Target-agnostic Source-free Domain Adaptation for Regression Tasks

Tianlang He[1], Zhiqiu Xia[2], Jierun Chen[1], Haoliang Li[3], S.-H. Gary Chan[1]
[1,2]The Hong Kong University of Science and Technology, Hong Kong S.A.R., China
[3]City University of Hong Kong, Hong Kong S.A.R., China
Email: [1]{theaf, jcheneh, gchan}@cse.ust.hk, [2]zxiaae@connect.ust.hk, [3]haoliang.li@cityu.edu.hk

2024 IEEE 40th International Conference on Data Engineering (ICDE) | 979-8-3503-1715-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDE60146.2024.00121

***Abstract*—Unsupervised domain adaptation (UDA) seeks to bridge the domain gap between the target and source using unlabeled target data. Source-free UDA removes the requirement for labeled source data at the target to preserve data privacy and storage. However, previous works on source-free UDA assume knowledge of domain gap, and hence is limited to either target-aware or classification task. To overcome it, we propose TASFAR, a novel target-agnostic source-free domain adaptation approach for regression tasks. Using prediction confidence, TASFAR estimates a label density map as the target label distribution, which is then used to calibrate the source model on the target domain. We have conducted extensive experiments on four regression tasks with various domain gaps, namely, pedestrian dead reckoning for different users, image-based people counting in different scenes, housing-price prediction at different districts, and taxi-trip duration prediction from different departure points. TASFAR demonstrates significant superiority over state-of-the-art source-free UDA approaches, achieving an average error reduction of 22% across the four tasks and comparable accuracy to source-based UDA, all without relying on source data.**

***Index Terms*—Domain adaptation, unsupervised learning, regression model, model uncertainty**

I. Introduction

Deep learning has shown promising results in various tasks, including location sensing [1], [2], [3], people counting [4], [5], and activity recognition [6], [7]. However, the performance of deep models often significantly degrades when the target data deviates from the input distribution of the training dataset, known as the source domain. To tackle this, unsupervised domain adaptation (UDA) has been proposed to learn a target model, i.e. aligning the model features extracted from both source and target domains using unlabelled target data.

In traditional UDA, the source dataset is made available to the target for adaptation. Due to the potential large size of the database (several gigabytes or more), it is not storage-efficient and unsuitable for devices with limited storage capacity. Though some works [8] [9] have tried to reduce such storage by compression, they still consume substantial storage with trade-off on the adaptation quality. To remove dataset storage, source-free UDA has been recently proposed, which is to adapt the source model pre-trained by source data directly with unlabeled target data. In this work, we consider one-shot source-free UDA; readers interested in consecutive adaptations may refer to studies on continual learning [10], [11], [12] and references therein.

[0]Code is available at https://github.com/Siriusize/TASFAR_DA.

In source-free UDA, the absence of source data causes complication in measuring the domain gap (the discrepancy between the input distribution of the source and target). Existing approaches bridge the gap based on either input data or features. The approaches based on input data require prior knowledge of domain gap and simulate such gap by means of data augmentation to extract invariant features [13], [14], [15]. On the other hand, the feature-based approach is generally applied to classification tasks by measuring and minimizing domain gaps in feature space [16], [17], [18], [19].

While impressive, previous approaches assume the knowledge of domain gap. As target domains are often agnostic when designing adaptation algorithms, domain gaps can be unknown, heterogeneous and complex to simulate [20]. For example, a source model may be deployed in different target scenarios in terms of user behaviors, device heterogeneity and operating environments. Furthermore, many machine learning tasks are regression in nature. In contrast to classification tasks where target data of the same label can be correlated in the feature space to shed light on domain gap [16], [18], the vast continuous label space of regression tasks without overlapping labels poses great difficulty for the deep regressor to adapt and converge.

We consider, for the first time, target-agnostic source-free UDA for regression tasks. The key observation is that the target label, like the input data that all conform to the target domain, also originates from the same target scenario. As an example, if a target user's stride length mostly falls into a certain range (say, 0.5 to 0.8m), his/her next stride length is likely within the range as well. Therefore, in contrast to the previous approaches that measure and bridge domain gaps in input data or feature space of the source model, we directly estimate the label distribution of the target scenario and use it to calibrate source models. By considering only the label distribution, we can achieve target-agnostic adaptation for regression tasks that is orthogonal to target domain.

We propose ***TASFAR***, a novel ***t***arget-***a***gnostic ***s***ource-***f***ree domain ***a***daptation approach for ***r***egression tasks. We show its overall system diagram in Figure 1. TASFAR first classifies the target data into confident data and uncertain data based on a confidence classifier depending on prediction confidence [21] [22]. Based on the confident data, TASFAR uses a label distribution estimator to generate a label density map. Then, a pseudo-label generator leverages the label density map

2375-026X/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDE60146.2024.00121

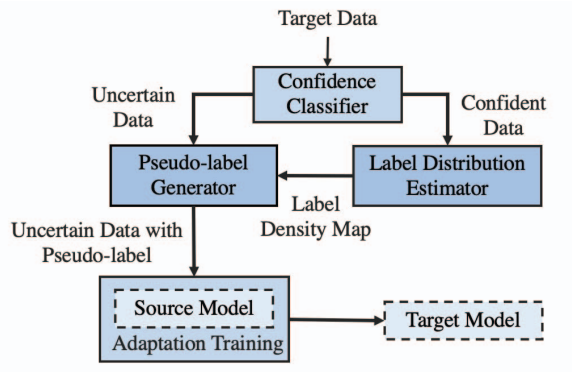

Fig. 1. System diagram of TASFAR. First, TASFAR uses a confidence classifier to classify the target data into confident and uncertain data. The confident data are used by a label distribution estimator to generate a label density map. The uncertain data are pseudo-labeled by a proposed pseudo-label generator based on the label density map. Then, TASFAR uses the pseudo-labeled uncertain data to train the source model to be the target model.

to pseudo-label the uncertain data. Finally, TASFAR uses the pseudo-labeled uncertain data to fine-tune the source model by supervised learning, after which the target model is delivered.

To the best of our knowledge, TASFAR is the first target-agnostic source-free regressor adaptation approach based on label distribution. Our contributions are the following:

- *Label distribution estimator using prediction confidence:* We estimate the label distribution of confident data to pseudo-label uncertain data. However, the target labels are unavailable in the setting of UDA. We thus propose a label distribution estimator to overcome it. To be specific, the proposed estimator utilizes the prediction confidence of the source model to estimate the target label distribution, which is represented as a label density map.
- *Pseudo-label generator based on label density map:* Generated from the same target scenarios, the label distribution of confident data can be the prior knowledge of the labels of uncertain data. Therefore, we propose a pseudo-label generator that utilizes the label density map to pseudo-label the uncertain data. Specifically, the pseudo-label generator pseudo-labels uncertain data by considering the joint distribution of label density map and source model prediction. To avoid generating low-quality pseudo-labels that cause accuracy degradation, the generator also weighs the pseudo-labels by evaluating their credibility based on the map densities.

We have conducted extensive experiments to validate TASFAR on four regression tasks: location sensing (pedestrian dead reckoning) [23]; image-based people counting [24]; and two prediction tasks [25], [26]. We compared TASFAR with the existing source-free UDA with pre-defined domain gap and traditional source-based UDA approaches (expectedly the best performance due to the availability of source dataset). Our experimental results show that, as compared with the state-of-the-art source-free UDA approaches, TASFAR achieves on average a substantial 14% and 24% reduction in localization error on different users and counting MSE on various crowd scenes, respectively, and 22% and 28% reduction of prediction errors on the two prediction tasks. Without access to source datasets, TASFAR impressively achieves similar accuracy as the source-based UDA approaches.

The remainder of this paper is organized as follows. We review related works in Section II and present TASFAR in Section III in terms of its confidence classifier, label distribution estimator, and pseudo-label generator. We discuss in Section IV illustrative experimental results, followed by conclusion in Section V. Finally, we discuss future works in Section VI.

II. Related Work

UDA for deep models has been extensively studied [27], [28], [29]. These works align the source and target domains either by input data or deep model features. Some pioneering works on data alignment [30], [31] reduce the domain gap by importance sampling on source data to simulate target data distribution. Recent data alignment approaches [32], [33], [34] study style transfer from target to source data through deep generative models. Instead of manipulating the input data directly, the feature alignment approach focuses on aligning the extracted deep features from both domains to minimize the feature discrepancy. This discrepancy can be measured using techniques such as maximum mean discrepancy (MMD) [35], [36], [37], adversarial neural networks [38], [39], or reconstruction loss [40]. However, these traditional UDA approaches require the coexistence of source and target datasets. This may raise concerns on source-data privacy and can be troublesome when deploying to resource-constrained devices.

To overcome that, some research works study transforming source data into lightweight forms [41], [42]. Works in [8], [43], [44] compress the source data into generative models and deploy the source model with the data generator to target scenarios for UDA. However, the deep generator may not protect data privacy [45] and still consumes the precious storage of resource-constrained devices. Other works align target features with the stored statistics of the source feature, such as feature prototype [46], feature histogram [9], and batch normalization parameters [47]. However, these methods are limited to addressing small domain gaps, and there is a trade-off in adaptation quality when compared to traditional source-based UDA approaches. This limitation arises because the proposed feature statistics inevitably experience information loss from the source datasets.

Source-free UDA further reduces the storage requirement by adapting the source model with only a set of unlabeled target data, which is more privacy-preserving and applicable for resource-constrained devices. Existing studies focus on either input data or feature alignment. The input data-based approaches [13] [14] [15] learn to extract domain-invariant features against data augmentation (e.g., image rotation) that simulates the domain gap from target to source, whilst they require target-specific knowledge that is usually unavailable

when designing adaptation algorithms. The feature-based approach studies to measure the similarity of the target feature to the source. Works in [17] [16] use the information entropy of classification score as an indicator of feature similarity, where low information entropy indicates source-like features. Other works [18] [19] optimize the compactness of the target features because the source features are usually clustered or correlated by classification categories. Nevertheless, the current source-free approaches either rely on target-specific information or the properties of classification, which cannot be extended to regression tasks whose target domains are unknown. In comparison, TASFAR explores the label distribution of target scenarios to calibrate source models, regardless of any classification properties or information of target domains.

III. TASFAR Design

In this section, we discuss the technical design of TASFAR. First, we overview TASFAR in Section III-A and introduce its confidence classifier in Section III-B. Then, we discuss the label distribution estimator in Section III-C and pseudo-label generator in Section III-D.

A. Overview

The problem entails a source regression model f_{θ_s} and a set of target data $(x_t, y_t) \in D_t (\subset \mathcal{D})$. The parameters of the source model $\theta_s \subset \Theta$ are learned from a source dataset $(x_s, y_s) \in D_s (\subset \mathcal{D})$. The ground truth (or label) of the target data y_t exists but is unknown. Even though the source model f_{θ_s} is performing the same task in both source and target scenarios, i.e., $Pr(x|y_s) = Pr(x|y_t)$, the statistical distribution of inputs (i.e. domain) of both datasets can be different, i.e., $Pr(x_s) \neq Pr(x_t)$, which is termed domain gap. Our objective is to adapt the parameters of the source model θ_s to be $\theta_t \subset \Theta$, so the target model f_{θ_t} minimizes the prediction error on target domain

$$\min \sum_{(x_t, y_t) \in D_t} \|f_{\theta_t}(x_t) - y_t\|_n . \tag{1}$$

In the problem setting of source-free UDA, the labels of the target data are unavailable. We need an alternative objective that complies with Equation 1. For this, previous works try to align accuracy on the target to the source, where they either simulate or measure from classifiers the domain gap and extract domain-invariant features against it. However, they either only work for specific target domains or are designed for classification tasks. In this paper, we aim to design an adaptation approach for regression models and consider a more practical setting where target domains are unknown in advance, which calls for a new objective.

Instead of extracting domain-invariant features, we directly aim for Equation 1 by replacing the target label y_t with a pseudo-label $\hat{y}_t$:

$$\min \sum_{(x_t, y_t) \in D_t} \|f_{\theta_t}(x_t) - \hat{y}_t\|_n , \tag{2}$$

where the pseudo-label $\hat{y}_t$ is supposed to be closer to the ground truth y_t compared with the source model prediction. In contrast to previous approaches that operate on the input or feature space of source models, this adaptation directly focuses on the label space of target scenarios. However, obtaining the pseudo-label $\hat{y}_t$ is not intuitively straightforward.

In most machine-learning paradigms, the labels that provide supervised information are generally regarded as independent, while their underlying meanings are essentially correlated in the real world. Take an image-based recognition task as an example. The categories of "dog" and "cat" naturally resemble each other in front of the label "boat", though such a pattern is not presented by their one-hot labels. Conceptually, the correlation among label classes is referred to as 'dark knowledge" in the field of knowledge distillation [48], [49], [50]. By exploring such correlation among label classes, works in model compression have successfully equipped small models with the accuracy of large models, especially for deep classifiers [51], [52]. Enlightened by knowledge distillation, we extend the idea of dark knowledge to source-free UDA for regression tasks, regarding regression as the classification task with infinite categories of labels. Nevertheless, the challenging issues remain: 1) how are the labels of target scenarios correlated, and 2) how to leverage the correlation to get the pseudo-label?

We observe that, in many tasks, target labels are inherently correlated due to the target scenarios. Specifically, due to the same target scenarios, the generation processes of the labels usually share commonalities, such as the same person, site, device, and so on. Therefore, just like the input data that all conform to the target domain (say, cartoon or realistic images), the target labels from the same scenario usually form a label distribution that characterizes the scenario. To illustrate this, we show one example in Figure 2 using the task of stride length estimation, where the label distribution reflects the walking pattern of the person.

From another perspective, the label distribution of a scenario can be viewed as the prior knowledge of predicting single labels, which is especially useful when a prediction is uncertain. For example, if an elder's stride length mostly falls into a range (say, 0.5 to 0.8m), his/her next stride length is highly likely to be within the range. Intuitively, when making a random guess, a stride length within the range is expected to be more accurate than the out-of-range one. Therefore, we can leverage the label distribution to generate pseudo-labels for those uncertain predictions, which are expected to be more accurate than the original ones. In this paper, we capture the density information of label distribution, which serves as the prior knowledge to calculate pseudo-labels. More details will be discussed in the following, where we will cover which data need pseudo-labels, how to estimate the label distribution of target scenarios, and how to leverage the distribution to generate the pseudo-labels.

B. Confidence Classifier

In this section, we discuss two important considerations to get our intuition down to earth, which leads to the design of the confidence classifier.

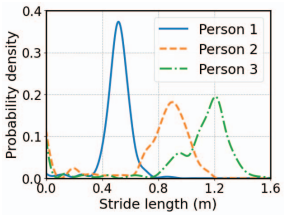

Fig. 2. Stride length distribution of different users: label distribution often characterizes target scenarios.

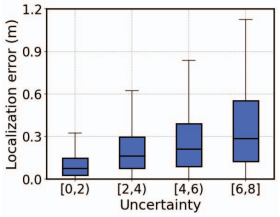

Fig. 3. Example of pedestrian dead reckoning: larger uncertainty tends to indicate larger errors.

How to estimate the label distribution of target scenarios without target labels? If the predictions from the source model are accurate, we can use these predictions to estimate the label distribution of the target. Unfortunately, due to the domain gap between the target and source, we cannot guarantee all source model predictions on target data are accurate. Thus, we need a recognition module to differentiate those accurate predictions from all source model predictions on the target data.

The accuracy of predictions is related to the prediction confidence – since the source model usually produces accurate predictions on familiar input data that lead to high prediction confidence, source model usually shows high confidence in its predictions with high accuracy. We thus use prediction confidence from source models to recognize the accurate predictions. Note that confidence (or uncertainty) estimation for deep learning has been well-studied. For example, the prediction variance caused by the Dropout layer [53] can be interpreted as prediction confidence [21]. More uncertainty estimation methods can be found in [22]. Since most of the uncertainty estimation approaches are orthogonal to both tasks and model performance, employing these approaches does not influence the generality of our approach.

Which kind of target data needs pseudo-labels? We consider the label distribution as the scenario's prior knowledge which is independent of the individual pieces of the target data (or inputs). Thus, it fits for calibrating the source model predictions when the predictions are uncertain. Specifically, the source model shows low prediction confidence when it has trouble analyzing a target input, indicating a failure to utilize the information from input data. In this case, we use the label distribution to calibrate the source model predictions when the prediction confidence is low.

Considering the two factors, we build a confidence classifier to differentiate the target data into confident data and uncertain data based on the source model predictions. The predictions on the confident data are utilized to estimate the label distribution of the target, which calibrates the predictions that the source model makes on the uncertain data. The criterion to differentiate the target data is actually related to how well the source model learns from the source data. In other words, the model's performance on the source data determines its level of confidence in making predictions. Therefore, we differentiate uncertain and confident data based on a threshold

Algorithm 1 Pseudo code of confidence classifier

Input: Target dataset $x_t \in D_t$, source model f_{θ_s}
Parameter: Uncertainty threshold τ
Output: Confident and uncertain data set SET_C, SET_U

```
Initialize SET_C, SET_U
for x_t in D_t do
    Calculate prediction uncertainty u_t using f_{θ_s}
    if u_t < τ then
        Save (f_{θ_s}(x_t), u_t) to SET_C
    else
        Save (f_{θ_s}(x_t), u_t) to SET_U
    end if
end for
return SET_C, SET_U
```

of prediction uncertainty τ whose value is determined by the model performance on source data.

Specifically, if a source model learns well from the source dataset, it should be confident about most of the predictions. Therefore, we regard it as a confident prediction if η (proportion) of the source data show uncertainty lower than τ. This threshold also applies to target data only if using the same source model. It can be determined after the source-model training. Finally, we present the confidence classifier by pseudo-code in Algorithm 1.

C. Label Distribution Estimator

In this section, we introduce our design on the label distribution estimator, which delivers a label density map M of the target scenario using the source model predictions $\tilde{y}_t = f_{\theta_s}(x_t)$ with prediction uncertainty u_t from the confident data. For a concise expression, we focus on the single-dimensional label and leave its extension to the multi-dimensional label at the end of Section III.

We refer to label density as the number of labels that appear in a unit region. Thus, we build a grid (or discrete) representation of the label density, named label density map. Formally, we denote the label density map as a set of label densities:

$$d_i = M(i), \tag{3}$$

where $i \in \mathcal{N}$ denotes the index for label density d_i. If target label y_t is available, the label density for index i is

$$d_i = 1/D \sum_{k=1}^{K} \mathbb{1}\left(\frac{y_t^{(k)} - y_0}{g} \in [i, i+1) \right), \tag{4}$$

where $\mathbb{1}(\cdot)$ denotes indicator function, y_0 is the smallest label value considered, g is the grid size of the label density map, K is the number of the confident data, and $1/D$ is a normalization term.

Unfortunately, the label of the target data is unavailable. Thus, we propose a label distribution estimator that uses confident data to estimate label density map. Specifically, the estimator is based on the correlation between prediction error and uncertainty since a higher uncertainty indicates a potential

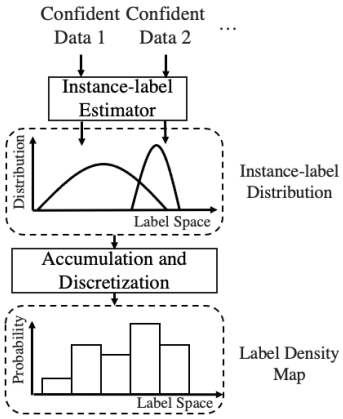

Fig. 4. Illustration on label distribution estimator. It first estimates the label distribution of each piece of confident data and then accumulates the estimated instance-label distributions into a label density map.

larger error. To make it more straightforward, we further show in Figure 3 an example (from location sensing) to illustrate the correlation.

This intuition leads to the technical design of the label distribution estimator, which is illustrated in Figure 4. The estimator first estimates the label distribution of each piece of confident data. Then, it accumulates and discretizes the estimated distributions as the label density map of the target scenario.

Estimating label distribution based on model prediction is to estimate prediction uncertainty, which is usually presented as the variance of prediction in uncertainty literature [22], [54], [53]. Thus, we can leverage the prediction variance to estimate label distribution. In this paper, we employ Gaussian distribution, where we also experimentally compare other error models in Figure 8. Specifically, we model the label distribution of a prediction $\tilde{y}_t^{(k)}$ as

$$y_t^{(k)} \sim \mathcal{N}\left(y|\tilde{y}_t^{(k)}, \sigma_t^{(k)}\right), \tag{5}$$

where we call it instance-label distribution.

Prediction uncertainty relates to model structure and training process [22]. Thus, existing uncertainty estimation approaches usually requires calibration to reflect prediction errors. Since the estimation of the instance-label distribution is based on that larger uncertainty indicates a larger error, we model their relationship by a polynomial function

$$\sigma_t^{(k)} = Q_s\left(u_t^{(k)}\right), \tag{6}$$

where Q_s can be modeled based on the source dataset before delivering to the target scenario.

We regard the modeling of Q_s as a curve-fitting problem. In particular, the standard deviation of the error σ_t entails that around 68% data whose errors should be less than σ_t. Thus, we learn Q_s so that, for each value of uncertainty u_t, around 68% predictions in the source datasets have errors lower than $Q_s(u_t)$. Nevertheless, σ_t is hard to directly determine since u_t is a continuous variable. To tackle this, we divide source data into q segments according to their prediction uncertainty (similar to Figure 3) and fit a parameterized curve to those segments

$$\min \sum_{q' \in q} \left\| Q_s\left(u_s^{(q')}\right) - e_\sigma^{(q')} \right\|_n, \tag{7}$$

where $u_s^{(q')}$ is the mean uncertainty of the segment q', and $e_\sigma^{(q')}$ is the estimated standard deviation of errors in the segment. For simplicity, we use a first-order linear regression model

$$Q_s(u_t) = a_0 + a_1 u_t, \tag{8}$$

where a_0 and a_1 are optimized by least square method [55]

$$\begin{cases} a_1 = \frac{\sum_{q' \in q} u_s^{(q')} e_\delta^{(q')} - |q| \bar{u}_s \bar{e}_\delta}{\sum_{q' \in q} \left(u_s^{(q')}\right)^2 - |q| \bar{u}_s^2}, \\ a_0 = \bar{e}_\delta - a_1 \bar{u}_s. \end{cases} \tag{9}$$

Overall, we name it instance-label estimator, which uses each piece of confident data to estimate the instance-label distribution.

With the function Q_s, we are able to model the label distribution of each piece of confident data by Equation 5. This enables us to assign the label to the density map by probability, i.e. accumulation and discretization. In particular, the probability of the k^{th} label in $M(i)$ is

$$d_i^{(k)} = \int_i^{i+1} S_k\left(y_0 + gI\right) dI, \tag{10}$$

where the Gaussian probability density

$$S_k(y) = \frac{1}{\sigma_t^{(k)}\sqrt{2\pi}} \exp\left(-\frac{\left(y - \tilde{y}_t^{(k)}\right)^2}{2\left(\sigma_t^{(k)}\right)^2}\right), \tag{11}$$

and the standard deviation $\sigma_t^{(n)} = Q_s\left(u_t^{(k)}\right)$. Totally, the label density map can be estimated by

$$d_i = 1/D \sum_{k=1}^{K} d_i^{(k)}, \tag{12}$$

based on Equation 4 with probability calculated from Equation 10. Finally, we present the pseudo-code of label distribution estimator in Algorithm 2.

D. Pseudo-label Generator

In this section, we first introduce how to create the pseudo-label using the label density map M. Then, we discuss evaluating the credibility of each pseudo-label. Both parts contribute to the loss function that supervises the adaptation training for the source model.

How to generate pseudo-labels using label density map? The basic idea of the pseudo-label generator is illustrated in Figure 5. We stick to the grid representation of label density map and denote the grid range as

$$Y_i = y_0 + g[i, i+1). \tag{13}$$

Algorithm 2 Pseudo code of label distribution estimator

Input: Confident data set SET_C
Parameter: Grid size g, label value range $y \in [y_0, y_m]$, function Q_s from Equation 6
Output: Label density map M

Calculate the number of grids $J = \left\lfloor \frac{y_m - y_0}{g} \right\rfloor$
Initialize $M(j)$ for $j = 1, 2..J$
for $(f_{\theta_s}(x_t), u_t) \in SET_C$ **do**
    Calculate $\sigma_t = Q_s(u_t)$
    for $j = 1, 2..J$ **do**
        $S(y) = \frac{1}{\sigma_t\sqrt{2\pi}} \exp\left(-\frac{(y - f_{\theta_s}(x_t))^2}{2(\sigma_t)^2}\right)$
        $M(j) = M(j) + \int_j^{j+1} S\left(y_0 + gI\right) dI$
    end for
end for
for $j = 1, 2, ...J$ **do** ▷ Normalization
    $M(j) = M(j) / |SET_C|$
end for
return M

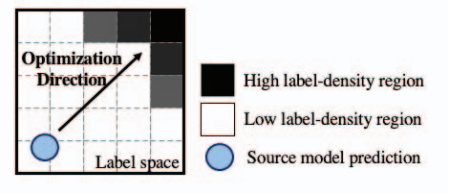

Fig. 5. Illustration of the basic idea of pseudo-label generator.

As mentioned, we regard the label distribution of confident data as the prior knowledge of the label of uncertain data. For an uncertainty data $x_t^{(j)}$, we estimate the posterior label distribution as the joint distribution of prior knowledge and its instance-label distribution

$$Pr\left(y_t^{(j)} \in Y_i\right) = Pr\left(y \in Y_i | \tilde{y}_t^{(j)}, u_t^{(j)}\right) \times Pr(Y_i), \quad (14)$$

where $\tilde{y}_t^{(j)} = f_{\theta_s}\left(x_t^{(j)}\right)$. On the right-hand side, we model the first probability as a Gaussian distribution similar to Equation 5 and the second probability using label density map $Pr(Y_i) = M(i)$.

Based on the posterior label distribution, we generate the pseudo-label that should be close to the grids with high probability. Instead of selecting the grid with the highest probability, we calculate the pseudo-label by interpolating grids according to their probability:

$$\hat{y}_t^{(j)} = 1/Y \sum_i Pr\left(y_t^{(j)} \in Y_i\right) \bar{Y}_i, \quad (15)$$

where $\bar{Y}_i$ denotes the center of the grid, and $1/Y$ is the normalization term. Through such an interpolation, the generated pseudo-label is naturally close to the dense grids when the label density map shows a clear local trend. Otherwise, it will be close to the source model prediction. This avoids causing accuracy degradation when the prior knowledge is not informative.

To understand the relationship between the estimated label distribution of confident data $Pr(y_{con})$ and uncertain data $Pr(y_{unc})$, we present $Pr(y_{unc})$ by the joint distribution of N independent samplings

$$Pr(y_{unc}) = \prod_{i=1}^{N} Pr\left(y_{unc}^{(i)}\right). \quad (16)$$

By considering Equation 12, their relationship can be represented as

$$\log Pr(y_{unc}) = N \log Pr(y_{con}) + \sum_{i=1}^{N} \log Pr(y_{unc}^{(i)}). \quad (17)$$

Intuitively, $Pr(y_{con})$ serves as the prior knowledge for estimating $Pr(y_{unc})$, while they do not have to be the same.

Should we equally trust all pseudo-labels? We regard the label density map as a preference (or prior knowledge of the scenario) when the source model is uncertain about its predictions. Therefore, we should place greater trust in the pseudo-label when the source model's prediction lacks confidence, and conversely. In order to evaluate the credibility of the pseudo-labels, we normalize the confidence of the source model predictions. In particular, we use the confidence threshold τ as a reference to normalize the confidence of source model prediction

$$I_d = \frac{\tau}{u_t^{(j)}}. \quad (18)$$

Also, we assign a higher credibility to pseudo-labels when a clear trend is formed in the label density, On one hand, a clear trend means that the local label densities of a prediction (in label space) should not be evenly distributed. This can be achieved by the interpolation method in Equation 15 because the uniform distribution of the local density will render the pseudo-label close to the prediction. On the other hand, the location where the pseudo-labels are calculated should have a high label density. We denote such a feature by local mean density $\bar{d}_l$ regarding the global mean density $\bar{d}_i$

$$I_l = \frac{\bar{d}_l}{\bar{d}_i}. \quad (19)$$

Here, we regard the locality as the grids whose centers are within three standard deviations from the prediction, i.e.

$$\left\| \bar{Y}_i - \tilde{y}_t^{(j)} \right\|_n < 3\sigma_t^{(j)}, \quad (20)$$

and the credibility of the pseudo-label is

$$\beta_t = \frac{I_l}{I_d}. \quad (21)$$

We use it as the weight of the pseudo-label in the adaptation training.

Overall, the loss function for the adaptation training is

$$\mathcal{L}_{ada} = \sum_{\substack{x_t \in D_t, \\ u_t > \tau}} \beta_t \mathcal{L}\left(f_{\theta_s}(x_t), \hat{y}_t\right), \quad (22)$$

Algorithm 3 Pseudo code of pseudo-label generator

Input: Label density map M, uncertain data set SET_U, grid size g, minimum label value y_0, grid number J
Parameter: Uncertainty threshold τ, function Q_s
Output: Pseudo-label set SET_P

Initialize SET_P
$\bar{d}_i = \sum_{j=1}^{J} M(j)/J$ ▷ Calculate global mean density
for $(f_{\theta_s}(x_t), u_t) \in SET_C$ **do**
  for j=1,2,...,J **do**
    $VAR_W = 0$, $VAR_Y = 0$, $\beta_t = 0$
    Initialize SET_M
    $\sigma_t = Q_s(u_t)$
    $y_m = y_0 + (j + 0.5)g$ ▷ Calculate grid center
    if $|y_m - f_{\theta_s}(x_t)| < 3\sigma_t$ **then**
      $S(y) = \frac{1}{\sigma_t\sqrt{2\pi}} \exp\left(-\frac{(y - f_{\theta_s}(x_t))^2}{2(\sigma_t)^2}\right)$
      $VAR_W += M(j) \times \int_j^{j+1} S\,(y_0 + gI)\,dI$
      $VAR_Y = VAR_Y + y_m \times VAR_W$
      Save $M(j)$ to SET_M
    end if
  end for
  $\hat{y}_t = VAR_Y / VAR_W$ ▷ Calculate pseudo-label
  $\beta_t = \frac{\bar{d}_i \times u_t}{\tau} \times \frac{\sum_{m \in SET_M} m}{|SET_M|}$ ▷ Calculate credibility
  Save $(\hat{y}_t, \beta_t)$ to SET_P
end for
return SET_P

where the pseudo-label is calculated from Equation 15, the loss weight β_t is from Equation 21, and $\mathcal{L}$ is task-dependent. Besides training on the uncertain data, we suggest as well involving the confident data in the adaptation training using the pseudo-label $\hat{y}_t = \tilde{y}_t$. Because the confident data also belongs to the target data, involving them in the training data facilitates the model adapting to the target scenario and avoids the catastrophic forgetting issue [11] where deep models may forget previous knowledge when learning new ones. The pseudo-code of the pseudo-label generator is presented in Algorithm 3.

Finally, we discuss extending the approach to tasks with *multi-dimensional labels*. It mainly distinguishes from the case of single-dimensional labels by requiring a label density map with a multi-dimensional index $i \in \mathcal{N}^m$ where m is the label dimension. This leads to a multivariate Gaussian distribution in Equation 5, which requires estimating the covariance matrix for Equation 7. For simplicity, we suggest treating label dimensions as independent if they are not coupled by the loss function during the training process.

IV. Illustrative Experimental Results

In this section, we demonstrate illustrative experimental results to verify TASFAR, where we introduce the experimental setting in Section IV-A and present illustrative results in Section IV-B.

A. Experimental Setting

To verify TASFAR, we first experiment it with two regression tasks – pedestrian dead reckoning [23] and image-based people counting [24] – because their target domains are usually heterogeneously different, and the applications often require a source-free adaptation due to storage and privacy concerns. Then, we additionally verify TASFAR on two prediction tasks – California housing price prediction [25] and New York City taxi trip duration prediction [26]. The two tasks further validate TASFAR's generality to different tasks.

Pedestrian dead reckoning (PDR) [23] is a task of location sensing. It aims to estimate the user's walking trajectory using the phone-mounted IMU sensors, specifically, the accelerometer and gyroscope. We employ RoNIN [23] as a baseline model to adapt, which is a state-of-the-art PDR model based on temporal-convolutional neural network (TCN). The model focuses on 2D trajectories, where we model the two dimensions independently. In the experiment, we adapt the baseline model to 25 users separately, wherein 15 users have contributed to the source datasets but perform differently in the tests (small domain gap), and the other 10 users are completely unseen by the baseline model (large domain gap).

Different users have different walking behaviors with random carriage states of the phone, causing heterogeneous domain gaps. Each user may contribute one or multiple trajectories – user in the seen group provide, on average, 250m trajectories, and that of the unseen group is 500m. To verify that TASFAR is applicable to not only the target data that have been adapted but all data from the target scenario, for each user, we use 80% trajectories for adaptation and the rest for testing. Note that the labels are unavailable in both adaptation and testing.

The evaluation for PDR focuses on how well the model recovers the trajectory. In this paper, we evaluate the PDR model by two metrics:

- *Step error (STE)*: The model outputs a displacement vector using IMU signals every two seconds (one step). We measure the Euclidean distance between model output and ground truth in each step and average them over a trajectory by
$$STE = 1/J \sum_{j \in J} \|y_j - \tilde{y}_j\|_2 , \tag{23}$$
where the trajectory has J steps;
- *Relative trajectory error (RTE)* [23]: RTE measures the localization error in terms of trajectory
$$RTE = \left\| \sum_{j \in J} y_j - \sum_{j \in J} \tilde{y}_j \right\|_2 \tag{24}$$
with an aligned starting point between the estimated trajectory and the ground-truth trajectory.

We also experiment TASFAR on *image-based people counting* [56] which counts the number of people from single images. We use MCNN [24] as our baseline model, which

is a classic and well-recognized people-counting approach based on convolutional neural network. In our experiment, the baseline model is trained on Part-A (482 images) of the Shanghaitech dataset [24] and are adapted to Part-B (716 images) of it, where the two parts are differentiated by scenes and people densities. The resolution of each image is 768×1024. Similar to the PDR experiment, we use 80% data for adaptation and the rest for testing. We follow the original paper to evaluate the experimental results by mean squared error (MSE) and mean absolute error (MAE).

To verify the generality of TASFAR to different tasks, we additionally apply it to predict California housing price [25] and New York City taxi trip duration [26]. Generally, the two tasks are using the provided features (such as house age and pickup date) to predict house prices in California and taxi trip duration in New York. To form domain gaps, we separate the two datasets spatially since both the house price and taxi trip duration are related to (house or take-off) location. Specifically, we separate California as coastal (target) and non-coastal (source) areas according to [57] and New York as Manhattan (target) and non-Manhattan (source) areas. We employ a MLP-based model [58] as baseline and evaluate the two tasks by mean squared error (MSE) and rooted mean squared logarithmic error (RMSLE), as provided by their datasets.

We compare TASFAR with the following state-of-the-art schemes:

- *MMD-based UDA (MMD):* Work in [36] proposes a traditional source-based UDA approach using source data. It measures the domain gap by MMD and aligns them in the feature space;
- *ADV-based UDA (ADV):* Work in [38] proposes a traditional source-based UDA approach using source data. It leverages a pre-trained adversarial neural network to bridge the domain gap in feature space;
- *UDA without source data (Datafree):* Work in [9] conducts UDA without using source data. Instead, it stores source feature distribution via a soft histogram and regards the feature distribution as a domain gap.
- *Augmentation-based source-free UDA (AUGfree):* Work in [14] is a source-free UDA approach based on data alignment. It requires a known domain gap and simulates the gap by data augmentation, where the domain-invariant features are extracted. In the experiment, we follow the original paper and employ the variance perturbation as the augmentation method.

In the experiment, we use the Dropout mechanism to calculate model uncertainty. Uncertainty is presented by the standard deviation of predictions from twenty samplings with a dropout rate of 0.2. To reduce randomness, we repeat each experiment five times and report the average result. Unless particularly specified, we show the results on the adaptation set.

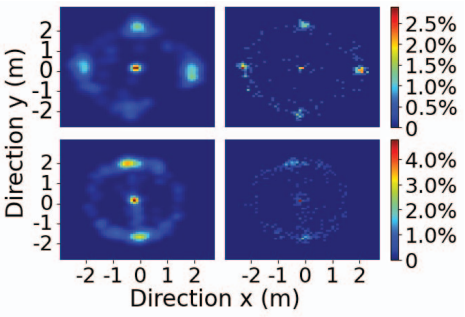

Fig. 6. Visualization of the estimated (left) and true (right) label density map based on two PDR users.

B. Illustrative Results

In this part, we experiment on the system parameters using PDR in Section IV-B1 and show performance of TASFAR with the comparison schemes in Section IV-B2. Then, we extend the experiments and analysis to people counting in Section IV-B3. Evaluations of the two prediction tasks are in Section IV-B4. A failure case is analyzed in Section IV-B5.

1) Study on System Parameters: We study how TASFAR performance varies with its system parameters. Unless specified, the experiments are on the seen group of PDR using the identical grid size on the two label dimensions of PDR label.

We first visualize the estimated label density maps and compare with their ground truth, using two sample users from PDR. In Figure 6, from the ground truths of the label density maps, both label density maps display ring-shaped patterns in the high-density grids, which indicates the users' regular walking speeds. Also, the clusters on rings indicate the users' walking patterns. From the figure, the estimated label density maps accurately capture the ring-shaped pattern and clustering information of the high-density grids. The larger ring of the upper figure shows that the walking speed of the user tends to be larger than the other user. And, the clustered regions of the high-density grids indicate that the upper user is more likely to make sharp turns than the other one. This confirms the effectiveness of the label distribution estimator and justifies the use of the estimated label density map to calibrate source models.

To verify the label distribution estimator (Equation 12), we present the mean absolute error (MAE) of the estimated label density map in Figure 7. As shown in the figure, the MAE converges to MAE=2/0 with extremely small/large grid sizes. This is because larger grids ease the estimation task, and vice versa. For instance, an extremely large grid would involve all target data in both the estimated and ground truth maps, leading to the same label density. Nevertheless, we will provide an explanation as to why using a large grid is not recommended in the subsequent analyses.

In Figure 8, we show how pseudo-label accuracy varies with grid size based on different distribution forms of error models. First, there is no significant difference among different error models, which verifies that TASFAR is compatible with

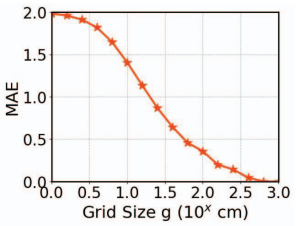

Fig. 7. Error of label distribution estimator varies with grid size: a larger grid size leads to a lower estimation error.

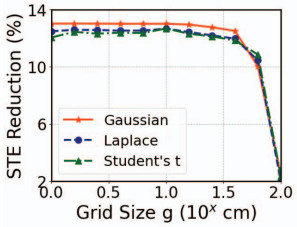

Fig. 8. Pseudo-label error varies with grid size: a large grid size is not preferred.

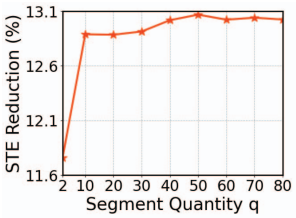

Fig. 9. Pseudo-label error varies with segment quantity q: a too small q is not preferred.

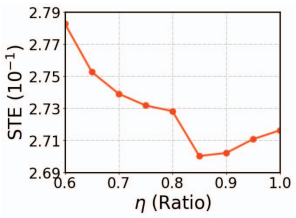

Fig. 10. Pseudo-label error varies with the ratio η.

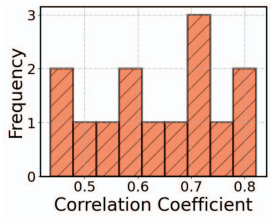

Fig. 11. Distribution of the correlation coefficient between credibility β_t and prediction error over different users.

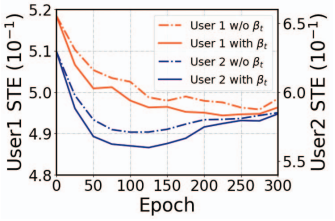

Fig. 12. Ablation study on the credibility β_t.

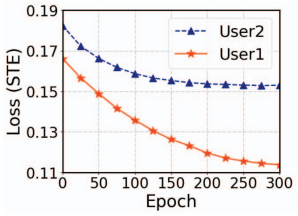

Fig. 13. Learning curves of adaptation training: early stop when the rate of error reduction slows down.

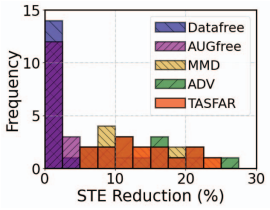

Fig. 14. Comparison on STE reduction rate.

different distribution forms, as long as it shows larger errors for high prediction uncertainty. In terms of pseudo-label accuracy, the figure suggests a small grid size, while it may lead to low accuracy as in Figure 7. This is because the grid interpolation of TASFAR (in Equation 15) makes it robust to the estimation error, while the performance will only degrade with an extremely large grid. Overall, the system performance is not sensitive to the choice of the grid size. Even though, an extremely small grid size is not preferred. Specifically, the computation complexity of constructing a label density map is $O(n/g)$ based on n pieces of confident data with grid size g, indicating that it consumes more computing resources to construct a label density map with smaller grid size. Since the accuracy flattens off when the grid size reduces, there is no need for a small grid size.

In figure 9, we set the grid size to be $10cm$ and investigate how pseudo-label accuracy varies with segment quantity q in Equation 7. The figure shows that the pseudo-label accuracy quickly converges with a small q. Therefore, only a few segments can capture the relationship between model uncertainty and prediction error. The convergence also shows that TASFAR works with a wide range of q. We empirically set $q = 40$ for the following experiments.

In Figure 10, we study how to select the confidence ratio η for the confidence classifier as discussed in Section III-B. The figure shows how pseudo-label error varies with η, where the pseudo-label error decreases when η is less than 0.9. As explained in Section III-B, a small η leads to a small confidence threshold τ so that the accurate predictions may be considered as the uncertain ones. Also, a too large η may decrease the numbers of uncertain data such that no data are available for adaptation. Even though, the figure shows a wide range of η to use. In this paper, we set η to be 0.9.

We validate the pseudo-label credibility β_t (from Equation 22) in Figure 11. For trajectory data (with multiple steps) of each person, we calculate the Pearson correlation coefficient of β_t and the pseudo-label accuracy and summarize them as a probability distribution function (PDF) over different users. As shown in the figure, the coefficients of all users exhibit a positive correlation, where most users' correlations are larger than 0.5. Therefore, TASFAR will assign large weights to the accurate pseudo-labels (in Equation 22), which avoid generating low-quality pseudo-labels that cause accuracy degradation.

We further conduct an ablation study of β_t in Figure 12. With or without using the weight β_t, the figure compares the STE varies with epochs in the adaptation training. Both curves show lower STEs with β_t, while the gaps are reduced with more training epochs. This is because the pseudo-labels with larger β_t tend to be more accurate than those with small weights. The model would stress more on the pseudo-labels with large β_t in the beginning because of the large weights. This explains the gap shown in the two curves. As the number of epochs increases, the gaps are reduced because the training losses of these pseudo-labels (with large β_t) are reduced when the model starts to focus on the ones with less accurate pseudo-labels. Thus, we should employ an early stop to improve adaptation performance.

As the adaptation training process is automatic and unsupervised, we study the early stop issue in Figure 13 We show

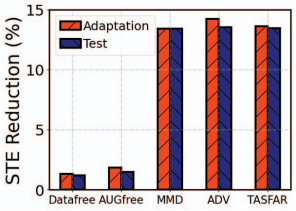

Fig. 15. Comparison on STE reduction between adaptation and test sets.

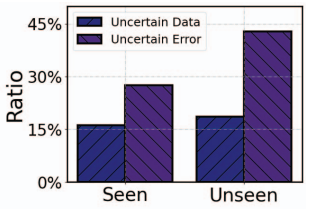

Fig. 16. The ratio of uncertain data and errors in the seen and unseen group.

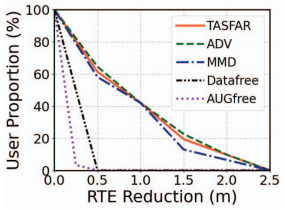

Fig. 17. How many users' RTE from the seen group are reduced?

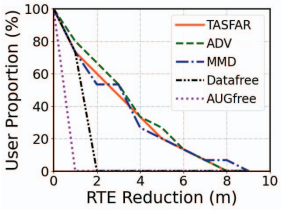

Fig. 18. How many users' RTE from the unseen group are reduced?

the learning curve of the same users as in Figure 12. In the figure, both curves show the regular patterns of deep model training: the speed of the training loss drops gradually reduces as the epoch increases. The significant training loss drops, at small epochs, shows that the adaptation training is bridging the gaps between source model predictions and pseudo-labels with large weights β_t (from Equation 22). Therefore, the change in the loss-dropping speed indicates a changing focus from the large β_t to the smaller one. So, we can early stop the adaptation training when the loss-dropping speed is significantly reduced, i.e. epoch 250 of user 1 and epoch 100 of user 2. This also agrees with the satisfactory stopping epochs from Figure 12.

2) Performance Analysis in PDR: In this part, we analyze the experimental results on PDR and compare TASFAR with the comparison schemes. Unless particularly specified, we demonstrate results on adaptation data.

We first evaluate the STE reduction in the seen group and show the reduction distribution over the individual user in Figure 14. The figure shows TASFAR achieves similar error reduction compared with source-based UDA approaches, i.e., MMD and ADV, while the improvement from other source-free approaches is insignificant. Datafree can only achieve small improvements because it merely aligns domains in terms of feature statistics. The adaptation performance of AUGfree varies across different users because its augmentation only fits a few users. In comparison, the STE of each person is significantly reduced by applying TASFAR because it directly calibrates the source model using label distribution of the target scenarios. Considering that different users have different signal distributions, this experiment has verified that TASFAR is practical and general to heterogeneous target domains.

We verify the performance consistency in adaptation and test sets in Figure 15. TASFAR achieves an averaged STE reduction of 13.6% in the adaptation set and 13.4% in the test set, and all schemes show similar error reductions between the two sets. Firstly, the schemes are not accessible to labels of both adaptation and test sets. Secondly data from both sets are generated from the same domain, where the test data distribution is similar to that of the adaptation. This explains the consistent performance of TASFAR in both adaptation and test sets and validates that the adaptation can be achieved by using a group of data from target domains.

Figure 16 shows the ratio of uncertain data and their prediction errors regarding the whole dataset. Due to the domain gaps between the target and source, the uncertain data ratios of both seen and unseen groups are larger than $\eta = 0.9$. The ratio of the unseen group 18.6% is larger than that of the seen group 16.2% due to its larger domain gap. From the figure, the error ratios are much larger than the data ratios in both groups because the errors are mainly incurred by uncertain data. Therefore, TASFAR only pseudo-labels the uncertain data, though, it can achieve commendable adaptation performance because the uncertain data group is the main source of the inaccuracy.

Besides using STE to show model accuracy on the uncertain data, we show the RTE of both confident and uncertain data in the test set. Specifically, we show how many localization errors are reduced over the test trajectories in Figure 17. The figure shows the numbers (in ratio) of trajectories whose error reduction is more than a threshold (x-axis), where TASFAR achieves $0.92m$ (or 7%) average error reduction for trajectories with an average length of $50m$. This is comparable with the source-based UDA, outperforming other source/data-free approaches. Also, the result conforms to the conclusion drawn from the STE experiments. Note that the localization error of PDR is temporally dependent, where the location of the next step depends on the last one. So, the errors sometimes cancel each other's bias over the trajectories. Therefore, it is possible that Datafree can outperform AUGfree in RTE while performing worse in STE.

We study the performance of TASFAR to both small (seen) and large (unseen) domain gaps in Figure 18. As users from the unseen group are not exposed to the source model in model training, the domain gaps of the unseen group are larger than those of the seen group. As the errors of PDR are cumulative, the error reduction is more significant in longer trajectories. From the figure, TASFAR still shows comparable RTE reduction with the source-based UDA approaches. It reduces around $3.13m$ of RTE for trajectories with an average length of $100m$. This validates that TASFAR is capable of handling both small and large domain gaps in terms of the input distribution because it explores label space that is decoupled from the input space.

3) Performance Analysis in People Counting: To show that TASFAR can work with multiple signal forms, we conduct experiments of image-based people counting. To capture the properties of target scenarios, we apply TASFAR to images

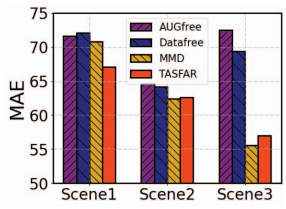

Fig. 19. Comparison of the different scenes of people counting on the test set.

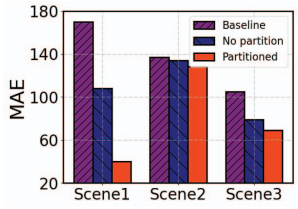

Fig. 20. TASFAR's performance with or without partitioning the test data.

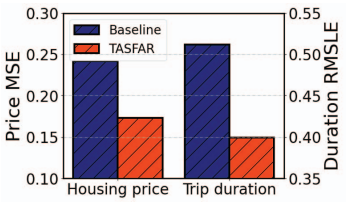

Fig. 21. TASFAR's performance on the two prediction tasks.

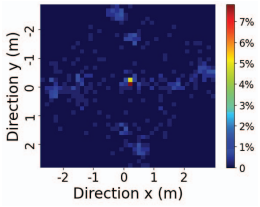

Fig. 22. Label distribution with two users in PDR.

TABLE I
COMPARISON ON CROWD COUNTING. TASFAR PERFORMS COMPARABLY WITH THE TRADITIONAL SOURCE-BASED UDA APPROACHES ON THE ADAPTATION SET, UNCERTAIN DATA FROM THE ADAPTATION SET, AND TEST SET.

	Adaptation (whole)		**Error Reduction (%)**		Adaptation (uncertain)		**Error Reduction (%)**		Test		**Error Reduction (%)**	
Scheme	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Baseline	56.4	86.8	-	-	114.4	143.8	-	-	71.7	141.5	-	-
MMD*	51.5	82.2	8.7	5.3	80.1	96.7	29.9	32.7	59.6	110.9	16.8	21.5
ADV*	52.1	81.9	7.6	5.6	76.4	97.2	33.2	32.4	59.7	111.0	16.7	21.6
AUGfree	56.3	86.9	0.2	0	113.6	142.5	0.7	0.9	71.5	141.0	0.2	0.3
Datafree	56.2	86.1	0.4	0.8	108.1	134.5	5.5	6.5	69.5	134.1	3.0	5.2
TASFAR	52.4	80.3	**7.0**	**7.5**	74.0	97.5	**35.3**	**32.2**	59.9	107.4	**16.5**	**24.1**

* Source-based UDA approach

Fig. 23. Sample images of sites from the people-counting dataset [24].

belonging to the same sites (streets) separately from the test dataset. The sample images from the three sites are shown in Figure 23, where scene 3 tends to be more crowded than the others from our observation.

In Table I, we compare the experimental results on the adaptation and test set. The source model (baseline) performs worse on the uncertain set than on the whole adaptation set because of the high prediction uncertainty. Although all schemes reduce more errors on the uncertain set than the whole adaptation set due to the large error, TASFAR significantly outperforms the other source-free approaches in both MSE and MAE and achieves comparable results with the source-based UDA approaches. After adaptation, the experimental results of the test set also come to a consistent conclusion: Datafree only reduces errors slightly and AUGfree does not perform well on people counting because its augmentation approach misfits the task, while TASFAR achieves 16.5% and 24.1% error reduction in terms of MAE and MSE that are comparable with the source-based UDA approaches.

We further compare TASFAR with the other approaches on different scenes in Figure 19. We only show MMD because it performs similarly to ADV. Similarly to the results in Table I, TASFAR achieves comparable performance with the source-based UDA on all three scenes, outperforming AUGfree and Datafree. Interestingly, TASFAR outperforms the existing source-free approaches in scene 2 and 3, and surpass them by a large margin in scene 1. This is because the crowded scene 3 maintains a stable pedestrian stream, forming a prominent feature in label distribution. In all, the accuracy improvement in all three scenes has verified that TASFAR can work with different crowd scenes.

In Figure 20, we discuss TASFAR's performance without partitioning the test dataset by scene. From the figure, TASFAR shows better performance in all three scenes when their adaptation sets are partitioned. This is because data from the same scenes are correlated by the target scenarios, providing prominent features in the label distribution that are leveraged by TASFAR. On the contrary, fusing data with multiple scenes may corrupt the features of each target, degrading the adaptation performance of TASFAR. Even though, TASFAR can still achieve good performance without partitioning because the crowd density of the Part B dataset is inherently correlated.

4) Performance Analysis in Prediction Tasks: To verify the generality of TASFAR in different tasks, we further show in Figure 21 its performance on two prediction tasks. On the target regions, TASFAR has reduced 22% MSE and 28% RMSLE separately in predicting housing price and trip duration. Since (house and take-off) location is a key factor of housing price and trip duration, the baseline models that are learned from one district cannot perform well in another district. Even though, the housing prices and trip duration in the target district are naturally correlated. TASFAR captures such correlation and improves the accuracy for the target district.

5) Failure Case Analysis: Finally, we show a failure case in the PDR task where the target model, calibrated by TASFAR, is only marginally better than the source model. Specifically,

we manually balance the target data by using two users' data as the target, upon which TASFAR only reduces around 1% STE. The performance is similar to those of other source-free approaches. To analyze it, we visualize target label distribution in Figure 22. As shown, the two users have different step lengths and walking patterns, so the label distribution displays a double-ring shape that differentiates the single-person case in Figure 6. However, the label distribution of one user usually cannot serve as the prior knowledge of the other, resulting in a failure of adaptation. To avoid causing accuracy degradation, the TASFAR would generate pseudo-labels that are close to the source-model predictions (due to the double-ring shape) and assign small weights to adaptation loss since the label densities are spread out over the map. Ideas to tackle such cases are further discussed in Section VI.

V. Conclusion

The traditional source-based unsupervised domain adaptation (UDA) uses both unlabeled target data and the training dataset (on the source domain) to overcome the domain gap between the target and source. To protect source data confidentiality and reduce storage requirements, source-free UDA replaces source data with a source model and adapts it to the target domain. Previous source-free UDA approaches measure and bridge domain gaps in input-data or feature space of the source model, which only works for specific domain gaps or classification tasks.

In this paper, we propose, for the first time, a ***t***arget-***a***gnostic ***s***ource-***f***ree domain ***a***daptation approach termed ***TASFAR*** for ***r***egression tasks. TASFAR is based on the observation that the target label, like target data that all conform to the target domain, also originates from the same target scenario. Therefore, in contrast to previous source-free UDA approaches, TASFAR directly estimates the label distribution of the target scenario and uses it to calibrate source models. Specifically, TASFAR classifies the target data into confident and uncertain data and proposes a label distribution estimator, based on the confident data, to estimate the target label distribution, represented as a label density map. Then, a pseudo-label generator utilizes the label density map to pseudo-label the uncertain data, which is used to fine-tune the source model based on supervised learning.

To validate TASFAR, we have conducted extensive experiments on four regression tasks, namely, pedestrian dead reckoning (using the inertial measurement unit), image-based people counting from single images, and two prediction tasks. We compare TASFAR with state-of-the-art source-free UDA and source-based UDA approaches. The experimental results show that TASFAR significantly outperforms the existing source-free UDA with around 14% and 24% reduction in localization error and mean absolute error (MSE) in the pedestrian dead reckoning of different users and people counting with various crowd scenes, respectively. In the two prediction tasks, TASFAR reduces 26% of the prediction errors. Without the need for any source data, its performance outperforms the previous source-free UDA approaches and is notably comparable with the source-based UDA approaches.

VI. Future Work

To achieve source-free domain adaptation for regression tasks on agnostic target domains, TASFAR explores the label properties that originate from target scenarios themselves, such as environmental features, behavioral patterns of users, cyclic events of the scenes, and so on. This observation makes TASFAR well-suited for adaptation in real-world scenarios, where the label distributions are naturally imbalanced because of the heterogeneous target scenarios. Consequently, its performance gain is not so marked in tasks where the target data comes from multiple sources or where labels are manually balanced, such as those datasets for data competitions. TASFAR may achieve only minimal accuracy improvement on such tasks since their scenario properties may be corrupted or intentionally reduced.

One direction of future works can focus on how to partition test data so as to better utilize the characteristics of the target scenario. This partition may depend on task-specific knowledge. When applying TASFAR to a specific task, we can partition the target data into multiple segments based on task-specific knowledge. In each segment, we independently assign pseudo-labels to the uncertain data. For example, in a surveillance-based people counting, TASFAR may perform better if we treat the morning and evening as two target scenarios. From this perspective, TASFAR may serve as a general framework to incorporate more task-specific knowledge to achieve better adaptation performance on real-world applications.

Finally, we discuss the potential application of TASFAR to classification tasks, despite its specific design for regression models. Technically, TASFAR may be straightforwardly applied to classification tasks. Without leveraging classification properties, however, TASFAR by itself is not expected to show advantages over those approaches in classification tasks. Nevertheless, TASFAR can be seamlessly integrated with other classification-based approaches as a plug-in module. Specifically, TASFAR may be used to explore the correlation among label classes of a classification task and generate soft pseudo-labels for uncertain data. Such kind of information (namely dark knowledge) has been successfully verified in the field of knowledge distillation. We thus believe it may be useful in source-free domain adaptation, which can be an interesting future work to study.

References

[1] H. Wu, Z. Mo, J. Tan, S. He, and S.-H. G. Chan, "Efficient Indoor Localization based on Geomagnetism," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 4, pp. 1–25, 2019.

[2] T. He, J. Tan, W. Zhuo, M. Printz, and S.-H. G. Chan, "Tackling Multipath and Biased Training Data for IMU-assisted BLE Proximity Detection," in *INFOCOM*. IEEE, 2022, pp. 1259–1268.

[3] W. Zhuo, K. H. Chiu, J. Chen, Z. Zhao, S.-H. G. Chan, S. Ha, and C.-H. Lee, "FIS-ONE: Floor Identification System with One Label for Crowdsourced RF Signals," in *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2023, pp. 418–428.

[4] H. Bai, H. He, Z. Peng, T. Dai, and S.-H. G. Chan, "Countr: An end-to-end transformer approach for crowd counting and density estimation," in *European Conference on Computer Vision*. Springer, 2022, pp. 207–222.

[5] V. A. Sindagi and V. M. Patel, "A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation," *Pattern Recognition Letters*, vol. 107, pp. 3–16, 2018.

[6] E. Ramanujam, T. Perumal, and S. Padmavathi, "Human Activity Recognition with Smartphone and Wearable Sensors Using Deep Learning Techniques: A Review," *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13 029–13 040, 2021.

[7] W. Lu, J. Wang, X. Sun, Y. Chen, and X. Xie, "Out-of-distribution Representation Learning for Time Series Classification," in *The Eleventh International Conference on Learning Representations*, 2022.

[8] V. K. Kurmi, V. K. Subramanian, and V. P. Namboodiri, "Domain Impression: A Source Data Free Domain Adaptation Method," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 615–625.

[9] C. Eastwood, I. Mason, C. K. Williams, and B. Schölkopf, "Source-free Adaptation to Measurement Shift via Bottom-up Feature Restoration," *ICLR*, 2022.

[10] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing Change: Continual Learning in Deep neural Networks," *Trends in Cognitive Sciences*, vol. 24, no. 12, pp. 1028–1040, 2020.

[11] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A Continual Learning Survey: Defying Forgetting in Classification Tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[12] Y. Li, Y. Shen, and L. Chen, "Camel: Managing Data for Efficient Stream Learning," in *Proceedings of the 2022 International Conference on Management of Data*, 2022, pp. 1271–1285.

[13] H. Yu, J. Huang, Y. Liu, Q. Zhu, M. Zhou, and F. Zhao, "Source-Free Domain Adaptation for Real-World Image Dehazing," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 6645–6654.

[14] L. Xiong, M. Ye, D. Zhang, Y. Gan, X. Li, and Y. Zhu, "Source Data-free Domain Adaptation of Object Detector through Domain-specific Perturbation," *International Journal of Intelligent Systems*, vol. 36, no. 8, pp. 3746–3766, 2021.

[15] N. Karim, N. C. Mithun, A. Rajvanshi, H.-p. Chiu, S. Samarasekera, and N. Rahnavard, "C-SFDA: A Curriculum Learning Aided Self-training Framework for Efficient Source Free Domain Adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 120–24 131.

[16] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, "Generalized Source-free Domain Adaptation," in *ICCV*, 2021, pp. 8978–8987.

[17] J. N. Kundu, N. Venkat, R. V. Babu *et al.*, "Universal Source-free Domain Adaptation," in *CVPR*, 2020, pp. 4544–4553.

[18] S. Yang, J. van de Weijer, L. Herranz, S. Jui *et al.*, "Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation," *NIPS*, vol. 34, pp. 29 393–29 405, 2021.

[19] Z. Zhang, W. Chen, H. Cheng, Z. Li, S. Li, L. Lin, and G. Li, "Divide and Contrast: Source-free Domain Adaptation via Adaptive Contrastive Learning," *NIPS*, 2022.

[20] Y. Wang, A. K.-s. Wong, S.-H. G. Chan, and W. H. Mow, "Leto: Crowdsourced Radio Map Construction With Learned Topology and a Few Landmarks," *IEEE Transactions on Mobile Computing*, 2023.

[21] Y. Gal *et al.*, "Uncertainty in Deep Learning," *PhD Thesis*, 2016.

[22] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, "A Survey of Uncertainty in Deep Neural Networks," *Artificial Intelligence Review*, 2021.

[23] H. Yan, S. Herath, and Y. Furukawa, "RONIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, and New Methods," *ICRA*, 2020.

[24] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image Crowd Counting via Multi-column Convolutional Neural Network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597.

[25] C. Nugent, "The California Housing Dataset," 2022. [Online]. Available: https://www.kaggle.com/datasets/camnugent/california-housing-prices/data

[26] M. Risdal, "New York City Taxi Trip Duration," 2017. [Online]. Available: https://kaggle.com/competitions/nyc-taxi-trip-duration

[27] S. Zhao, X. Yue, S. Zhang, B. Li, H. Zhao, B. Wu, R. Krishna, J. E. Gonzalez, A. L. Sangiovanni-Vincentelli, S. A. Seshia *et al.*, "A Review of Single-source Deep Unsupervised Visual Domain Adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[28] G. Wilson and D. J. Cook, "A Survey of Unsupervised Deep Domain Adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, jul 2020. [Online]. Available: https://doi.org/10.1145/3400066

[29] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A Brief Review of Domain Adaptation," *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

[30] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola, "Correcting Sample Selection Bias by Unlabeled Data," *NIPS*, vol. 19, 2006.

[31] S. J. Phillips, M. Dudík, J. Elith, C. H. Graham, A. Lehmann, J. Leathwick, and S. Ferrier, "Sample Selection Bias and Presence-only Distribution Models: Implications for Background and Pseudo-absence Data," *Ecological applications*, vol. 19, no. 1, pp. 181–197, 2009.

[32] S. Benaim and L. Wolf, "One-sided Unsupervised Domain Mapping," *NIPS*, vol. 30, 2017.

[33] H. Fu, M. Gong, C. Wang, K. Batmanghelich, K. Zhang, and D. Tao, "Geometry-consistent Generative Adversarial Networks for One-sided Unsupervised Domain Mapping," in *CVPR*, 2019, pp. 2427–2436.

[34] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CYCADA: Cycle-consistent Adversarial Domain Adaptation," in *ICML*. Pmlr, 2018, pp. 1989–1998.

[35] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond Sharing Weights for Deep Domain Adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 801–814, 2018.

[36] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep Transfer Learning with Joint Adaptation Networks," in *ICML*. PMLR, 2017, pp. 2208–2217.

[37] 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.

[38] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial Discriminative Domain Adaptation," in *CVPR*, 2017, pp. 7167–7176.

[39] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial Neural Networks," *NIPS*, 2014.

[40] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep Reconstruction-classification Networks for Unsupervised Domain Adaptation," in *ECCV*. Springer, 2016, pp. 597–613.

[41] J. Chen, T. He, W. Zhuo, L. Ma, S. Ha, and S.-H. G. Chan, "Tvconv: Efficient translation variant convolution for layout-aware visual processing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 548–12 558.

[42] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 021–12 031.

[43] Y. Hou and L. Zheng, "Visualizing Adapted Knowledge in Domain Transfer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 824–13 833.

[44] R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu, "Model Adaptation: Unsupervised Domain Adaptation without Source Data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9641–9650.

[45] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial Attacks and Defences: A Survey," *arXiv preprint arXiv:1810.00069*, 2018.

[46] Z. Qiu, Y. Zhang, H. Lin, S. Niu, Y. Liu, Q. Du, and M. Tan, "Source-free Domain Adaptation via Avatar Prototype Generation and Adaptation," *IJCAI*, 2021.

[47] C. Yang, X. Guo, Z. Chen, and Y. Yuan, "Source Free Domain Adaptation for Medical Image Segmentation with Fourier Style Mining," *Medical Image Analysis*, vol. 79, p. 102457, 2022.

[48] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[49] G. Hinton, O. Vinyals, and J. Dean, "Dark Knowledge," *Presented as the keynote in BayLearn*, vol. 2, no. 2, 2014.

[50] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational Knowledge Distillation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3967–3976.

[51] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient Knowledge Distillation for BERT Model Compression," *EMNLP*, 2019.
[52] A. Polino, R. Pascanu, and D. Alistarh, "Model Compression via Distillation and Quantization," *ICLR*, 2018.
[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[54] L. Oala, C. Heiß, J. Macdonald, M. März, W. Samek, and G. Kutyniok, "Interval Neural Networks: Uncertainty Scores," *arXiv preprint arXiv:2003.11566*, 2020.
[55] Å. Björck, "Least Squares Methods," *Handbook of numerical analysis*, vol. 1, pp. 465–652, 1990.
[56] Q. Wang, J. Gao, W. Lin, and X. Li, "NWPU-crowd: A Large-scale Benchmark for Crowd Counting and Localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 2141–2149, 2020.
[57] M. Woloszyn, J. E. Bell, A. Sheffield, R. Lookadoo, and K. Hansen, "Advancing Drought Science and Preparedness Across the Nation: Focus on Public Health," in *AGU Fall Meeting Abstracts*, vol. 2022, 2022, pp. SY16A–03.
[58] M. Poongodi, M. Malviya, C. Kumar, M. Hamdi, V. Vijayakumar, J. Nebhen, and H. Alyamani, "New York City Taxi Trip Duration Prediction Using MLP and XGBoost," *International Journal of System Assurance Engineering and Management*, pp. 1–12, 2022.