DIVERSIFY TO GENERALIZE: LEARNING GENERALIZED REPRESENTATIONS FOR TIME SERIES CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series classification is an important problem in real world. Due to its non-stationary property that the distribution changes over time, it remains challenging to build models for generalization to unseen distributions. In this paper, we propose to view the time series classification problem from the *distribution* perspective. We argue that the temporal complexity attributes to the unknown latent distributions within. To this end, we propose DIVERSIFY to learn generalized representations for time series classification. DIVERSIFY takes an iterative process: it first obtains the worst-case distribution scenario via adversarial training, then matches the distributions between all segments. We also present some theoretical insights. Extensive experiments on gesture recognition, speech commands recognition, and sensor-based human activity recognition demonstrate that DIVERSIFY significantly outperforms other baselines while effectively characterizing the latent distributions by qualitative and quantitative analysis.

1 INTRODUCTION

Time series classification is one of the most challenging problems in machine learning and statistics community. Example applications include sensor-based human activity recognition, Parkinson's disease diagnosis, and electronic power consumption (Fawaz et al., 2019). One important nature of time series is non-stationary property, which means that its statistical features are changing over time. For years, there have been tremendous efforts to tackle the time series classification problem, such as hidden Markov models (Fulcher & Jones, 2014), RNN-based methods (Hüsken & Stagge, 2003), and Transformer-based approaches (Li et al., 2019).

In this paper, we are specifically interested in modeling time series from the *distribution* perspective. More precisely, we aim to learn representations for time series that can generalize to *unseen* distributions. Note that this scenario has been extensively studied in existing literature of domain generalization (Muandet et al., 2013; Wang et al., 2021a) and out-of-distribution generalization (Krueger et al., 2021), where researchers are keen to bridge the gap between known and unknown distributions, thus generalize well. While most of the efforts are done in the image context, few of them focus on the time series domain, which is more challenging. Although time series share a similar goal as image data in domain generalization, it naturally brings more challenges due to its non-stationary property: the distribution keeps changing over time, which contains diverse distribution information that should be harnessed well for better generalization.

We show an illustrative example in Figure 1. Domain generalization in image data often involves several domains and the domain information is known (subfigure (a)). Thus, we can leverage such domain information to build generalization models. However, in Figure 1 (b), we see that in time series data, although the distribution is changing dynamically over time, its domain information is not available. This will dramatically impede the modeling of existing domain generalization algorithms as they typically assume access to domain information (subfigure (c)).

In order to learn a generalized time series model under this setting, we propose DIVERSIFY, a domain generalization algorithm to characterize the latent distributions inside the time series data. Concretely speaking, our method consists of a min-max adversarial game that: on one hand, it learns to segment the time series data into several latent sub-domains by maximizing the segment-wise

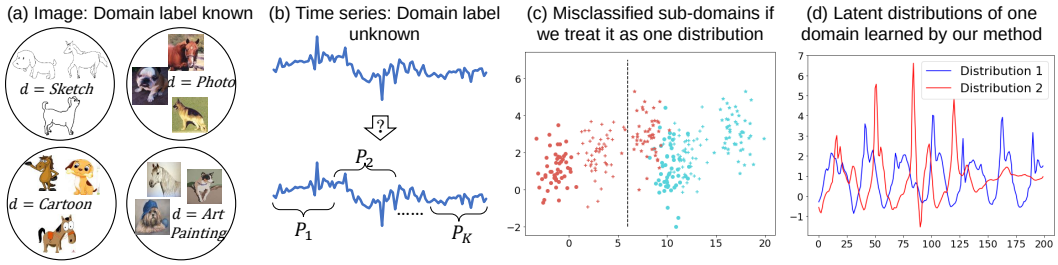


Figure 1: Illustration of DIVERSIFY: (a) Domain generalization for image data require known domain labels. (b) Domain label is unknown for time series. (c) If we treat the time series data as one single domain, the sub-domains will be misclassified. Different colors and shapes correspond to different classes and domains. (d) Finally, our DIVERSIFY can automatically learn the latent distributions from the dataset.

distribution gap to preserve diversities, i.e., the *worst-case* distribution scenario; on the other hand, it learns domain-invariant representations by reducing the distribution divergence for the worst-case scenario. Such diversification naturally exists in a complicated dataset where the data from multiple people naturally follow several latent distributions. Moreover, it is also surprising to find that even the data of one person still has such diversification: it can also be split into several latent distributions. Obviously, our DIVERSIFY can automatically characterize such latent distributions to learn more generalized representations (Figure 1 (d)).

We evaluate our approach in three types of time series classification tasks: gesture recognition, speech command recognition, and sensor-based human activity recognition. Extensive experiments demonstrate the superiority of DIVERSIFY on several challenging scenarios: on difficult tasks, significantly diverse datasets, and limited data. Moreover, the existence of latent distributions is certified in detailed analysis and visualization, providing experiences for future research.

2 METHODOLOGY

2.1 PROBLEM FORMULATION

We are given a time-series dataset $\mathcal{D}^{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ as the training dataset, where N is the number of samples, $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^p$ is the p-dimensional instance (sliding window) and $y_i \in \mathcal{Y} = \{1, \dots, C\}$ is its label. We use $\mathbb{P}^{tr}(\mathbf{x}, y)$ on $\mathcal{X} \times \mathcal{Y}$ to denote the joint distribution of the training dataset. Our goal is to learn a generalized model from $\mathcal{D}^{tr}$ to predict well on an *unseen* target dataset, $\mathcal{D}^{te}$, which is inaccessible in training. Like $\mathcal{D}^{tr}$, time series in $\mathcal{D}^{te}$ also be split into short series. In our problem, the training and test datasets have the same input and output spaces but different distributions, i.e., $\mathcal{X}^{tr} = \mathcal{X}^{te}$, $\mathcal{Y}^{te} = \mathcal{Y}^{te}$, but $\mathbb{P}^{tr}(\mathbf{x}, y) \neq \mathbb{P}^{te}(\mathbf{x}, y)$. We aim to train a model h from $\mathcal{D}^{tr}$ to minimize the risk on $\mathcal{D}^{te}$: $\min_h \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}^{te}}[h(\mathbf{x}) \neq y]$.

Note that due to the non-stationary property, the training dataset may be composed of several unknown latent distributions, instead of one fixed distribution, i.e., $\mathbb{P}^{tr}(\mathbf{x}, y) = \sum_{i=1}^K \pi_i \mathbb{P}^i(\mathbf{x}, y)$, where $\mathbb{P}^i(\mathbf{x}, y)$ is the distribution of the i-th sub-domains of the training data and π_i is its weight. K is the number of sub-domains that is unknown, and $\sum_{i=1}^K \pi_i = 1$.

2.2 MOTIVATION

The labeled time series data can be composed of several latent distributions (domains) that are challenging to characterize, even if the dataset is fully labeled. For instance, data collected by sensors of three persons may belong to two different distributions when considering their similarities. Moreover, even for data from one single person, different segments of one sequence may follow different distributions. In a nutshell, in reality, there often exist several sub-domains in one time series dataset.

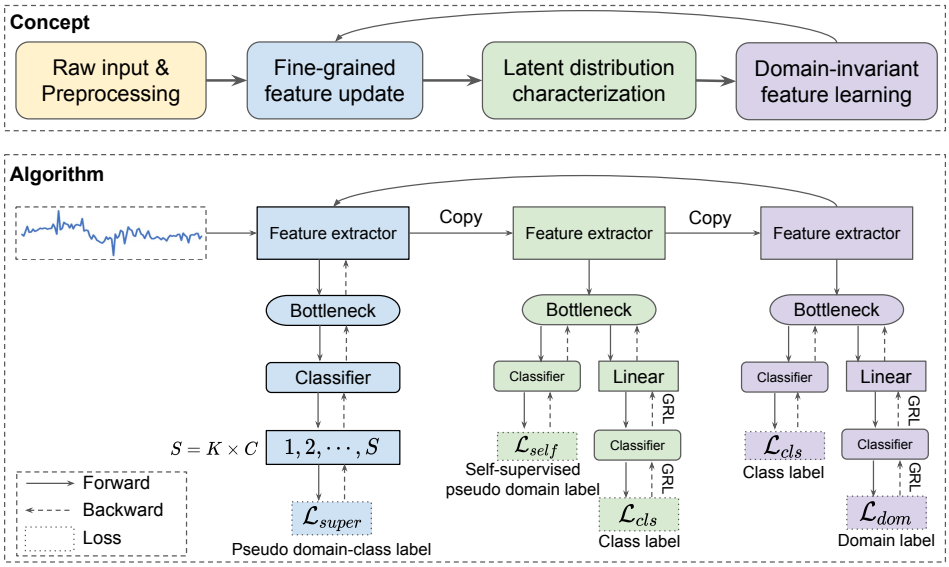


Figure 2: The framework of DIVERSIFY.

To ensure good generalization performance on the test dataset, it is important to learn *distribution-invariant*, or domain-invariant representations from the training dataset by characterizing its *latent* distributions. These latent distributions may contain both benign and malignant knowledge that influences generalization on the target dataset. In Figure 1 (c), we assume the source domain contains two sub-domains (circle and plus points). Directly learning from the entire source domain by treating it as one distribution may generate the black margin. Although green star data points and red star points can be classified easily, red star data points are misclassified to the green class when predicting on the out-of-distribution domain (star points) with the learned model. While other methods fail, our method can characterize the latent distributions, which will be introduced later.

2.3 DIVERSIFY

In this paper, we propose DIVERSIFY to learn generalized representations for time series classification. The core of DIVERSIFY is to characterize the latent distributions in a time series dataset and then to minimize the distribution divergence between each two. To characterize the diverse information contained in time series for better generalization, DIVERSIFY utilizes an iterative process: it first obtains the worst-case distribution scenario from a given dataset, then bridges the distribution gaps between each pair of latent distributions. Why the worst-case scenario? We argue that the worst-case scenario will maximally preserve the diverse information of each latent distribution, thus benefiting generalization.

Figure 2 describes the main procedures of DIVERSIFY:

1. Pre-processing: this step adopts the sliding window to split the entire training dataset into fixed-size windows. We believe that the data from one window is the smallest domain unit.
2. Fine-grained feature update: this step updates the feature extractor using the proposed *pseudo domain-class* labels as the supervision.
3. Latent distribution characterization: this step aims to identify the domain label for each instance to acquire the latent distribution information. It tries to *maximize* the different distribution gaps to enlarge diversity.
4. Domain-invariant representation learning: this step utilizes pseudo domain labels acquired from the last step to learn domain-invariant representations of data and train a generalizable classification model.

Note that steps $2 \sim 4$ are iterative as shown in Figure 2. Now we elaborate their details.

Fine-grained Feature Update Before we characterize the latent distributions, we perform fine-grained feature update to obtain useful representations. As shown in Figure 2 (blue), in order to fully utilize the knowledge contained in the domains and classes, we propose a new concept: *pseudo domain-class label*, which serves as the supervision for feature extractor. This will ensure that the feature extractor can learn fine-grained information which benefits the later steps.

At the first iteration, there is no domain label d' and we simply initialize $d' = 0$ for all samples. We treat per class per domain as a new class, and set the label to $s \in \{1, 2, \cdots, S\}$. We have $S = K \times C$ where K is the pre-defined number of latent distributions that can be tuned in experiments. We perform the following pseudo domain-class label assignment to get discrete values for supervision:

$$s = d' \times C + y. \tag{1}$$

Let $h_f^{(2)}, h_b^{(2)}, h_c^{(2)}$ be feature extractor, bottleneck, and classifier, respectively (we use superscript to denote step number), then, the supervised loss is computed using the cross-entropy loss ℓ:

$$\mathcal{L}_{super} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}^{tr}} \ell \left(h_c^{(2)}(h_b^{(2)}(h_f^{(2)}(\mathbf{x}))), s \right). \tag{2}$$

Latent Distribution Characterization This step characterizes the latent distributions contained in one dataset. As shown in Figure 2 (green), we employ an adversarial training strategy to disentangle the domain labels from the class labels. However, there are no actual domain labels provided, which hinders such disentanglement. Inspired by (Liang et al., 2020), we adopt a *self-supervised pseudo-labeling* strategy to obtain domain labels.

Firstly, we attain the centroid for each domain with class-invariant features:

$$\tilde{\mu}_k = \frac{\sum_{\mathbf{x}_i \in \mathcal{X}^{tr}} \delta_k(h_c^{(3)}(h_b^{(3)}(h_f^{(3)}(\mathbf{x}_i))))h_b^{(3)}(h_f^{(3)}(\mathbf{x}_i))}{\sum_{\mathbf{x}_i \in \mathcal{X}^{tr}} \delta_k(h_c^{(3)}(h_b^{(3)}(h_f^{(3)}(\mathbf{x}_i))))}, \tag{3}$$

where $h_f^{(3)}, h_b^{(3)}, h_c^{(3)}$ are feature extractor, bottleneck, and classifier, respectively. $\tilde{\mu}_k$ is the initial centroid of the kth latent sub-domains while δ_k is the kth element of the logit soft-max output. Then, we obtain the pseudo domain labels via the nearest centroid classifier using a distance function D:

$$\tilde{d}'_i = \arg\min_k D(h_b^{(3)}(h_f^{(3)}(\mathbf{x}_i)), \tilde{\mu}_k). \tag{4}$$

Then, we compute the centroids based on the new pseudo labels and obtain the updated pseudo domain labels:

$$\begin{aligned} \mu_k &= \frac{\sum_{\mathbf{x}_i \in \mathcal{X}^{tr}} \mathbb{I}(\tilde{d}'_i = k) h_b^{(3)}(h_f^{(3)}(\mathbf{x}))}{\sum_{\mathbf{x}_i \in \mathcal{X}^{tr}} \mathbb{I}(\tilde{d}'_i = k)}, \\ d'_i &= \arg\min_k D(h_b^{(3)}(h_f^{(3)}(\mathbf{x}_i)), \mu_k), \end{aligned} \tag{5}$$

where $\mathbb{I}(a) = 1$ when a is true, otherwise 0. After we obtain d', we can compute the goal of step 2:

$$\mathcal{L}_{self} + \mathcal{L}_{cls} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}^{tr}} \ell(h_c^{(3)}(h_b^{(3)}(h_f^{(3)}(\mathbf{x}))), d') + \ell(h_{adv}^{(3)}(R_{\lambda_1}(h_b^{(3)}(h_f^{(3)}(\mathbf{x})))), y), \tag{6}$$

where $h_{adv}^{(3)}$ is the discriminator for step 3 that contains several linear layers and one classification layer. R_{λ_1} is the gradient reverse layer with hyperparameter λ_1 (Ganin et al., 2016). After this step ends, we can obtain pseudo domain label d' for $\mathbf{x}$.

Domain-invariant Representation Learning After obtaining the latent distributions, we learn domain-invariant representations for generalization. In fact, this step (Figure 2 purple) is simple: we borrow the idea from DANN (Ganin et al., 2016) and directly use adversarial training to update the classification loss $\mathcal{L}_{cls}$ and domain classifier loss $\mathcal{L}_{dom}$ using GRL:

$$\mathcal{L}_{cls} + \mathcal{L}_{dom} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}^{tr}} \ell(h_c^{(4)}(h_b^{(4)}(h_f^{(4)}(\mathbf{x}))), y) + \ell(h_{adv}^{(4)}(R_{\lambda_2}(h_b^{(4)}(h_f^{(4)}(\mathbf{x})))), d'), \tag{7}$$

where ℓ is the cross-entropy loss and R_{λ_2} is the gradient reverse layer with hyperparameter λ_2 (Ganin et al., 2016).

We repeat the above three steps until convergence or max epochs. The final model is selected via a validation dataset split from the source domain (Gulrajani & Lopez-Paz, 2021).

As for inference, we predict its label with the modules from the last step.

2.4 THEORETICAL INSIGHTS

We present some theoretical insights to show that our approach is well motivated in theory.

Proposition 2.1. *(Proposition 3.1 in (Sicilia et al., 2021)) Let $\mathcal{X}$ be a space and let $\mathcal{H}$ be a class of hypotheses corresponding to this space. Let $\mathbb{Q}$ and the collection $\{\mathbb{P}_i\}_{i=1}^k$ be distributions over $\mathcal{X}$ and let $\{\varphi_i\}_{i=1}^k$ be a collection of non-negative coefficients with $\sum_i \varphi_i = 1$. Let the object $\mathcal{O}$ be a set of distributions such that for every $\mathbb{S} \in \mathcal{O}$ the following holds*

$$\sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{S}) \leq \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j). \tag{8}$$

Then, for any $h \in \mathcal{H}$, the following equality holds:

$$\varepsilon_{\mathbb{Q}}(h) \leq \lambda_\varphi + \sum_i \varphi_i \varepsilon_{\mathbb{P}_i}(h) + \frac{1}{2} \min_{\mathbb{S}\in\mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q}) + \frac{1}{2} \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j), \tag{9}$$

where $\lambda_\varphi = \sum_i \varphi_i \lambda_i$ and each λ_i is the error of an ideal joint hypothesis for $\mathbb{Q}$ and $\mathbb{P}_i$. $\varepsilon_{\mathbb{P}}(h)$ is the error for a hypothesis h on a distribution $\mathbb{P}$. $d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q})$ is $\mathcal{H}$-divergence which measure differences in distribution (Ben-David et al., 2010).

The first item in Eq. 9, λ_φ, is often neglected since it is small in reality. The second item, $\sum_i \varphi_i \varepsilon_{\mathbb{P}_i}(h)$, exists in almost all methods and can be minimized via supervision from class labels with cross-entropy loss in Eq. 7. Our main purpose is to minimize the last two items in Eq. 9. Here $\mathbb{Q}$ corresponds to the unseen out-of-distribution target domain.

The last term $\frac{1}{2} \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j)$ is common in DA and DG which measures the maximum differences among source domains. This corresponds to step 3 in our approach.

We finally analyze the third item, $\frac{1}{2} \min_{\mathbb{S}\in\mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q})$, which explains why we exploit sub-domains in step 2. Since our goal is to learn a model which can perform well on an unseen target domain, we cannot obtain $\mathbb{Q}$. To minimize $\frac{1}{2} \min_{\mathbb{S}\in\mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q})$, we can only enlarge the range of $\mathcal{O}$. We have to $\max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j)$ according to Eq. 8, corresponding to step 2 in our method which tries to segment the time series data into several latent sub-domains by maximizing the segment-wise distribution gap to preserve diversities, i.e., the worst-case distribution scenario.

3 EXPERIMENTS

We extensively evaluate our approach in three time series classification applications: gesture recognition, speech commands recognition, and sensor-based human activity recognition. These applications represent diverse situations of time series data and thus can thoroughly reflect the advantage of our method. We will introduce the detail of each application later.

For all experiments, we conduct the training-domain-validation strategy following DomainBed (Gulrajani & Lopez-Paz, 2021). The training data are split by $8:2$ for training and validation. For comparison, we re-implement eight recent strong comparison methods following DomainBed: ERM, DANN (Ganin et al., 2016), CORAL (Sun & Saenko, 2016), Mixup (Zhang et al., 2018), GroupDRO (Sagawa et al., 2020), RSC (Huang et al., 2020), ANDMask (Parascandolo et al., 2021), and GILE (Qian et al., 2021). Note that most methods require the domain labels unknown in training while our does not, which is more challenging and practical. We extensively tune all methods to report the averaged best performance of three trials.

3.1 GESTURE RECOGNITION

We first evaluate our method on a type of time-series data, Electromyography (EMG) that is based on bioelectric signals. We use EMG for gestures Data Set (Lobov et al., 2018) which contains raw EMG data recorded by MYO Thalmic bracelet. The bracelet is equipped with eight sensors equally spaced around the forearm that simultaneously acquire myographic signals. Data of 36 subjects are collected while they performed series of static hand gestures and the number of instances is about 40,000-50,000 recordings in each column. The dataset contains 7 classes in total and we select 6 common classes for our experiments. We randomly divide 36 subjects into four domains and each domain contains data of 9 persons. Experiments are done in three random trials, thus mitigating influence of unfair splits.

Since EMG data comes from bioelectric signals, it is affected by many factors. The bioelectric signals are very weak and require lots of processing before they can be used. EMG data are scene and device-dependent, which means the same person may generate different data when performing the same activity with the same device at a different time or with the different devices at the same time. Therefore, the benchmark of EMG is more challenging. From Table 1, we can see that our method achieves the best average performance on this benchmark and is $\mathbf{4.3}\%$ better than the second-best method.

Table 1: Results on EMG dataset.

Target	0	1	2	3	AVG
ERM	62.6	69.9	67.9	69.3	67.4
DANN	62.9	70.0	66.5	68.2	66.9
CORAL	66.4	74.6	71.4	<u>74.2</u>	71.7
Mixup	60.7	69.9	70.5	68.2	67.3
GroupDRO	67.6	<u>77.4</u>	<u>73.7</u>	72.5	<u>72.8</u>
RSC	<u>70.1</u>	74.6	72.4	71.9	72.2
ANDMask	66.5	69.1	71.4	68.9	69.0
DIVERSIFY	**71.7**	**82.4**	**76.9**	**77.3**	**77.1**

3.2 SPEECH COMMANDS

Then, we adopt a regular speech recognition task, the Speech Commands dataset (Warden, 2018). The dataset consists of one-second audio recordings of both background noise and spoken words such as 'left', 'right', etc. It is collected from more than 2,000 persons, thus is more complicated. Following (Kidger et al., 2020), we only use 34,975 time series corresponding to ten spoken words so as to produce a balanced classification problem. 20% of selected data serve as validation dataset, 20% serve as testing dataset, and the rest are for training. Since this dataset is collected from multiple persons, thus the training and test distributions are different, which is also an OOD problem with one training domain.

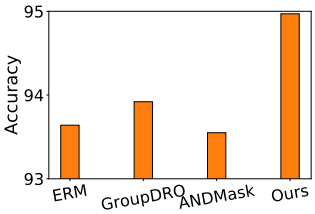


Figure 3: Results on Speech commands dataset.

Figure 3 shows the recognition results. Although it is completely a regular benchmark, our method still achieves the best performance. There are too many subjects and each subject only records few audios. In addition, an audio recording is so short that we do not split each sample. Compared with GroupDRO, DIVERSIFY still has over $\mathbf{1}\%$ improvement (its baseline is already very strong). The results on this benchmark demonstrate the superiority of our method on a regular time-series benchmark composed of massive distributions.

3.3 SENSOR-BASED HUMAN ACTIVITY RECOGNITION

Finally for large-scale and more detailed evaluation, we conduct more experiments on several popular sensor-based human activity recognition datasets. UCI daily and sports dataset (**DSADS**) (Barshan & Yüksek, 2014) consists of 19 activities collected from 8 subjects wearing body-worn sensors on 5 body parts. USC-SIPI human activity dataset (**USC-HAD**) (Zhang & Sawchuk, 2012) is composed of 14 subjects (7 male, 7 female, aged from 21 to 49) executing 12 activities with a sensor tied on the front right hip. **UCI-HAR** (Anguita et al., 2012) is collected by 30 subjects performing 6 daily living activities with a waist-mounted smartphone. PAMAP2 physical activity monitoring dataset **PAMAP2** (Reiss & Stricker, 2012) contains data of 18 different physical activities, performed by 9 subjects wearing 3 sensors.

We construct *four* different settings representing different degrees of generalization: (1) **Cross-person generalization**: This setting utilizes DSADS, USC-HAD, PAMAP2[1] datasets to construct three benchmarks. Within each dataset, we randomly split the data into four groups[2] and then use three groups as training data to learn a generalized model for the last group. (2) **Cross-position generalization**: this setting utilizes DSADS dataset and data from each position corresponds to a different domain. Each sample contains three sensors with nine dimensions. We treat one position as the test domain while the others are for training. (3) **Cross-dataset generalization**: this setting utilizes all four datasets, and each dataset corresponds to a different domain. Six common classes

[1]We do not use UCI-HAR in cross-person setting since its baseline is good enough.

[2]For DSADS, 8 subjects are divided into four groups and each sample has three sensors on five positions (15 sensors in total) with 45 dimensions. For USC-HAD, 14 subjects are divided into four groups and each sample has two sensors on one position with six dimensions. For PAMAP2, 9 subjects are divided into four groups and each sample has three sensors on three positions (9 sensors in total) with 27 dimensions. Six classes are discarded due to too few samples.

are selected. Two sensors from each dataset that belong to the same position are selected and data is down-sampled to ensure the dimension of data same. (4) **One-Person-To-Another**. This setting utilizes USC-HAD dataset to randomly select two persons where one is the training and the other is the test domain. For simplicity, we use $0, 1, \cdots$ to denote different domains. All experiments are done in three random trials, mitigating influence of unfair splits or constructions.

Table 2: Classification accuracy on cross-person generalization.

	DSADS					USC-HAD					PAMAP					ALL
Target	0	1	2	3	AVG	0	1	2	3	AVG	0	1	2	3	AVG	AVG
ERM	83.1	79.3	87.8	71.0	80.3	81.0	57.7	74.0	65.9	69.7	90.0	78.1	55.8	84.4	77.1	75.7
DANN	89.1	84.2	85.9	83.4	85.6	81.2	57.9	76.7	70.7	71.6	82.2	78.1	55.4	87.3	75.7	77.7
CORAL	91.0	85.8	86.6	78.2	85.4	78.8	58.9	75.0	53.7	66.6	86.2	77.8	49.0	87.8	75.2	75.7
Mixup	89.6	82.2	89.2	**86.9**	87.0	80.0	**64.1**	74.3	61.3	69.9	89.4	80.3	58.4	87.7	79.0	78.6
GroupDRO	**91.7**	85.9	87.59	78.3	85.9	80.1	55.5	74.7	60.0	67.6	85.2	77.7	56.2	85.0	76.0	76.5
RSC	84.9	82.3	86.7	77.7	82.9	81.9	57.9	73.4	65.1	69.6	87.1	76.9	60.3	**87.8**	78.0	76.9
ANDMask	85.0	75.8	87.0	77.6	81.4	79.9	55.3	74.5	65.0	68.7	86.7	76.4	43.6	85.6	73.1	74.4
GILE	81.0	75.0	77.0	66.0	74.7	78,0	62.0	77.0	63.0	70.0	83.0	68.0	42.0	76.0	67.5	70.7
DIVERSIFY	90.4	**86.5**	**90.0**	86.1	**88.2**	**82.6**	63.5	**78.7**	**71.3**	**74.0**	**91.0**	**84.3**	**60.5**	87.7	**80.8**	**81.0**

Table 3: Classification accuracy on cross-position and cross-dataset generalization.

	Cross-position generalization						Cross-dataset generalization				
Target	0	1	2	3	4	AVG	0	1	2	3	AVG
ERM	41.5	26.7	35.8	21.4	27.3	30.6	26.4	29.6	44.4	32.9	33.3
DANN	45.4	25.3	38.1	28.9	25.1	32.6	29.7	45.3	46.1	43.8	41.2
CORAL	33.2	25.2	25.8	22.3	20.6	25.4	39.5	41.8	39.1	36.6	39.2
Mixup	**48.8**	**34.2**	37.5	29.5	29.9	36.0	37.3	**47.4**	40.2	23.1	37.0
GroupDRO	27.1	26.7	24.3	18.4	24.8	24.3	**51.4**	36.7	33.2	33.8	38.8
RSC	46.6	27.4	35.9	27.0	29.8	33.3	33.1	39.7	45.3	45.9	41.0
ANDMask	47.5	31.1	39.2	30.2	29.9	35.6	41.7	33.8	43.2	40.2	39.7
DIVERSIFY	47.7	32.9	**44.5**	**31.6**	**30.4**	**37.4**	48.7	46.9	**49.0**	**59.9**	**51.1**

Table 2, Table 3, and Figure 4 show the results on four settings for HAR. For all settings for HAR, our method achieves the best average performances. For Cross-person, Cross-dataset, Cross-position, and One-Person-To-Another settings, our method significantly outperforms the second-best baseline by $\mathbf{2.4}\%$, $\mathbf{9.9}\%$, $\mathbf{1.4}\%$ and $\mathbf{3.3}\%$ respectively. These results all demonstrate the superiority of DIVERSIFY in different generalization settings.

We observe more insightful conclusions. (1) *When the task is difficult:* In the Cross-Person setting, USC-HAD may be the most difficult task. Although it includes more samples, it contains 14 subjects with only two sensors on one position, which may bring more sub-domains in this situation. The experimental results prove the above argument that all methods perform terribly on this benchmark while ours has the largest improvement. (2) *When datasets are significantly more diverse:* Compared to Cross-Person and Cross-Position settings, Cross-Dataset may be more difficult since all datasets are totally different that samples are influenced by subjects, devices, sensor positions, and some other factors. In this setting, our method is substantially better than others. (3) *Limited data:* Compared with Cross-Person setting, One-Person-To-Another is more difficult since it has fewer data samples. Therefore, in this situation, enhancing diversity can bring a remarkable improvement and our method can boost the performance.

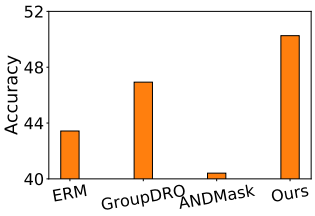
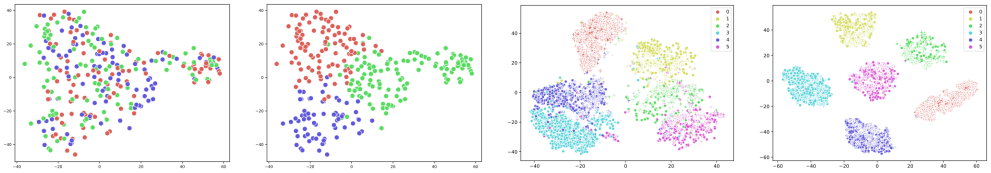


Figure 4: The results on One-Person-To-Another.

3.4 ANALYSIS

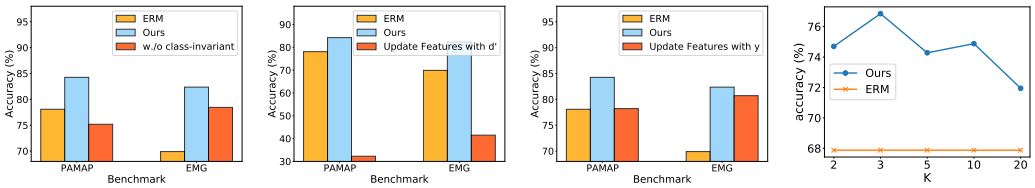
Visualization Study We present some visualizations to show the rationales of our method and more results are in appendix. Data points with different initial domain labels are mixed together in Figure 5(a) while our DIVERSIFY can characterize different latent distributions to separate them well in Figure 5(b). It illustrates that our method can find better latent sub-domains. From Figure 5(d) and Figure 5(c), we can see that our method can learn better domain-invariant representations compared to the latest method ANDMask. Therefore, DIVERSIFY can find better latent sub-domains and then find domain-invariant representation from split domains, which can finally enhance generalization.



(a) Initial splits (b) Ours (c) ANDMask (d) Ours

Figure 5: t-SNE visualizations for domain splits ((a) and (b)) and classification ((c) and (d)).(EMG)

Ablation Study *Why obtaining pseudo domain labels with class-invariant features in step 3?* If we obtain pseudo domain labels with common features, domain labels may have correlations with class labels, which may introduce contradictions when learning domain-invariant representations and lead common performance. This is certified by the results in Figure 6(a).



(a) Class-invariant feat. (b) Update feat. with d' (c) Update feature with y (d) #Latent domains K

Figure 6: Ablation study.

Why using fine-grained domain-class labels in step 2? If we utilize pseudo domain labels to update the feature net, it may make the representations seriously biased towards domain-related features and thereby leads to terrible performance on classification, which is proved in Figure 6(b). If we only utilize class labels to update the feature net, it may make representations biased to class-related features, make DIVERSIFY unable to obtain true latent sub-domains, and thereby has unremarkable performance, which is proved in Figure 6(c). Therefore, we should utilize fine-grained domain-class labels to obtain representations with both domain and class information.

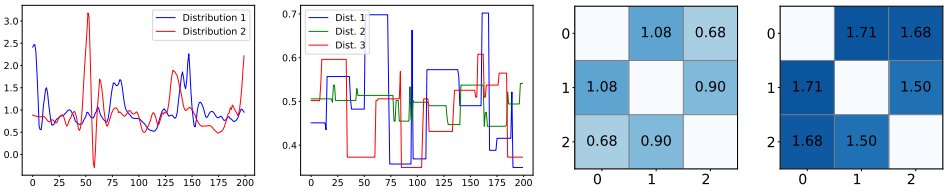
The more latent sub-domains, the better? More latent sub-domains may not bring better results as shown in Figure 6(d). A dataset may only contain a few sub-domains and introducing more may have a conflict with the intrinsic data property. Moreover, introducing more latent sub-domains also brings difficulty to obtain pseudo domain labels and learn domain-invariant features, which may have negative effects.

Existence of latent distributions. Latent sub-domains widely exist in time series, even in data collected from one person when performing the same activity. As shown in Figure 7(a), for a subject in USC-HAD dataset, there is more than one latent distribution for his walking activities. Figure 7(b) demonstrates similar phenomena exist in EMG dataset where our algorithm found three latent distributions.

Quantitative analysis for worst-case distributions. We present quantitative analysis by computing the $\mathcal{H}$-divergence (Ben-David et al., 2010) to show the effectiveness of our 'worst-case distribution'. As shown in Figure 7(c) and Figure 7(d), compared with initial domain splits, latent sub-domains generated by our method have larger $\mathcal{H}$-divergence among each other. According to Prop. 2.1, larger $\mathcal{H}$-divergence among domains can bring better generalization.

4 RELATED WORK

Time series classification is a challenging problem for years. Existing researches mainly focus on the modeling of temporal relations using either RNN-based methods or the recently-proposed



(a) USC-HAD (b) EMG (c) Initial splits (d) Our splits

Figure 7: (a) (b) Latent distributions obtained by our approach on two datasets. (c) (d) $\mathcal{H}$-divergence among domains with initial splits and our splits on PAMAP.

Transformer architecture. To our best knowledge, there is only one recent work (Du et al., 2021) that studied time series from the distribution level. However, their method is not end-to-end trainable since they developed a two-stage method that is non-differential.

Domain / OOD generalization (Wang et al., 2021a) mainly consists of data augmentation (Zhang et al., 2018; Yue et al., 2019), domain-invariant feature learning (Muandet et al., 2013; Ganin et al., 2016), and meta-learning (Li et al., 2018) techniques. The work of (Matsuura & Harada, 2020) also studied DG without knowing the domain labels by clustering with the style features for image data. However, that work is not applied to time series and it is not end-to-end trainable due to the clustering approaches. Disentanglement (Peng et al., 2019; Wang et al., 2021b) tries to disentangle the domain and label information, but they also assume access to domain information.

Single domain generalization is similar to our problem setting that also involves one training domain (Fan et al., 2021; Li et al., 2021; Qiao et al., 2020; Wang et al., 2021c). Most of them adopted generative models and other data augmentation strategies. However, they treated the single domain as *one* distribution and did not explore the latent sub-distributions. Specifically, Wang et al. (2021c) proposed a 'learning-to-diversify' algorithm, whose name is similar to ours, but they focused on mutual information maximization which is different from ours.

Mixture models (Rasmussen et al., 1999) are models representing the presence of subpopulations within an overall population, e.g., Gaussian mixture models. Our approach belongs to this category in general, while our focus is on domain generalization. Moreover, our approach does not rely on generative models as most of the mixture models do.

Subpopulation shift is a new setting introduced in (Koh et al., 2021), which refers to the situation that the training and test domains overlap, but their relative proportions differ. Our problem does not belong to this setting since we assume that the training and test distributions do not overlap.

Distributionally robust optimization (Delage & Ye, 2010) shares a similar paradigm with our work, whose paradigm is also to seek a distribution that has the worst performance within a range of the raw distribution. However, we study the internal distribution shift instead of seeking a global distribution close to the original one. GroupDRO (Sagawa et al., 2020) studied the DRO at a group level, which is significantly different from our idea.

5 Conclusions

We proposed DIVERSIFY to characterize the latent distributions from one domain for better generalization. DIVERSIFY employs an adversarial game that maximizes the worst-case distribution scenario while minimizing their distribution divergence. Our experiments on generalization from different degrees of difficulty demonstrated its effectiveness. We are surprised by the observations that not only a mixed dataset, but one dataset from a single person can also contain several latent distributions. Characterizing such latent distributions will great improve the generalization performance on unseen test datasets.

In the future, we plan to extend our algorithm to make it simpler in concept and easier to train. This can be done by seeking new architectures and distance metrics for computation and distribution matching. Moreover, we also plan to apply our algorithm to other time series-related applications such as forecasting and video classification.

REFERENCES

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*, pp. 216–223. Springer, 2012.

Billur Barshan and Murat Cihan Yüksek. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *The Computer Journal*, 57(11): 1649–1667, 2014.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. *arXiv preprint arXiv:2108.04443*, 2021.

Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8208–8217, 2021.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

Ben D Fulcher and Nick S Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3026–3037, 2014.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 124–140. Springer, 2020.

Michael Hüsken and Peter Stagge. Recurrent neural networks for time series classification. *Neurocomputing*, 50:223–235, 2003.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural Controlled Differential Equations for Irregular Time Series. *Advances in Neural Information Processing Systems*, 2020.

Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 224–233, 2021.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32:5243–5253, 2019.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 6028–6039. PMLR, 2020.

Sergey Lobov, Nadia Krilova, Innokentiy Kastalskiy, Victor Kazantsev, and Valeri A Makarov. Latent factors limiting the performance of semg-interfaces. *Sensors*, 18(4):1122, 2018.

Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *AAAI*, 2020.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. In *ICLR*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. volume 32, pp. 8026–8037, 2019.

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*, pp. 5102–5112. PMLR, 2019.

Hangwei Qian, Sinno Jialin Pan, Chunyan Miao, H Qian, SJ Pan, and C Miao. Latent independent excitation for generalizable sensor-based cross-person activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11921–11929, 2021.

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12556–12565, 2020.

Carl Edward Rasmussen et al. The infinite gaussian mixture model. In *NIPS*, volume 12, pp. 554–560. Citeseer, 1999.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pp. 108–109. IEEE, 2012.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Anthony Sicilia, Xingchen Zhao, and Seong Jae Hwang. Domain adversarial neural networks for domain generalization: When it works and how to improve. *arXiv preprint arXiv:2102.03924*, 2021.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*, 2021a.

Yufei Wang, Haoliang Li, Lap-Pui Chau, and Alex C Kot. Variational disentanglement for domain generalization. *arXiv preprint arXiv:2109.05826*, 2021b.

Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *ICCV*, 2021c.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2100–2110, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Mi Zhang and Alexander A Sawchuk. Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pp. 1036–1043, 2012.

A Datasets Information

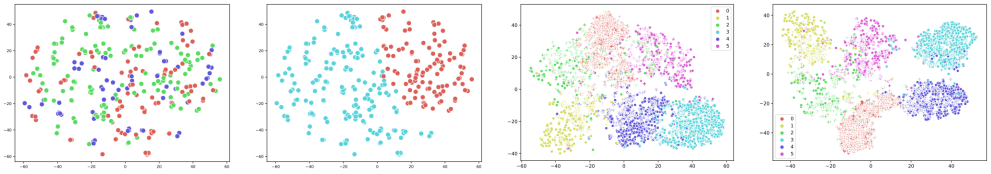
Table 4 shows the statistical information on each dataset.

Table 4: Information on HAR datasets.

Dataset	Subjects	Sensors	Classes	Samples
EMG	36	1	7	33,903,472
SPCMD	2618	-	35	105,829
DSADS	8	3	19	1,140,000
USC-HAD	14	2	12	5,441,000
UCI-HAR	30	2	6	1,310,000
PAMAP	9	3	18	3,850,505

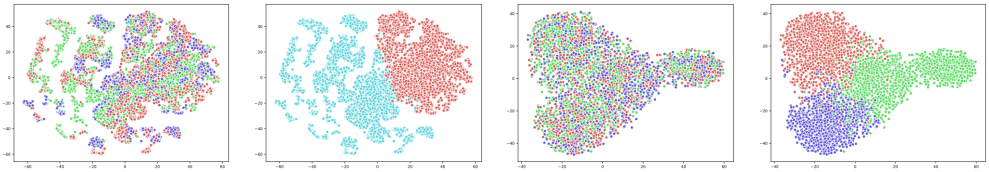
B Network Architecture and Hyperparameters

For the architecture, the model contains two blocks, and each has one convolution layer, one pool layer, and one batch normalization layer. A single-fully-connected layer is used as the bottleneck layer while another fully-connected layer serves as the classifier. All methods are implemented with PyTorch (Paszke et al., 2019). The maximum training epoch is set to 150. The Adam optimizer with weight decay 5×10^{-4} is used. The learning rate for GILE is 10^{-4}. The learning rate for the rest methods is 10^{-2} for the first three settings and 10^{-3} for the last setting. We tune hyperparameters for each method.



(a) Initial splits (b) Ours (c) ERM (d) RSC

Figure 8: Visualization of the t-SNE embeddings for domain splits and classification.



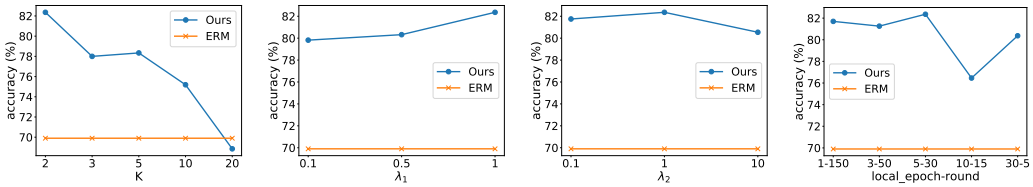
(a) Initial splits on USC (b) Ours on USC (c) Initial splits on EMG (d) Ours on EMG

Figure 9: Visualization of the t-SNE embeddings for domain splits.

C VISUALIZATION STUDY

We show more visualization study in this part. As shown in Figure 9(c) and Figure 9(d), compare with initial domain splits, latent sub-domains generated by our method are better separated. As shown in Figure 9(c) and Figure 9(d), both ERM and RSC also cannot obtain fine domain-invariant representations. Figure 9 shows complete visualization for domain splits.

D PARAMETER SENSITIVITY

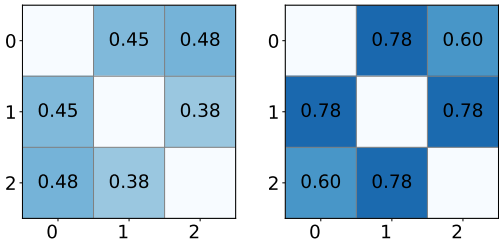


(a) #Latent domains K (b) λ_1 (c) λ_2 (d) Local epoch and Round

Figure 10: Parameter sensitivity analysis.

There are mainly four hyperparameters in our method: K which is the number of latent sub-domains, λ_1 for the adversarial part in step 3, λ_2 for the adversarial part in step 4, and local epochs and total rounds. For fairness, the product of local epochs and total rounds is the same value. We evaluate the parameter sensitivity of our method in Figure 10 where we change one parameter and fix the other to record the results. From these results, we can see that our method achieves better performance in a wide range, demonstrating that our method is robust.

E $\mathcal{H}$-DIVERGENCE



(a) Initial splits (b) Our splits

Figure 11: $\mathcal{H}$-divergence among domains with initial splits and our splits on EMG.

Figure 11 shows $\mathcal{H}$-divergence among domains with initial splits and our splits on EMG, which demonstrates our splits have larger $\mathcal{H}$-divergence and thereby can bring better generalization.