# arXiv:2208.13986v1 [cs.CV] 30 Aug 2022

# Uncertainty-Induced Transferability Representation for Source-Free Unsupervised Domain Adaptation

Jiangbo Pei\*, Zhuqing Jiang\*, Aidong Men, Liang Chen, Yang Liu, Qingchao Chen<sup>∞</sup>

Abstract—Source-free unsupervised domain adaptation (SFUDA) aims to learn a target domain model using unlabeled target data and the knowledge of a well-trained source domain model. Most previous SFUDA works focus on inferring semantics of target data based on the source knowledge. Without measuring the transferability of the source knowledge, these methods insufficiently exploit the source knowledge, and fail to identify the reliability of the inferred target semantics. However, existing transferability measurements require either source data or target labels, which are infeasible in SFUDA. To this end, firstly, we propose a novel Uncertainty-induced Transferability Representation (UTR), which leverages uncertainty as the tool to analyse the channel-wise transferability of the source encoder in the absence of the source data and target labels. The domain-level UTR unravels how transferable the encoder channels are to the target domain and the instance-level UTR characterizes the reliability of the inferred target semantics. Secondly, based on the UTR, we propose a novel Calibrated Adaption Framework (CAF) for SFUDA, including i) the source knowledge calibration module that guides the target model to learn the transferable source knowledge and discard the non-transferable one, and ii) the target semantics calibration module that calibrates the unreliable semantics. With the help of the calibrated source knowledge and the target semantics, the model adapts to the target domain safely and ultimately better. We verified the effectiveness of our method using experimental results and demonstrated that the proposed method achieves state-of-the-art performances on the three SFUDA benchmarks. Code is available at https://github.com/SPIresearch/UTR.

# I. INTRODUCTION

Deep neural networks have achieved state-of-the-art performance in a variety of image processing and computer vision applications when the testing data and training data are drawn from the same distribution (domain). When the model needs to be deployed in a new target domain (e.g. a new user uploads photos to a social media website), the recommendation or retrieval model often suffers from huge

Jiangbo Pei and Aidong Men are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China. Jiangbo Pei is also affiliated with the National Institute of Health Data Science, Peking University. (e-mail: jiangbop@bupt.edu.cn; menad@bupt.edu.cn).

Zhuqing Jiang is with Beijing Key Laboratory of Network System and Network Culture, and also with Beijing University of Posts and Telecommunications, Beijing 100876, China, (e-mail: jiangzhuqing@bupt.edu.cn).

Liang Chen is with School of Mathematical Sciences, Peking University, Beijing 100871, China, (e-mail: clandzyy@pku.edu.cn).

Yang Liu is with Wangxuan Institute of Computer Technology at Peking University, Beijing, 100080, China, (email: yangliu@pku.edu.cn).

Qingchao Chen is with the National Institute of Health Data Science, Peking University, Beijing, 100191, China. (e-mail: qingchao.chen@pku.edu.cn).

This work is supported by Peking University Medicine Seed Fund for Interdisciplinary Research (BMU2022MX011), the Fundamental Research Funds for the Central Universities and PKU-OPPO Innovation Fund BO202103.

\* Equally contributed first author.

Corresponding author.



Fig. 1. (a): Most existing SFUDA methods directly transfer all source knowledge to the target model at the start of training, infer the semantics (labels) of target data using the model, and update the model with the inferred semantics. Without identifying the source knowledge's transferability, the target model receives less-transferable knowledge (for example, the feature "Horse hoof" which is learned to classify humans and horses in the real-world source domain but may not be suitable for the target cartoon domain). The less-transferable knowledge hinders the model to infer the semantics of the target data (e.g. Misclassified Horse Image). (b): The  $UTR_D$  identifies how transferable each channel of the source encoder is to the target domain. (c): The  $UTR_I$  characterizes the reliability of the inferred semantics of each target sample.

performance degradation due to the cross-user domain gap. Unsupervised domain adaptation (UDA) is an effective solution to tackle the domain gap, which aims at adapting a model to a target domain where labels are not available with the help of a labeled source domain dataset. However, the vanilla UDA assumes the source data is accessible during adaptation, which is not always practical. On the one hand, data privacy protection is increasingly important because data often contain personal information. Sharing source domain data will endanger personal privacy and is strictly prohibited in many applications, especially in social media, medicine and biometrics. On the other hand, transmitting source data is costly such as video data or high-resolution images.

Source-free unsupervised domain adaptation (SFUDA) is proposed as a promising task to tackle previous issues. SFUDA aims to learn a discriminative target domain model, given the unlabeled target domain data and a pre-trained source model but without any source data or labels. To address SFUDA, as shown in Fig. 1 (a), most existing works [1]– [5] directly transfer all source knowledge to the target model at the start of training, infer the semantics (labels) of target data using the model and turn back to update the model with the inferred semantics. However, these methods suffer two limitations. *Firstly, the utilization of the source knowledge is limited.* On the one hand, the way that transfers all source knowledge to the target model *ignores discarding the nontransferable knowledge.* On the other hand, they transfer source knowledge at the start of training only, which wastes the valuable transferable knowledge learned from the wellannotated source domain. *Secondly, the semantics of the target data inferred by the source knowledge is risky*, if using the non-transferable knowledge (taking the "human hoof" in Fig. 1 (a) as an example). Updating the model using these risky semantics rarely learns a discriminative target model.

Therefore, the key common challenge and the missing part of existing SFUDA methods is to *measure the transferability of the source knowledge to the target domain in the absence of source data and target labels.* To our best knowledge, only Wang et al. [6] proposed to search for domain-invariant/transferable model parameters. They explore the transferability of source model parameters based on calculating their variations after each adaptation procedure in the stochastic optimization. However, their measurement is susceptible to the quality of the adaptive procedures.

Beyond SFUDA, the transferability of the deep neural network has been studied intensively [7]–[10].

Nevertheless, existing transferability measurements require either source data or target labels, which are not applicable to SFUDA.

To tackle the key challenge, we proposed a novel Uncertainty-induced Transferability Representation (UTR), which provides a transferability measurement to the source knowledge in the absence of source data and target labels. Specifically, we develop the uncertainty as a tool to measure the transferability of the source model, inspired by the theory of distributional uncertainty [11]-[13] that measures how "unfamiliar" a trained model is with any input data, and the model uncertainty [11], [12], [14] that reflects the degree to which the model fits its training distribution. Intuitively, the two uncertainties reveal the probability of the input data that are sampled from the training distribution of the modelthat is, an implicit Uncertainty Distance (UD) between the input data and the training one. This measurement provides us solid theoretical supports and more importantly, we propose to bridge the uncertainty and the source model transferability in the SFUDA setup: if a lower UD of the target data and source model is obtained, we made the following conjectures: the target data is "closer" to the source domain distribution in the encoding space of the model, which indicates the source knowledge (source model parameters) can more effectively eliminate the domain discrepancy between the two domains, reflecting the source knowledge is more transferable to the target domain. On the other hand, stemming from our finding that different channels of the source features have different transferability to the target domain, we propose to measure the transferability of the source encoder (the feature encoder of the source model) channel-wisely. Intuitively, the transferability of different channels reflects the transferability of the "partial" source knowledge that encodes the features in these channels,

facilitating us to explore which "partial" source knowledge is transferable or non-transferable.

Our UTR can be considered as a transferability spectrum, consisting of the instance and channel axis, where the instance axis denotes which target data is used to calculate the UD for the transferability measuring, while the channel axis represents the transferability of different channels of the source encoder. To facilitate the UTR to address the previous two limitations in SFUDA, we designed the following variants. Specifically, for the first limitation, the UTR on the domain-level, namely  $UTR_D$ , integrates the UTR over all of the target instances, which measures the transferability of different channels more accurately than the UTR of each target instance, thus can efficiently guide the utilization of the knowledge of the source encoder. For the second limitation, the instance-level namely  $UTR_I$  integrates UTR of a particular instance over all channels, which is proven to characterize the reliability of the inferred target semantics of each target instance. The usages of the  $UTR_D$  and  $UTR_I$  are illustrated in Fig. 1 (b) and (c).

Based on the introduced domain-level and instance-level UTR, we proposed a novel Calibrated Adaptation Framework to address the two limitations of existing SFUDA works. Firstly, a source knowledge calibration module is designed, which uses  $UTR_D$  to identify the transferability of different channels of the source encoder, and calibrates the source knowledge that transferred to the target domain by distilling the knowledge in transferable channels and discard the knowledge in less-transferable ones. Secondly, a target semantic calibration module is proposed based on our  $UTR_I$ to detect unreliable target semantics and calibrate them by designing a semantic calibration loss. The semantic calibration loss encourages the model to "forget" the unreliable semantics and "discover" the true ones. With the calibrated source knowledge and target semantics, we safely adapt the model to the target domain, therefore summarizing a better-performing target model.

Our main contributions are summarized as follows: **Firstly**, we propose an *Uncertainty-induced Transferability Representation* (UTR) to explore the source model transferability in the absence of source data and target labels, which is beneficial to the SFUDA community. **Secondly**, we design a novel Calibrated Adaptation Framework (CAF) to calibrate the source knowledge and the inferred target semantics, allowing the target model to fully and safely exploit the source knowledge and target data, hence learning a better-performing target model. **Finally**, we verified the effectiveness of our method with extensive experimental results and demonstrated that the proposed method achieves state-of-the-art performances on the three SFUDA benchmarks.

# II. RELATED WORK

# A. Source free unsupervised domain adaptation

Recent years have witnessed great achievements in the vanilla UDA [9], [10], [15]–[23]. However, they assume that the source data is accessible during the adaptation, which is not always practical. SFUDA aims to adapt a source-trained model to an unlabeled target domain without access to source data [5], [24], [25]. Without the labeled source

data, some methods propose to generate labeled data by generative adversarial net (GAN) [26]. Kurmi et al. [27] generate source data using the source-trained classifier, so that the vanilla UDA methods can be applied. Li et al. [28] leverage a conditional GAN to directly produce training samples in the target style. These methods use the source model as auxiliary supervisions to control the label of the generated data. Nevertheless, the source model is ineffective for the data generation process, due to the instability training of GAN [29]. Most existing SFUDA methods directly transfer all the source knowledge to the target model at the start of training, infer the semantic information of target data using the target model, and update the target model with the inferred semantic information. SHOT [5] and ISFDA [30] predict the target category using the pseudo-labeling strategy. CPGA [31] and BAIT [4] propose to align the samples with category-wise prototypes in a contrastive learning framework. NRC [1] and LSC-SDA [2] aim at propagating the categorical semantics from the neighborhood/cluster structure to the feature space. Xia et al. [3] focus on the disagreements between target data and the source model. They select partial target data with high agreements with the source model and apply the source model to these samples. However, without measuring the transferability of the source knowledge, these methods fail to control over discarding non-transferable knowledge and preserving transferable knowledge. Additionally, they fail to identify risks of applying the source model to infer target semantics. To our best knowledge, Wang et al. [6] is the most similar work as ours. They explored to transfer only partial source model parameters based on calculating the parameter variations after each adaptation procedure in the stochastic optimization. However, their measurement is susceptible to the quality of the adaptive procedures. In contrast, our method measures the transferability using only the source model and unlabeled target data, which is irrelevant to the adaptation procedure, so that can avoid the hazards of the unreliable adaptation.

## B. Uncertainty

Uncertainty is an important criterion to measure the robustness of a deep model [11], [32]–[34]. Given an annotated sample (x, y) and a model parameterized by  $\theta$  trained on domain D, the uncertainty can be decomposed into the following equation:

$$P(y|x,D) = \iint \underbrace{P(y|\mu)}_{Data} \underbrace{P(\mu|x,\theta)}_{Distributional} \underbrace{P(\theta|D)}_{Model} d\theta d\mu, \quad (1)$$

where  $\mu = \theta(x)$  is the predicted label distribution and the three probability density functions represent the data uncertainty, model uncertainty, and distributional uncertainty respectively [11]–[13]. The data uncertainty is almost irreducible which arises from the natural complexity of the data, such as the class overlap, label noise, homoscedastic and heteroscedastic noise. The model uncertainty measures how well the model fits to its training distribution [11], [12], [14]. The distributional uncertainty measures the probability of an input instance that is sampled from a region that the model is "unfamiliar" with. Its characteristic has prompted its usage in the out-of-distribution detection [35]–[37] and also in vanilla UDA methods [13],

# [38]. To our best knowledge, our work is the first to propose the use of uncertainty to explore transferability in SFUDA. *C. Transferability*

It is essential to asses and measure the model transferability and there are two mainstream methods in the deep learning community. Firstly, the transferability of a model is measured by how much it can bridge the domain discrepancy between the source and the target domain [39]. It can be calculated by domain discrepancy measurements such as Proxy A-distance [9] and Maximum Mean Discrepancy (MMD) [10]. In addition, Chen et al. [39] propose the Corresponding Angle to measure the transferability. However, these methods require the access to the source data which is not suitable for SFUDA. Secondly, some transfer learning methods investigated the transferability of pre-trained source representations to the target domain. Existing works in this line of research have been proposed, such as the NCE [40], LEEP [7] and LogME [8]. Nevertheless, they need the target data annotations which are not applicable in SFUDA. In contrast, our proposed method can estimate the transferability in the absence of source data and target data labels that fits the challenging SFUDA setup.

# III. UNCERTAINTY-INDUCED TRANSFERABILITY

REPRESENTATION

It is essential to analyse the source knowledge transferability for SFUDA, *however*, existing transferability measurements are not applicable in SFUDA. To tackle this problem, in Section III-A, we develop the Uncertainty Distance (UD) as a tool to estimate the general transferability in the absence of source data and target annotations. In Section III-B, we introduce the channel-wise transferability analysis and propose the Uncertainty-induced Transferability Representation (UTR). In Section III-C, we derive the domain-level UTR and the instance-level UTR and state their effectiveness for the SFUDA community.

A. Transferability measurement using Uncertainty Distance

Not all knowledge in the source model is transferable and discriminative to the target domain. Therefore, it brings risks if we do not measure and quantify the transferability of source knowledge but deploy it directly in the target domain. However, previous transferability measurements require some matched information, either both the source and target data, or data-annotation pairs of the target domain. These requirements are infeasible in SFUDA where only unmatched source model and target data are provided. The unmatched information makes it extremely challenging to measure the transferability by acquiring "known and certain" information as the supervision signal.

To this end, our work alternatively explores and exploits the uncertainty as a fundamental tool, and proposes an Uncertainty Distance (UD) to address these challenges. The UD is an implicit distance between the target instance  $x_t$  and the source domain  $D_s$ . A low UD demonstrates that  $x_t$  is "close" to  $D_s$  given a source model parameterized by  $\theta_s$ , which reflects that it is efficient for the  $\theta_s$  to reduce the domain discrepancy between the source and target domains. Therefore it suggests that the  $\theta_s$  is transferable to the target domain and a high UD indicates the opposite.



Fig. 2. (a): The model uncertainty measures the degree to which a model's fitted region covers its training distribution. (b) The distributional uncertainty measures the probability of an input instance that is sampled from a region that unfitted/unfamiliar by the model, which reveals how far the sample is from the fitted region of the model. (c): The distributional and model uncertainties reveal an implicit uncertainty distance (UD) from the target instance to the source data distribution, which reveals the ability of the source model in reducing the domain discrepancy between the target and source domains, therefore suggesting the transferability of the source model to the target domain. In SFUDA, UD could be approximated by distributional uncertainty as the model uncertainty is small. (d) The UTR leverages the distributional uncertainty to estimate the transferability of these channels more accurately. (f) The instance-level UTR integrates UTR on the channel axis, which identifies the risk of using source knowledge to predict the semantics of the target instance.

(e)

Our consideration is shown in Fig. 2 (a)-(c). Given the source model parameterized by  $\theta_s$ , the model uncertainty characterizes the degree to which the fitted region of  $\theta_s$  covers its training distribution (i.e. the source domain  $D_s$ ). While given both the  $\theta_s$  and the target instance  $x_t$ , the distributional uncertainty reveals how far the  $x_t$  is from the fitted region of the  $\theta_s$ . Previous observations inspired us that the cooperation of the two uncertainties reveals the distance between the target instance  $x_t$  and the source domain  $D_s$ . Such a distance implicitly reflects the contributions of the source model to reduce the domain discrepancy. It can also be used to probe and measure the transferability of the source model to the target instance for SFUDA.

By incorporating the distributional uncertainty and the model uncertainty, we first formulate the UD as:

$$UD(x_t, \theta_s, D_s) = M(\underbrace{P(\theta_s(x_t)|x_t, \theta_s)}_{Distributional} \underbrace{P(\theta_s|D_s)}_{Model}),$$
(2)

where  $M(\cdot)$  is the uncertainty measurement function such as the Sensitivity Analysis [41], the Deep Ensembles [42] and the MC dropout [43].

Although it requires the  $D_s$  in Equation 2 to measure the model uncertainty, we argue that *it is still feasible to estimate transferability using UD in the SFUDA*. The reason is that the source model has been well-trained in the source domain so that the  $\theta_s$  fits  $D_s$  well. As shown in Fig. 2 (c), in this case, the model uncertainty is small enough to be ignored, and the UD in SFUDA can be approximated by the distributional uncertainty calculated by the target instance  $x_t$  and source model parameters  $\theta_s$ :

$$UD(x_t, \theta_s) = M(P(\theta_s(x_t)|x_t, \theta_s)).$$
(3)  
B. Channel-wise Transferability Analysis

The proposed UD in Equation 3 essentially measures the transferability of the **whole source knowledge** (i.e., the whole  $\theta_s$ ) to the target instances. Nevertheless, as motivated in the introduction, only partial knowledge is useful for the target domain. Therefore we proposed to analyse the transferability of the knowledge in a finer-grained manner: to determine which part of the learned source parameters are transferable to the target domain. A straight-forward method is to measure the transferability of the partial and individual source parameters

 $\theta$  using  $UD(x_t, \theta)$ , where  $\theta \subset \theta_s$ . *However*, it is well-known that most deep neural networks belong to the end-to-end "black-box" system, where the knowledge is highly abstract and entangled. Individual parameters generally make no sense, let alone analyzing their transferability.

(f)

To tackle this challenge, we propose to estimate the transferability of *different channels of the source encoder rather than different model parameters*, as shown in Fig. 2 (d). In this way, the transferability of a particular channel natural represents the transferability of the "partial" source knowledge (relevant parameter) that encodes the feature of this channel. More specifically, we propose the Uncertainty-induced Transferability Representation (UTR), a transferability spectrum, composed of the instance axis and the channel axis, which is formulated as:

 $UTR(x_t, h_s) = [UD(x_t, h_s^1), ..., UD(x_t, h_s^d)],$  (4) where  $UD(x_t, h_s^i) = M(P(z_i|x_t, h_s^i)), x_t$  is the target instance,  $z = h_s(x_t), z \in \mathbb{R}^d$  denotes the *d*-channel target features produced by the source encoder  $h_s$ , and  $z_i = (h_s(x_t))^i = h_s^i(x_t)$  is the target feature of the  $i^{th}$ channel,  $h_s^i$  is the potential source parameter to encode  $z_i$ . The instance axis of UTR denotes which target data is used to calculate the UD for the transferability estimating. The channel axis represents the transferability of different channels of the source encoder. The  $i^{th}$  channel of the UTR (i.e.,  $UD(x_t, h_s^i)$ ) indicates the transferability of  $z_i$  to the target domain. A low value of  $UD(x_t, h_s^i)$  indicates that the target instance  $x_t$  is close to the source one in the space of  $z_i$  and suggests that the source knowledge to encode  $z_i$  (i.e., the parameters  $h_s^i$ ) is highly transferable across the two domains.

To calculate the UTR, we adopt the sensitivity analysis method [41] as the uncertainty measurement  $M(\cdot)$  for Equation 4. To be specific, the model parameters of  $h_s$  are perturbed for T times randomly as follows:  $\{h_{s;T} = (1+r_t)*h_s\}_{t=1}^T$  are firstly calculated by inserting T random perturbations  $\{r_t\}_{t=1}^T$ to original parameter  $\theta_{hs}$ . Then the uncertainty is estimated by calculating the variance of the T outputs of the  $i^{th}$  dimension feature:

$$M(P(z_i|x_t, h_s^i)) = Var_{h_s \sim h_{s:T}}((h_s(x_t))^i).$$
(5)

#### C. The Domain-level and Instance-level UTR

Given a source model parameterized by  $\theta_s$  and a target instance  $x_t$ , the UTR (in Equation (4)) is able to quantify the fine-grained transferability of the instance-level target features. In order to tackle the limitations in the SFUDA community: 1) measuring the transferability of source knowledge to target domain to sufficiently exploit it; 2) measuring the risk of inferring semantic information of target instances using the source knowledge, we design two variants of the UTR on two levels: the *domain-level* UTR namely the  $UTR_D$  and the *instance-level* UTR namely the  $UTR_I$ .

The  $UTR_D$  describes the domain-level transferability estimation over the channel axis, which identifies how transferable each channel of the source encoder is to the target domain using the UD of the source model to all target instances. The  $UTR_I$  characterizes the instance-level trasferability over all the target instances, which identifies the instance-level risk of inferring target semantic labels. The two are useful measurements proposed to fit in the later on adaptation framework for SFUDA problem.

Specifically, as shown in Fig. 2 (e), the  $UTR_D$  is calculated by integrating the  $UTR(x_t, h_s)$  of all  $n_t$  target instances over the target domain  $D_t$ . Detailed formulation is as follows:

$$UTR_D(h_s) = \mathbb{E}_{x_t \sim \mathcal{X}_t} UTR(x_t, h_s) = \frac{1}{n_t} [\sum_{i=0}^{n_t} UD(x_t^i, h_s^1), ..., \sum_{i=0}^{n_t} UD(x_t^i, h_s^d)]$$
(6)

As for the instance-level transferability spectrum, the  $UTR_I$  is calculated by integrating the  $UTR(x_t, h_s)$  over all the *d*-channels of the source encoder  $h_s$ , as shown in Fig. 2 (f). The detailed formulation of the  $UTR_I$  is as follows:

 $UTR_I(x_t) = \mathbb{E}_{z \sim \mathbb{R}^d} UTR(x_t, h_s)$ 

$$= \frac{1}{d} \left[ \sum_{i=0}^{d} UD(x_t, h_s^i) \right]$$
(7)

# IV. CALIBRATED ADAPTATION FRAMEWORK A. Notation

In this paper, we focus on the K-way visual object classification task. SFUDA provides a well-trained source model parameterized by  $\theta_s$  to the target domain  $D_t$ , where  $\theta_s = g_s \circ h_s$ ,  $h_s$  is the parameter of the source encoder, and  $g_s$  is the parameter of the source classifier. The target domain  $D_t = \{x_t^i\}_{i=1}^{n_t}$  consists of  $n_t$  unlabeled target instances. The SFUDA aims to learn a discriminative target model parameterized by  $\theta_t = g_t \circ h_t$  using the  $\theta_s$  and  $D_t$ .

# B. Overall

Most existing SFUDA methods directly transfer all source knowledge to the target model at the start of training, infer the semantic information (target labels) of target data using the model, and directly update the model using the inferred semantic information. *However, they are limited as follows:* 1) the utilization of the source knowledge is limited. Directly transferring all source knowledge to the target model ignores discarding the less-transferable one. And updating the models using the inferred target semantics failed to preserve the discriminative knowledge in the source model. 2) the target semantic information inferred by the source model is risky To this end, we introduce the Calibrated Adaptation Framework (CAF). To tackle the first limitation, we propose to *calibrate the source knowledge that transferred to the target model* using our  $UTR_D$ . To tackle the second, we propose to *calibrate the inferred semantic information of target instances* based on our  $UTR_I$ . Finally, we adapt the model based on the calibrated source knowledge and target semantics. The overview of CAF is shown in Fig. 3. The pseudo-code of the whole algorithm is described in Algorithm 1.

Source knowledge calibration. To address the limitation 1, instead of directly inheriting all source knowledge, we design a transferability-controlled knowledge distillation loss  $\mathcal{L}_{kd}$ , which used  $UTR_D$  to control the knowledge distillation by quantifying different channels' transferability and assigning more transferable channels larger weights. On the one hand, it prompts the target model to learn transferable source knowledge and discard less-transferable ones. On the other hand, it constrains the updated target model by unceasingly distilling the transferable source knowledge along the whole training process, rather than at the beginning only.

Target semantics calibration. The less-transferable knowledge is prone to lead to incorrect semantics (labels) inferred by the model. Considering that it is fundamental in SFUDA to update the target model based on the inferred semantics of target instances, calibrating the incorrect semantics is essential to learn a discriminative target model. Specifically, after inferring target semantics (using the source model at the beginning of training, and the target model later), we use the  $UTR_I$  to select instances whose inferred semantics are unreliable. Then, a semantic calibration loss  $L_{sc}$  is designed to calibrate their model predictions. Specifically, on the one hand, as the semantics inferred by the feature of less-transferable channels tend to be wrong, we proposed to use to "forget" the current semantics by minimizing a negative cross-entropy  $L_c$ . It implicitly guides the model to re-initialize the parameters representing the less-transferable knowledge. On the other hand, we minimize the entropy of the prediction probability distribution of these instances to force their predictions close to a new and appropriate class category. This procedure discovers the new and proper semantics of these instances.

*Adaptation.* With the above two steps, the target model "safely" integrates the source knowledge and target semantics. The adaptation step finally refines the target model using inferred semantics from the transferable knowledge, therefore summarizing a better-performing discriminative model.

# C. Source Knowledge Calibration and Distillation

Not all source knowledge is transferable and discriminative to the target domain. Directly transferring all source knowledge to the target model without dealing with the lesstransferable parts of it is detrimental to the adaptation of the target domain. To this end, we instruct the target model to selectively learn the features of transferable channels of the source encoder, therefore, to inherit transferable knowledge from the source encoder. Given the source encoder  $h_s$ , the  $UTR_D(h_s)$  is calculated following the Equation (6) to estimate the transferability of each channel in  $h_s$ , where a lower



Fig. 3. Overview of Our Calibrated Adaptation Framework. (a) Source knowledge absorption calibration. The  $UTR_D$  is calculated and used to estimate the transferability of the knowledge of the source encoder  $h_s$ . Then knowledge in  $h_s$  is distilled into the target encoder  $h_t$  with  $L_{kd}$ , which controls the target encoder to absorb transferable source knowledge and neglect less-transferable knowledge according to  $UTR_D$ . (b) Target semantics calibration. (b.1) Infer target semantics with target model. (b.2) Select instances whose inferred semantics are risk (the red point) according to their  $UTR_I$  and threshold  $\tau$ . (b.3) The forget objective  $L_f$  of the semantics calibrate loss minimizes the negative cross-entropy to risk instances, forcing [Li: it] to forget the current unreliable semantics. (b.4) The discover objective  $L_d$  of the semantics calibrate loss guides to discover their true semantics by minimize the entropy of the prediction probability distribution of the target instances. (c) Adaptation. (c.1) Re-infer target semantics. (c.2) Refine the model with the adapt loss  $L_a$ .

 $UTR_D$  value suggests stronger transferability. Then, the target model learn the source knowledge based on the identified transferability. We proposed a novel transferability-controlled knowledge distillation loss as the objective:

$$L_{kd} = \mathbb{E}_{x_t \sim \mathcal{X}_t} [\|Q(UTR_D(h_s)) \odot [h_s(x_t) - h_t(x_t)]\|_2],$$
(8)

where Q(x) = sigmoid(-x) is a monotone minus function,  $\odot$  is the Hadamard product. The  $Q(UTR_D(h_s))$  weights the mean squared error term  $||h_s(x_t) - h_t(x_t)||_2$  to distill knowledge within  $h_s$  to  $h_t$ , aiming to assign large weights to features with low  $UTR_D$  while small ones to those with high  $UTR_D$ , guiding the target model to learn more transferable knowledge from the source model and discard less-transferable ones in a well-controlled manner.

# D. Target Semantics Calibration

Refining a target model based on the inferred semantics (labels) of target instances is a fundamental and important step for the adaptation in SFUDA. Due to the less-transferable source knowledge, the predicted target semantics may be incorrect, which greatly hinders the adaptation to the target domain. To this end, we design the target semantics calibration module to calibrate the target semantics.

First, the inferred semantics of a target instance  $x_t$  is  $\hat{y} = \arg \max p(x_t)$  with probability  $p^{\hat{y}}(x_t)$ , where  $p(x_t) = \sigma(\theta_s(x_t)/\theta_t(x_t))$  is the source/target model predicted probability distribution,  $\sigma(.)$  is the softmax function. Note that we use the source model to infer target semantics at the first epoch, and turn to use the target model later since the target model will be more discriminative to the target domain after adaptation.

Second, we leverage  $UTR_I$  to detect risk target instances whose semantic is prone to be incorrectly inferred that satisfies  $\{x_t : UTR_I(x_t) > \tau\}$  as  $\mathcal{X}_{t;risk}$ , where  $\tau$  denotes the threshold. Following the first step, the feature encoder that calculates  $UTR_I(x_t)$  (Equation 7) changes from  $h_s$  to  $h_t$  after the first epoch.

Third, based on the detected instances  $\mathcal{X}_{t;risk}$ , we propose a semantics calibrated loss  $L_{sc}$  to calibrate the semantics of these instances. Since their semantics is prone to be inaccurate, we train the target model to firstly forget these semantics by minimizing the negative cross-entropy loss. The forget objective  $L_f$  is represented as follows:

$$L_f = \mathbb{E}_{x_t \sim \mathcal{X}_{t;risk}} - CE(x_t, \hat{y}).$$
(9)

As illustrated in Fig. 3 (b.3), optimizing this term decreases the prediction probability to the misclassified category  $\hat{y}$ .

On the other hand, we guide the target model to discover the true semantic by the following discover objective  $L_d$ :

$$L_d = -\mathbb{E}_{x_t \sim \mathcal{X}_t} \sum_{k=1}^{K} p(x_t) log p(x_t), \tag{10}$$

where  $p(x_t) = \sigma(\theta_t(x_t))$  is the target model predicted probability distribution.  $L_d$  aims to minimize the entropy of the  $p(x_t)$ , thus guiding the model to assign the prediction to an appropriate class. Note that, instead of only minimizing the entropy on  $\mathcal{X}_{t;risk}$ , we calculate  $L_d$  on all target instances  $\mathcal{X}_t$ . In this way, the semantic information of instances with low  $UTR_I$ , where the model tends to make the right predictions, is also introduced to help the semantic discovery of instances in  $\mathcal{X}_{t;risk}$ .

Such a "forget-discover" process implicitly guides the model to free itself from the shackles of less-transferable knowledge and facilitates the discovery of the true semantics of the target data, and the semantic calibration loss can be denoted as:

$$L_{sc} = \gamma L_f + L_d, \tag{11}$$

where  $\gamma$  is the scale coefficient of the  $L_f$ .

E. Adaptation

With the above two steps to calibrate the source knowledge and target semantics, the target model then can safely adapt to the target model. In the adaptation step, we re-infer the target semantics by the model and use it to refine the target model, ultimately adapting the model to the target domain.

In this step, we adopt the pseudo-label strategy in [5] to re-infer the semantic  $\hat{y}$  of the target instance  $x_t$  consider its simplicity and effectiveness. Given  $x_t$  and  $\hat{y}$ , we optimize the model with the cross-entropy loss and the objective of the adapt step can be formulated as:

$$L_{a} = \mathbb{E}_{x_{t} \sim \mathcal{X}_{t}} CE(x_{t}, \hat{y}). \tag{12}$$
  
F. Training Steps

In this subsection, we summarize the training steps of CAF framework. The two calibration steps are separate with

Algorithm 1 Calibrated Adaptation Framework

<b>Require:</b> Source model parameterized by $\theta_s = g_s \circ h_s$ , target
model parameterized $\theta_t = g_t \circ h_t$ , unlabeled target instances
$D_t$
<b>Require:</b> hyperparameter $\tau$ , $\lambda$ , $\gamma$
Calculate $UTR_D(h_s)$
while $i < max$ epoch do
In the $i^{th}$ epoch
Sample batch $T$ from $D_t$
Calculate the $L_{kd}$
Infer target semantics
Calculate $UTR_I(x_t)$ , select $\mathcal{X}_{t;risk}$ with $\tau$
Calculate $\gamma L_f + L_d$
Train the target model by optimizing $\lambda L_{kd} + \gamma L_f + L_d$
In the $i + 1^{th}$ epoch
Sample batch T from $D_t$
Infer target semantics
Calculate $L_a$
Train the target model by optimizing $L_a$
i = i + 2
end while

the adapt step. Specifically, in the  $i^{th}$  epoch, perform two calibration steps to calibrate transferable source knowledge and target semantics by:

$$\min_{a} \lambda L_{kd} + \gamma L_f + L_d, \tag{13}$$

where  $\lambda$  and  $\gamma$  is the scale coefficient.

And in the  $i+1^{th}$  epoch, conduct the adaption step to adapt the target model to the target domain:

$$\min_{\theta_t} \lambda L_a. \tag{14}$$

# V. RESULTS

We evaluate our SFUDA method using the following three benchmarks: Office-31 [44], the Office-Home [45] and the VisDA [46]. Office-31 [44] contains 4,652 images in 31 categories from three domains: Amazon (A), Webcam (W) and DSLR (D). Office-Home [45] consists of four domains, i.e., Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Rw), with 65 classes and a total of 15,500 images. VisDA [46] is a more challenging dataset, whose source domain contains 152k synthetic images generated by rendering 3D models while the target domain has 55k real object images sampled from Microsoft COCO [47].

#### B. Implementations

A. Datasets

For fair comparisons with existing methods, we adopt the backbone of ResNet-50 [48] for Office-31 and Office-Home and ResNet-101 for VisDA. Following the setups in [1], [5], along with the backbones, we used a fully-connected (fc) layer with the output channels of 256 as the encoder. A fc layer with the weight normalization as the classifier. The source model is trained following the same strategy with [1], [5]. **The pre-trained source model is used to adapt to the target domain but without using any labeled source data.** In the optimization, we adopt SGD with momentum 0.9 and

 TABLE I

 CLASSIFICATION ACCURACIES (%) ON OFFICE-31 DATASET.

Method	A→D	$A \rightarrow W$	D→A	$D{\rightarrow}W$	$W \rightarrow A$	$W \rightarrow D$	Avg.
Source-model	80.4	76.5	60.2	95.6	63.4	98.6	79.1
SoFA [50]	73.9	71.7	53.7	96.7	54.6	98.2	74.8
SFDA [51]	92.2	91.1	71.0	98.2	71.2	99.5	87.2
SHOT [5]	94.0	90.1	74.7	98.4	74.3	99.9	88.6
3C-GAN [28]	92.7	93.7	75.3	98.5	77.8	99.8	89.6
BAIT [4]	92.0	94.6	74.6	98.1	75.2	100.0	89.1
NRC [1]	96.0	90.8	75.3	99.0	75.0	100.0	89.4
HCL [52]	94.7	92.5	75.9	98.2	77.7	100	89.8
AAA [53]	95.6	94.2	75.6	98.1	76.0	99.8	89.9
A2Net [3]	94.5	94.0	76.7	99.2	76.1	100.0	90.1
DIPE [6]	96.6	93.1	75.5	98.4	77.2	99.6	90.1
Ours	95.0	93.5	76.3	99.1	78.4	100.0	90.3

batch size of 64 on all datasets. For the Office-31 and office-Home datasets, the learning rates used to train the ResNet-50 backbone and the newly added layers are 1e-3 and 1e-2 respectively. The learning rate is 1e-4 for VisDA. We trained 40, 60 and 50 epochs for Office-31, Office-Home and VisDA respectively. Note that the mixup [49] data augmentation is used in the adaptation step. The threshold of  $UTR_I$ , i.e.  $\tau$ , is set to be 3. The weight  $\lambda$  of the transferability-controlled knowledge distillation loss is set to 10 at the beginning. As the training procedure progresses, the model is gradually adapted to the target domain, requiring less source knowledge. Therefore, after 10 epochs, we decrease  $\lambda$  to zero. The weight  $\gamma$  of the "forget" loss is set to 0.9. For the uncertainty measurement (Equation 5), T is set to 2, and  $r_t$  is randomly sampled from the uniform distribution U(-0.05, 0.05).

# C. Comparison with State-of-the-Art Methods

We report the results on Office-31, Office-Home, and VisDA, in Tables I, II, and III, respectively.

On Office-31 tasks, in terms of the average accuracy of 6 transfer tasks, our method outperforms the state-of-the-art work A2Net [3] and DIPE [6] by 0.2%, improving from 90.1% to 90.3%. We also achieve the state-of-the-art results on  $W \rightarrow A$  and  $W \rightarrow D$ . For other transfer directions of Office-31, we achieved very competitive results. We hypothesize the reason of the results is that our method brings transferability risk quantification to SFUDA and integrates the "safe-to-transfer" source knowledge to the target domain for better adaptation. We also argue that our method is more useful and brings more improvements for challenging adaptation tasks, where the cross-domain transfer risk is high. *Instead*, the Office-31 transfer tasks are easy and bring less risks (considering that the average accuracy of the source-only model is 79.1%), so our method improvement is competitive and not that significant.

As expected, on the more challenging Office-Home tasks and VisDA tasks (the mean accuracy of the source models are 60.0% and 48.0%), our method brings larger improvement. In the Office-Home tasks, we achieve the state-ofthe-art performance on 8 of 12 tasks, and also outperform the prior work in terms of the average accuracy of 0.6%, improving from 72.6% (by A2Net [3]) to 73.2%. Particularly, we have achieved significant improvements in two difficult tasks Ar $\rightarrow$ Cl and Re $\rightarrow$ Cl and outperform the second best one by 1.2% and 0.7%, respectively. On the VisDA tasks, our method outperforms others among 10 out of 12 tasks and

TABLE II Classification accuracies (%) on Office-Home dataset (ResNet-50). AC denote the task Ar→CL.

Method	AC	AP	AR	CA	CP	CR	PA	PC	PR	RA	RC	RP	Avg.
Source-model	44.8	67.4	75.1	52.3	63.4	63.7	53.6	39.5	72.7	64.1	45.2	77.6	60.0
SHOT [5]	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
SFDA [51]	48.4	73.4	76.9	64.3	69.8	71.7	62.7	45.3	76.6	69.8	50.5	79.0	65.7
SoFA [50]	-	74.1	77.6	-	71.8	75.1	-	-	-	-	-	-	-
BAIT <sup>[4]</sup>	57.4	77.5	82.4	68.0	77.2	75.1	67.1	55.5	81.9	73.9	59.5	84.2	71.6
PS [54]	57.8	77.3	81.2	68.4	76.9	78.1	67.8	57.3	82.1	75.2	59.1	83.4	72.1
AAA [53]	56.7	78.3	82.1	66.4	78.5	79.4	67.6	53.5	81.6	74.5	58.4	84.1	71.8
NRC [1]	57.7	80.3	82.0	68.1	79.8	78.6	65.3	56.4	83.0	71.0	58.6	85.6	72.2
DIPE [6]	56.5	79.2	80.7	70.1	79.8	78.8	67.9	55.1	83.5	74.1	59.3	84.8	72.5
A2Net [3]	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.6
Ours	59.8	81.2	83.2	67.2	79.2	80.1	68.4	56.4	83.0	73.7	61.2	85.9	73.2
-													

TABLE III CLASSIFICATION ACCURACIES (%) ON VISDA-C DATASET (RESNET-101),  $Per_c$  denotes the per-class accuracy.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	$Per_{a}$
Source-model	64.6	28.9	47.2	63.5	67.2	12.4	82.5	23.5	61.7	31.4	82.1	11.1	48.0
3C-GAN [28]	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
SHOT [5]	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
SFDA [51]	86.9	81.7	84.6	63.9	93.1	91.4	86.6	71.9	84.5	58.2	74.5	42.7	76.7
BAIT [4]	93.7	83.2	84.5	65.0	92.9	95.4	88.1	80.8	90.0	89.0	84.0	45.3	82.7
DIPE [6]	95.2	87.6	78.8	55.9	93.9	95.0	84.1	81.7	92.1	88.9	85.4	58.0	83.1
HCL [52]	93.3	85.4	80.7	68.5	91.0	88.1	86.0	78.6	86.6	88.8	80.0	74.7	83.5
PS [54]	95.3	86.2	82.3	61.6	93.3	95.7	86.7	80.4	91.6	90.9	86.0	59.5	84.1
AAA [53]	94.4	85.9	74.9	60.2	96.0	93.5	87.8	80.8	90.2	92.0	86.6	68.3	84.2
A2Net [3]	96.1	88.3	85.5	74.1	97.1	95.4	89.5	79.4	95.4	92.9	89.1	42.6	85.4
NRC [1]	96.8	91.3	82.4	62.4	96.2	95.9	86.1	80.6	94.8	94.1	90.4	59.7	85.9
Ours	98.0	92.9	88.3	78.0	97.8	97.7	91.1	84.7	95.5	91.4	91.2	41.1	87.3

surpasses the SOTA method NRC by a large margin (2.4%) in the per-class accuracy.

Compared with the most related work DIPE [6], our method also obtains performance improvement in all three benchmarks, including 0.2% in Office-31, 0.7% in Office-Home and 4.2% in VisDA. The reported results clearly demonstrate the efficacy of our method.

# VI. ANALYSIS

# A. Ablation Study

Ablation study on the Source Knowledge Calibration. We designed the transferability-controlled knowledge distillation loss  $L_{kd}$  to distill transferable source knowledge. To prove its effectiveness, the ablation results of this module are reported in the first three rows of Table IV. It can be seen that the v2  $(L_a + L_{kd})$  outperforms v1  $(L_a)$  by 2.1% in Ar $\rightarrow$ Cl and 1.7% in Ar $\rightarrow$ Re respectively. However, it induces negative effects on Ar $\rightarrow$ Pr. We hypothesize the reasons to be that the transferable knowledge to the target domain may not discriminate the target samples and may infer wrong semantics in the Ar $\rightarrow$ Pr. To prove our hypothesis, by comparing results of v4 and ours in Table IV, it proves that by adding two semantics calibration losses  $L_f$  and  $L_d$ , the  $L_{kd}$  is more effective and brings significant improvements on all transfer directions Ar -> Cl,  $Ar \rightarrow Re$  and  $Ar \rightarrow Pr$  (see more details in the following two comparisons:  $[v2 \leftrightarrow v3]$  and  $[v4 \leftrightarrow Ours]$ ). It may prove that calibrating the target semantics helps calibrate transferable source knowledge.

Ablation study on the Target Semantics Calibration We designed a forget loss  $L_f$  and a discover loss  $L_d$  to calibrate the target semantics subsequently. By comparing the results of v1 and v3 in Table IV, we may conclude that adding  $L_d$  in v1 boosts the performances by 4.2%, 3.8% and 5.0% on the three tasks respectively.

In addition, using the  $L_f$  only without the  $L_d$  brings negative effects (see the comparison [v6  $\leftrightarrow$  Ours]). However,



Fig. 4. The visualization of the prediction probability, prediction accuracy of different samples and their  $UTR_I$  in VISDA by (a): source model (b) CAF without  $L_f$ , and (c) CAF. Blue point represent samples that the model predicted correctly, and red indicates that the prediction is wrong. The vertical axis represents the prediction probability of the sample, and the horizontal axis represents  $UTR_I$ .



Fig. 5. The Accuracy- $UTR_I$  curve of source model to target samples on Ar $\rightarrow$ Cl and VisDA tasks. The horizontal axis is the threshold of  $UTR_T$ , denotes samples that satisfied  $UTR_I(x_t) > \tau$ . The vertical axis represents the predicted semantic accuracy of the model for these samples. For a better illustration, we select samples to which the max prediction probability of the source model are larger than 0.5.

from comparisons  $[v3 \leftrightarrow v4]$  and  $[v5 \leftrightarrow Ours]$  in Table IV, we verify that  $L_f$  is only effective when combined with  $L_d$ .

Moreover, we notice that the forget loss  $L_f$  is more effective on challenging tasks, e.g. Ar $\rightarrow$ Cl. We hypothesize that it is because the predictions on challenging tasks tend to be wrong and therefore forgetting model prediction completely brings more improvements.



Fig. 6. The Accuracy- $UTR_I$  curves in Ar $\rightarrow$ Cl where the  $UTR_I$  is calculated on (a) different layers ( the last, penultimate, and antepenultimate bottlenecks of Resnet-50), (b) other model structures (VGG16 and AlexNet) and (c) the target model. For a better illustration, we report the prediction accuracy of the source model for samples where the max prediction probability are larger than 0.5, 0.9 for (a) and (b), respectively.

 TABLE IV

 Ablation study on three Office-Home task.

Method	Module	Ar→Cl	Ar→Pr	Ar→Re	
source	-	44.6	67.3	74.0	
v1	La	50.4	73.3	75.6	
v2	$L_a + L_{kd}$	52.5	72.4	77.3	
v3	$L_a + L_d$	54.6	77.1	80.6	
v4	$L_a + L_f + L_d$	57.1	77.2	81.1	
v5	$L_a + L_d + L_{kd}$	58.5	79.2	82.0	
v6	$L_a + L_f + L_{kd}$	46.6	70.1	74.3	
Ours	$L_a + L_f + L_d + L_{kd}$	59.8	81.2	83.2	
Ours-Merge	$L_a + L_f + L_d + L_{kd}$	57.7	79.3	82.1	
Ours-Online	$L_a + L_f + L_d + L_{kd}$	59.2	81.2	83.1	
Ours-Ensemble	$L_a + L_f + L_d + L_{kd}$	59.1	80.7	82.9	
Ours-MC dropout	$L_a + L_f + L_d + L_{kd}$	59.3	80.5	82.7	

TABLE V

HYPERPARAMETER(HPR) ANALYSIS. THE RESULTS ARE SHOWN IN FORM OF VALUE/ACC(%).

HPR			Ar-	≻Pr		
$\tau$	1.5m/76.2	2m/78.6	2.5m/81.2	3m/81.2	3.5m/80.6	4m/79.6
$\lambda$	0.5/78.56	2.5/77.91	5/79.08	7.5/80.1	10/81.2	12.5/80.4
$\gamma$	0.1/79.3	0.5/80.7	0.9/81.2	1.0/80.8	1.5/79.5	2/79.4

To further understand the forget loss  $L_f$  and prove the previous arguments, we visualize the correct/incorrect case of target prediction and the  $UTR_I$  obtained by the source model, our CAF model without the forget loss and our CAF model in Fig. 4 (a), (b) and (c) respectively. It can be seen in (a) that the source model predicts many wrong semantics in the target domain (the red points) due to the less-transferable knowledge. By observing many red points on the upper right of the Fig. 4 (b), it seems that we can not calibrate the wrong semantics of samples with high prediction probability without  $L_f$ , since the model is confident in its inferred semantics and tends to maintain these semantics. Finally, from Fig. 4 (c), it can be seen that after adding the forget term  $L_f$ , the model forgets the wrong semantics and finally calibrates their semantics. The above experiments verify the effectiveness of the Target Semantics Calibration.

Ablation study on merging different steps. In our CAF framework, the two calibration steps integrate "transferable" source knowledge and the reliable target semantics *first* and *after that* the adaptation step refines the model using the calibrated knowledge and target semantics. Therefore, it is necessary and better to perform the two calibration steps before the adaptation step. To prove the necessary, we conduct extra experiments performing the calibration and adaptation steps simultaneously. The results are denoted as "Ours-Merge" in Table IV. It can be observed that the "Ours" result outperforms the "Ours-merge" result by 2.1%, 1.9% and 1.1% in Ar $\rightarrow$ Cl, Ar $\rightarrow$ Pr and Ar $\rightarrow$ Re respectively.

## B. Hyperparameter Analysis

We analyse the sensitivity of the following hyperparameters: the  $UTR_I$  threshold  $\tau$  and the weights  $\lambda$  and  $\gamma$  of the losses  $L_{KD}$  and  $L_f$  respectively. The results in Table V demonstrate that our method is stable to the choices of hyperparameters in a wide range.

TABLE VI Improvement to existing SFUDA methods.

Method	$Cl \rightarrow Ar$	Cl→Pr	Cl→Re
Ours	67.2	79.2	80.1
SHOT [5]	68.0	78.2	78.1
SHOT+Ours	68.5	79.3	80.3
NRC [1]	68.1	79.8	78.6
NRC+Ours	68.9	80.1	80.2

# C. Calibration on other existing methods

Without measuring the transferability, current SFUDA methods [1]–[5] directly perform adaptation but ignore calibration steps in our CAF framework. Our two calibration steps fill in the gap, and are flexible and "plug-and-play". Therefore, we add our calibration modules on existing SFUDA works [1], [5] and report the experimental results in Table VI. It can be seen that adding our calibration modules on SHOT/NRC methods improves SHOT/NRC by 0.9/0.8%, 1.2/0.3%, and 2.2/2.1% on Cl $\rightarrow$ Ar, Cl $\rightarrow$ Pr and Cl $\rightarrow$ Re respectively. It proves that our CAF method is "plug-and-play" and effective to different SFUDA baselines.

## D. Empirical Analysis of UTR

The effectiveness of  $UTR_D$ . The  $UTR_D$  describes the domain-level transferability over the channel axis, which identifies how transferable each channel of the source encoder is to the target domain. To evaluate the effectiveness of UD, we conduct the following experiments.

Implementation. The experiments are conducted on the Office-31, Office-Home, and VisDA tasks. For the Office-31 tasks and the Office-Home tasks, the backbone of ResNet-50 along with a fc layer is the source encoder, whose output channel d = 256. A fc layer with weight normalization is the classifier. For the VisDA tasks, we replace the ResNet-50 with the ResNet-101 and keep the other settings the same. We follow [1], [5] to train the source model. For each task, we feedforward all target data to the pre-trained source model and finally calculate  $UTR_D(h_s)$  according to Equation 6, where T = 2,  $r_t$  is randomly sampled from U(-0.05, 0.05).

Comparison Protocols. We evaluate our  $UTR_D$  by comparing it with the existing transferability measurements, that are inapplicable in the SFUDA, including: MMD [10], A-Distance [9], Corresponding Angle [39], LogME [8], LEEP [7] and NCE [40]. In addition, the performance (prediction accuracy) is also considered as an extra intuitive measurement. Considering that existing transferability measurements are not suitable for a single channel's feature, we design the following comparison protocol. Specifically, we sort the 256 channels representations  $z = h_s(x)$  and split them into two separate 128 channels vectors  $Z_{low}$  and  $Z_{high}$ , representing the channels with the 128 smallest  $UTR_D^i(h_s)$  and the 128 largest  $UTR_D^i(h_s)$  respectively. In other words, the conclusion of  $UTR_D(h_s)$  is that  $Z_{low}$  is more transferable than  $Z_{high}$ . Then we calculate the existing transferability measurements on  $Z_{low}$  and  $Z_{high}$  and report the consistency of their conclusion with ours. Note that these source data and target annotation are given when using these measurements. The results are reported in Table VII and VIII.

First, we quantitatively measure the transferability of  $Z_{low}$ and  $Z_{high}$  with the two vanilla UDA methods, the MMD and the A-Distance, which requires the source data. These methods measure the ability to bridge the domain discrepancy between the source and target domain. The lower MMD/A-Distance, the more transferable the model is. From Table VII and VIII, it can be seen that in 15 out of 19 adaptation tasks, the MMD of  $Z_{low}$  is lower than that of  $Z_{high}$ , and in 17 tasks, the A-Distance of  $Z_{low}$  is lower than that of  $Z_{high}$ . The result

Comparison with different transferability measurements on the Office-Home tasks.  $z_{low}$  and  $z_{high}$  are features with low  $UTR_D$ and high  $UTR_D$ , respectively. The  $\uparrow/\downarrow$  indicates the larger/smaller the value, the higher the transferability. In each task, current transferability measurements are calculated on  $Z_{low}$  and  $Z_{high}$ , respectively. The more transferable one is bolded.

	Ar	→Cl	Ar	→Pr	Ar	→Re	Cl-	→Ar	Cl	→Pr	Cl-	→Re	Pr-	→Ar	Pr	→Cl	Pr-	→Re	Re-	→Ar	Re-	→Cl	Re-	→Pr
Measurement	$Z_{low}$	$Z_{high}$																						
MMD↓	0.38	0.41	0.11	0.11	0.50	0.56	0.71	0.74	0.20	0.21	0.17	0.18	0.48	0.54	0.50	0.57	0.32	0.30	0.30	0.31	0.58	0.54	0.19	0.24
A-Distance↓	1.47	1.51	1.23	1.35	0.80	0.86	1.40	1.43	1.20	1.25	1.40	1.44	1.33	1.45	1.47	1.52	0.84	0.87	1.00	1.07	1.47	1.45	0.85	0.88
Corresponding Angle↑	-0.14	-0.15	-0.05	-0.13	-0.72	-0.71	0.97	0.94	0.98	0.97	0.99	0.97	0.27	0.09	-0.09	-0.49	0.31	0.28	0.29	0.21	-0.12	0.40	0.23	0.08
LogME↑	0.83	0.82	0.93	0.92	0.89	0.87	0.85	0.81	0.84	0.82	0.84	0.81	0.83	0.82	0.81	0.80	0.86	0.84	0.84	0.83	0.84	0.82	0.92	0.90
LEEP↑	-3.66	-3.79	-3.30	-3.35	-3.25	-3.34	-2.81	-2.98	-2.39	-2.65	-2.38	-2.54	-3.51	-3.63	-3.75	-3.85	-3.15	-3.28	-3.20	-3.36	-3.51	-3.61	-3.12	-3.24
NCE↑	-2.05	-2.17	-1.21	-1.31	-1.12	-1.51	-1.71	-1.99	-1.43	-1.58	-1.44	-1.43	-1.82	-2.07	-2.30	-2.55	-1.21	-1.39	-2.43	-2.54	-2.09	-2.41	-0.93	-1.05
Accuracy(%)↑	49.5	47.1	60.3	58.2	62.9	61.2	48.6	47.2	59.5	57.9	61.7	60.4	48.4	45.3	38.7	33.9	68.8	67.5	61.8	60.2	43.5	39.6	75.4	73.1

TABLE VIII Comparison with different transferability measurements on the Office-31 and VisDA tasks.  $z_{low}$  and  $z_{high}$  are features with low  $UTR_D$  and high  $UTR_D$ , respectively. The  $\uparrow \downarrow$  indicates the larger/smaller the value, the higher the transferability. In each task, current transferability measurements are calculated on  $Z_{low}$  and  $Z_{high}$ , respectively. The more transferable one is bolded.

	A-	→D	A-	→W	D-	→A	D-	→W	W	→A	W	→D	Synthet	ic→Real
Measurement	$Z_{low}$	$Z_{high}$												
MMD↓	0.95	0.96	0.23	0.26	0.50	0.55	0.50	0.53	0.16	0.17	0.30	0.27	0.55	0.61
A-Distance↓	1.72	1.81	1.70	1.78	1.64	1.77	0.93	1.35	1.42	1.45	0.93	1.2	0.16	0.17
Corresponding Angle↑	0.7	0.61	0.15	0.11	0.10	-0.05	0.57	0.12	0.27	-0.02	-0.2	-0.3	0.38	0.11
LogME↑	0.74	0.70	0.75	0.72	0.61	0.60	0.74	0.63	0.64	0.63	0.78	0.76	0.21	0.20
LEEP <sup>↑</sup>	-2.51	-2.65	-2.11	-2.41	-3.01	-3.55	-2.44	-2.51	-3.14	-3.00	-2.00	-2.01	-0.20	-0.22
NCE↑	-0.55	-0.79	-0.69	-0.84	-1.64	-1.55	-0.28	-0.38	-1.51	-1.65	-0.14	-0.15	-1.03	-0.32
Accuracy(%)↑	80.3	73.1	74.8	68.4	54.6	50.0	92.5	89.1	59.6	55.4	98.7	97.1	51.3	45.9

indicates that  $UTR_D$  is consistent with these domain discrepancy measurements in most case, which suggest features in channels with less  $UTR_D$  are more effective to bridge the domain discrepancy, therefore; they are more transferable.

Second, we compared with the Corresponding Angle, which is proposed by Chen et al. [39] according to their observation that the eigenvectors with the largest singular values will dominate the feature transferability. We can observe that in 17 cases, the Corresponding Angle of  $Z_{low}$  is larger than that of  $Z_{high}$  in all experiments, demonstrating the consistency of  $UTR_D$  with the Corresponding Angle.

Third, we compare the consistency of our method with the transferability measurements LogME, NCE, and LEEP that estimate the potential of the source model parameter in learning a well-performed target model by refining. In Table VII and VIII, it can be seen that  $UTR_D$  is consistent with LogME in all tasks, and also consistent with NCE and LEEP in 18 and 17 cases, respectively. These results denote that the relevant source model parameters to encode  $Z_{low}$  is more transferable than  $Z_{high}$ .

Finally, We also calculate the classification performances of  $Z_{low}$  and  $Z_{high}$  on the target domain. Using  $Z_{low}$ , for example, we set the features of channels which not belongs to  $Z_{low}$ to zero, feedforward the modified feature into the classifier to get prediction and calculate the prediction accuracy.  $Z_{high}$  is evaluated in the same way. As shown in Table VII and VIII, the prediction using  $Z_{low}$  is more accurate than  $Z_{high}$  in target domain on all adaptation tasks. For example, the accuracy of  $Z_{low}$  outperform  $Z_{high}$  by 3.1%, 4.8% and 1.3% on Pr $\rightarrow$ Ar,  $Pr \rightarrow Cl$  and  $Pr \rightarrow Re$ , respectively. Note that we did not extra train the source model but only split it into two part of channels  $Z_{low}$  and  $Z_{high}$  according to our  $UTR_D$ . The significant performance gap between the two parts indicates that  $Z_{low}$ with less  $UTR_D$  is more transferable to the target domain than  $Z_{high}$ . The experimental observations from the series of studies above illustrate that 1) the proposed  $UTR_D$  is strongly consistent with current transferability measurements and can estimate the transferability, 2) Our method can effectively



Fig. 7. (a)-(c): The effectiveness of the  $UTR_D$  for channels of different layers. (a): At the last layer. (b) At the penultimate layer. (c) At the antepenultimate layer. (d)-(f): The effectiveness of the  $UTR_D$  to the target model during the adaptation process (trained with 5, 10, 15, 20, 25 and 30 epochs) on (d)  $Ar \rightarrow Cl$ , (e)  $Ar \rightarrow Pr$ , and (f)  $Ar \rightarrow Re. Z_{low}$  and  $Z_{high}$  represent the channels with the 128 smallest  $UTR_D^i(h_s)$  and the 128 largest  $UTR_D^i(h_s)$  respectively.

analyse the internal transferability of the source model, the channels of source encoder with lower  $UTR_D$  is more transferable to the target domain, which allows us to leverage the source knowledge more efficiently and safely, which proves our motivation.

The effectiveness of  $UTR_I$ . The  $UTR_I$  identifies the instance-level reliability of inferring target semantics using the source model. To evaluate its effectiveness, we draw the Accuracy- $UTR_I$  curve in Fig. 5, which describes the relationship between the  $UTR_I$  of different target samples and the prediction accuracy of the source model to these samples. It can be seen that the source model is more accurate to samples with small  $UTR_I$ . And the prediction accuracy tends to decrease with the increase of the  $UTR_I$ . This phenomenon demonstrates the effectiveness of  $UTR_I$  to identify the instance-level risk of inferring target semantic labels.

**Extension to other layers.** In our previous experiments, UTR is calculated using the last layer output of the feature extractor (the FC layer). Here, we explore the feasibility of extending the UTR to channels of other layers. To this end, we use



Fig. 8. The Grad-CAM [55] visualization of the features of channels with the smallest  $UTR_D$  on the source and target domains.



Fig. 9. The Grad-CAM [55] visualization of the features of channels with the largest  $UTR_D$  (b) on the source and target domains.

the average pooling to extract features of different channels  $z \in \mathbb{R}^{2048}$  from the last, penultimate, and antepenultimate bottleneck of the ResNet-50 backbone, respectively. Then we calculate the UTR of these channels and evaluate the effectiveness of  $UTR_D$  and  $UTR_I$ . For the  $UTR_D$ , the evaluation method is similar to the previous one: that is, the feature z is divided into two 1024 channels vectors  $Z_{low}$ and  $Z_{high}$  according to  $UTR_D(h_s)$ , and their corresponding angles are compared to evaluate their transferability. The results are shown in Fig. 7. We can see that the corresponding angle of  $Z_{low}$  with lower  $UTR_D$ , is larger than  $Z_{hiqh}$  with higher  $UTR_D$ . Therefore, the  $UTR_D$  is effective at the last, penultimate, and antepenultimate bottlenecks of Resnet-50 as well. The similar trends among multiple layers' features prove that our transferable index has the potential to extend to the feature representation of other layers. For the  $UTR_I$ , the Accuracy- $UTR_I$  curve is shown in Fig. 6 (a). It can be seen that the  $UTR_I$  is satisfying in the last bottleneck of the ResNet-50 backbone. On the penultimate, and antepenultimate bottlenecks, it may be invalid, such as when  $\tau = 2.5$  but is effective overall.

Extension to other network architectures. In this section, we evaluate the effectiveness of UTR on different backbone models including the VGG16 [56] and AlexNet [57]. The experiments are conducted on Office-Home task  $Ar \rightarrow Cl$ . The



Fig. 10. The t-SNE visualizations of features of  $Z_{low}$  and  $Z_{high}$  on the target domain (Cl) of the Office-Home task Ar $\rightarrow$ Cl. For a better illustration, we choose features in the first 6 classes, and different color denotes different class. Best viewed in colors.



Fig. 11. The t-SNE visualizations of different methods on the target domain (Ar) of the Office-Home task Cl $\rightarrow$ Ar, including: the source model, Shot, NRC and Ours. For a better illustration, we choose features in the first 6 classes, and different color denotes different class. Best viewed in colors.

TABLE IX Comparison with existing transferability measurements with different model structures on office-home tasks  $AR \rightarrow CL$ . The  $\uparrow/\downarrow$  indicates the larger/smaller the value, the higher the transferability.

Measurement	MMD↓	A-Distance↓	CA↑	LogME↑	LEEP↑	NCE↑
VGG16 Z <sub>low</sub>	0.60	1.36	-0.01	0.81	-3.61	-2.36
VGG16 $Z_{high}$	0.64	1.44	-0.21	0.80	-3.63	-2.43
AlexNet $Z_{low}$	0.51	1.31	0.50	0.76	-3.80	-2.88
AlexNet $Z_{high}$	0.57	1.28	0.25	0.75	-3.85	-2.97

results of  $UTR_D$  are reported in Table IX. The results of  $UTR_I$  is shown in Fig. 6 (b). It can be seen that using two different backbone architectures, the  $UTR_D$  is consistent with the most recent transferability measurements. In addition, the  $UTR_I$  is able to reveal the target semantics risk, which demonstrates that our UTR method is able to apply to different model architectures.

**Extension to the target model.** We have investigated the effectiveness of our UTR on the source model. In this subsection, we evaluate it on the target model in the adaptation process. The results of  $UTR_D$  and  $UTR_I$  are shown in Fig. 7 (d)-(f) and 6 (c), respectively. From Fig. 7 (d)-(f), we can observe that the  $UTR_D$  is also effective for the target model in the first few steps of adaptation.

To be specific, we can see that in the first 5 steps, the  $Z_{low}$  has a larger Corresponding Angle between the source and the target domain than  $Z_{high}$ , which indicates that it is more transferable than  $Z_{high}$ . However, it can be seen that after training for a period, it is inadequate to use the  $UTR_D$  for identifying the target model. For example, in the epoch 10/30 of Fig. 7 (d), the corresponding angle of  $Z_{low}$  is lower than that of  $Z_{high}$ .

The same phenomenon can be observed for  $UTR_I$ . From Fig. 6 (c), we can observe that the  $UTR_I$  is effective in the epoch 0 and epoch 5, but became invalid in the epoch 15. The main reason may be that after a period of training, the model gradually adapts to the target domain. Thus, it no longer needs or even actively abandons the source knowledge. It is worth noting that if the target model does not fit the source domain well, the model uncertainty in the Equation 2 can not be ignored. Therefore the UD may not be calculated without accessing to the source data.

Calculating the  $UTR_D$  stochastically. In our implementation, the calculation of  $UTR_D$  requires to feed-forward all target samples. As a statistical measurement,  $UTR_D$  can also be adapted to the online version, where  $UTR_D$  is updated using the moving average method widely used in Batch Normalization [58]. We conducted new experiments using the moving-average calculation of  $UTR_D$ , with their results denoted as "Ours-Online". We set the momentum to 0.1 and conduct the experiment on three office-home tasks: Ar $\rightarrow$ Cl, Ar $\rightarrow$ Pr, and Ar $\rightarrow$ Re. The results in Table IV show that the performances of the online version on the three tasks are 59.2%, 81.2% and 83.1%, respectively. These are very similar to the original version "Ours".

Uncertainty Estimation. We evaluate the performances of using different uncertainty implementation methods to calculate  $UTR_D$ , i.e., the M(.) in Equation 5, including the sensitivity analysis [41] and Deep Ensembles [42] and Monte Carlo dropout [43], denoted as "Ours-Ensemble" and "Ours-MC drooout", respectively. Table IV shows that our method is not sensitive to various uncertainty implementation methods. Visualization. Fig. 8 and Fig. 9 illustrate the Grad-CAM [55] feature visualization of a source model on the source and target data. Fig. 8 visualizes the feature of the most transferable channel selected by our proposed  $UTR_D$  (i.e., with the smallest  $UTR_D$ ). Fig. 8 shows the feature of the most non-transferable one (i.e., with the largest  $UTR_D$ ). It can be observed that the feature in Fig. 8 captures the semantic information "screen" on the source domain and it remains the same semantic information on the target domain, which indicates that it is transferable to the target domain. However, the feature in Fig. 8 seems to focus on "keyboard" in the source domain, but fails to capture the same semantics on the target data, which suggests it is non-transferable to the target domain.

Fig. 10 shows the t-SNE [59] visualizations of the features of  $Z_{low}$  (128 channels' features with low  $UTR_D$ ) and  $Z_{high}$ (128 channels' features with high  $UTR_D$ ) on the task Cl $\rightarrow$ Al. We can see that the semantic information extracted by  $Z_{low}$ in the target domain is more discriminative than  $Z_{high}$ . It qualitatively proves that it is more transferable to the target domain. The above phenomenons demonstrate that  $UTR_D$  is effective to estimate the transferability of the knowledge in the source encoder.

By estimating the transferability of different channels of the source encoder, our method can incorporate more valuable knowledge into the target domain to learn a more discriminative target model. To prove it, we provide the t-SNE visualizations of the feature obtained by the original source model, SHOT, NRC and our method on the task  $Cl \rightarrow Al$  in Fig. 11. As expected, the feature extracted by our method is more semantically discriminative.

#### VII. METHOD LIMITATION

In this paper, we propose the Uncertainty-induced Transferability Representation (UTR) to explore the transferability of the source model in the absence of source data and target annotations. We prove the effectiveness and universality of the domain-level UTR and the instance-level UTR, which help the SFUDA community leverage the knowledge of the source model and target data fully and safely. However, it also has the following two limitations.

First, we use the distributional uncertainty to approximate the implicit uncertainty distance, which assumes that the model uncertainty is small enough to be ignored. As we discussed in Section VI-D: "Extension to the target model", because the model uncertainty represents how much the pre-trained model covers the training distribution, the assumption may be violated somehow with the model gradually adapted to the target domain. In this paper, it will be our future work to quantify when the previous assumption is violated.

Second, we demonstrate the consistency of  $UTR_D$  with existing domain discrepancy measurements. However, at present it is only a way to analyse the transferability, but not a rigorous domain distribution divergence yet that can be explicitly optimized, such as MMD and A-Distance. We hope that future research will address this limitation.

#### VIII. CONCLUSIONS

In this paper, we develop a novel measurement termed Uncertainty-induced Transferability Representation (UTR) which uses uncertainty distance as a tool to estimate transferability in the absence of source data and target annotations. The domain-level UTR describes how transferable each source feature dimension is to the target domain, and the instancelevel UTR identifies the reliability of the inferred target semantics. Based on the UTR, we propose a novel Calibrated Adaption Framework (CAF) for SFUDA, including a source knowledge calibration module to control the target model to learn transferable knowledge and discard non-transferable one, and a target semantics calibration module calibrates the target semantics. The calibrated source knowledge and target semantics help the target model fully and safely leverage the source knowledge and target data, ultimately prompting to better adapt to the target domain. We verified the effectiveness of our method using experimental results and demonstrated that the proposed method achieves state-of-the-art performances on three SFUDA benchmarks.

#### REFERENCES

- S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, "Exploiting the intrinsic neighborhood structure for source-free domain adaptation," *arXiv preprint arXiv:2110.04202*, 2021.
- [2] —, "Generalized source-free domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8978–8987.
- [3] H. Xia, H. Zhao, and Z. Ding, "Adaptive adversarial network for sourcefree domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9010–9019.
- [4] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, "Unsupervised domain adaptation without source data by casting a bait," arXiv preprint arXiv:2010.12427, 2020.
- [5] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6028–6039.
- [6] F. Wang, Z. Han, Y. Gong, and Y. Yin, "Exploring domain-invariant parameters for source free domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7151–7160.
- [7] C. Nguyen, T. Hassner, M. Seeger, and C. Archambeau, "Leep: A new measure to evaluate transferability of learned representations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7294–7305.

- [8] K. You, Y. Liu, J. Wang, and M. Long, "Logme: Practical assessment of pre-trained models for transfer learning," in *International Conference* on Machine Learning. PMLR, 2021, pp. 12133–12143.
- [9] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.
- [10] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," *Advances in neural information processing systems*, vol. 19, pp. 513–520, 2006.
- [11] 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," *arXiv preprint arXiv:2107.03342*, 2021.
- [12] J. Nandy, W. Hsu, and M. L. Lee, "Towards maximizing the representation gap between in-domain & out-of-distribution examples," arXiv preprint arXiv:2010.10474, 2020.
- [13] J. Gao, Y. Hua, G. Hu, C. Wang, and N. M. Robertson, "Reducing distributional uncertainty by mutual information maximisation and transferable feature learning," in *European Conference on Computer Vision*. Springer, 2020, pp. 587–605.
- [14] A. Malinin and M. J. Gales, "Predictive uncertainty estimation via prior networks," in *NeurIPS*, 2018.
- [15] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," arXiv preprint arXiv:1705.10667, 2017.
- [16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [17] S. Chhabra, P. B. Dutta, B. Li, and H. Venkateswara, "Glocal alignment for unsupervised domain adaptation," in *Multimedia Understanding with Less Labeling on Multimedia Understanding with Less Labeling*, 2021, pp. 45–51.
- [18] S. Chhabra, H. Venkateswara, and B. Li, "Iterative image translation for unsupervised domain adaptation," in *Multimedia Understanding with Less Labeling on Multimedia Understanding with Less Labeling*, 2021, pp. 37–44.
- [19] Z. Han, H. Sun, and Y. Yin, "Learning transferable parameters for unsupervised domain adaptation," *IEEE Transactions on Image Processing*, 2022.
- [20] J. Moon, D. Das, and C. G. Lee, "A multi-stage framework with mean subspace computation and recursive feedback for online unsupervised domain adaptation," *IEEE Transactions on Image Processing*, 2022.
- [21] Z. Deng, K. Zhou, D. Li, J. He, Y.-Z. Song, and T. Xiang, "Dynamic instance domain adaptation," arXiv preprint arXiv:2203.05028, 2022.
- [22] W. Deng, Q. Liao, L. Zhao, D. Guo, G. Kuang, D. Hu, and L. Liu, "Joint clustering and discriminative feature alignment for unsupervised domain adaptation," *IEEE Transactions on Image Processing*, vol. 30, pp. 7842–7855, 2021.
- [23] H. Xu, M. Yang, L. Deng, Y. Qian, and C. Wang, "Neutral cross-entropy loss based unsupervised domain adaptation for semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 4516–4525, 2021.
- [24] M. Ye, J. Zhang, J. Ouyang, and D. Yuan, "Source data-free unsupervised domain adaptation for semantic segmentation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2233– 2242.
- [25] B. Yang, H.-W. Yeh, T. Harada, and P. C. Yuen, "Model-induced generalization error bound for information-theoretic representation learning in source-data-free unsupervised domain adaptation," *IEEE Transactions* on *Image Processing*, vol. 31, pp. 419–432, 2021.
- [26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
- [27] 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.
- [28] 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.
- [29] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.
- [30] X. Li, J. Li, L. Zhu, G. Wang, and Z. Huang, "Imbalanced sourcefree domain adaptation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3330–3339.

- [31] Z. Qiu, Y. Zhang, H. Lin, S. Niu, Y. Liu, Q. Du, and M. Tan, "Sourcefree domain adaptation via avatar prototype generation and adaptation," *arXiv preprint arXiv:2106.15326*, 2021.
- [32] N. Kwon, H. Na, G. Huang, and S. Lacoste-Julien, "Repurposing pretrained models for robust out-of-domain few-shot learning," arXiv preprint arXiv:2103.09027, 2021.
- [33] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" arXiv preprint arXiv:1703.04977, 2017.
- [34] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *arXiv preprint arXiv:1806.01768*, 2018.
- [35] A. Sedlmeier, T. Gabor, T. Phan, L. Belzner, and C. Linnhoff-Popien, "Uncertainty-based out-of-distribution classification in deep reinforcement learning," arXiv preprint arXiv:2001.00496, 2019.
- [36] S. Padhy, Z. Nado, J. Ren, J. Liu, J. Snoek, and B. Lakshminarayanan, "Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks," *arXiv preprint* arXiv:2007.05134, 2020.
- [37] R. McAllister, G. Kahn, J. Clune, and S. Levine, "Robustness to outof-distribution inputs via task-aware generative uncertainty," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 2083–2089.
- [38] J. Liang, R. He, Z. Sun, and T. Tan, "Exploring uncertainty in pseudolabel guided unsupervised domain adaptation," *Pattern Recognition*, vol. 96, p. 106996, 2019.
- [39] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *International conference on machine learning*. PMLR, 2019, pp. 1081–1090.
- [40] A. T. Tran, C. V. Nguyen, and T. Hassner, "Transferability and hardness of supervised classification tasks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1395–1405.
- [41] Z. K. Nagy and R. Braatz, "Distributional uncertainty analysis using power series and polynomial chaos expansions," *Journal of Process Control*, vol. 17, no. 3, pp. 229–240, 2007.
- [42] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," arXiv preprint arXiv:1612.01474, 2016.
- [43] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.
- [44] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.
- [45] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.
- [46] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "Visda: The visual domain adaptation challenge," arXiv preprint arXiv:1710.06924, 2017.
- [47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision* and pattern recognition, 2016, pp. 770–778.
- [49] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," arXiv preprint arXiv:1710.09412, 2017.
- [50] H.-W. Yeh, B. Yang, P. C. Yuen, and T. Harada, "Sofa: Source-data-free feature alignment for unsupervised domain adaptation," in *Proceedings* of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 474–483.
- [51] Y. Kim, D. Cho, K. Han, P. Panda, and S. Hong, "Domain adaptation without source data," *IEEE Transactions on Artificial Intelligence*, 2021.
- [52] J. Huang, D. Guan, A. Xiao, and S. Lu, "Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3635–3649, 2021.
- [53] J. Li, Z. Du, L. Zhu, Z. Ding, K. Lu, and H. T. Shen, "Divergenceagnostic unsupervised domain adaptation by adversarial attacks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [54] Y. Du, H. Yang, M. Chen, J. Jiang, H. Luo, and C. Wang, "Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation," *arXiv preprint arXiv:2109.04015*, 2021.

- [55] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 618-626.
- [56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
- [57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, 2012. [58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep
- network training by reducing internal covariate shift," in International *conference on machine learning.* PMLR, 2015, pp. 448–456. [59] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal*
- of machine learning research, vol. 9, no. 11, 2008.