Abstract—Human Context Recognition (HCR) from smartphone sensor data is an essential task in Context-Aware (CA) systems including those targeting healthcare and security. Two types of smartphone HCR studies (and datasets) have become popular for training HCR models: a) scripted and b) Unscripted/In-the-wild. Supervised machine learning HCR models can achieve good performance on scripted datasets due to their high quality labels but such models generalize poorly to in-the-wild datasets which are more representative of real-world scenarios. In-the-wild datasets are often imbalanced, have missing or wrong labels, with a diversity of phone placements and smartphone models. Lab-to-field approaches try to train HCR models to learn a robust data representation from a high-fidelity, scripted dataset that is used to improve performance on noisy in-the-wild datasets that have similar labels without having to incur the high expense of gathering high-quality labeled dataset. In this paper, leveraging coincident datasets with the same HCR labels collected in separate scripted and unscripted studies, we propose Triplet-based Domain Adaptation for context Recognition (Triplet-DARE), a novel lab-to-field neural networks method with three key components: 1) a domain alignment loss to learn domain-invariant embeddings, 2) a classification loss to maintain task-discriminative features, 3) a joint fusion triplet loss designed to increase intra-class compactness and inter-class separation in the embedding space of multi-labeled datasets. In rigorous evaluation, Triplet-DARE improved on the F1-score and classification accuracy of state-of-the-art HCR baselines by 6.3% and 4.5%, respectively, and on HCR models with no adaptation by 44.6% and 10.7%, respectively.

Index Terms—Human context recognition, domain adaptation, ubiquitous computing.

I. INTRODUCTION

Context-Aware (CA) systems that can adapt their behavior based on the user’s current context (situation) have significant potential in numerous domains, including healthcare, smart homes, and security systems [1]. Human Context Recognition (HCR), the task of detecting a user’s current situation, is essential for CA systems. While several definitions exist in the literature, in this work, we define Human Context as an (Activity, Prioception) tuple, which consists of the user’s current activity (e.g., walking, running) and the phone placement (e.g., in a pocket, hand, or bag). We focus on CA and HCR on smartphones that are now ubiquitously owned and are equipped with a rich collection of sensors such as accelerometers, gyroscopes, and position sensors. Two study designs are common in human subjects studies to gather HCR datasets for supervised machine learning: 1) scripted [2] or 2) in-the-wild [3]. In scripted studies, participants perform tasks in a pre-planned order under the supervision of a human proctor while a smartphone app continuously records smartphone sensor readings. Afterward, the human proctors annotate users’ sensor data with labels of the contexts they visited. In contrast, in-the-wild studies involve collecting data for several days in the real world as subjects live their lives. A smartphone app continuously gathers sensor data and periodically prompts the smartphone owner to report their current context, which is then used to annotate their sensor data.

Issues with In-the-wild datasets that reduce HCR performance: Supervised machine learning classification HCR models typically achieve high accuracy on scripted datasets due to their high-fidelity sensor data and high-quality context labels. For instance, DeepContext, a state-of-the-art deep learning HCR model, achieved 91.2% accuracy on a scripted dataset[4]. However, scripted datasets are not realistic as the contexts visited and visit patterns are not representative of real life. It is crucial that HCR models are accurate on in-the-wild datasets, which are more representative of real-world deployment scenarios. However, HCR models achieve lower performance when trained directly on more realistic, in-the-wild datasets. For instance, Vaizman achieved 71.7% accuracy using a Multi-Layer Perceptron (MLP) HCR model trained directly on an in-the-wild dataset. This represents a 19.5% drop in accuracy of state-of-the-art HCR models on scripted vs. in-the-wild datasets. This discrepancy in performance is caused by in-the-wild dataset issues including:

1) Diversity of Phone Placements: or positions in which smartphones are placed (prioceptions). Sensor signals have different signatures for the same activity when the phone is carried in different prioceptions [5]. In fact, prioception is one of the most significant sources of variability in smartphone context sensor data [3], as illustrated in Figure (2). Smartphone users may choose to carry their smartphones in a bag, in their hand, or in their coat pocket while performing a given activity (e.g., walking).

2) Weak, noisy, and missing context labels: as users stop providing labels when their lives get busy, or they may erroneously provide wrong labels [6], which presents a challenge for supervised machine learning algorithms [7].

3) Diversity of smartphone models: Unlike scripted HCR studies where subjects use a single study phone model provided by the proctor, subjects in-in-the-wild HCR studies typically use their own phones. The sensor values recorded
for a given context by different smartphone models can differ by as much as 30% [8], presenting an additional challenge for machine learning classifiers.

**Lab-to-field methods**: have recently emerged as viable solutions to achieve good HCR performance on in-the-wild datasets that have noisy, low-quality labels [9]. Lab-to-field approaches try to train highly accurate machine learning models on scripted datasets, which are adapted for *in-the-wild* datasets with the hope of maintaining good performance. However, in general, the performance of HCR models trained naively on scripted datasets often drops when tested on in-the-wild datasets (Going from Lab-to-field). This performance drop is because in addition to the in-the-wild dataset issues listed above, the contexts visited by subjects in the scripted study, as well as their context visit order and visit duration differ significantly from in-the-wild scenarios. Consequently, there are significant differences between the distribution of features extracted from scripted vs. in-the-wild datasets, also known as the *covariate shift problem* [9]–[11].

**Domain Adaptation (DA)** is a transductive transfer learning method and one of the main solutions used to adapt neural networks to mitigate the covariate shift problem. DA has been employed in various related domains, including object detection in computer vision and the problems caused by the variability of wearable sensor placement in ubiquitous computing[5], [11]. Unsupervised DA (UDA) tries to learn a deep learning model using a combination of a labeled source (e.g. scripted dataset) and unlabeled target (e.g. in-the-wild) samples with different distributions to achieve accurate predictions on previously unseen, unlabeled (e.g. in-the-wild) samples [5], [12]. Figure (1) provides a high-level overview of the problem, challenges, and our approach.

**Challenges.** Two key challenges must be addressed for using UDA for Lab-to-Field generalization of smartphone context recognition. First, the previously described data issues with in-the-wild datasets (the diversity of placements, weak and noisy labels, and diverse smartphone types) must be overcome. Secondly, it is challenging to develop a robust method for transferring knowledge from a scripted dataset to a more realistic but considerably noisier in-the-wild dataset with sparse labels.

**Our approach.** We are motivated by the recent empirical success of triplet loss in face identification [13], [14], where variations of the same person’s face images are mapped closely in the learned embedding space. We believe sensor data can benefit from the same approach where there is often variation in sensor signatures corresponding to the same context. Our belief is also consistent with Khaertdinov et al.’s findings where triplet loss was applied recently to mitigate the effects of subject heterogeneity and improve model generalizability [15].

We propose Triple-DARE, a deep Lab-to-field UDA method that is able to leverage the tremendous amounts of unlabeled in-the-wild smartphone HCR data, decreasing the need for human-annotated labels. To facilitate our DA approach, we utilized coincident scripted and in-the-wild HCR datasets in which similar context labels were gathered in both studies [1]. These coincident datasets and similar context labels ensure that there exists a feature representation of contexts that is common between the scripted and in-the-wild datasets, a key requirement for the DA approach. We demonstrate our method’s applicability to HCR models deployed in realistic environments by using context labels gathered in a scripted study only during model development and using DA to mitigate the influence of potentially noisy labels and retain HCR performance on an in-the-wild dataset. *Triple-DARE* outperforms state-of-the-art baselines with 3.79% and 1.89% increases in F1-score and classification accuracy, respectively, and also achieves improvements of 39% and 14.7% in F1-score and classification accuracy, respectively, HCR models without *Triple-DARE*.

**Contributions.** The main contributions of this paper are:

1) We propose *Triple-DARE*, a novel UDA deep-learning framework, which uses a scripted dataset to improve the HCR accuracy of predicting in-the-wild contexts. *Triple-DARE* utilizes a domain alignment loss to learn domain-invariant features, a classification loss to maintain task-discriminative features, and a joint fusion triplet loss to increase intra-class compactness and inter-class separation.

2) We rigorously evaluated *Triple-DARE*, comparing it to multiple state-of-the-art unsupervised domain methods,
including DAN[16], CORAL[17], and HDCNN[18], and bench-marking improvements in HCR performance on target domains in several use cases.

3) We demonstrate that Triple-DARE mitigates in-the-wild dataset challenges when compared to state-of-the-art DA methods, achieving high prediction performance on the target (in-the-wild) domain without the need for large amounts of source labeled samples.

The remainder of this paper is organized as follows. Section II lists related work. Section III introduces our proposed approach. Section IV demonstrates our evaluation and results. Finally, Section V concludes the paper.

II. RELATED WORK

Lab-to-field generalization. Lab-to-field methods previously proposed to handle covariate shifts include importance re-weighting[9], [19] and Positive Unlabeled (PU) classifiers[1]. There is very little work on lab-to-field generalization for HCR. However, a related work that dealt with this problem on wearable electrocardiogram (ECG) data [9], used importance re-weighting to adapt a linear logistic regression model. However, these methods have achieved lower performance when applied to deep neural networks[20].

DA for wearable sensor data. Several DA methods have been proposed to adapt a trained model for use on another related dataset in ubiquitous computing [5], [18], [21]. DA has previously been used to address the issue of variability in the placement of wearable sensors, learning domain-invariant accelerometer [5], [18] and gyroscope[5] features from sensor data by minimizing a discrepancy distance in the Convolutional Neural Network (CNN) embedding. HDCNN[18] investigated adapting a pre-trained model on smartphone data to work on unlabeled smartwatch data. A discrepancy-based approach using Kullback-Leibler (KL) divergence was employed to adapt the model that was pre-trained on smartphone data, for unlabeled smartwatch data. Prior work focuses only on reducing the global distribution discrepancy while learning common feature representations across domains [5], [18]. Our work builds on prior work by utilizing a joint fusion triplet loss to improve intra-class compactness and inter-class separability [12], [13].

III. PROPOSED Triple-DARE METHODOLOGY

A. Problem Formulation

In this work, we utilize data from two coincident datasets in which similar context labels were gathered: 1) a scripted dataset (source) with high-quality labels and 2) an in-the-wild dataset (target) with similar context (⟨Activity, Phone Placement⟩) labels, shown in Table II. Initially, the HCR model is trained using the labeled source dataset. Afterwards, the trained HCR model is utilized to recognize unlabeled contexts in the target dataset. With regards to UDA, there are labeled samples for the source domain and unlabeled samples for the target domain, which have different data distributions. Our goal is to learn a classifier from a combination of labeled source and unlabeled target data, which generalizes well on the target domain. Formally, we have labeled samples \(D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}\) and unlabeled samples \(D_t = \{(x_i^t)\}_{i=1}^{n_t}\) where \(n_s\) and \(n_t\) represent the number of samples in the source and target domains, respectively. Both the source and the target domain share the same feature space \(X_s = X_t\) and label space \(Y_s = Y_t\), but differ in the marginal distribution \((P_s(x_s) \neq P_t(x_t))\). The conditional distributions are presumed to be equal \(P_s(y_t|x_s) = P_t(y_t|x_t)\). We denote \(x\) as a feature vector and \(y\) as a human context represented by a multi-label output vector, where each label produced is a binary output (E.g walking vs not walking). The source and target tasks are presumed to be the same.

B. Overview

As illustrated in Figure (3), Triple-DARE has two types of feature sources extracted from both the source scripted and target in-the-wild datasets: 1) Time and frequency based handcrafted features handled by a feed-forward network and 2) raw three-axial sensors fed into an attention-based CNN that extracts salient features from raw sensor data using a soft attention mechanism. Triple-DARE has three major learning components: 1) A domain alignment loss \(L_d\) to extract embeddings that are invariant across domains. 2) A classification loss \(L_{cls}\) to maintain task-discriminative features, 3) a joint fusion triplet loss \(L_{tri}\) to increase intra-class compactness and inter-class separation in the learned embedding space by learning similar contexts represented by variations of sensor inputs. Triple-DARE’s final output is used for multi-labeled context predictions. For example, based on our <Activity, Phone Placement> definition of context, example contexts include (“Sitting”,”In Bathroom” with “Phone In Hand”). In order to perform our context predictions by learning discriminative and domain-invariant embeddings, our final objective is to minimize the cost function \(C(\cdot)\)

\[
C(\theta) = \lambda_1 L_{cls}^\theta + \lambda_2 L_d^\theta + \lambda_3 L_{tri}^\theta, \quad (1)
\]

where \(\theta\) are model parameters, \(\lambda_1, \lambda_2,\) and \(\lambda_3\) are balancing coefficients. Each of these types of losses are described in more detail in subsequent subsections.

C. Feature Generation

For a given smartphone context dataset, two views are created from raw sensor input data. View 1 is a vector obtained by extracting handcrafted features from all available sensors. View 2 consists of raw tri-axial sensor data. Different feature encoders were used for each type of input view: 1) Multi-Layer Perceptron (MLP) encoder for handcrafted features, and 2) An attention-based CNN encoder for raw sensor data. Finally, a joint fusion encoding is obtained by concatenating the two feature encodings generated. In work on our prior DeepContext HCR model, we found this two-view feature generation approach to be effective [4].

We use data from 5 sensors: accelerometer, gyroscope, GPS, magnetometer, and phone state (e.g., is phone screen locked/unlocked?). We compute statistical, time- and frequency-based features for each sensor type at a sliding
window level. A 10-second window was selected based on findings of prior work [4]). Z-score normalization $z_i = \frac{x_i - \bar{x}}{\sigma}$ was then applied. These 188 handcrafted features [3] were used to construct a vector that was fed into a feed-forward network.

The CNN auto-learns a representation from raw sensor data from three axial sensors (accelerometer, gyroscope, and magnetometer). Adapted from our previous work, DeepContext [4], the CNN we leveraged has a soft attention mechanism that learns salient features, while giving higher weights to regions of the raw sensor data that are more predictive of the user’s context. As a result, the model is able to highlight discriminative CNN features or different contexts.

### D. Domain Alignment Loss

The goal of the domain alignment loss is to map the source and target feature encodings into a standard feature distribution space and learn common feature representations across domains. We utilized Multi Kernel Maximum Discrepancy Mean (MK-MMD), an extension for Maximum Discrepancy Mean (MMD) [23] as our domain alignment loss. MMD is a non-parametric distance measure that may be used to assess the discrepancy between the data representation for the source and target domains, extracted from the $l$th layer in the network.

The formulation of MK-MMD is defined as:

$$\mathcal{L}^{l}_{\text{MK-MMD}} = \frac{1}{r^2} \sum_{l=1}^{N} \sum_{i,j=1}^{N} k(l_{i,j}) - \frac{1}{r^2} \sum_{l=1}^{N} \sum_{i,j=1}^{N} k(l_{i,j})$$

where $k(\cdot)$ is the RKHS norm and $\phi(\cdot)$ is a feature map defined as a combination of multiple positive kernels. The domain alignment loss can be obtained by:

$$\mathcal{L}^{l}_{\text{alignment}} = \frac{1}{r^2} \sum_{l=1}^{N} \sum_{i,j=1}^{N} k(l_{i,j}) - \frac{1}{r^2} \sum_{l=1}^{N} \sum_{i,j=1}^{N} k(l_{i,j})$$

MK-MMD is calculated per layer of the network to quantify the distance between the data representation for the source and target domains. $N_l$ is the number of layers, and $(X^s_l, X^t_l)$ are the distributions of the source and target domains, extracted from the $l$th layer in the network.

### E. Classification Loss

The classification loss aims to leverage source domain labels in discovering features that are discriminative for context predictions. Optimizing our model for classifying contexts on the source domain guides the overall learning process. Since the labels of $D_s$ are available, the classification loss is defined as:

$$\mathcal{L}^{\text{cls}} = \frac{1}{N_s} \sum_{i=1}^{N_s} \ell_{\phi}(f_\theta(x^s_i), y^s_i),$$

where $f_\theta(\cdot)$ is a classifier, $N_s$ is the number of labeled training samples and $\ell_{\phi}$ is a binary cross-entropy function weighted by inverse class frequency to account for class imbalance where infrequently-occurring classes get higher weights than frequently occurring classes.

### F. Triplet Loss

The triplet loss is used to pull samples belonging to the same or similar classes together, while pushing away samples belonging to different classes in an embedding space[13], [14]. Given three types of samples: 1) an anchor sample $x_a$ (i.e. a
HCR on the in-the-wild dataset (target), which is then used to improve the scripted dataset (source), which forms multiple UDA use cases on coincident scripted and in-the-wild datasets after concatenating them into one mini-batch. For constructing triplets suitable to our multi-labeling settings, we lower the distribution disparity. As a result, the quality of the pseudo label can automatically improve. We create triplets from two mini-batches of samples from the source and target domains after concatenating them into one mini-batch. For constructing triplets suitable to our multi-labeling settings, we need a notion of similarity between multi-labeled vectors. Because our dataset is extremely imbalanced, we consider all the positive examples in constructing the triplets. We follow a similar strategy to [13] that focuses on triplets contributing the most to the learning process but modified using our compatibility score to select triplets that satisfy this condition:

\[
d(x_a, x_p) + \alpha > d(x_a, x_n) \quad \text{and} \quad c(y_a, y_p) > c(y_a, y_n)
\]  

IV. Experiments

We evaluated Triple-DARE and baseline models for performing multiple UDA use cases on coincident scripted and in-the-wild smartphone HCR datasets. The overarching goal was to use Triple-DARE to learn a robust representation from the scripted dataset (source), which is then used to improve HCR on the in-the-wild dataset (target).

G. Joint-Fusion Triplet Mining

Triplet mining is the process of constructing triplets (anchor, positive, and negative) for triplet loss calculations. Following the practice in [13], we adopt the online triplet mining strategy which does not require a complete pass on the training set beforehand. Since finding triplets across two domains requires the existence of target domain labels, the classifier trained on the source domain is used to construct pseudo labels for target domain samples during training the classifier, which is one of the most common solutions for UDA problems. [12]. We re-assign pseudo labels every few iterations because the classifier accuracy on the target dataset increases steadily during training. Additionally, the domain alignment loss can also improve the classifier’s accuracy on the target dataset by lowering the distribution disparity. As a result, the quality of the pseudo label can automatically improve. We create triplets from two mini-batches of samples from the source and target domains after concatenating them into one mini-batch. For constructing triplets suitable to our multi-labeling settings, we need a notion of similarity between multi-labeled vectors. We first define a compatibility score between two contexts \(y_1, y_2\) that are both represented as binary labels, as the dot product between them:

\[
c(y_1, y_2) = y_1 \cdot y_2
\]

Because our dataset is extremely imbalanced, we consider all the positive examples in constructing the triplets. We follow a similar strategy to [13] that focuses on triplets contributing the most to the learning process but modified using our compatibility score to select triplets that satisfy this condition:

\[
d(x_a, x_p) + \alpha > d(x_a, x_n) \quad \text{and} \quad c(y_a, y_p) > c(y_a, y_n)
\]
was initialized to 1e-1, balancing coefficients were initialized as $\lambda_1 = 1$, $\lambda_2 = 0$, and $\lambda_3 = 0$. The balancing coefficients and the learning rate were increased or decreased following the schedule mentioned in [24], making our model highly confident on source labels and less sensitive to low-quality pseudo labels at the early stages of the training. The batch size was set to 256 and the Adam optimizer was used. The back-bone layers used in our DA method were shared across all experiments: handcrafted-features with a 2-layer MLP, each layer having 16 hidden dimensions and an MLP domain classifier with one layer with 32 hidden dimensions, and CNN with attention blocks for separate and merged sensors layers, followed by an average pooling layer, adopted from [4]. All raw sensor data were input to a 3-layer CNN. Then their outputs are concatenated and forwarded to another 3-layer CNN. Attention blocks are used to focus on salient regions in inputs [4], [25]. Euclidean distance was used for computing pairwise distances in triplet mining and $\alpha$ was set to 0.1. The final context prediction layer has LeakyReLU activation, followed by Sigmoid activation.

2) Evaluation Protocol Due to the class imbalance in our context datasets, in addition to classification accuracy, we used the F1 metric to evaluate HCR performance in the UDA setting. As the sizes of the source and target domain datasets may not be the same, we iterate through the target domain dataset with random sampling. However, we evaluate our model on all samples in the target domain dataset.

D. Results and Findings

1) Overall Results: The overall performance scores for our Triple-DARE compared to baseline models is reported in Table (III). Triple-DARE outperforms the baseline methods in the overall UDA tasks and Lab-to-field UDA task by 4.5% increase in F1-score and 6.3% increase in classification accuracy. The result, shown in Figure (4), demonstrate the performance per label aggregated over all the UDA tasks, showing that our approach outperforms state-of-the-art methods across several context labels. In general, the advantage of using UDA methods can be observed over classifiers that are solely trained on the source domain without leveraging unlabeled data. In particular, UDA methods helped a lot in the Jogging, Running, Going Up and Down Stairs labels where the user is likely to stop providing labels while performing these activities in the wild. Triple-DARE improves adaptation using the high fidelity labels acquired in the scripted study.

2) Training under insufficient labels: As presented in Table (IV), we studied the performance of our model when the number of labels from the source domain is varied. Triple-DARE achieves higher prediction scores on the target domain, with small amounts of source labels, outperforming baseline methods in almost all UDA tasks.

3) Intra-class compactness and inter-class separation: To provide a measure of compactness and separation in the learned feature embeddings, we utilized the Silhouettescore $\text{Score} = \frac{\min(b_i - a_i) - a_i}{\max(b_i - a_i)}$, where $b_i$ is the shortest mean distance between a point to all other points in any other cluster, whereas $a_i$ is the mean distance of $i$ and all data points from the same cluster. This score measures both compactness and separation. To calculate the Silhouette scores on the learned feature embeddings, we assign each instance with cluster labels using one of the binary context labels. Then, we averaged the scores over labels. The scores are reported in Figure (5), which shows that Triple-DARE achieves better compactness and separation scores in most UDA tasks. CORAL achieves higher scores than DAN in most cases.

V. CONCLUSION

Several issues reduce the performance of machine learning HCR models on in-the-wild datasets, including the diversity of phone placements and smartphone models and weak, noisy, or missing labels. Lab-to-field methods try to improve the performance of HCR models by first training them on similar scripted datasets, then adapting them for use in predicting context labels in in-the-wild datasets. We designed DA strategies that are susceptible to covariate shifts between the scripted and in-the-wild datasets, improving lab-to-field generalization. This paper proposed Triple-DARE, a UDA deep-learning model for HCR on smartphones, comprised of three parts: 1) domain alignment loss using MK-MMD 2) a classification loss and, 3) joint-fusion triplet loss designed for multi-labeled datasets. Triple-DARE learns domain-invariant features common to both datasets, reducing the influence of highly noisy in-the-wild data by using its attention mechanism to focus on salient regions in sensor inputs, achieving a high F1-score for various UDA tasks on our scripted and in-the-wild context datasets. Using its domain alignment loss, Triple-DARE is able to map the source and target feature encoding into a standard feature distribution space with better performance than state-of-the-art baseline methods. Furthermore, the triplet loss improves discrimination, increasing intra-class compactness and inter-class separation while leveraging massive amounts of unlabeled data. Triple-DARE outperforms other state-of-the-art DA baselines, improving on their F1-score and classification accuracy by 4.6% and 1.89%, respectively, and improving on models with no adaptations by 10.7% and 14.7%, respectively. In future work, we plan to leverage our proposed method in representation learning for smartphone sensor data. One limitation in our model is the assumption that an identical number of sensors are available in both scripted and in-the-wild datasets. We could improve our framework by increasing model robustness against missing sensors during model inference.

### Table III: Overall context prediction

<table>
<thead>
<tr>
<th>Overall UDA Tasks</th>
<th>Accuracy</th>
<th>F1-micro</th>
<th>Lab-to-field UDA Tasks</th>
<th>Accuracy</th>
<th>F1-micro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triple-DARE</td>
<td>0.879</td>
<td>0.366</td>
<td>CORAL</td>
<td>0.845</td>
<td>0.188</td>
</tr>
<tr>
<td>CORAL</td>
<td>0.806</td>
<td>0.302</td>
<td>DAN</td>
<td>0.674</td>
<td>0.294</td>
</tr>
<tr>
<td>HDCNN</td>
<td>0.816</td>
<td>0.3215</td>
<td>HDA</td>
<td>0.708</td>
<td>0.146</td>
</tr>
<tr>
<td>Source (no adaptation)</td>
<td>0.433</td>
<td>0.239</td>
<td>Source (no adaptation)</td>
<td>0.552</td>
<td>0.133</td>
</tr>
</tbody>
</table>
TABLE IV: F-1 scores - comparing different methods for various UDA tasks, varying the amounts of source labels used.

<table>
<thead>
<tr>
<th>Training %</th>
<th>Method</th>
<th>$S_{Bag}$</th>
<th>$S_{Bag}$</th>
<th>$S_{Hand}$</th>
<th>$S_{Hand}$</th>
<th>$S_{Pocket}$</th>
<th>$S_{Pocket}$</th>
<th>Avg</th>
<th>$W_{Bag}$</th>
<th>$W_{Hand}$</th>
<th>$W_{Pocket}$</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>Triple-DARE</td>
<td>0.500</td>
<td>0.651</td>
<td>0.213</td>
<td>0.318</td>
<td>0.652</td>
<td>0.467</td>
<td>0.467</td>
<td>0.101</td>
<td>0.080</td>
<td>0.326</td>
<td>0.169</td>
</tr>
<tr>
<td></td>
<td>CORAL</td>
<td>0.357</td>
<td>0.328</td>
<td>0.357</td>
<td>0.428</td>
<td>0.352</td>
<td>0.378</td>
<td>0.567</td>
<td>0.089</td>
<td>0.087</td>
<td>0.150</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>DAN</td>
<td>0.341</td>
<td>0.436</td>
<td>0.285</td>
<td>0.265</td>
<td>0.418</td>
<td>0.403</td>
<td>0.358</td>
<td>0.079</td>
<td>0.077</td>
<td>0.165</td>
<td>0.107</td>
</tr>
<tr>
<td>0.4</td>
<td>Triple-DARE</td>
<td>0.557</td>
<td>0.617</td>
<td>0.444</td>
<td>0.492</td>
<td>0.767</td>
<td>0.511</td>
<td>0.565</td>
<td>0.118</td>
<td>0.143</td>
<td>0.359</td>
<td>0.207</td>
</tr>
<tr>
<td></td>
<td>CORAL</td>
<td>0.380</td>
<td>0.584</td>
<td>0.455</td>
<td>0.451</td>
<td>0.633</td>
<td>0.484</td>
<td>0.499</td>
<td>0.092</td>
<td>0.075</td>
<td>0.165</td>
<td>0.111</td>
</tr>
<tr>
<td></td>
<td>DAN</td>
<td>0.452</td>
<td>0.509</td>
<td>0.488</td>
<td>0.451</td>
<td>0.721</td>
<td>0.459</td>
<td>0.502</td>
<td>0.101</td>
<td>0.090</td>
<td>0.244</td>
<td>0.146</td>
</tr>
<tr>
<td>0.6</td>
<td>Triple-DARE</td>
<td>0.497</td>
<td>0.588</td>
<td>0.570</td>
<td>0.653</td>
<td>0.744</td>
<td>0.542</td>
<td>0.599</td>
<td>0.111</td>
<td>0.112</td>
<td>0.341</td>
<td>0.188</td>
</tr>
<tr>
<td></td>
<td>CORAL</td>
<td>0.577</td>
<td>0.608</td>
<td>0.505</td>
<td>0.505</td>
<td>0.754</td>
<td>0.448</td>
<td>0.580</td>
<td>0.110</td>
<td>0.123</td>
<td>0.210</td>
<td>0.148</td>
</tr>
<tr>
<td></td>
<td>DAN</td>
<td>0.540</td>
<td>0.634</td>
<td>0.428</td>
<td>0.459</td>
<td>0.635</td>
<td>0.429</td>
<td>0.524</td>
<td>0.100</td>
<td>0.084</td>
<td>0.209</td>
<td>0.127</td>
</tr>
<tr>
<td></td>
<td>HDCNN</td>
<td>0.345</td>
<td>0.561</td>
<td>0.575</td>
<td>0.540</td>
<td>0.645</td>
<td>0.445</td>
<td>0.518</td>
<td>0.094</td>
<td>0.102</td>
<td>0.285</td>
<td>0.160</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.518</td>
<td>0.619</td>
<td>0.409</td>
<td>0.488</td>
<td>0.721</td>
<td>0.507</td>
<td>0.544</td>
<td>0.111</td>
<td>0.112</td>
<td>0.341</td>
<td>0.188</td>
</tr>
<tr>
<td></td>
<td>No Adaptation</td>
<td>0.319</td>
<td>0.469</td>
<td>0.476</td>
<td>0.470</td>
<td>0.260</td>
<td>0.315</td>
<td>0.385</td>
<td>0.108</td>
<td>0.110</td>
<td>0.180</td>
<td>0.133</td>
</tr>
</tbody>
</table>

VI. ACKNOWLEDGMENT

This material is based on research sponsored by DARPA under agreement number FA8750-18-2-0077. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

REFERENCES


