., Tseng, V. S., Zheng, Y., Chen, L., and Xiong, H. Large models for time series and spatio-temporal data: A survey and outlook. *arXiv preprint arXiv:2310.10196* (2023).

[15] Jin, X., Park, Y., Maddix, D., Wang, H., and Wang, Y. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning* (2022), PMLR, pp. 10280–10297.

[16] Kwapisz, J. R., Weiss, G. M., and Moore, S. A. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter 12*, 2 (2011), 74–82.

[17] Lai, K.-H., Wang, L., Chen, H., Zhou, K., Wang, F., Yang, H., and Hu, X. Context-aware domain adaptation for time series anomaly detection. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)* (2023), SIAM, pp. 676–684.

[18] Lessmeier, C., Kimotho, J. K., Zimmer, D., and Sextro, W. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM Society European Conference* (2016), vol. 3.

[19] Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[20] Li, Y., Chen, Z., Zha, D., Du, M., Ni, J., Zhang, D., Chen, H., and Hu, X. Towards learning disentangled representations for time series. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), pp. 3270–3278.

[21] Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., Chowdhury, T., Li, Y., Cui, H., Zhao, T., et al. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv 2305* (2023).

[22] Liu, Q., and Xue, H. Adversarial spectral kernel matching for unsupervised time series domain adaptation. In *IJCAI* (2021), pp. 2744–2750.

[23] Luo, C., Chen, Z., Tang, L.-A., Shrivastava, A., Li, Z., Chen, H., and Ye, J. Tinet: learning invariant networks via knowledge transfer. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), pp. 1890–1899.

[24] Luo, D., Cheng, W., Wang, Y., Xu, D., Ni, J., Yu, W., Zhang, X., Liu, Y., Chen, Y., Chen, H., et al. Time series contrastive learning with information-aware augmentations. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2023), vol. 37, pp. 4534–4542.

[25] Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* (2018).

[26] Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations* (2022).

[27] Peng, X., Huang, Z., Sun, X., and Saenko, K. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning* (2019), PMLR, pp. 5102–5112.

[28] Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *International Conference on Machine Learning* (2019), PMLR, pp. 5171–5180.

[29] Purushotham, S., Carvalho, W., Nilanon, T., and Liu, Y. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations* (2016).

[30] Ragab, M., Eldele, E., Chen, Z., Wu, M., Kwoh, C.-K., and Li, X. Self-supervised autoregressive domain adaptation for time series data. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[31] Ragab, M., Eldele, E., Tan, W. L., Foo, C.-S., Chen, Z., Wu, M., Kwoh, C.-K., and Li, X. Adatime: A benchmarking suite for domain adaptation on time series data. *ACM Transactions on Knowledge Discovery from Data 17*, 8 (2023), 1–18.

[32] Rahman, M. M., Fookes, C., Baktashmotlagh, M., and Sridharan, S. On minimum discrepancy estimation for deep domain adaptation. *Domain Adaptation for Visual Understanding* (2020), 81–94.

[33] Shu, R., Bui, H., Narui, H., and Ermon, S. A dirt-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations* (2018).

[34] Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., Sonne, T., and Jensen, M. M. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems* (2015), pp. 127–140.

[35] Sun, B., and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14* (2016), Springer, pp. 443–450.

[36] Sun, C., Li, Y., Li, H., and Hong, S. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241* (2023).

[37] Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[38] Wang, D., Chen, Z., Fu, Y., Liu, Y., and Chen, H. Incremental causal graph learning for online root cause analysis. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023), pp. 2269–2278.

[39] Wang, D., Chen, Z., Ni, J., Tong, L., Wang, Z., Fu, Y., and Chen, H. Interdependent causal networks for root cause localization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023), pp. 5051–5060.

[40] Wang, J., and Zhao, L. Multi-instance domain adaptation for vaccine adverse event detection. In *Proceedings of the 2018 World Wide Web Conference* (2018), pp. 97–106.

[41] Wang, Y., Chauhan, J., Wang, W., and Hsieh, C.-J. Universality and limitations of prompt tuning. In *Proceedings of Advances in Neural Information Processing Systems 36 (NeurIPS 2023)* (2023).

[42] Wilson, G., Doppa, J. R., and Cook, D. J. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (2020), pp. 1768–1778.

[43] Wu, Z., Wang, S., Gu, J., Hou, R., Dong, Y., Vydiswaran, V. V., and Ma, H. Idpg: An instance-dependent prompt generation method. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2022), pp. 5507–5521.

[44] Xue, H., and Salim, F. D. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[45] Yang, L., and Hong, S. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning* (2022), PMLR, pp. 25038–25054.

[46] Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems 32* (2019).

[47] Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2022), vol. 36, pp. 8980–8987.

[48] Zhao, M., Yue, S., Katabi, D., Jaakkola, T. S., and Bianchi, M. T. Learning sleep stages from radio signals: A conditional adversarial architecture. In *International Conference on Machine Learning* (2017), PMLR, pp. 4100–4109.

[49] Zhao, S., Li, B., Xu, P., and Keutzer, K. Multi-source domain adaptation in the deep learning era: A systematic survey. *arXiv preprint arXiv:2002.12169* (2020).

[50] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B.,

Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[51] Zhou, T., Niu, P., Wang, X., Sun, L., and Jin, R. One fits all: Power general time series analysis by pretrained lm. *arXiv preprint arXiv:2302.11939* (2023).

[52] Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., Xiong, H., and He, Q. Deep subdomain adaptation network for image classification. *IEEE transactions on neural networks and learning systems 32*, 4 (2020), 1713–1722.

Appendix

A MATHEMATICAL PROOFS

A.1 Proofs of Properties 1 and 2

Property 1 (Preserving Fidelity). *If $\Delta P_j^{(S_i)}$ minimizes Equation (2), the mutual information between $\Delta P_j^{(S_i)}$ and the label $Y_j^{(S_i)}$ is equivalent to that between the time series input $X_j^{(S_i)}$ and the label $Y_j^{(S_i)}$, i.e., $MI(\Delta P_j^{(S_i)}, Y_j^{(S_i)}) = MI(X_j^{(S_i)}, Y_j^{(S_i)})$.*

Proof. From the definition of mutual information, we have:

$$
\begin{aligned}
&MI(\Delta P_j^{(S_i)}, Y_j^{(S_i)})\\
&= H(Y_j^{(S_i)}) - H(Y_j^{(S_i)}|\Delta P_j^{(S_i)})\\
&= H(Y_j^{(S_i)}) + \sum_{\Delta P_j^{(S_i)}, Y_j^{(S_i)}} p(\Delta P_j^{(S_i)}, Y_j^{(S_i)}) \log \frac{p(\Delta P_j^{(S_i)}, Y_j^{(S_i)})}{p(\Delta P_j^{(S_i)})}\\
&= H(Y_j^{(S_i)}) + \sum_{X_j^{(S_i)}, Y_j^{(S_i)}} \sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)}, Y_j^{(S_i)})\\
&\log \frac{p(\Delta P_j^{(S_i)}, Y_j^{(S_i)})}{p(\Delta P_j^{(S_i)})}.
\end{aligned}
$$

where $\mathbb{V}(X_j^{(S_i)})$ is the set of generated prompts of a time series input $X_j^{(S_i)}$. Because $\Delta P_j^{(S_i)}$ is a function of $X_j^{(S_i)}$ only, this means that $p(\Delta P_j^{(S_i)}|X_j^{(S_i)}, Y_j^{(S_i)}) = p(\Delta P_j^{(S_i)}|X_j^{(S_i)})$. Since the mapping between $X_j^{(S_i)}$ and $\Delta P_j^{(S_i)}$ is one to many, for each $\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})$, we have $p(\Delta P_j^{(S_i)}, Y_j^{(S_i)}) = p(\Delta P_j^{(S_i)}, X_j^{(S_i)}, Y_j^{(S_i)})$, and $p(\Delta P_j^{(S_i)}) = p(\Delta P_j^{(S_i)}|X_j^{(S_i)})p(X_j^{(S_i)})$. Therefore, we have

$$
\begin{aligned}
\frac{p(\Delta P_j^{(S_i)}, Y_j^{(S_i)})}{p(\Delta P_j^{(S_i)})} &= \frac{p(\Delta P_j^{(S_i)}, X_j^{(S_i)}, Y_j^{(S_i)})}{p(\Delta P_j^{(S_i)}|X_j^{(S_i)})p(X_j^{(S_i)})}\\
&= \frac{p(\Delta P_j^{(S_i)}|X_j^{(S_i)}, Y_j^{(S_i)})p(X_j^{(S_i)}, Y_j^{(S_i)})}{p(\Delta P_j^{(S_i)}|X_j^{(S_i)})p(X_j^{(S_i)})}\\
&= \frac{p(\Delta P_j^{(S_i)}|X_j^{(S_i)})p(X_j^{(S_i)}, Y_j^{(S_i)})}{p(\Delta P_j^{(S_i)}|X_j^{(S_i)})p(X_j^{(S_i)})}\\
&= \frac{p(X_j^{(S_i)}, Y_j^{(S_i)})}{p(X_j^{(S_i)})}.
\end{aligned}
$$

$$
\begin{aligned}
&MI(\Delta P_j^{(S_i)}, Y_j^{(S_i)})\\
&= H(Y_j^{(S_i)}) + \sum_{X_j^{(S_i)}, Y_j^{(S_i)}} \sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)}, Y_j^{(S_i)})\\
&\log \frac{p(X_j^{(S_i)}, Y_j^{(S_i)})}{p(X_j^{(S_i)})}\\
&= H(Y_j^{(S_i)}) + \sum_{X_j^{(S_i)}, Y_j^{(S_i)}} [\sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)}, Y_j^{(S_i)})]\\
&\log \frac{p(X_j^{(S_i)}, Y_j^{(S_i)})}{p(X_j^{(S_i)})}\\
&= H(Y_j^{(S_i)}) + \sum_{X_j^{(S_i)}, Y_j^{(S_i)}} p(X_j^{(S_i)}, Y_j^{(S_i)}) \log \frac{p(X_j^{(S_i)}, Y_j^{(S_i)})}{p(X_j^{(S_i)})}\\
&= MI(X_j^{(S_i)}, Y_j^{(S_i)}).
\end{aligned}
$$

□

Property 2 (Adding New Information). *By minimizing Equation (2), the generated prompt $\Delta P_j^{(S_i)}$ contains new information comparing to the time series input $X_j^{(S_i)}$, i.e., $H(\Delta P_j^{(S_i)}) \geq H(X_j^{(S_i)})$.*

Proof. Without loss of generality, we assume that a finite number of prompts are generated for each time series input, and each prompt is generated independently. Then we have $p(X_j^{(S_i)}) = \sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)})$. It follows that

$$
\begin{aligned}
&H(X_j^{(S_i)})\\
&= -\sum_{X_j^{(S_i)}} p(X_j^{(S_i)}) \log(p(X_j^{(S_i)}))\\
&= -\sum_{X_j^{(S_i)}} [\sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)})] \log([\sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)})])\\
&= -\sum_{X_j^{(S_i)}} \sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)}) \log([\sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)})])\\
&\leq -\sum_{X_j^{(S_i)}} \sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)}) \log(p(\Delta P_j^{(S_i)})) \quad \text{(Jensen's Inequality)}\\
&= -\sum_{\Delta P_j^{(S_i)} \in \mathbb{V}(X_j^{(S_i)})} p(\Delta P_j^{(S_i)}) \log(p(\Delta P_j^{(S_i)})) = H(\Delta P_j^{(S_i)})
\end{aligned}
$$

□

A.2 Proofs of Theorems 1 and 2

To prove Theorems 1 and 2, we follow the similar procedure of [41]. To make proofs self-contained, we first mathematically formulate our simplified POND model f. Without loss of generality, we assume that f has only one expert transformer network, and it consists of an attention layer and an MLP layer. The attention layer and the transformer layer are defined as follows [41]:

Definition 1 (Attention Layer)**.** The $\mathbf{h}$-head attention layer between a time-stamp $\mathbf{x}$ and a time series $\mathbf{X}$ is defined as follows:

$$Att(\mathbf{x}, \mathbf{X}) = \sum_{i=1}^{\mathbf{h}} \mathbf{W}_o^i \mathbf{W}_v^i \mathbf{X} \sigma((\mathbf{W}_k^i \mathbf{X})^T \mathbf{W}_q^i \mathbf{x})$$

where $\mathbf{W}_q^i$, $\mathbf{W}_k^i$, $\mathbf{W}_v^i$ and $\mathbf{W}_o^i (i = 1, \cdots, \mathbf{h})$ are parameterized weights, and σ is a softmax operator. The normalizing factor of $\frac{1}{\sqrt{\mathbf{d}_{kq}}}$ is subsumed in the weight matrices $\mathbf{W}_k^i$ for notational simplicity.

We then define the cross-attention between two time series $\mathbf{X} \in \mathbb{R}^{n \times L}$ and $\mathbf{X}' \in \mathbb{R}^{n \times L}$:

$$Att(\mathbf{X}, \mathbf{X}') = [Att(\mathbf{X}_{:,1}, \mathbf{X}'), Att(\mathbf{X}_{:,2}, \mathbf{X}'), \cdots, Att(\mathbf{X}_{:,L}, \mathbf{X}')]$$

where $\mathbf{W}_{:,j}$ is the j-th column of $\mathbf{W}$.

Definition 2 (Simplified POND Model)**.** The simplified POND model f is shown as follows:

$$\begin{aligned} MLP(\mathbf{X}) = [&\mathbf{W}_2 RELU(\mathbf{W}_1 \mathbf{X}_{:,1}) + \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{X}_{:,1}, \cdots, \\ &\mathbf{W}_2 RELU(\mathbf{W}_1 \mathbf{X}_{:,n}) + \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{X}_{:,n}] \\ f(\mathbf{X}) = &MLP(Att(\mathbf{X}, \mathbf{X}) + \mathbf{X}). \end{aligned}$$

where $RELU(\cdot)$ is the ReLU activation function.

Theorem 1 (Universality of our POND Model)**.** *Let $1 \leq q < \infty$ and $\varepsilon > 0$, and $\mathcal{F}^{(S_i)} : [0, 1]^{n \times L} \longrightarrow [0, 1]^{|C|}$ is a time series classifer, which is trained from source domain S_i and is $\mathcal{L}$-Lipschitz, there exist a prompt length m and a POND model f such that for any $\mathcal{F}^{(S_i)}$, we can find a domain-specific prompt generator $g^{(S_i)} : [0, 1]^{n \times L} \longrightarrow \mathbb{R}^{n \times m}$ from source domain S_i with $d_q(f([P+g^{(S_i)}(\cdot), \cdot]), \mathcal{F}^{(S_i)}) < \varepsilon$ for all $S_i (i = 1, 2, \cdots M)$.*

Proof. Let the common prompt $P = 0$, the prompt generator $g^{(S_i)}$ be constant $P^{(S_i)}$, and f be a transformer with two heads of size one and four hidden units, then this theorem can be directly derived from Theorem 1 in [41]. □

To prove Theorem 2, we need two assumptions on our simplified POND model f, which are shown as follows:

Assumption 1 (Assumption on the Attention Layer)**.** $Att(X_1^{(S_1)}, X_1^{(S_1)}) + X_1^{(S_1)} \neq Att(X_1^{(S_2)}, X_1^{(S_2)}) + X_1^{(S_2)}$ in Theorem 2, and $\mathbf{W}_o$, $\mathbf{W}_k$, $\mathbf{W}_q$, and $\mathbf{W}_v$ are full rank.

Assumption 2 (Assumption on the MLP Layer)**.** $Y_1^{(S_i)} (i = 1, 2)$ in Theorem 2 are in the range set of MLP. Moreover, the number of channels $n \geq 2 + dim((MLP^{-1}(Y_1^{(S_1)}) - \mathcal{X}_1) \cup (MLP^{-1}(Y_1^{(S_2)}) - \mathcal{X}_2))$ in Theorem 2. Here $dim(\mathbf{S})$ measures the dimension of the subspace spanned by vectors in a set $\mathbf{S}$ and $MLP^{-1}(\mathbf{y}) = \{\mathbf{x} : MLP(\mathbf{x}) = \mathbf{y}\}$.

Aside from two assumptions, the following Lemma is also useful to prove Theorem 2.

Lemma 1. *(Lemma 7 in [41]) Given $\mathbf{c} \in \mathbb{R}^{n \times L}$ and full-rank attention weights $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$, there are $\mathbf{x}_0$ almost everywhere for which there exists $\mathbf{x}_1 \in \mathbb{R}^{n \times L}$ such that $Att(\mathbf{x}_0, [\mathbf{x}_0, \mathbf{x}_1]) || \mathbf{c}$.*

Theorem 2 (Flexibility of of our POND Model)**.** *Consider two labeled time series pairs $(X_1^{(S_1)} = [\mathcal{X}_1, \mathcal{X}_0], Y_1^{(S_1)})$ and $(X_1^{(S_2)} = [\mathcal{X}_2, \mathcal{X}_0], Y_1^{(S_2)})$ from two source domains S_1 and S_2, respectively, where $Y_1^{(S_1)} \neq Y_2^{(S_1)}$. For some proposed POND model f:*
(a).[The limitation of prompt tuning] There exists no prompt P such that $f([P, X_1^{(S_i)}]) = Y_1^{(S_i)} (i = 1, 2)$.
(b).[Our POND Model handles this limitation] There exist the common prompt P and the prompt generators $g^{(S_i)} (i = 1, 2)$ such that $f([P + g^{(S_i)}(X_1^{(S_i)}), X_1^{(S_i)}]) = Y_1^{(S_i)} (i = 1, 2)$.

Proof. (a). Firstly, we consider the prompt P only (*i.e.*, without the prompt generator $g^{(S_i)}$), we pass $X_1^{(S_1)}$ and $X_1^{(S_2)}$ to the attention layer to obtain:

$$\begin{aligned} Att(\mathcal{X}_0, [P, X_1^{(S_1)}]) = \lambda(X_1^{(S_1)}, \mathcal{X}_0, [P, X_1^{(S_1)}]) Att(\mathcal{X}_0, X_1^{(S_1)}) \\ + \lambda(P, \mathcal{X}_0, [P, X_1^{(S_1)}]) Att(\mathcal{X}_0, P) \qquad (8) \end{aligned}$$

$$\begin{aligned} Att(\mathcal{X}_0, [P, X_1^{(S_2)}]) = \lambda(X_1^{(S_2)}, \mathcal{X}_0, [P, X_1^{(S_2)}]) Att(\mathcal{X}_0, X_1^{(S_2)}) \\ + \lambda(P, \mathcal{X}_0, [P, X_1^{(S_2)}]) Att(\mathcal{X}_0, P) \qquad (9) \end{aligned}$$

where $\lambda(\mathbf{X}, \mathbf{X}', \mathbf{X}'' = [\mathbf{X}_1, \mathbf{X}_2]) \in (0, 1)$ is a positive scalar defined as:

$$\lambda(\mathbf{X}_1, \mathbf{X}', \mathbf{X}'') = \frac{\sum_j \exp((\mathbf{W}_k \mathbf{X}_{:,j})^T (\mathbf{W}_q \mathbf{X}'))}{\sum_j \exp((\mathbf{W}_k \mathbf{X}''_{:,j})^T (\mathbf{W}_q \mathbf{X}'))}$$

Based on Equations (8) and (9), we learn that $Att(\mathcal{X}_0, P)$ is the intersection of $Cone(Att(\mathcal{X}_0, [P, X_1^{(S_1)}]), Att(\mathcal{X}_0, X_1^{(S_1)}))$ and $Cone(Att(\mathcal{X}_0, [P, X_1^{(S_2)}], Att(\mathcal{X}_0, X_1^{(S_2)}))$, where $Cone(\mathbf{a}_1, \cdot, \mathbf{a}_k) = \{x | x = \sum_{i=1}^k c_i \mathbf{a}_i, c_i > 0 (i = 1, \cdots, k)\}$ is a convex cone formed by $\mathbf{a}_i (i = 1, \cdots, k)$. However, due to the same deduction by the proof of Theorem 2 in [41], $Cone(Att(\mathcal{X}_0, [P, X_1^{(S_1)}]), Att(\mathcal{X}_0, X_1^{(S_1)}))$ and $Cone(Att(\mathcal{X}_0, [P, X_1^{(S_2)}], Att(\mathcal{X}_0, X_1^{(S_2)}))$ have no intersection based on Assumption 2, which contradicts the existence of $Att(\mathcal{X}_0, P)$. Therefore, there exists no common prompt P such that $f([P, X_1^{(S_i)}]) = Y_1^{(S_i)} (i = 1, 2)$.
(b). Secondly, we illustrate the case when both the common prompt P and prompt generators $g^{(S_i)}$ are available. In this case, Equations (8) and (9) become the following:

$$\begin{aligned} &Att(\mathcal{X}_0, [P + g^{(S_1)}(X_1^{(S_1)}), X_1^{(S_1)}]) \\ &= \lambda(X_1^{(S_1)}, \mathcal{X}_0, [P + g^{(S_1)}(X_1^{(S_1)}), X_1^{(S_1)}]) Att(\mathcal{X}_0, X_1^{(S_1)}) \\ &+ \lambda(P + g^{(S_1)}(X_1^{(S_1)}), \mathcal{X}_0, [P + g^{(S_1)}(X_1^{(S_1)}), X_1^{(S_1)}]) \\ &Att(\mathcal{X}_0, P + g^{(S_1)}(X_1^{(S_1)})) \qquad (10) \end{aligned}$$

$$\begin{aligned} &Att(\mathcal{X}_0, [P + g^{(S_2)}(X_1^{(S_2)}), X_1^{(S_2)}]) \\ &= \lambda(X_1^{(S_2)}, \mathcal{X}_0, [P + g^{(S_2)}(X_1^{(S_2)}), X_1^{(S_2)}]) Att(\mathcal{X}_0, X_1^{(S_2)}) \\ &+ \lambda(P + g^{(S_2)}(X_1^{(S_2)}), \mathcal{X}_0, [P + g^{(S_2)}(X_1^{(S_2)}), X_1^{(S_2)}]) \\ &Att(\mathcal{X}_0, P + g^{(S_2)}(X_1^{(S_2)})) \qquad (11) \end{aligned}$$

Obviously, the role of $g^{(S_i)}$ is to find $Att(\mathcal{X}_0, P + g^{(S_1)}(X_1^{(S_1)})) \in Cone(Att(\mathcal{X}_0, [P+g^{(S_1)}(X_1^{(S_1)}), X_1^{(S_1)}]), Att(\mathcal{X}_0, X_1^{(S_1)}))$ and $Att(\mathcal{X}_0, P + g^{(S_2)}(X_1^{(S_2)})) \in Cone(Att(\mathcal{X}_0, [P+g^{(S_2)}(X_1^{(S_2)}), X_1^{(S_2)}]), Att(\mathcal{X}_0, X_1^{(S_2)}))$ so that these two cones have no intersections, and therefore the contradiction mentioned in (a) can be addressed. □