SELECTIVE CROSS-DOMAIN CONSISTENCY REGULARIZATION FOR TIME SERIES DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization aims to learn models robust to domain shift, with limited source domains at training and without any access to target domain samples except at test time. Current domain alignment methods seek to extract features invariant across all domains, but do not consider inter-domain relationships. In this paper, we propose a novel representation learning methodology for time series classification that selectively enforces prediction consistency between source domains estimated to be closely-related. Specifically, we view a domain shift as a form of data transformation that preserves labels but not necessarily class relationships, and we regularize the predicted class relationships to be shared only by closely-related domains instead of all domains to prevent negative transfer. We conduct comprehensive experiments on two public real-world datasets. The proposed method significantly improves over the baseline and achieves better or competitive performance in comparison with state-of-the-art methods.

1 INTRODUCTION

Increasing accessibility to data has spurred the use of data-driven machine learning methods, especially deep learning. In practical deployments, models need to be robust to data distribution shifts between training and test data, also known as domain shift (Gulrajani & Lopez-Paz, 2021; Hendrycks et al., 2020). Domain shift may occur as data collection is subject to resource constraints and may not provide sufficient coverage, or is conducted in controlled settings that do not fully assimilate real environments (Li et al., 2018; Zhang et al., 2020; Gupta et al., 2018).

Examples of domains are operating conditions for machine fault detection (Zheng et al., 2020), human subjects for activity recognition (Wilson et al., 2020), product category for sentiment analysis (Ganin et al., 2016; Guo et al., 2018; Balaji et al., 2018), art styles for image classification (Li et al., 2019; Mancini et al., 2018; Somavarapu et al., 2020; Li et al., 2018; Carlucci et al., 2019), and different equipments and practices of data collection (Dou et al., 2019; Gong et al., 2019; Mahajan et al., 2020). Taking machine fault detection for example, a domain shift occurs when the available samples for training are collected when the machine is operating under a limited set of conditions *(source domains)*, while at test time the machine is subject to a different condition *(target domain)* where faults may manifest differently from those in the training conditions.

We consider the *domain generalization* problem in time series classification in this work, where we want to learn a model robust to domain shift without any access to target domain samples during training. This differs from *domain adaptation* where unlabelled target domain samples are available for training. While there are many works studying domain generalization in image classification tasks (Hendrycks et al., 2020; Gulrajani & Lopez-Paz, 2021), there is limited literature and limited evaluation of existing methods for time series classification. We find it important to fill this gap to develop suitable methods for time series applications. We work in the conventional setting with multiple source domains, and where all domains share the same label space.

In this paper, we view domain shifts as data transformations that are label preserving, but do not necessarily preserve class relationships. For example, in classifying between 'walking' and 'running' actions, samples from subjects who tend to walk at a faster pace are expected to generate more

class confusion. Hence, we assume there exists latent inter-domain relationships, such that similar domains should result in similar predicted class relationships.

We propose selective cross-domain consistency regularization to encourage similar model predictions on source domains that are estimated to be more closely related to each other. We further use domain-wise time series data augmentation to increase robustness to time series perturbations. Our consistency regularization is applied on the logits (pre-softmax classifier outputs). As far we know, existing domain generalization methods regularize on either features or soft label predictions (Ganin et al., 2016; Li et al., 2018; Sun & Saenko, 2016; Ben-David et al., 2010; Dou et al., 2019; M. & H., 2020; Gulrajani & Lopez-Paz, 2021; Wang & Deng, 2018; Wang et al., 2020b; Ahmed et al., 2021) or on logits in conjunction with features (Kim et al., 2021), which we find empirically produce worse generalization in our time series classification experiments. Moreover, we regularize subsets of source domains separately based on their similarity, instead of regularizing all source domains together as common in existing domain alignment methods (Gulrajani & Lopez-Paz, 2021; Wang et al., 2021), to prevent negative transfer. Figure 1 and 2 provide an overview of our proposed method. We introduce two versions of the method for when auxiliary domain metadata is available or unavailable. When such metadata is available, inter-domain relationships amongst source domains can be directly inferred based on application-specific knowledge. Otherwise, we estimate latent relationships by the inherent prediction similarity amongst source domains.

Our contributions in this work are:

- We propose a new domain generalization methodology for time series classification that selectively enforces prediction consistency between source domains estimated to be closely-related, which helps to calibrate the model from being over-confident in its predictions;
- The proposed method is easy to implement with domain-wise time series data augmentation and logit regularization on top of empirical risk minimization. We provide two versions for when source domain metadata are known or unknown to estimate domain relationships;
- We provide new benchmark evaluation results of existing domain generalization methods on two public time series datasets, and we show that the proposed method demonstrates better or competitive performance compared to state-of-the-art methods.

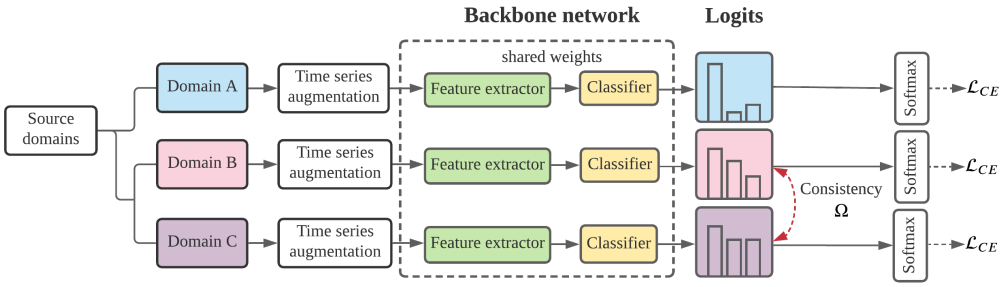
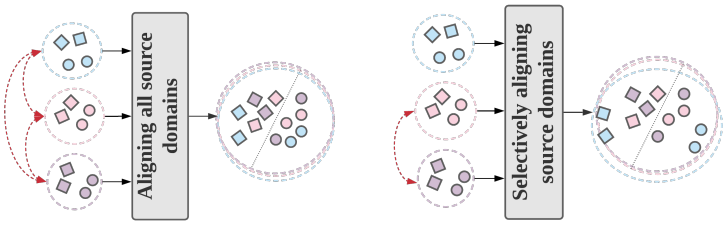


Figure 1: Overview: Considering multiple source domains, our proposed method applies time series augmentations on input samples and enforces prediction consistency between similar domains through selective regularization of logits (pre-softmax classifier outputs). We encourage similar domains to share predicted class relationships, while allowing diverse predicted class relationships across dissimilar domains.

2 Related Works

There is limited literature on domain generalization for time series classification. We include literature on general domain generalization in the review. Aside from one line of work that ensemble model predictions (Mancini et al., 2018; Vinyals et al., 2016; D'Innocente & Caputo, 2019; Guo et al., 2018) which is potentially not scalable with multiple source domains, domain generalization methods mainly attempt to learn more robust representations with a single model. In the following we discuss related works according to strategies used to learn these representations.



(a) Convention (b) Proposed method

Figure 2: Conventional domain alignment methods align all source domains equally and can result in over-regularization. Our proposed method allows greater diversity in output predictions.

Data augmentation and generation: Enriching training sample diversity naturally helps the model to generalize. (Volpi et al., 2018; Shankar et al., 2018; Stutz et al., 2019) generates new samples by adversarial perturbation of original training samples, and (Yan et al., 2020; Wang et al., 2020a) interpolate between samples of different domains. Due to advancements in style transfer (Jing et al., 2020) and generative adversarial networks (Goodfellow et al., 2014), many image augmentation techniques are developed (Hendrycks et al., 2020; Zhou et al., 2020; Nam et al., 2019; Gong et al., 2019). However, advanced techniques for image augmentation may not readily apply to time series.

Learning domain-invariant features: An approach to learn invariant features that do not carry any domain-specific information is to train the feature extractor such that the same classifier is optimal for all domains (Arjovsky et al., 2019; Rosenfeld et al., 2020). A large number of works learn invariant features by aligning the distribution of representations from all source domains (Albuquerque et al., 2020; Li et al., 2018; Mahajan et al., 2020). This technique builds on theoretical results in domain adaptation where the target risk is bounded by a divergence between source and target domains (Ben-David et al., 2010; Ganin et al., 2016). Some works directly minimize the distance between the learned features or soft labels of source domains by a distance measure or adversarial networks (Motiian et al., 2017; Wilson et al., 2020; M. & H., 2020; Li et al., 2018; Long et al., 2018), and some others use meta-learning to simulate domain shift during training (Balaji et al., 2018; Dou et al., 2019; Li et al., 2018). However, theoretical results from domain adaptation cannot apply directly since target domain samples are not available.

Robustness: By deliberately perturbing the model during training, the model learns to be more robust to domain shifts at test-time. (Li et al., 2019) episodically switches the feature extractor or classifier to domain-specific counterparts, and (Huang et al., 2020) zeros out features associated with the highest gradient in the final classification layer so that the model learns more diverse features. Another branch of work aims to minimize the worst-case risk over all domains such that the training objective is a weighted average of source domain losses (Sagawa et al., 2019; Krueger et al., 2020). However, the worst-performing domain may be improved at the cost of other domains.

3 Proposed Method

In this section, we introduce our proposed method that applies selective cross-domain consistency regularization on top of supervised task loss and stochastic augmentations of time series inputs. We cover the two scenarios where regularization selection can be made based on domain metadata or learned when such metadata is not available.

3.1 Preliminaries: Notations

We denote total N observed samples from M source domains as $\{(\boldsymbol{x}_n, \boldsymbol{y}_n, d_n)\}_{n=1}^{N}$, where for the n-th sample, $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ are the predictor and response respectively, and $d_n \in \{1, \ldots, M\}$ is the domain label. $\boldsymbol{y}_n$ is a one-hot vector of the true class label in L classes. Samples in each domain d are independently and identically distributed (i.i.d.) according to a domain-dependent data distribution as $(\boldsymbol{x}, \boldsymbol{y}) \sim P(\mathcal{X}, \mathcal{Y}|\mathcal{D} = d)$. We denote the number of samples in domain d as N_d, and $\sum_{d=1}^{M} N_d = N$. Domain labels $\{d_n\}$ are not available at test-time.

A neural network model is composed of feature extractor $f(\cdot)$ parameterized by Θ that yields learned features $\boldsymbol{z} = f(\boldsymbol{x}; \Theta)$, and classifier $h(\cdot)$ parameterized by Ψ that yields logits output $\boldsymbol{g} = h(\boldsymbol{z}; \Psi)$. Soft labels or vector of estimated class probabilities are obtained by applying softmax function on the logits i.e. $\boldsymbol{s} = softmax(\boldsymbol{g})$ where $\boldsymbol{s}[i] = \frac{\exp(\boldsymbol{z}[i])}{\sum_{\ell=1}^{L} \exp(\boldsymbol{z}[\ell])}$. The final predicted class is the one with the highest probability in $\boldsymbol{s}$.

3.2 Selective Cross-Domain Consistency Regularization

We consider a domain shift as a form of data transformation that is label-preserving, but does not necessarily preserves class relationships. As a toy example, consider vector input $\boldsymbol{x} = [x_1, x_2, \ldots, x_r]^T$, $r > L$ and ground-truth model that is an identity map on the first L elements of the input i.e. logits $\boldsymbol{g} = h(f([x_1, x_2, \ldots, x_r]^T)) = [x_1, x_2, \ldots, x_L]^T$. We can view $x_1, x_2, \ldots, x_L$ as features with true correlations with class probability. Augmenting $x_{L+1}, \ldots, x_r$ is guaranteed to preserve both original label and class relationship, but augmenting elements in $x_1, \ldots, x_L$ can change class relationships even when label is preserved. Unlike artificial data augmentations where we can control the perturbations introduced while preserving class-correlated information we want to retain in the data, there is no guarantee that such information is preserved through naturally-occurring domain shifts. For instance, some features can be obscured in different domains. We hypothesize, however, that class-correlated information is shared across domains to different extents according to latent inter-domain relationships.

More formally, for $\boldsymbol{x}^{d^{(i)}} \sim \mathbb{P}_{\ell}^{d^{(i)}}$ and $\boldsymbol{x}^{d^{(j)}} \sim \mathbb{Q}_{\ell}^{d^{(j)}}$, we aim to learn model parameters such that $\mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_{\ell}^{d^{(i)}}}[h(f(\boldsymbol{x}; \Theta); \Psi)] = \mathbb{E}_{\boldsymbol{x} \sim \mathbb{Q}_{\ell}^{d^{(j)}}}[h(f(\boldsymbol{x}; \Theta); \Psi)]$, for closely-related domains $d^{(i)}$ and $d^{(j)}$, for each class ℓ. This helps to prevent the model from being over-confident in its predictions due to overfitting on spurious features of individual domains, and increase robustness to domain shifts.

We propose a selective logit regularization $\Omega(\Theta, \Psi)$ such that the learned model is encouraged to preserve prediction consistency between similar domains:

$$\Omega(\Theta, \Psi) = \sum_{d^{(i)}=1}^{M} \sum_{d^{(j)}=1}^{M} w(d^{(i)}, d^{(j)}) \sum_{\ell=1}^{L} \|\bar{\boldsymbol{g}}^{(d^{(i)}, \ell)} - \bar{\boldsymbol{g}}^{(d^{(j)}, \ell)}\|_2^2 \tag{1}$$

where $\bar{\boldsymbol{g}}^{(d^{(i)}, \ell)}$ is the mini-batch average of logits for domain $d^{(i)}$ class ℓ, which we refer to as the class-conditional domain centroid. Weights $w(d^{(i)}, d^{(j)}) \geq 0$ depends on pairwise domain similarity between domains $d^{(i)}$ and $d^{(j)}$ to impose greater regularization on more similar domains. We describe estimation procedures for $w(d^{(i)}, d^{(j)})$ in Section 3.3.

The final objective function is then:

$$L(\Theta, \Psi) = L_{CE}(\Theta, \Psi) + \lambda \Omega(\Theta, \Psi) \quad \text{where} \quad L_{CE}(\Theta, \Psi) = -\sum_{n=1}^{N} \boldsymbol{y}_n \cdot \log(\boldsymbol{s}_n) \tag{2}$$

and $L_{CE}(\Theta, \Psi)$ is the supervised classification cross-entropy loss.

3.3 Regularization Selection based on Estimated Domain Relationships

Inter-domain relationships need to be determined in order to select closely-related domains for consistency regularization. We consider two scenarios: when domain metadata is available, and when it is not; domain metadata are descriptions of source domains to provide (possibly limited) context of the environments in which data is collected.

3.3.1 Fixed Selection

With expert knowledge of the application, users can directly use available metadata to infer relationships and group the domains into clusters. Domains across clusters do not share class relationships and hence are not regularized. We denote the function $clust : \{1, \ldots, M\} \rightarrow \{1, \ldots, K\}$ as the map from domain index to cluster index for K clusters, and $\mathcal{S}_c = \{d | clust(d) = c, d \in \{1, \ldots, M\}\}$ the

set of domains in cluster c. The consistency regularization function in Equation 1 can be restated as

$$\Omega(\Theta, \Psi) = \sum_{c=1}^{K} \sum_{d^{(i)} \in \mathcal{S}_c} \sum_{d^{(j)} \in \mathcal{S}_c} \frac{1}{2|\mathcal{S}_c|} \sum_{\ell=1}^{L} \|\bar{\boldsymbol{g}}^{(d^{(i)},\ell)} - \bar{\boldsymbol{g}}^{(d^{(j)},\ell)}\|_2^2 = \sum_{c=1}^{K} \sum_{d \in \mathcal{S}_c} \sum_{\ell=1}^{L} \|\bar{\boldsymbol{g}}^{(d,\ell)} - \boldsymbol{\gamma}^{(c,\ell)}\|_2^2 \tag{3}$$

$$\text{where } \gamma^{(c,\ell)} = \frac{1}{|\mathcal{S}_c|} \sum_{d \in \mathcal{S}_c} \bar{\boldsymbol{g}}^{(d,\ell)} \tag{4}$$

We refer to $\gamma^{(c,\ell)}$ of cluster c class ℓ as the class-conditional cluster centroid. The equivalence between Equation 1 and 3 is by setting $w(d^{(i)}, d^{(j)}) = \frac{1}{2|\mathcal{S}|_c}$ if both $d^{(i)}$ and $d^{(j)}$ are in cluster $c \in \{\mathbf{1}, \dots K\}$, and 0 otherwise. The scaling factor ensures that when domain centroids are equidistant to their cluster centroids, the amount of contribution each cluster makes to the regularization term $\Omega(\Theta, \Psi)$ is proportional to its size.

3.3.2 Learned Selection

When no domain metadata is available, we propose estimating domain relationships by inter-domain distances during training. We measure the distance between two domains for class ℓ as the squared L_2 distance between their class-conditional domain centroids, and $d^{(j)}$ is defined as the nearest neighbor domain to $d^{(i)}$ if it is nearest to $d^{(j)}$ for the most number of classes. We estimate the nearest neighbor domain as the most similar domain to $d^{(i)}$ at fixed intervals during training (every 100 iterations in our experiments), and we enforce prediction consistency between each domain and its nearest neighbor domain. That is, we set the weights $w(d^{(i)}, d^{(j)})$ as per Equation 5 if $d^{(j)}$ is the nearest neighbor and 0 otherwise, where:

$$w(d^{(i)}, d^{(j)}) = \frac{1}{L} \sum_{\ell=1}^{L} \exp\left(\frac{-\|\bar{\boldsymbol{g}}^{(d^{(i)},\ell)} - \bar{\boldsymbol{g}}^{(d^{(j)},\ell)}\|_2^2}{2\xi^2}\right) \tag{5}$$

by applying RBF kernel on the inter-domain distance with hyperparameter ξ. We block gradients on the weights to prevent the weights and inter-domain distance function in Equation 1 from updating in opposing directions.

3.4 Domain-wise Time Series Augmentation

To achieve additional robustness to data perturbations, we apply time series augmentations on the input samples at training with 0.5 probability. For each source domain, we sample an augmentation function from a pre-defined distribution at each iteration, and apply the function on all samples from the domain. The domain-wise augmentation simulates potential test-time domain shifts.

Table 1: Time series augmentations.

Augmentation	**General Expression**
mean shift	$a_{mean}(x) = x - \mu + \mu_{new}$
scaling	$a_{scale}(x) = \left(\frac{x-\mu}{\sigma}\right) * \sigma_{new} + \mu$
masking	$a_{mask}(x[i]) = \begin{cases} x[i] & \text{w.p. } 0.9 \\ \mu & \text{w.p. } 0.1 \end{cases}$

We consider 3 time series augmentations, namely mean shift, scaling and masking, in Table 1. The choice of augmentations depends on the dataset to avoid perturbing characteristics known to be important for classification. We provide augmentation details for each dataset in Section 4.

4 Experiments and Results

We compare with baseline ERM and state-of-the-art domain generalization methods: GroupDRO (Sagawa et al., 2019), VREx (Krueger et al., 2020), IRM (Arjovsky et al., 2019), Interdomain Mixup (Yan et al., 2020), RSC (Huang et al., 2020), MTL (Blanchard et al., 2021), MLDG (Li et al., 2018) and Correlation (Arpit et al., 2019). We also reformulate 4 popular domain adaption methods for domain generalization following (Gulrajani & Lopez-Paz, 2021): MMD-DG (Li et al., 2018),

CORAL-DG (Sun & Saenko, 2016), DANN-DG (Ganin et al., 2016) and CDANN-DG (Li et al., 2018; Long et al., 2018).

We evaluate the proposed method on two real-world datasets for fault detection and human activity recognition. For each dataset, we use leave-one-domain-out evaluation where we treat each domain as the unseen target domain in turn and train with rest as source domains. Each source-target combination is ran over 3 seeds (0, 1 and 2) and consequently splits where 80% source samples are randomly chosen for training and the remaining 20% are used for validation, and 80% target samples are randomly chosen for testing. For each split, each method is tuned with 20 random hyperparameter configurations, and the best configuration is selected by the highest validation accuracy. The largest regularization hyperparameter is picked when there is a tie. Overall, on a single dataset, we run each method $60 \times (\text{number of domains})$ times to ensure comprehensive evaluation. All methods use the same backbone networks. We use convolutional neural network (CNN) as feature extractor and fully-connected network (FCN) as classifier. We use Adam optimizer with learning rate 0.001 and weight decay 5×10^{-5}, and batch size 32 per domain. Models are trained for 3000 iterations, with learning rate reduced by a factor of 10 after 2400 iterations. Further details of backbone networks and hyperparameters are provided in the Appendix.

Table 2: Domain attributes. Each row of domains is regularized in fixed regularization selection.

(a) Bearings

Loc.	**Loading torque**			
	0	1	2	3
Drive	A	B	C	D
Fan	E	F	G	H

(b) HHAR

User	**Phone model**			
	Nexus	S3	S3 mini	S+
User 1	A	B	C	D
User 2	E	F	G	H
User 3	I	J	K	L

4.1 Fault Detection

The Bearings dataset[1] from Case Western Reserve University is widely used for predictive maintenance. It contains vibration signals at 12kHz sampling rate to detect rolling element bearings faults in rotating machines (Smith & Randall, 2015). We extract samples of length 4096 by a sliding window with stride 290 (Zhang et al., 2017). There are 1 healthy class and 9 fault classes: inner-race fault (IF), outer-race fault (OF), and ball fault (BF) with each further divided into dimensions 0.007, 0.014 and 0.021 inches. We apply a combination of mean shift, scaling and masking data augmentations by setting $\mu = \bar{x}$, $\mu_{new} = 0$, $\sigma = sd(x)$ and $\sigma_{new} = 1$ in Table 1. Samples are augmented with probability 0.5, and additional augmentations beyond the first one are applied with probability 0.5 to allow a mixture of perturbations. There are 8 domains: drive end and fan end location with each operated at 0, 1, 2, and 3 loading torques as in Table 2. For fixed regularization selection, consistency regularization is applied on domains with the same location. From domain generalization performance in Table 3, the proposed method improves over the baseline ERM in almost all cases. On average, ERM has accuracy 82.2%, and the proposed method attains the best performance across all methods with 87.9% and 89.1% given fixed and learned regularization selection, respectively.

4.2 Human Activity Recognition

The HHAR dataset (Stisen et al., 2015) consists of multi-channel sensor readings to classify six activities, namely Biking, Standing, Sitting, Walking, Stair down, and Stair up. Following a recent work in domain adaptation (Wilson et al., 2020), we focus on accelerometer readings in the x, y and z direction on smartphones and extract samples of length 128 by a sliding window with no overlap. All samples are scaled by $\frac{1}{20}$ so that readings for all 3 channels approximately fall between -1 and 1. For this application, mean and standard deviation are known to be important features for classification and activities such as Standing are sensitive to abrupt changes in sensor readings (Seto et al., 2015), hence we apply limited data augmentation i.e. scaling with $\mu = 0$, $\sigma = 1$ and $\sigma_{new} \sim Unif(0.8, 1.2)$. To keep the number of domains to a limited level suitable for leave-one-domain-out evaluation, we use 12 domains as in Table 2: the first 3 users each with 4

[1] `https://csegroups.case.edu/bearingdatacenter/pages/welcome-case-western-reserve-university-bearing-data-center-website`

Table 3: Bearings: Classification accuracy on target domain using leave-one-domain-out testing and training on remaining domains.

Method	**Accuracy** (%)								
	A	B	C	D	E	F	G	H	Avg
ERM	65.4	93.8	96.0	71.2	68.4	83.0	94.0	86.2	82.2 ± 1.2
IRM	60.7	87.0	89.2	76.0	62.8	80.9	92.8	88.5	79.7 ± 1.8
GroupDRO	55.7	70.0	77.8	74.8	60.7	59.2	65.6	50.5	64.3 ± 2.2
Interdomain Mixup	62.0	86.5	96.8	76.0	82.0	95.4	97.7	87.2	85.4 ± 1.3
MLDG	62.8	77.6	85.9	72.8	63.3	58.5	60.7	55.0	67.1 ± 5.4
MTL	35.3	64.5	66.6	48.3	47.6	48.2	36.7	44.9	49.0 ± 2.1
Correlation	46.5	79.0	90.0	69.1	71.5	85.5	80.5	83.9	75.7 ± 1.9
VREx	63.8	90.5	97.2	81.6	70.3	83.9	92.0	84.3	83.0 ± 1.1
RSC	62.4	94.4	98.0	86.5	73.4	87.4	97.3	85.7	85.6 ± 2.5
DANN-DG	56.2	84.7	92.2	80.2	70.0	79.1	89.1	90.5	80.3 ± 2.3
CDANN-DG	56.0	80.8	94.8	80.2	70.7	81.4	90.3	84.0	79.8 ± 3.3
CORAL-DG	62.5	77.9	90.0	76.0	63.1	79.2	74.5	83.0	75.8 ± 3.6
MMD-DG	53.9	67.7	84.2	67.6	63.2	74.3	74.7	56.7	67.8 ± 3.5
proposed (fixed sel.)	86.8	95.3	97.6	79.8	77.4	82.7	93.4	90.8	87.9 ± 0.6
proposed (learned sel.)	89.1	97.9	97.1	75.8	81.5	85.3	94.4	91.8	**89.1** ± 1.1

phone models. For fixed regularization selection, consistency regularization is applied on domains with the same user. From domain generalization performance in Table 4, the proposed method improves over the baseline ERM in almost all cases, and has the best performance of 88.5% on average. The second-best performing method RSC encourages learning more diverse features by feature masking, and applying the proposed method on RSC by alternating between the two methods further improves average performance to 88.9% given fixed regularization selection. We chose the alternating procedure (Zhang et al., 2021) so that strategies from the two methods do not directly interfere with each other.

Table 4: HHAR: Classification accuracy on target domain using leave-one-domain-out testing and training on remaining domains.

Method	**Accuracy** (%)												
	A	B	C	D	E	F	G	H	I	J	K	L	Avg
ERM	86.4	91.6	81.0	91.7	71.3	96.9	96.4	85.9	85.0	88.1	86.6	89.5	87.5 ± 0.5
IRM	87.2	92.0	80.0	90.5	71.7	96.5	96.6	85.3	84.3	88.4	87.3	89.6	87.4 ± 0.5
GroupDRO	80.4	76.7	52.4	74.6	63.6	77.3	76.6	75.0	86.3	86.8	82.8	70.3	75.2 ± 1.2
Interdomain Mixup	80.2	68.9	61.4	69.7	55.3	71.1	81.4	64.7	87.8	85.0	84.9	71.5	73.5 ± 2.0
MLDG	81.7	75.8	58.8	79.3	58.4	70.7	70.9	68.0	86.8	87.7	84.7	70.4	74.4 ± 0.9
MTL	79.6	75.8	60.9	77.1	62.8	75.7	80.2	72.7	85.7	81.1	79.3	71.9	75.2 ± 0.3
Correlation	80.3	91.0	81.7	87.9	69.1	95.4	95.8	88.5	85.1	85.6	88.6	88.9	86.5 ± 0.5
VREx	87.1	90.6	80.5	92.2	71.0	96.5	96.7	85.5	85.5	88.7	87.5	90.2	87.7 ± 0.3
RSC	87.3	90.5	84.4	92.2	73.9	96.7	96.9	86.2	86.8	87.5	88.5	90.1	88.4 ± 0.1
DANN-DG	84.7	89.5	72.4	92.8	71.2	95.1	94.8	84.2	81.6	84.3	84.9	86.7	85.2 ± 1.2
CDANN-DG	85.6	86.0	79.8	89.6	72.4	93.6	95.6	83.0	81.3	87.0	83.4	85.8	85.2 ± 0.6
CORAL-DG	80.5	76.8	58.4	74.3	62.5	77.5	85.8	74.2	86.8	79.7	86.2	69.1	76.0 ± 0.5
MMD-DG	81.9	74.0	52.9	75.4	60.3	76.8	79.0	74.6	86.9	85.6	83.7	68.9	75.0 ± 0.4
proposed (fixed sel.)	87.4	91.0	80.7	94.6	75.7	96.5	97.1	86.2	85.0	89.2	89.1	89.8	88.5 ± 0.2
proposed (learned sel.)	87.3	90.6	85.3	93.5	76.0	96.1	96.7	86.0	85.1	88.1	88.6	88.9	88.5 ± 0.3
RSC													
+ proposed (fixed sel.)	87.2	89.9	86.0	93.6	75.6	96.6	96.4	86.1	85.7	88.6	90.4	90.6	**88.9** ± 0.0
+ proposed (learned sel.)	86.3	89.0	84.3	93.3	74.4	96.8	96.9	86.8	87.0	86.7	90.4	89.2	88.5 ± 0.4

5 Further Analysis

We perform ablation studies and further experiments to verify the effects of each component in the proposed method.

Ablation study: In Table 5, we see that applying data augmentations and consistency regularization individually improves model performance over ERM for both datasets. Performance increases further when both strategies are applied together. The synergy between the strategies may be explained by the smoothness assumption in semi-supervised learning, which states that two samples that are close in the input space should share the same labels, and consequently preferentially learns decision boundaries in low-density regions (van Engelen & Hoos, 2019).

Table 5: Effect of regularization and time series augmentation strategies.

(a) Effect of regularization and / or time series augmentation strategies

Strategy		**Avg Accuracy** (%)	
Regularization	Augmentation	Bearings	HHAR
✗	✗	82.2	87.5
✗	✓	86.5	88.1
(fixed sel.)			
✓	✗	87.1	88.5
✓	✓	87.9	88.5
(learned sel.)			
✓	✗	86.8	88.3
✓	✓	89.1	88.5

(b) Augmentation effects on Bearings, without consistency regularization

Augmentation	**Avg Accuracy** (%)
None	82.2
Mean shift	83.0
Scale	82.4
Mask	82.4
All	86.5

Effect of selective cross-domain consistency regularization: To further study the effect of regularization in isolation, we apply the proposed method with fixed regularization selection and without time series augmentations. We experiment with 3 distance functions, namely squared L_2 distance, cosine distance and KL-divergence, either between individual samples and cluster centroids or between domain and cluster centroids. Regularization is applied on the features $\boldsymbol{z}$, logits $\boldsymbol{g}$ or soft labels $\boldsymbol{s}$. Comparing the generalization performance in Table 6, domain-level regularization tends to have higher accuracy, possibly because it allows greater diversity of representations in each domain. Regularizing on logits results in higher accuracy for most cases. It allows more flexibility since both feature extractor and classifier are directly regularized while preserving class-relationships. Soft labels are normalized logits and have limited variability across source domains for further alignment. The feature space is generally much larger than the logit or label space, and hence possibly more difficult for effective alignment. We observe that all choices of distance functions and representations attain better performance than ERM (82.2%), with regularization of logits at the domain-level achieving the best accuracy of 87.1%.

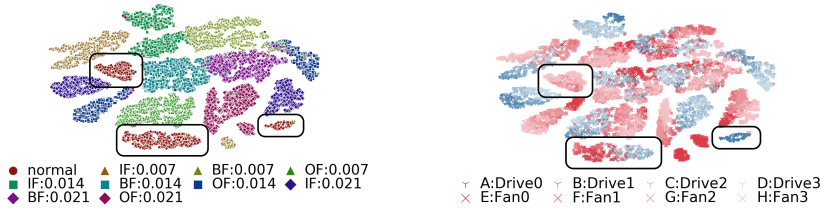
Next, we study how different cluster assignments affect domain generalization performance on each target domain in the Bearings dataset in Table 7. Consistency regularization tends to improve performance over ERM across all target domains. While regularization with 4 clusters and 1 cluster both improves average accuracy over ERM by 2.2% and 3.4% respectively, regularization with 2 clusters achieves the highest improvement of 4.9%. Domains in each of the two clusters contain data collected from the same machine location, and hence can be expected to be closely-related with similar class relationships. This shows we can improve domain generalization performance given good cluster assignments.

Table 6: Bearings: Average classification accuracy of proposed method with fixed consistency regularization using different regularization functions (squared L_2 distance, cosine distance, KL-divergence) between samples or domains and cluster centroids, without time series augmentations.

	Avg Accuracy (%)					
	Sample-level			*Domain-level*		
Regularize on	L2	cos	KL	L2	cos	KL
Features $\boldsymbol{z}$	83.9	84.9	N/A	85.4	82.3	N/A
Logits $\boldsymbol{g}$	86.2	84.2	N/A	**87.1**	85.9	N/A
Soft labels $\boldsymbol{s}$	82.9	83.3	82.8	82.5	83.8	83.8

Table 7: Bearings: Target domain classification accuracy of the proposed method given different cluster assignments for fixed regularization selection, without time series augmentations. Placing each domain in a separate cluster is equivalent to ERM.

Cluster assignment	**# clusters**	**Accuracy** (%)								
		A	B	C	D	E	F	G	H	Avg
{A},{B},{C},{D},{E},{F},{G},{H}	8	65.4	93.8	96.0	71.2	68.4	83.0	94.0	86.2	82.2
{A,B},{C,D},{E,F},{G,H}	4	61.2	93.3	97.3	79.6	70.3	91.4	94.3	87.5	84.4
{A,B,C,D},{E,F,G,H}	2	66.8	92.4	96.9	85.4	78.3	90.6	94.4	92.5	**87.1**
{A,B,C,D,E,F,G,H}	1	65.8	97.0	97.1	80.5	79.4	86.8	92.3	86.1	85.6



(a) ERM; Marked by class (b) ERM; Marked by domain

(c) Proposed; Marked by class (d) Proposed; Marked by domain

Figure 3: Bearings: t-SNE plots of features from ERM (top row) and proposed method for fixed regularization selection, no time series augmentation (bottom row) trained with target domain H, marked by class (left column) and domain (right column; blue is drive-end, red is fan-end).

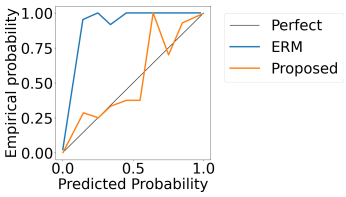
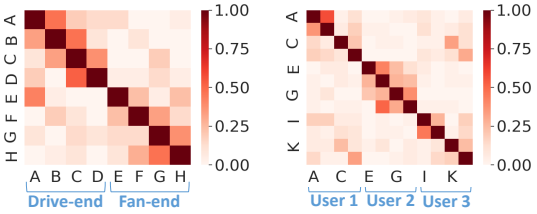


Figure 4: Bearings: Reliability diagram for 'BF:0.007' class of ERM and proposed method for fixed regularization selection, no time series augmentation trained with target domain H.



(a) Bearings (b) HHAR

Figure 5: For each i, j entry, shade corresponds to proportion of runs domain j is the nearest neighbor of domain i at end of training. Diagonal entries are set to 1.

We visualize the features learned by ERM and the proposed method in t-SNE plots in Figure 3, where the last domain H is the unseen target. Domains from the same location (i.e. drive or fan end) tend to cluster even in ERM, but not consistently across classes. For instance, 'normal' class has 3 clusters in ERM, and is regularized by the proposed method such that domains from the same location are closer to each other. Model calibration is also improved as seen in Figure 4.

Visualization of learned domain relationships: The performance of the proposed method depends on the estimation of inter-domain relationships when domain metadata are not provided. Figure 5 plots the proportion of runs each pair of domain is estimated to be closest neighbors at the end of training, with the diagonal entries set to 1. For Bearings, the estimated clusters approximately match the manually specified ones in Table 2. For HHAR, the variation between phone models (i.e. Nexus and S3 versus S3 mini and S+) appears larger than that between some users (i.e. User 1 versus User 3). Domain relationships estimated from data can be different from those inferred from metadata descriptions and can contain finer measures of inter-domain similarity, and hence may be a more preferable approach to set the selective consistency regularization. In fact, using learned selection obtains higher accuracy (89.1%) than using fixed metadata-inferred selection (87.9%) for Bearings.

6 Conclusion

In this work, we introduced a representation learning method for domain generalization for time series classification. We applied time series augmentations to improve robustness, and selective consistency regularization to enforce similar predictions for similar domains. From comprehensive experiments, we showed that the proposed method significantly improves over baseline ERM and performs better than or comparably to state-of-the-art methods. For future work, we will study incorporating techniques from computer vision literature that benefit generalization for time series.

ETHICS STATEMENT

All datasets used in this paper are publicly available. Our proposed method contributes to the effort to reduce prediction bias due to domain shift.

REPRODUCIBILITY STATEMENT

All datasets used in this paper are publicly available. We provide description of data processing steps and experiment setup details in Section 4. We provide details of deep network architectures and hyperparameter tuning distributions in the Appendix Section A.

REFERENCES

F. Ahmed, Y. Bengio, H. v. Seijen, and A. C. Courville. Systematic generalisation with group invariant predictions. In *ICLR*, 2021.

I. Albuquerque, J. Monteiro, M. Darvishi, T. H. Falk, and I. Miltliagkas. Generalizing to unseen domains via distribution matching. In *ArXiv*, 2020.

M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. In *ArXiv*, 2019.

D. Arpit, C. Xiong, and R. Socher. Predicting with high correlation features. *ArXiv*, 2019.

Y. Balaji, S. Sankaranarayanan, and R. Chellappa. MetaReg: Towards domain generalization using meta-regularization. In *NeurIPS*. 2018.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J.W. Vaughan. A theory of learning from different domains. In *Machine Learning*, volume 79, pp. 151–175, 2010.

G. Blanchard, A. A. Deshmukh, U. Dogan, G. Lee, and C. Scott. Domain generalization by marginal transfer learning. *Journal of Machine Learning Research*, 22(2):1–55, 2021.

F. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

A. D'Innocente and B. Caputo. Domain generalization with domain-specific aggregation modules. In *Pattern Recognition*, pp. 187–198, 2019.

Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, 2019.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. In *Journal of Machine Learning Research*, volume 17, pp. 59:1–59:35, 2016.

R. Gong, W. Li, Y. Chen, and L. Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

I. Gulrajani and D. Lopez-Paz. In Search of Lost Domain Generalization. In *ICLR*, 2021.

J. Guo, D. Shah, and R. Barzilay. Multi-source domain adaptation with mixture of experts. In *EMNLP*, 2018.

A. Gupta, A. Murali, D. Gandhi, and L. Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *NIPS*, 2018.

D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. L. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ArXiv*, 2020.

Z. Huang, H. Wang, E. Xing, and D. Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020.

Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. Neural style transfer: A review. In *IEEE Transactions on Visualization and Computer Graphics*, volume 26, pp. 3365–3385, 2020.

D. Kim, S. Park, J. Kim, and J. Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *ArXiv*, 2021.

D. Krueger, E. Caballero, J. Jacobsen, A. Zhang, J. Binas, R. L. Priol, and A. C. Courville. Out-of-distribution generalization via risk extrapolation (REx). In *ArXiv*, 2020.

D. Li, Y. Yang, Y-Z. Song, and T. M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018.

D. Li, J. Zhang, Y. Yang, C. Liu, Y-Z. Song, and T. M. Hospedales. Episodic training for domain generalization. In *ICCV)*, 2019.

H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018.

Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018.

Y. Liu, Y-M. Zhang, X-Y. Zhang, and C-L. Liu. Adaptive spatial pooling for image classification. In *Pattern Recognition*, volume 55, pp. 58–67. Elsevier, 2016.

M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

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

D. Mahajan, S. Tople, and A. Sharma. Domain generalization using causal matching. In *Uncertainty and Robustness in Deep Learning (ICML Workshop)*, 2020.

M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci. Best sources forward: Domain generalization through source-specific nets. In *ICIP*, 2018.

S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017.

H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo. Reducing domain gap via style-agnostic networks. In *ArXiv*, 2019.

E. Rosenfeld, P. Ravikumar, and A. Risteski. The risks of invariant risk minimization. In *ArXiv*, 2020.

S. Sagawa, P. W. Koh, T. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *ArXiv*, 2019.

S. Seto, W. Zhang, and Y. Zhou. Multivariate time series classification using dynamic time warping template selection for human activity recognition. *IEEE Symposium Series on Computational Intelligence*, pp. 1399–1406, 2015.

S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi. Generalizing across domains via cross-gradient training. In *ICLR*, 2018.

W. A Smith and R. B Randall. Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. In *Mechanical Systems and Signal Processing*, volume 64, pp. 100–131. Elsevier, 2015.

N. Somavarapu, C-Y. Ma, and Z. Kira. Frustratingly simple domain generalization via image stylization. In *ArXiv*, 2020.

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

D. Stutz, M. Hein, and B. Schiele. Disentangling adversarial robustness and generalization. In *CVPR*, 2019.

B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016.

J. E. van Engelen and H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109: 373–440, 2019.

O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, 2018.

J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin. Generalizing to unseen domains: A survey on domain generalization. In *IJCAI*, 2021.

M. Wang and W. Deng. Deep visual domain adaptation: A survey. In *Neurocomputing*, volume 312, pp. 135–153, 2018.

Y. Wang, H. Li, and A. Kot. Heterogeneous domain generalization via domain mixup. In *ICASSP*, 2020a.

Z. Wang, M. Loog, and J. Gemert. Respecting domain relations: Hypothesis invariance for domain generalization. *ICPR*, 2020b.

G. Wilson, J. R. Doppa, and D. J. Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *SIGKDD*, 2020.

S. Yan, H. Song, N. Li, L. Zou, and L. Ren. Improve unsupervised domain adaptation with mixup training. In *ArXiv*, 2020.

W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. In *Sensors*, volume 17, pp. 425. Multidisciplinary Digital Publishing Institute, 2017.

W. Zhang, S. Seto, and D. K. Jha. Cazsl: Zero-shot regression for pushing models by generalizing through context. In *IROS*, 2020.

W. Zhang, M. Ragab, and R. Sagarna. Robust domain-free domain generalization with class-aware alignment. *ICASSP*, 2021.

H. Zheng, R. Wang, Y. Yang, Y. Li, and M. Xu. Intelligent fault identification based on multisource domain generalization towards actual diagnosis scenario. In *IEEE Transactions on Industrial Electronics*, volume 67, pp. 1293–1304, 2020.

K. Zhou, Y. Yang, T. Hospedales, and T. Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, 2020.

A IMPLEMENTATION DETAILS

We provide additional details on the implementation for our experiments and the datasets used.

A.1 HYPERPARAMETERS

We fix learning rate 0.001, weight decay 5×10^{-5}, and batch size 32 per domain. Models are trained for 3000 iterations, with learning rate reduced by a factor of 10 after 2400 iterations. All other hyperparameters are tuned by random sampling from distributions in Table 8. All experiments are run using Adam optimizer with the NVIDIA container image for PyTorch, release 20.03.

Method	**Hyperparameter**	**Distribution**
IRM	Regularization λ	$10^{Unif(-1,5)}$
	Iterations of penalty annealing	$\lfloor 10^{Unif(0,4)} \rfloor$
GroupDRO	Group weight temperature η	$10^{Unif(-3,-1)}$
Interdomain Mixup	Beta shape parameter α	$10^{Unif(-1,1)}$
MTL	Embedding averaging proportion	$\{0.5,0.9,0.99,1\}$
MLDG	Meta-learning loss β	$10^{Unif(-1,1)}$
Correlation	Regularization λ	10^{-5}
CORAL-DG, MMD-DG	Regularization λ	$10^{Unif(-3,-1)}$
DANN-DG, CDANN-DG	Discriminator learning rate	$10^{Unif(-5,-3.5)}$
	Discriminator weight decay	$10^{Unif(-6,-2)}$
	Discriminator Adam β_1	$\{0, 0.5\}$
	Discriminator steps	$\lfloor 2^{Unif(0,3)} \rfloor$
	Discriminator gradient penalty	$10^{Unif(-2,1)}$
	Adversarial regularization λ	$10^{Unif(-2,2)}$
VREx	Regularization λ	$10^{Unif(-1,5)}$
	Iterations of penalty annealing	$\lfloor 10^{Unif(0,4)} \rfloor$
RSC	Feature drop percentage p	$Unif(0, 0.5)$
	Batch percentage	$Unif(0, 0.5)$
proposed	Regularization λ	$10^{Unif(-3,-1)}$
	RBF kernel parameter ξ	$10^{Unif(-2,2)}$

Table 8: Setup for hyperparameter tuning.

A.2 DATASETS AND NETWORK ARCHITECTURES

We provide details on the sample size of the datasets. Backbone network architectures used for each dataset is given in Table 9.

Bearings: All domains have the same number of samples. For each domain, the sample size of each class is 'normal': 416, 'IF:0.007': 371, 'BF:0.007': 409, 'OF:0.007': 417, 'IF:0.014': 387, 'BF:0.014': 408, 'OF:0.014': 398, 'IF:0.021': 407, 'BF:0.021': 383, 'OF:0.021': 404. We use a 6-layer CNN as feature extractor and a 3-layer FCN as classifier.

HHAR: Sample size differs across domain according to availability of data per user and device, as in Table 10. We use a 3-layer CNN as feature extractor and a 1-layer fully-connected network as classifier (Liu et al., 2016).

(a) Network for Bearings

Layer	**Operation**	**Specifications**
Convolution	Conv BatchNorm LeakyReLU	8 (filter: 64×1, stride: 2, pad: 1)
Convolution (3 times)	Conv BatchNorm LeakyReLU	8 (filter: 3×8, stride: 2, pad: 1)
Convolution	Conv LeakyReLU	8 (filter: 3×8, stride: 2, pad: 1)
Convolution	Conv	8 (filter: 8×8, stride: 1, pad: 1)
Fully connected	FC	32
Fully connected (2 times)	FC ReLU	32
Fully connected	FC	10

(b) Network for HHAR

Layer	**Operation**	**Specifications**
Convolution	Conv BatchNorm LeakyReLU	128 (filter: 8×3, stride: 1, pad: 1)
Convolution	Conv BatchNorm LeakyReLU	256 (filter: 5×128, stride: 1, pad: 1)
Convolution	Conv BatchNorm LeakyReLU	128 (filter: 3×256, stride: 1, pad: 1)
Pooling	Average pooling	1 (filter: 121, stride:121)
Fully connected	FC	6

Table 9: Backbone network architectures for each dataset. Convolution operation is abbreviated as 'Conv' and fully connected operation is abbreviated as 'FC'.

Domain	**Class**					
	Biking	Standing	Sitting	Walking	Stair down	Stair up
A	626	933	652	676	874	778
B	346	468	316	341	435	376
C	175	234	162	176	207	212
D	298	237	264	226	223	275
E	999	682	681	771	692	1013
F	487	387	312	407	370	495
G	234	196	161	199	186	245
H	385	251	267	298	253	331
I	539	817	628	768	723	857
J	293	427	312	374	358	445
K	147	213	168	211	164	244
L	275	229	264	265	248	300

Table 10: HHAR: Sample size distribution per domain.

B FURTHER EXPERIMENT RESULTS

We additionally evaluate on a more challenging setting where target domain conditions are not combinations of source domain conditions. For Bearings, we use only domains from drive-end location, so each domain has a distinct loading torque. For HHAR, we use only domains from the first user, so each domain has a distinct phone model. We fix the hyperparameter of our proposed method with learned selective regularization to $\lambda = 0.01$ and $\xi = 0.1$. From Table 11 and 12, our proposed method outperforms ERM on all target domains in Bearings, and on average in HHAR.

Method	**Accuracy** (%)				
	A	B	C	D	Avg
ERM	84.4	94.9	98.9	86.0	91.0 ± 3.5
proposed (learned sel.)	**93.9**	**97.8**	**99.0**	**88.7**	**94.8** ± 3.8

Table 11: Bearings drive-end domains: Classification accuracy on target domain using leave-one-domain-out testing and training on remaining domains. Standard error is taken over 3 seeds. Train/test sample splits in all domains are also varied by seed.

Method	**Accuracy** (%)				
	A	B	C	D	Avg
ERM	**79.0**	80.5	**71.5**	80.3	77.8 ± 0.6
proposed (learned sel.)	78.7	**81.8**	70.8	**82.0**	**78.3** ± 0.6

Table 12: HHAR user 1: Classification accuracy on target domain using leave-one-domain-out testing and training on remaining domains. Standard error is taken over 3 seeds. Train/test sample splits in all domains are also varied by seed.

Ablation study: We study the effect of neighbor domain selection and weight function in learned selective regularization on Bearings. Hyperparameters are fixed at $\lambda = 0.01$ and $\xi = 0.1$. In Table 13, regularizing each domain with its nearest neighbor achieves higher accuracy than regularizing random pairs of domains. While nearest neighbor selection and a fixed weight of 1 has best accuracy in this study, we note that the RBF hyperparameter ξ can be tuned and the RBF weight approaches 1 as $\xi \to 0$.

Selected neighbor	**Weight function**	**Avg Accuracy** (%)
Random	1	80.9
Random	RBF ($\xi = 0.1$)	81.2
Nearest	1	**84.9**
Nearest	RBF ($\xi = 0.1$)	<u>83.2</u>

Table 13: Bearings: Regularization strategies of the proposed method for learned selective regularization, without time series augmentation.