ADAPT^2: Adapting Pre-Trained Sensing Models to End-Users via Self-Supervision Replay

Hyungjun Yoon*, Jaehyun Kwak*, Biniyam Aschalew Tolera*, Gaole Dai†,
Mo Li‡, Taesik Gong§, Kimin Lee* and Sung-Ju Lee*
*KAIST, †Nanyang Technological University,
‡Hong Kong University of Science and Technology, §Nokia Bell Labs
Email: {hyungjun.yoon, jaehyun98, biasc, kiminlee, profsj}@kaist.ac.kr,
gaole001@e.ntu.edu.sg, lim@cse.ust.hk, taesik.gong@nokia-bell-labs.com

***Abstract*—Self-supervised learning has emerged as a method for utilizing massive unlabeled data for pre-training models, providing an effective feature extractor for various mobile sensing applications. However, when deployed to end-users, these models encounter significant domain shifts attributed to user diversity. We investigate the performance degradation that occurs when self-supervised models are fine-tuned in heterogeneous domains. To address the issue, we propose *ADAPT*^{2}, a few-shot domain adaptation framework for personalizing self-supervised models. *ADAPT*^{2} proposes self-supervised meta-learning for initial model pre-training, followed by a user-side model adaptation by replaying the self-supervision with user-specific data. This allows models to adjust their pre-trained representations to the user with only a few samples. Evaluation with four benchmarks demonstrates that *ADAPT*^{2} outperforms existing baselines by an average F1-score of 8.8%p. Our on-device computational overhead analysis on a commodity off-the-shelf (COTS) smartphone shows that *ADAPT*^{2} completes adaptation within an unobtrusive latency (in three minutes) with only a 9.54% memory consumption, demonstrating the computational efficiency of the proposed method.**

***Index Terms*—Human Activity Recognition, Domain adaptation, Self-supervised learning, Meta-learning**

I. Introduction

The integration of deep learning into mobile sensing has broadened the scope of ubiquitous applications, such as contactless authentication [1], [2], sign language translation [3], [4], and mobile health applications [5], [6]. Nevertheless, a major challenge in mobile sensing is the scarcity of labeled data, which is often expensive to acquire. As a breakthrough, self-supervised learning [7] has been explored for its ability to train models with unlabeled data and transfer knowledge to downstream tasks. Self-supervised learning methods, such as Contrastive Predictive Coding (CPC) [8], [9], similarity-based contrastive learning (SimCLR) [10], and Multi-Task Learning [11], have been showing their effectiveness in sensory applications without needing explicit labels.

A challenge arises when fine-tuning is performed by an end-user showing heterogeneity from the pre-training data. In mobile sensing, data collected in distinct environments varies significantly across individual users, devices, and settings (e.g., position and sampling rate) [12]. Thus, the end-user data might have heterogeneous characteristics from those used for pre-training. This gap results in *domain shift*, where models trained in one domain underperform when applied to another [13]. To illustrate this challenge within the self-supervised learning setting, we conducted an empirical analysis using CPC pre-trained models [9] (experimental details in §IV-F5). In Fig. 1, the results show CPC is useful in downstream tasks when models are pre-trained in the same domain as the target domain (in-domain). However, there is a substantial decline in performance when models are pre-trained with different domains from the target domain (out-of-domain). This result emphasizes the challenges of deploying self-supervised models to diverse mobile sensing environments.

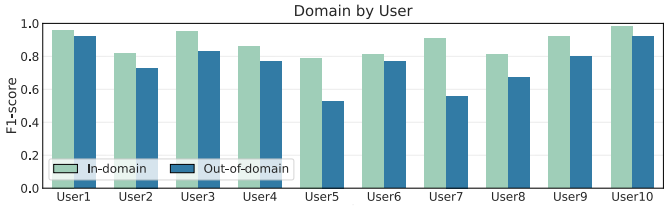

Fig. 1. Efficacy of CPC [9] pre-trained models on ICHAR [18] dataset, comparing F1-scores between in-domain and out-of-domain pre-training scenarios. Ten participants with different devices are shown as domains. Domain shifts deteriorate the performance of self-supervised models in heterogeneous environments.

Training a domain-specific model with data from the end user could be a straightforward solution, but this is infeasible considering the efforts and cost of gathering enough data from an individual user. To address the challenge, existing research encompasses domain generalization methods [14]–[16], which strive to train domain-invariant features, or domain adaptation techniques [17]–[20] that leverage a small portion of target domain data to achieve domain-specific performance. However, they primarily rely on labeled data for training, making them difficult to be applied to self-supervised learning which utilizes only unlabeled data for pre-training.

To mitigate the issue, we propose *ADAPT*^{2} as a few-shot domain adaptation framework to refine the self-supervised model into a personalized model specific to the end user. Inspired by the capability of meta-learning to "learn to learn" from few-shot data, we adapt the concept to our unsupervised pre-training setting. We introduce *self-supervised meta-learning*

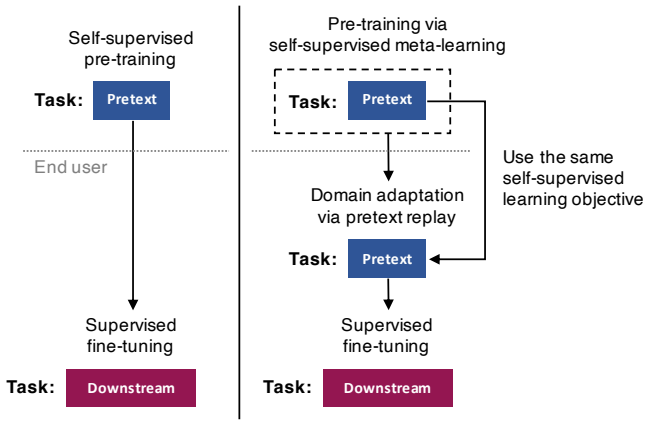

Fig. 2. A high-level comparison between the standard self-supervised learning (left) and *ADAPT*^{2}(right).

for pre-training, which enables the model to "learn to self-supervise" with only a few data. This results in the model being adaptable to the learning task of self-supervised pre-training, *i.e.*, *pretext task*. Next, we synergistically harness the adaptability by involving *pretext replay* as a pivotal domain adaptation step. The end-user engages in adapting the pre-trained model by replaying the pretext task using its own few-shot data. This process effectively aligns the model to the representation reflecting the end user's domain property with very few data. It is noteworthy that our framework is designed to be agnostic to the self-supervision algorithm, and thus, it can be integrated into existing self-supervision methods. Fig. 2 outlines a comparative overview of *ADAPT*^{2}'s operation against the standard self-supervised learning.

For evaluation, we used four Human Activity Recognition datasets [13], [18], [21], [22] that are representative datasets used for evaluating the performance under domain shifts. Our experiments demonstrate the superior performance of *ADAPT*^{2} to domain generalization and adaptation baselines [23], [24], exhibiting a high F1-score that shows an 8.83%p improvement on average. To further assess the practical viability of executing fine-tuning with domain adaptation on mobile devices, we measure the computational overhead of *ADAPT*^{2} on an off-the-shelf smartphone. Our evaluation reveals that the adaptation step with pretext replay can be completed within three minutes, indicating marginal user-side computational overhead while achieving improved performance.

We summarize our main contributions as follows:

- We investigate the domain shift problem arising when self-supervised models are deployed to diverse users for fine-tuning. Our research reveals that even after fine-tuning the pre-trained model with target domain data, the domain shift from the pre-training data can lead to significant performance degradation.
- We introduce *ADAPT*^{2}, a few-shot domain adaptation framework that can be integrated into existing self-supervised learning methods. *ADAPT*^{2} incorporates self-supervised meta-learning for the pre-training and target-side pretext replay, enabling the adaptation of pre-trained representations to the target domain using only a few target samples.
- We perform rigorous evaluations using four mobile sensing datasets with diverse domain characteristics. We compare *ADAPT*^{2} against domain generalization/adaptation baselines, that shows *ADAPT*^{2} achieves an average F1-score improvement of 8.83%p.
- We assess the computational overhead of deploying *ADAPT*^{2} on real-world mobile applications and demonstrate that the one-time target-side adaptation can be executed within three minutes by consuming a mere 9.54% of the device's memory on a COTS smartphone.

II. Related Work

A. Self-Supervised Learning

Self-supervised learning trains models using an auxiliary task that can be defined without labels, which enables learning generalized features of the data. Among numerous approaches, we focus on the methods applied for mobile sensing [7]. Multi-Task Learning [11] utilized multiple types of synthetic augmentations on the data and trained task prediction networks to infer the occurrence of the augmentation. Sensor-specific augmentations were selected to make the model learn sensory properties. Recent work focuses on using contrastive learning [25], which generates augmented views of data and trains the model with the objective of maximizing the similarity between the augmented views. Existing methods such as MoCo [26] and SimCLR [27] were applied to mobile sensing [10], [28] by using sensory augmentations to generate views. The temporal property of time-series data is utilized for generating views in a recent work [29]. Taking into account the multi-modality, Cosmo [30], COCOA [31], and ColloSSL [32] utilized contrastive learning to maximize the similarity between the embeddings driven from different modalities in the same context. Contrastive predictive coding (CPC) [8], [9] defined another type of task, predicting the embedding of future segments within the data based on the previously aggregated embeddings. In a similar context, masked-reconstruction-based methods [33], [34] have been explored for mobile sensing, using the task of reconstructing the synthetically masked segment within the data.

While models trained through self-supervised learning are known to be generalizable across diverse tasks, the potential performance decline when applied to the task of different domains (shown in §IV-F5) is overlooked. Our work differs from the prior research in exploring the domain shift problem between pre-training and fine-tuning data.

B. Domain Generalization and Domain Adaptation

Domain generalization (DG) [35] has been explored as a solution to mitigate domain shifts. DG includes approaches to learn domain-invariant features by adjusting the optimization objective [36], [37], performing adversarial training to discriminate domains [38], and processing the data to minimize the domain information from itself [39], [40]. Recent research incorporated meta-learning [41], [42] and

self-supervised learning [43], [44] to define domain-invariant training objectives. In mobile sensing, GILE [14] introduced a training scheme disentangling domain-specific information, while SDMix [16] employed semantic-aware augmentations to achieve DG tailored to activity recognition. Domain adaptation (DA) methods [45] offer a more suitable fit for our scenario, as they allow the utilization of fine-tuning data collected by end-users. Most approaches target to utilize unlabeled or a limited number of labeled target domain data [46], [47] to adapt the model to the target domain. In activity recognition, DA has been approached as an efficient transfer learning problem [20], [48], [49], and methods employing feature matching and confusion maximization [17] have been proposed. MetaSense [18] introduced a meta-learning-based model training approach followed by few-shot adaptation to create domain-specific models. DAPPER [50] is proposed as another line of research for estimating the expected performance of DA in mobile sensing.

However, these approaches assume the availability of labels in the source domain, making them incompatible with our unsupervised pre-training scenario. DARLING [23] addresses the domain shift from unsupervised learning and covers the problem by integrating conditional optimization that optimizes the contrastive loss per domain. However, our approach differs from DARLING in that our method uses the available target domain data (*i.e.*,fine-tuning data) to train domain-specific models. In our evaluation (§IV-F2), we demonstrate that this utilization results in superior performance.

C. Unsupervised Meta-Learning

We consider unsupervised meta-learning (UML) [24], [51], [52] methods due to their effectiveness in few-shot adaptation, which is also applicable to our unsupervised pre-training scenario. Traditional methods employ pseudo-labeling data through augmentation [51] or generative methods [52], followed by supervised meta-learning [53] using the generated labels. Set-SimCLR [24], during pre-training, trains a set encoder by creating sets of augmented samples from the same data, employing contrastive learning to maximize agreement between set embeddings. In fine-tuning, it composes sets of data by classes, generating class prototypes using the set encoder to initialize the classifier's parameters. These prototypes enable rapid adaptation for further few-shot fine-tuning. However, our approach differs in that we perform the adaptation to refine the encoder for the target domain, while Set-SimCLR primarily focuses on making the following classifier adaptable to few-shot fine-tuning. Our evaluation (§IV-F2) demonstrates the superior performance of our approach in mobile sensing scenarios.

III. *ADAPT*^{2}: Few-Shot Adaptation With Meta-Learned Pretext Replay

A. Overview

We present *ADAPT*^{2} as a framework designed to effectively tune a deployed self-supervised model into a personalized model on an end user's device. Fig. 3 shows *ADAPT*^{2} unfolds through two core stages: (1) pre-training models via *self-supervised meta-learning* and (2) adapting pre-trained models via *pretext replay*. The process begins with a model provider, utilizing a large amount of unlabeled data to train the model with a self-supervised objective optimized via meta-learning. This method is designed to enable the models to adapt effortlessly to the few-shot self-supervised learning of the end user. Once trained, these models are deployed to the edge devices of end users. Each user then engages in a domain adaptation process, replaying the learning task of the self-supervised pre-training (*i.e.*, pretext task) with the user's own few-shot data. This results in the model achieving a personalized representation, akin to having been pre-trained with the same user's data. The tuned models are subsequently fine-tuned again with few-shot labeled data, this time for downstream tasks, similar to the original self-supervised learning. Further details about these processes will be elaborated in the following sections.

B. Problem Formulation

Our target scenario is the same as typical self-supervised learning, where a model is pre-trained through self-supervision with unlabeled data, followed by fine-tuning on a target task (in our case, this is an end user's edge device) using very few labeled data. This approach is designed to reflect two key challenges: the difficulty of obtaining costly annotations for pre-training, and the constraints faced by end users in gathering enough data and executing large-scale training on resource-limited edge devices. We explore a setup where data for self-supervised pre-training originate from domains that are completely separate from that of the end user. We assume the availability of domain labels in the pre-training data. Domain labels involve the type of measurement device or anonymized user annotation, which are generally more accessible to acquire as metadata.

Notations. We use x, y, and d to indicate the sensory feature, label, and domain information (*e.g.*, anonymized user or device), respectively. We denote the unlabeled pre-training dataset with N^{pt} samples as $S^{\text{pt}} = \{(x_i, d_i) \,|\, i = 1...N^{\text{pt}}\}$. The set of d_i in S^{pt} composes the source domain, referred as D^s. The fine-tuning dataset with N^{ft} samples is defined as $S^{\text{ft}} = \{(x_i, y_i) \,|\, i = 1...N^{\text{ft}}\}$, where label y_i is available. Data x_i in S^{ft} originates from a single target domain D^t, which is distinct from D^s, *i.e.*, $D^t \cap D^s = \emptyset$. The fine-tuned model is tested in the same target domain D^t.

C. Self-Supervised Meta-Learning

In *ADAPT*^{2}, we pre-train an *adaptable* self-supervised model through the integration of meta-learning. Meta-learning [53], referred to as the method for "learning to learn," is renowned for its efficacy in fine-tuning models for unseen environments using only a few data. However, the incorporation of meta-learning into our framework presents a challenge, as it traditionally relies on supervised training, whereas our approach assumes the absence of labeled data. *ADAPT*^{2} redefines meta-learning by shifting from a supervised to a self-supervised training objective. This shift implies that

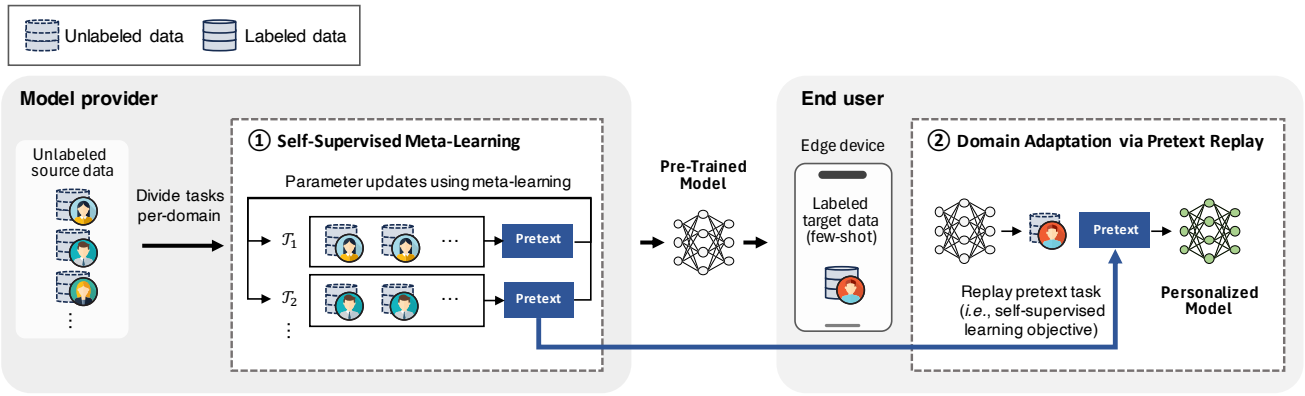

Fig. 3. Overview of *ADAPT*^{2} framework.

the model transitions from "learning to supervise" to "learning to self-supervise." Our design is grounded on the intuition that meta-learning, effective for enhancing supervised learning with few-shot data, can be equally potent in a self-supervised context, allowing our pre-trained models to adapt effectively in a self-supervised manner, even with only a few data. Therefore, we devise a unique adaptation approach for our meta-learned model; we replay the self-supervised training with the end user's few-shot data by performing the same pretext task. We detail this *pretext replay* in §III-D.

Our meta-learning implementation is based on Model-Agnostic Meta-Learning (MAML) [54]. We aim to identify the model parameters that effectively perform pretext tasks within new domains in minimal gradient steps. Our objective closely aligns with the core idea of MAML, training initial parameters for optimal performance on new tasks with minimal gradient updates. It is noteworthy that the model-agnostic nature of MAML enables *ADAPT*^{2} to be adapted flexibly, regardless of the type of self-supervised learning methods.

MAML structures a set of tasks, $\mathcal{T}$, to emulate the process of few-shot training and subsequent testing. Each task, $\mathcal{T}_i$, involves training the model using a small number of examples to optimize task-specific weights, θ_i. The dataset for training these weights is denoted as the *support set*, *i.e.*, $\mathcal{S}_i$. The trained task-specific weights are then evaluated on a distinct dataset termed the *query set*, *i.e.*, $\mathcal{Q}_i$. The evaluation results from each task are aggregated over all tasks and summed into a loss value to optimize the actual model weights.

Task Generation. We design the meta-learning task $\mathcal{T}_i$ to facilitate domain-specific few-shot adaptation. As we aim to adapt the model to data from a single target domain, we configure each task to compose the data samples in $\mathcal{S}_i$ and $\mathcal{Q}_i$ from a single domain d. We call these composed tasks as domain-specific tasks. d is randomly selected from D^s. We further mitigate the risk of overfitting to domains based solely on D^s by leveraging multi-conditioned tasks. Drawing on a prior study [55], we note that incorporating a small proportion (*e.g.*, 30%) of tasks comprised of samples from various random domains can enhance the robustness of the meta-learned model. These multi-conditioned tasks, formed from domain-agnostic random samples, function as new synthetic domains. While not representing real domains, we ensure they are not the primary, accounting for only a small fraction of the total task set $\mathcal{T}$. Algorithm 1 summarizes our meta-learning task generation.

Algorithm 1 *ADAPT*^{2}'s Task Generation for Meta-Learning

Inputs: Pre-train dataset S^{pt}, number of tasks M and number of domain-specific tasks $M^{\texttt{dom}} < M$

for $j \in \{1, 2, ..., M\}$ **do**
  if $j \leq M^{\texttt{dom}}$ **then**
    Select domain d from D^s uniformly at random
    Select K samples with domain d randomly: $\mathcal{S} = \{x_i \in S^{\text{pt}} | d_i = d\}$ such that $|\mathcal{S}| = K$
    Select another set of K samples with d randomly: $\mathcal{Q} = \{x_i \in S^{\text{pt}} | d_i = d\}$ such that $|\mathcal{Q}| = K$
  else
    Select K samples randomly: $\mathcal{S} = \{x_i \in S^{\text{pt}}\}$ such that $|\mathcal{S}| = K$
    Select another set of K samples randomly: $\mathcal{Q} = \{x_i \in S^{\text{pt}}\}$ such that $|\mathcal{Q}| = K$
  end if
  Update a set of task $\mathcal{T} \leftarrow \mathcal{T} \cup (\mathcal{S}, \mathcal{Q})$
end for
return Task set $\mathcal{T}$

Parameter Updates. Our framework trains adaptable parameters to optimize for minimal evaluation loss in a single domain, through simulating few-shot self-supervised learning tasks. For each task $\mathcal{T}_i$, composed from a random single domain. It trains task-specific weights, θ_i, using $\mathcal{S}_i$ and evaluates the corresponding task-specific performance using $\mathcal{Q}_i$. The loss function $\mathcal{L}_{\texttt{SSL}}$, used for both training and evaluation, is tailored to self-supervised learning. Importantly, our framework is designed to be agnostic to self-supervised learning methods, allowing for a versatile loss function application. During the training, task-specific losses are aggregated across all tasks and then utilized to optimize the model parameters, θ. The iteration of this process results in model parameters that are highly adaptable for few-shot self-supervised learning across different domains. Algorithm 2 outlines the process of parameter updates via self-supervised meta-learning. α and β

denote the learning rates for task-specific training and model parameter update.

Algorithm 2 *ADAPT*^{2}'s Self-Supervised Meta-Learning

Inputs: Parameters θ and self-supervised loss $\mathcal{L}_{\texttt{SSL}}$

for epochs **do**
  $\mathcal{T} = \text{TaskGeneration}(S^{\text{pt}})$
  for $\mathcal{T}_i = (\mathcal{S}_i, \mathcal{Q}_i) \in \mathcal{T}$ **do**
    Optimize task-specific weights using $\mathcal{S}_i$: $\theta_i \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\texttt{SSL}}(\theta; \mathcal{S}_i)$
    Evaluate loss of updated model on $\mathcal{Q}_i$: Compute $\mathcal{L}_{\texttt{SSL}}(\theta_i; \mathcal{Q}_i)$
  end for
  Update parameters $\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}_{\texttt{SSL}}(\theta_i; \mathcal{Q}_i)$
end for

D. Adapting Pre-trained Models via Pretext Replay

We introduce *pretext replay* as the core methodology for our few-shot domain adaptation. This adaptation involves refining the model using the same self-supervised learning objective (*i.e.*, pretext task) as in the pre-training step but with the end user's data. This step serves as an additional training stage on the user's edge device, facilitating domain adaptation prior to the actual fine-tuning for the subsequent supervised downstream task (e.g., Human Activity Recognition).

Importantly, pretext replay demonstrates its effectiveness when coupled with self-supervised meta-learning. On the contrary, typical self-supervised models are prone to overfitting when exposed to few-shot training through pretext replay. Similarly, relying solely on self-supervised meta-learning without the subsequent pretext replay step does not yield optimal adaptability for supervised fine-tuning. We propose pretext replay as the unique key bridging the self-supervised pre-training and the subsequent supervised fine-tuning, enabled by the integration of meta-learning in self-supervised pre-training. This synergy between the self-supervised meta-learning and pretext replay is the core aspect of our approach, and its efficacy is validated in an ablation study detailed in §IV-F4.

The model weights, θ, are updated using the parameterized function and the corresponding loss $\mathcal{L}_{\texttt{SSL}}$, originating from the self-supervised meta-learning. Pretext replay uses the fine-tuning data set S^{ft}, without labels, and is performed in a few steps (e.g., 10) with a fixed learning rate, as formulated in the equation:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\texttt{SSL}}(\theta; S^{\text{ft}}). \tag{1}$$

After the adaptation, the model undergoes fine-tuning for the downstream task, which is similar to the normal self-supervised learning setup. S^{ft} with the label information is employed to fine-tune θ towards the downstream task. We utilize the popular linear evaluation protocol [56] for fine-tuning, only training a simple classification head attached after the frozen encoder. The final model trained through this fine-tuning is used for the end-user application.

IV. Experiments

A. Datasets

We set Human Activity Recognition (HAR) as the representative benchmark for evaluating *ADAPT*^{2}. HAR is known for its heterogeneity in users and devices [12], [13], making it an effective benchmark for demonstrating the challenges associated with domain shifts. Datasets are selected to assess the effectiveness of *ADAPT*^{2} across various domain types.

ICHAR [18] comprises inertial measurement unit (IMU) data for classifying nine types of daily activities, such as walking, running, and stair climbing. Data is measured from ten participants using different models of mobile devices (seven smartphones and three watches). Each participant is treated as a unique domain.

HHAR [13] is designed for classifying six daily activities collected from nine users with a combination of four smartwatches and eight smartphones. We define domains based on distinct user-device pairs.

PAMAP2 [21] classifies 12 different activity types using data collected from IMUs placed on three body locations: the wrist, chest, and ankle. Domains are divided by device positions.

DSA [22] encompasses a wide array of 19 daily and sports activities data, gathered from eight participants wearing five IMUs on the torso, arms, and legs. We used device positions as domains.

B. Data Preprocessing

We preprocess all datasets using a fixed window size of 256 and an overlap of 128, following a prior work [18]. Additionally, we standardize the entire dataset to fall within the range of -1 to 1. We focus exclusively on the 3-channel accelerometer data from all available sources. We exclude data from domains with fewer than 500 samples to ensure enough data for training. As a result, we use 20 domains for HHAR, representing combinations of five users and four devices. For all datasets, we first split 70% of the data and use 90% of the splitted data for pre-training and the 10% as the corresponding validation. With the remaining 30% split, we use only a few samples for the few-shot fine-tuning and divide the remaining data with a $5:5$ ratio to compose the validation and test sets. We ensure that the pre-training, fine-tuning, and testing data are completely separated.

C. Baselines

We benchmark *ADAPT*^{2} against baselines chosen for their efficacy in mitigating domain shift between the unsupervised pre-training and the following fine-tuning. Note that most existing domain generalization [14]–[16] and adaptation [17]–[20] methods assume labeled data for pre-training and thus they are not applicable to our scenario. We found two approaches that fit our scenario.

DARLING [23] is a domain generalization method that uses contrastive learning. It optimizes the contrastive loss by composing intra-domain negative samples. Encouraging the intra-domain discrimination for every sample during the training, the model learns domain-invariant features.

Set-SimCLR [24] is an unsupervised meta-learning method that employs a set encoder to enhance the agreement between sets of augmented samples originating from identical source data. Both an instance encoder and the set encoder are trained through contrastive learning. In fine-tuning, the set encoder generates class prototypes from sample sets by class and is used to set the initial weights of the classifier. The resulting classifier, adjusted by the prototypes, facilitates rapid adaptation to novel conditions with the initial weights.

D. Implementation Details

ADAPT^2 serves as a model- and method-agnostic framework applicable to different self-supervised learning approaches. For a fair comparison with baselines, we selected SimCLR, designed for wearables [10], as the primary self-supervised method. This selection aligns with DARLING and Set-SimCLR, which are also presented based on contrastive learning. To maintain consistency, all baseline models utilize a network architecture identical to *ADAPT*^{2}, ensuring compatibility.

We implement *ADAPT*^{2} based on additional self-supervised learning methods, further showing its generalizability. We incorporate two other self-supervised learning approaches popular in HAR: CPC [9] and Multi-Task Learning [11]. Note that we do not extend *ADAPT*^{2} to be built on DARLING and Set-SimCLR. DARLING mainly offers an intra-domain negative sampling method, a concept we already integrate within *ADAPT*^{2} but with a different approach, focusing on creating small tasks for meta-learning. As for Set-SimCLR, it requires the training of a separate set encoder, which requires an additional training component to *ADAPT*^{2}, deviating from our framework's design.

To implement the backbone networks corresponding to the self-supervised learning methods, we used 1D convolutional neural networks (CNN) followed by a projection head of a fully connected layer. Our network design, including the architectures and hyperparameters, aligns with established practices in the prior assessment study [7]. Specifically for CPC, we utilized the latest state-of-the-art version designed for HAR [9]. We conducted a grid search to optimize the hyperparameters for each baseline. For pre-training, we explored learning rates from {1e-4, 5e-4, 1e-3, 5e-3}, batch sizes from {64, 128, 256}, with specific values of {1024, 2048, 4096} for SimCLR-based methods, and weight decays from {0, 1e-4}. During fine-tuning, we used a fixed learning rate of 0.005 for the linear evaluation protocol and a reduced learning rate of 0.001 for end-to-end fine-tuning, which exhibited superior performance. We trained models with 100 epochs for pre-training and 20 epochs for fine-tuning using Adam optimizer. We utilized parameters within the same range as the baselines for learning rate and weight decay for meta-learning. We searched for the task-specific learning rate from {1e-3, 5e-3, 1e-2} and the inner iteration steps for pretext replay from {10, 20, 30}. We kept the number of domain-specific tasks fixed at eight and multi-conditioned tasks at four. The task size was set to 128 for our evaluation. Note that meta-learning requires larger epochs for convergence than conventional training as fewer parameter updates happen inside an epoch; we set 5K epochs for meta-learning training. We implemented the methods using PyTorch, and the training was performed on NVIDIA TITAN Xp GPUs.

E. Evaluation Protocol and Metric

We employed the leave-one-domain-out setting [14]. For each domain in the dataset, we designate it as the target domain for fine-tuning and testing purposes while utilizing all other domains for pre-training. We conduct evaluations in this manner for each domain, rotating through each one as the target domain and then averaging the results. We select few-shot (*e.g.*, 1, 2, 5, 10) samples for fine-tuning, and then evaluate the performance in the same target domain. We set two fine-tuning protocols: linear evaluation and end-to-end fine-tuning. We consider linear evaluation the basic method for fine-tuning, which uses the pre-trained encoder as a frozen feature extractor and exclusively trains a following linear classification layer. All experiments using *ADAPT*^{2} used the linear evaluation protocol. As *ADAPT*^{2} involves the pretext replay step refining the encoder parameters, we further use end-to-end fine-tuning protocol for baseline methods by fine-tuning the entire network without freezing the encoder. It is for a comparison in a fair setting when encoder parameter update is allowed in both cases. The evaluation is performed with five random seeds, and we report the average value and standard deviations. To measure performance, we utilize the macro-averaged F1-score as our evaluation metric. This metric is chosen due to its ability to handle the class imbalance present in the data.

F. Results

1) Integration With Different Self-Supervised Learning Methods: Table I depicts the evaluation against various self-supervised learning methods, SimCLR, CPC, and Multi-Task Learning. Fine-tuning is performed with ten-shot samples, and *ADAPT*^{2} frameworks are implemented on each baseline for a fair comparison. *ADAPT*^{2} consistently enhances F1-scores across different self-supervised learning methods: 8.8%p for SimCLR, 7.1%p for CPC, and 4.1%p for Multi-Task Learning on average. The level of improvement varies depending on the self-supervised learning method, indicating that the impact of domain shift is different among self-supervised learning methods. We explore this topic further in our subsequent analysis (see §IV-F5). Nevertheless, *ADAPT*^{2} consistently outperforms all methods as a flexible framework. Furthermore, we observe that the best-working base method varies across datasets. For example, SimCLR excels in ICHAR, while Multi-Task Learning performs better in DSA. This variability highlights *ADAPT*^{2}'s method-agnostic versatility, allowing flexible integration with the most effective self-supervised learning method for specific applications.

2) Comparison with Baselines: Table II shows the performance of *ADAPT*^{2} compared with the domain generalization and adaptation baselines. The experiment was based on ten-shot fine-tuning. The bold font values indicate the highest value in the same column. Overall, *ADAPT*^{2} consistently

TABLE I

F1-SCORES OF *ADAPT*^{2} BUILT UPON DIFFERENT SELF-SUPERVISED LEARNING METHODS. RESULTS ARE COMPARED WITH THE BASE METHODS. THE HIGHEST AND COMPARABLE SCORES ARE HIGHLIGHTED IN BOLD.

		Domain: User		Domain: Position		
Pre-train	Fine-tune	ICHAR	HHAR	PAMAP2	DSA	Avg.
SimCLR [10]	Linear eval.	0.745 ± 0.024	0.866 ± 0.008	0.549 ± 0.016	0.391 ± 0.006	0.638 ± 0.014
	End-to-end	0.663 ± 0.028	0.836 ± 0.029	**0.589 ± 0.046**	0.253 ± 0.022	0.585 ± 0.031
SimCLR + *ADAPT*^{2} (ours)		**0.836 ± 0.011**	**0.903 ± 0.004**	**0.639 ± 0.030**	**0.526 ± 0.019**	**0.726 ± 0.016**
CPC [9]	Linear eval.	0.765 ± 0.016	0.846 ± 0.005	0.379 ± 0.017	0.371 ± 0.005	0.590 ± 0.011
	End-to-end	**0.816 ± 0.013**	**0.849 ± 0.021**	0.484 ± 0.026	0.352 ± 0.017	0.625 ± 0.019
CPC + *ADAPT*^{2} (ours)		**0.826 ± 0.008**	**0.871 ± 0.005**	**0.527 ± 0.017**	**0.419 ± 0.008**	**0.661 ± 0.009**
Multi-task [11]	Linear eval.	0.716 ± 0.010	0.877 ± 0.003	0.630 ± 0.003	0.456 ± 0.004	0.670 ± 0.005
	End-to-end	0.718 ± 0.019	**0.865 ± 0.030**	**0.636 ± 0.015**	0.378 ± 0.021	0.649 ± 0.021
Multi-Task + *ADAPT*^{2} (ours)		**0.794 ± 0.015**	**0.891 ± 0.005**	**0.659 ± 0.016**	**0.578 ± 0.011**	**0.731 ± 0.012**

TABLE II

F1-SCORES OF *ADAPT*^{2} AND BASELINE METHODS FOR 10-SHOT FINE-TUNING ACROSS FOUR DATASETS, WITH THE HIGHEST AND COMPARABLE SCORES HIGHLIGHTED IN BOLD.

		Domain: User		Domain: Position		
Pre-train	Fine-tune	ICHAR	HHAR	PAMAP2	DSA	Avg.
SimCLR [10]	Linear eval.	0.745 ± 0.024	0.866 ± 0.008	0.549 ± 0.016	0.391 ± 0.006	0.638 ± 0.014
	End-to-end	0.663 ± 0.028	0.836 ± 0.029	**0.589 ± 0.046**	0.253 ± 0.022	0.585 ± 0.031
Set-SimCLR [24]	Linear eval.	0.758 ± 0.010	0.814 ± 0.004	0.487 ± 0.011	0.283 ± 0.007	0.585 ± 0.008
	End-to-end	0.747 ± 0.029	0.848 ± 0.016	0.573 ± 0.015	0.165 ± 0.012	0.583 ± 0.018
DARLING [23]	Linear eval.	0.749 ± 0.019	0.831 ± 0.003	0.551 ± 0.012	0.399 ± 0.008	0.633 ± 0.011
	End-to-end	0.656 ± 0.019	0.844 ± 0.026	**0.580 ± 0.042**	0.258 ± 0.024	0.584 ± 0.028
SimCLR + *ADAPT*^{2} (ours)		**0.836 ± 0.011**	**0.903 ± 0.004**	**0.639 ± 0.030**	**0.526 ± 0.019**	**0.726 ± 0.016**

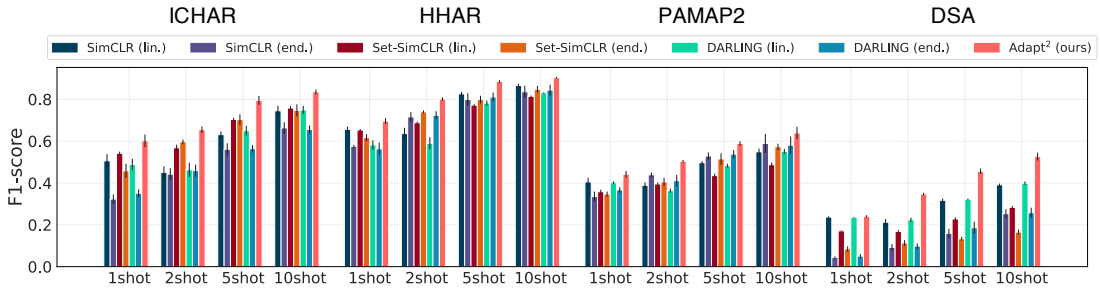

Fig. 4. Average F1-scores of *ADAPT*^{2} and the baselines across different shot numbers (1, 2, 5, 10).

shows the highest F1 scores for all datasets, regardless of the domain type. Set-SimCLR and DARLING occasionally produce comparable scores; however, their enhancements are inconsistent and lack overall improvement. This outcome suggests that the baselines struggle to capture the domain-specific features, given their reliance on a fixed encoder pre-trained out-of-domain. End-to-end fine-tuning sometimes boosts performance; its efficacy depends on the fine-tuning dataset, leading to unpredictable performance enhancements. We attribute this to the sensitivity of few-shot fine-tuning, which is highly influenced by the data quality. End-to-end fine-tuning does not help the baselines, as their design lacks the adaptability of the encoder for few-shot adaptation. With an average increase of 8.8%p in the F1-score, *ADAPT*^{2} works as a robust domain adaptation method.

3) Generalization across Different Shot Numbers: We evaluate the generalizability of *ADAPT*^{2} across different fine-tuning shot numbers. Fig. 4 illustrates the F1-scores for *ADAPT*^{2} and the baseline methods, showcasing average improvements in the F1-score of 4.4%p, 15.5%p, 11.4%p, and 8.8%p across varying shot numbers ($k \in \{1, 2, 5, 10\}$). Fig. 5 presents results when *ADAPT*^{2} is built upon various self-supervised learning methods, consistently demonstrating superior performance. This indicates the efficacy of our approach, making it suitable for extreme data-scarce scenarios, such as one-shot learning.

4) Effect of Self-Supervised Meta-Learning and Pretext Replay: We conduct an ablation study to investigate the impact of technical components within *ADAPT*^{2}. These components include *self-supervised meta-learning* for pre-training and *pretext replay* for domain adaptation. We compare the performance of *ADAPT*^{2} with and without the components to assess their contributions. We build *ADAPT*^{2} upon SimCLR, CPC, and Multi-Task Learning and compare it to three variants: (1) self-supervised learning without any *ADAPT*^{2} components (baseline), (2) self-supervised learning with pretext replay but no meta-learning, (3) self-supervised learning with meta-learning but no pretext replay.

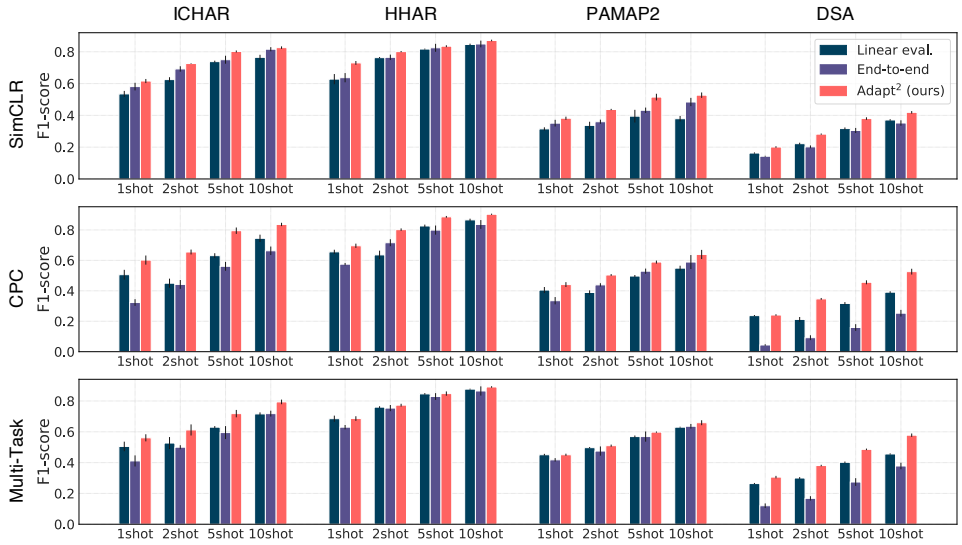

Fig. 5. Average F1-scores of *ADAPT*^{2}, built upon different self-supervised learning methods across different shot numbers (1, 2, 5, 10).

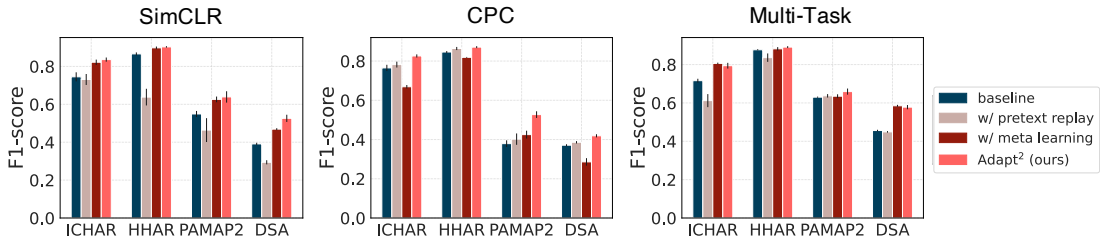

Fig. 6. F1-scores of the ablation study. We compare *ADAPT*^{2} against 1) the baseline self-supervised learning method, 2) the baseline only with pretext replay, and 3) the baseline only with meta-learning. Ten-shot fine-tuning is performed.

Fig. 6 illustrates the results of our ablation study, revealing that each component has a varying impact depending on the self-supervised learning method. For SimCLR, applying only pretext replay to the base model results in a significant performance drop. However, when meta-learning is incorporated, *ADAPT*^{2} outperforms the setting without pretext replay. This suggests that *meta-learning is the key enabler for the domain adaptation through pretext replay* for SimCLR. Additionally, SimCLR with meta-learning achieves outstanding results compared with the baseline, indicating that self-supervised meta-learning alone results in learning meaningful features. We attribute this to the feature learning capability of meta-learning [57], enabling the acquisition of meaningful common features across domains for rapid adaptation.

In the case of CPC, pretext replay enhances the baseline without meta-learning, while using only meta-learning yields poor results. Yet, when pretext replay is applied to the meta-learned model, performance significantly increases, exceeding the improvement observed from the baseline. This suggests that the adaptation step with pretext replay generally aids in learning meaningful features, and our meta-learning significantly amplifies this effect.

For Multi-Task Learning, pretext replay results in performance degradation in the baseline, while meta-learning improves it. Unlike SimCLR or CPC, the impact of pretext replay on the meta-learned model is minimal. Pretext replay slightly improves performance in HHAR and PAMAP2 while even slightly decreasing performance in DSA and ICHAR. This indicates that user-specific adaptation does not provide substantial benefits in the case of Multi-Task Learning. Importantly, meta-learning prevents pretext replay from significantly decreasing performance due to overfitting.

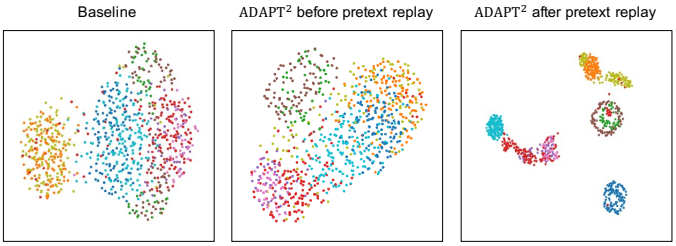

Fig. 7. 2D UMAP visualizations of embeddings from CPC pre-trained models. Embeddings are color-coded by class.

Fig. 7 visualizes the effectiveness of pretext replay when combined with meta-learning, on a domain of ICHAR. We plot 2D UMAP visualizations [58] of embeddings from CPC pre-trained models, color-coded by class. Both CPC baseline and *ADAPT*^{2} before pretext replay lack distinct class-based clustering. It suggests that the models pre-trained out-of-domain fail to grasp meaningful features in the target domain. Meanwhile, *ADAPT*^{2} after pretext replay creates well-defined clusters aligned with class boundaries, demonstrating that

pretext replay significantly enhances the model's ability to capture meaningful features within the target domain.

In summary, our self-supervised meta-learning effectively supports pretext replay, enhancing target-domain performance by training the model to learn general features that can be rapidly refined into domain-specific features. Another finding is that the level of improvement varies depending on the self-supervised learning method. This observation leads us to hypothesize that self-supervised learning methods are differently affected by domain shifts, which we analyze further in the following section (§IV-F5).

5) Domain Shift Effect of Different Self-Supervised Learning Methods: Our ablation study demonstrated varying effectiveness of pretext replay across different self-supervised learning methods. To understand the discrepancies, we delved deeper into how domain shifts affect different self-supervised learning methods. We pre-trained models using CPC, SimCLR, and Multi-Task Learning in a leave-one-domain-out setup and then fine-tuned them in a novel domain. We term it as an out-of-domain setting. For comparison, we established an in-domain setting for baseline, where pre-training and fine-tuning occur within the same domain. When moving from in-domain to out-of-domain, the performance drop quantifies each method's vulnerability to domain shift. While our initial motivational analysis (in §I) compared models pre-trained with different data sizes (the out-of-domain setting had larger data), this study ensures a fair comparison by equalizing the data size across both in-domain and out-of-domain settings.

Fig. 8 illustrates the impact of domain shift on various self-supervised learning methods. When models are fine-tuned and tested on data that differ from their pre-training domain, a common trend of performance decline is observed, highlighting the negative effects of domain shifts. However, the level of this decline varies with the self-supervised learning method. Specifically, models pre-trained with CPC show a notable average drop of 19.15%p F1-score, whereas SimCLR exhibits a 6.7%p decrease and Multi-Task Learning with a more modest decline of 4.95%p.

This variability suggests that the impact of domain shift is dependent on the self-supervised learning method employed during pre-training. CPC's predictive task—forecasting future segments based on past segments—tends to be highly domain-specific. For example, a model pre-trained with CPC on data from a younger user group showing increasing activity levels over time might not perform well on data from old users who exhibit a decline in activity levels. With a different temporal trend, the model's predictive features may not generalize across these domain variations. Conversely, Multi-Task Learning's pretext task involves identifying the type of augmentation applied to the data and is less domain-specific. In the previous example, knowledge acquired from detecting augmentations in one user group's data could transfer more effectively to another group, since recognizing augmented data characteristics (*e.g.*, rotation) does not necessarily depend on user-specific trends.

These findings teach us two insights: (1) domain shift negatively impacts the fine-tuning performance of pre-trained models, and (2) the degree of this impact depends on the pretext task's sensitivity to domains. Our ablation study verifies this, showing that the effectiveness of our pretext replay varies across different self-supervised learning methods. While we have assessed the domain shift effects on three popular methods, our results underscore the necessity to investigate various self-supervised learning methods.

TABLE III
COMPUTATIONAL OVERHEAD OF FEW-SHOT ADAPTATION (ADAPT) AND FINE-TUNING (TUNE) FOR SELF-SUPERVISED LEARNING METHODS.

	CPC		SimCLR		Multi-Task	
Metric	Adapt	Tune	Adapt	Tune	Adapt	Tune
Time (sec)	179.20	84.03	36.05	61.80	18.76	62.33
CPU (%)	44.53	13.86	31.86	10.57	32.15	10.22
Mem (%)	9.54	16.28	4.84	7.26	4.28	6.16

6) Computational Overhead: Our primary goal is to enable the practical deployment of pre-trained models to end-users, particularly those with resource-constrained edge devices. To this end, we assess the computational feasibility of *ADAPT*^{2} for mobile devices, focusing on its user-side operations: pretext replay and fine-tuning to downstream tasks. Our few-shot adaptation strategy is designed to minimize the computational load during these stages. We performed on-device training with a Samsung Galaxy S20 Ultra device equipped with 8 CPU cores and 12GB RAM to evaluate its effectiveness. Our on-device training is implemented via Termux [59], a Linux terminal emulator for Android, to execute PyTorch-based training code. The training protocol follows the same setting of our main evaluation in §IV-F2. The measurement metrics are execution time in seconds and CPU and memory consumption in percentage.

Table III presents the overhead from our pretext replay (Adapt) and fine-tuning (Tune), measured independently across different self-supervised learning methods. Supported by the few-shot setting, operations take minute-level overhead inside the device. Notably, adaptation for SimCLR and Multi-Task Learning was completed in under 40 seconds, consuming less than 5% of memory, even smaller than the overhead of fine-tuning. As a result, all user-side with *ADAPT*^{2} can be performed within 2 minutes. While CPC's adaptation is more resource-intensive due to its complex network architecture, it can still be completed within three minutes and use less than 10% of memory. Our findings confirm that the end-users can conduct all necessary operations of *ADAPT*^{2} on the device—within a 2-minute window for SimCLR and Multi-Task Learning, and under five minutes for CPC, which we believe is manageable computational overhead.

V. DISCUSSION

A. Expansion for Varied Self-Supervised Learning Methods

Our findings underscore the impact of domain shift on different self-supervised learning methods. We observed that the improvement from our domain adaptation varies with the type of self-supervised learning method applied. This suggests a need for a deeper understanding of how domain shifts

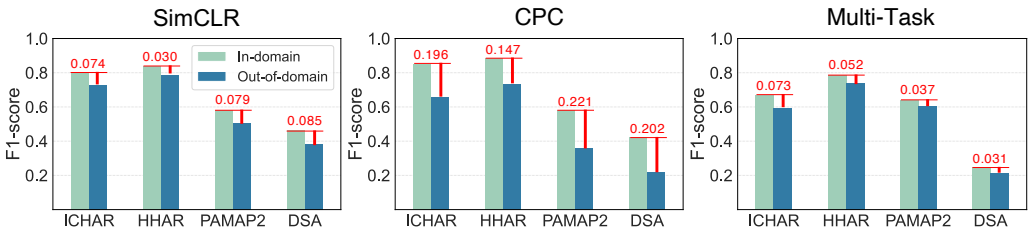

Fig. 8. Fine-tuning performance comparison between models pre-trained in-domain and out-of-domain settings. Ten-shot fine-tuning is performed for all settings. Performance drops between the settings are shown in red.

affect various self-supervised learning approaches. Although our results shed light on the domain shift effects for well-established methods such as SimCLR, CPC, and Multi-Task Learning, the behavior of numerous other self-supervised learning methods [29]–[34] under domain shifts remains unexplored. Addressing this as future work is an essential step in the field.

B. Covering Wider Range of Domains

Our meta-learning task generation is currently crafted to reflect end-user environmental conditions as closely as possible, by composing tasks by user and device domains. However, the variability of real-world data calls for a more complicated approach to task generation. For instance, pre-trained models can be applied to an application with a different type of modality. Possible future work could involve integrating diverse data modalities and user-specific contexts as domains, thus creating meta-learning tasks that better capture the nature of real-world applications. Moving forward, refining task generation will be a priority to ensure our meta-learning approach remains robust across real-world applications.

C. Application to Continuously Changing Environment

We designed our framework with a single domain adaptation step once pre-trained models are deployed to end-users. However, data characteristics within a single domain can change over time due to changing environments. This implies that the adapted model might not perform optimally as domain characteristics change continuously. We anticipate the potential for domain adaptation through pretext replay in such scenarios, as our adaptation step does not require user labels. This allows us to continuously adapt the model using the ongoing data stream from the user. Enhancing the efficiency and effectiveness of this approach in such dynamic scenarios is a direction for future work.

VI. Conclusion

We investigate the domain shift challenge in mobile sensing, where models pre-trained via self-supervised learning are fine-tuned to heterogeneous domains. To address the challenge, we propose *ADAPT*^{2}, a framework that enables few-shot domain adaptation for self-supervised models. Inspired by the observation that models pre-trained on homogeneous domains show superior performance, we refine the pre-trained model to better fit the target domain by replaying the pretext task on the target side. To facilitate few-shot adaptation in this step, our self-supervised pre-training is performed via meta-learning. Our evaluations, conducted across four Human Activity Recognition datasets, indicate that *ADAPT*^{2} consistently outperforms established self-supervised learning and domain generalization methods, achieving an average F1-score improvement of 8.8%p. Moreover, *ADAPT*^{2} proves to be computationally efficient, with the adaptation process being able to be completed on a COTS smartphone in under a few minutes. The findings validate *ADAPT*^{2} as a practical framework that boosts the performance of pre-trained models for end-users while ensuring minimal computational burden.

References

[1] Z. Wang, S. Tan, L. Zhang, Y. Ren, Z. Wang, and J. Yang, "An ear canal deformation based continuous user authentication using earables," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 819–821.

[2] S. Liu, W. Shao, T. Li, W. Xu, and L. Song, "Recent advances in biometrics-based user authentication for wearable devices: A contemporary survey," *Digital Signal Processing*, vol. 125, p. 103120, 2022.

[3] J. Hou et al., "Signspeaker: A real-time, high-precision smartwatch-based sign language translator," in *The 25th Annual International Conference On Mobile Computing And Networking*, 2019, pp. 1–15.

[4] H. Park, Y. Lee, and J. Ko, "Enabling real-time sign language translation on mobile platforms with on-board depth cameras," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–30, 2021.

[5] H. Zhang et al., "Pdmove: Towards passive medication adherence monitoring of parkinson's disease using smartphone-based gait assessment," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 3, no. 3, pp. 1–23, 2019.

[6] X. Song et al., "SpiroSonic: monitoring human lung function via acoustic sensing on commodity smartphones," in *Proceedings Of The 26th Annual International Conference On Mobile Computing And Networking*, 2020, pp. 1–14.

[7] H. Haresamudram, I. Essa, and T. Plötz, "Assessing the state of self-supervised human activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–47, 2022.

[8] H. Haresamudram, I. Essa, and T. Plötz, "Contrastive predictive coding for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–26, 2021.

[9] H. Haresamudram, I. Essa, and T. Plötz, "Investigating enhancements to contrastive predictive coding for human activity recognition," in *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2023, pp. 232–241.

[10] C. I. Tang, I. Perez-Pozuelo, D. Spathis, and C. Mascolo, "Exploring contrastive learning in human activity recognition for healthcare," *arXiv preprint arXiv:2011.11542*, 2020.

[11] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task self-supervised learning for human activity detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–30, 2019.

[12] Y. E. Ustev, O. Durmaz Incel, and C. Ersoy, "User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, 2013, pp. 1427–1436.

[13] A. Stisen et al., "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*, 2015, pp. 127–140.

[14] H. Qian, S. J. Pan, and C. Miao, "Latent independent excitation for generalizable sensor-based cross-person activity recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, no. 13, pp. 11921–11929.

[15] X. Qin, J. Wang, Y. Chen, W. Lu, and X. Jiang, "Domain generalization for activity recognition via adaptive feature fusion," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 1, pp. 1–21, 2022.

[16] W. Lu, J. Wang, Y. Chen, S. J. Pan, C. Hu, and X. Qin, "Semantic-discriminative mixup for generalizable sensor-based cross-domain activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, pp. 1–19, 2022.

[17] Y. Chang, A. Mathur, A. Isopoussu, J. Song, and F. Kawsar, "A systematic study of unsupervised domain adaptation for robust human-activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–30, 2020.

[18] T. Gong, Y. Kim, J. Shin, and S.-J. Lee, "Metasense: few-shot adaptation to untrained conditions in deep mobile sensing," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 2019, pp. 110–123.

[19] Z. Zhou et al., "Xhar: Deep domain adaptation for human activity recognition with smart devices," in *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2020, pp. 1–9.

[20] J. Wang, Y. Chen, L. Hu, X. Peng, and S. Y. Philip, "Stratified transfer learning for cross-domain activity recognition," in *2018 IEEE international conference on pervasive computing and communications (PerCom)*, 2018, pp. 1–10.

[21] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th international symposium on wearable computers*, 2012, pp. 108–109.

[22] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, 2010.

[23] X. Zhang, L. Zhou, R. Xu, P. Cui, Z. Shen, and H. Liu, "Towards unsupervised domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4910–4920.

[24] D. B. Lee, S. Lee, J. Ko, K. Kawaguchi, J. Lee, and S. J. Hwang, "Self-Supervised Set Representation Learning for Unsupervised Meta-Learning," *International Conference On Learning Representations*, 2023.

[25] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[26] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[27] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, 2020, pp. 1597–1607.

[28] J. Wang, T. Zhu, J. Gan, L. L. Chen, H. Ning, and Y. Wan, "Sensor data augmentation by resampling in contrastive learning for human activity recognition," *IEEE Sensors Journal*, vol. 22, no. 23, pp. 22994–23008, 2022.

[29] E. Eldele et al., "Self-supervised contrastive representation learning for semi-supervised time-series classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[30] X. Ouyang et al., "Cosmo: contrastive fusion learning with small data for multimodal human activity recognition," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 324–337.

[31] S. Deldari, H. Xue, A. Saeed, D. V. Smith, and F. D. Salim, "Cocoa: Cross modality contrastive learning for sensor data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–28, 2022.

[32] Y. Jain, C. I. Tang, C. Min, F. Kawsar, and A. Mathur, "Collossl: Collaborative self-supervised learning for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 1, pp. 1–28, 2022.

[33] H. Haresamudram et al., "Masked reconstruction based self-supervision for human activity recognition," in *Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 45–49.

[34] H. Xu, P. Zhou, R. Tan, M. Li, and G. Shen, "Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 220–233.

[35] J. Wang et al., "Generalizing to unseen domains: A survey on domain generalization," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[36] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International conference on machine learning*, 2013, pp. 10–18.

[37] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1446–1455.

[38] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5400–5409.

[39] Y. Wang, H. Li, and A. C. Kot, "Heterogeneous domain generalization via domain mixup," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3622–3626.

[40] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *International Conference On Learning Representations*, 2021.

[41] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32, no. 1.

[42] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12556–12565.

[43] D. Kim, Y. Yoo, S. Park, J. Kim, and J. Lee, "Selfreg: Self-supervised contrastive regularization for domain generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9619–9628.

[44] X. Yao et al., "Pcl: Proxy-based contrastive learning for domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7097–7107.

[45] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 5, pp. 1–46, 2020.

[46] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*, 2015, pp. 1180–1189.

[47] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Correlation-aware adversarial domain adaptation and generalization," *Pattern Recognition*, vol. 100, p. 107124, 2020.

[48] W. Lu, Y. Chen, J. Wang, and X. Qin, "Cross-domain activity recognition via substructural optimal transport," *Neurocomputing*, vol. 454, pp. 65–75, 2021.

[49] M. A. A. H. Khan, N. Roy, and A. Misra, "Scaling human activity recognition via deep learning-based domain adaptation," in *2018 IEEE international conference on pervasive computing and communications (PerCom)*, 2018, pp. 1–9.

[50] T. Gong et al., "DAPPER: Label-Free Performance Estimation after Personalization for Heterogeneous Mobile Sensing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 7, no. 2, Jun. 2023, doi: 10.1145/3596256.

[51] S. Khodadadeh, L. Boloni, and M. Shah, "Unsupervised meta-learning for few-shot image classification," *Advances in neural information processing systems*, vol. 32, 2019.

[52] S. Khodadadeh, S. Zehtabian, S. Vahidian, W. Wang, B. Lin, and L. Bölöni, "Unsupervised meta-learning through latent-space interpolation in generative models," *arXiv preprint arXiv:2006.10236*, 2020.

[53] M. Andrychowicz et al., "Learning to learn by gradient descent by gradient descent," *Advances in neural information processing systems*, vol. 29, 2016.

[54] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*, 2017, pp. 1126–1135.
[55] T. Gong, Y. Kim, R. Choi, J. Shin, and S.-J. Lee, "Adapting to unknown conditions in learning-based mobile sensing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3470–3485, 2021.
[56] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.
[57] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of maml," *International Conference On Learning Representations*, 2020.
[58] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.
[59] "Microsoft Termux," https://termux.dev/en/