Applied Intelligence
https://doi.org/10.1007/s10489-022-04364-9

Self-training transformer for source-free domain adaptation

Guanglei Yang[1] · Zhun Zhong[2] · Mingli Ding[1] · Nicu Sebe[2] · Elisa Ricci[2]

Accepted: 24 November 2022

Abstract

In this paper, we study the task of source-free domain adaptation (SFDA), where the source data are not available during target adaptation. Previous works on SFDA mainly focus on aligning the cross-domain distributions. However, they ignore the generalization ability of the pretrained source model, which largely influences the initial target outputs that are vital to the target adaptation stage. To address this, we make the interesting observation that the model accuracy is highly correlated with whether attention is focused on the objects in an image. To this end, we propose a generic and effective framework based on Transformer, named TransDA, for learning a generalized model for SFDA. First, we apply the Transformer blocks as the attention module and inject it into a convolutional network. By doing so, the model is encouraged to turn attention towards the object regions, which can effectively improve the model's generalization ability on unseen target domains. Second, a novel self-supervised knowledge distillation approach is proposed to adapt the Transformer with target pseudo-labels, further encouraging the network to focus on the object regions. Extensive experiments conducted on three domain adaptation tasks, including closed-set, partial-set, and open-set adaption, demonstrate that TransDA can significantly improve the accuracy over the source model and can produce state-of-the-art results on all settings. The source code and pretrained models are publicly available at: https://github.com/ygjwd12345/TransDA.

Keywords Transformer · Source-free · Domain adaption

1 Introduction

Deep learning has enabled several advances in various computer vision tasks, such as image classification, object detection and semantic segmentation. However, deep models suffer from significant performance degradation when applied to an unseen target domain due to the well-documented domain shift problem. To solve this problem, domain adaptation was introduced, aiming to transfer knowledge from a fully labeled source domain to a target domain [1–3]. A common strategy in domain adaptation is to align the feature distributions between the source and target domains by minimizing the domain shift through various metrics. These metrics include Maximum Mean Discrepancy [4, 5], Sliced Wasserstein Discrepancy [6], and Enhanced Transport Distance [7]. Another popular paradigm leverages the idea of adversarial learning to minimize cross-domain discrepancy [8–12].

Despite the success of current domain adaptation methods, they work under the strict condition that the source data are always available during training. However, this condition has two drawbacks that hinder the application of these methods. First, the source datasets, such as VisDAand GTAV, usually are large and thus require high saving and loading costs. This restricts their usage on specific platforms, especially portable devices. Second, fully accessing the source data may violate data privacy

✉ Zhun Zhong
zhun.zhong@unitn.it

Guanglei Yang
yangguanglei@hit.edu.cn

Mingli Ding
dingml@hit.edu.cn

Nicu Sebe
niculae.sebe@unitn.it

Elisa Ricci
e.ricci@unitn.it

[1] School of Instrument Science and Engineering, Harbin Institute of Technology (HIT), Harbin, China

[2] Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

Published online: 11 December 2022

policies. Considering these two factors, companies or organizations prefer to provide the learned models rather than the data. Therefore, designing a domain adaptation method without requiring source datasets has great practical value. To this end, in this paper, we aim to address the recently introduced problem of source-free domain adaptation [13] (SFDA), in which only the model pretrained on the source and the unlabeled target dataset are provided for target adaptation.

Today, amount of works [9, 13, 14] have been proposed for SFDA, aiming to align the source and target distributions by learning with underlying constraints, such as information maximization and label consistency. However, all of these methods perform adaptation with a model pretrained on the source data, while neglecting the source model's generalization ability. In SFDA, the adaptation process largely relies on the accuracy of the source model on the target domain. Without a source model that generalizes well, the generated pseudo-labels may contain significant noise, and learning with them will undoubtedly harm the model's performance.

In this paper, we attempt to improve the generalization ability of the source model for SFDA. Different from the existing out-of-domain generalization methods [15, 16], which aim to improve the model generalization by augmenting the diversity of the source samples, in this paper, we introduce a new perspective for building a robust source model motivated by the following observation. In Fig. 1, we directly apply the source model to the unseen target samples (A→W and A→D on Office-31) and produce the heatmaps by Grad-CAM. We use Amazon Mechanical Turk and ask annotators to label the samples with "focused / non-focused" according to whether the heatmap (red region) is localized on the object. Examples are shown in Fig. 1(a). We observe that the accuracy of the focused samples is much higher than that of the non-focused samples (see Fig. 1(b, c)). The indicates that when a network can effectively focus on the objects in the images, it commonly can provide a high prediction accuracy on these images. This finding is intuitive, since the model can capture more informative and domain-invariant feature when the object in the image is focused by the model. Otherwise, the model will tend to capture unhelpful information of noise and background, which may hamper the performance.

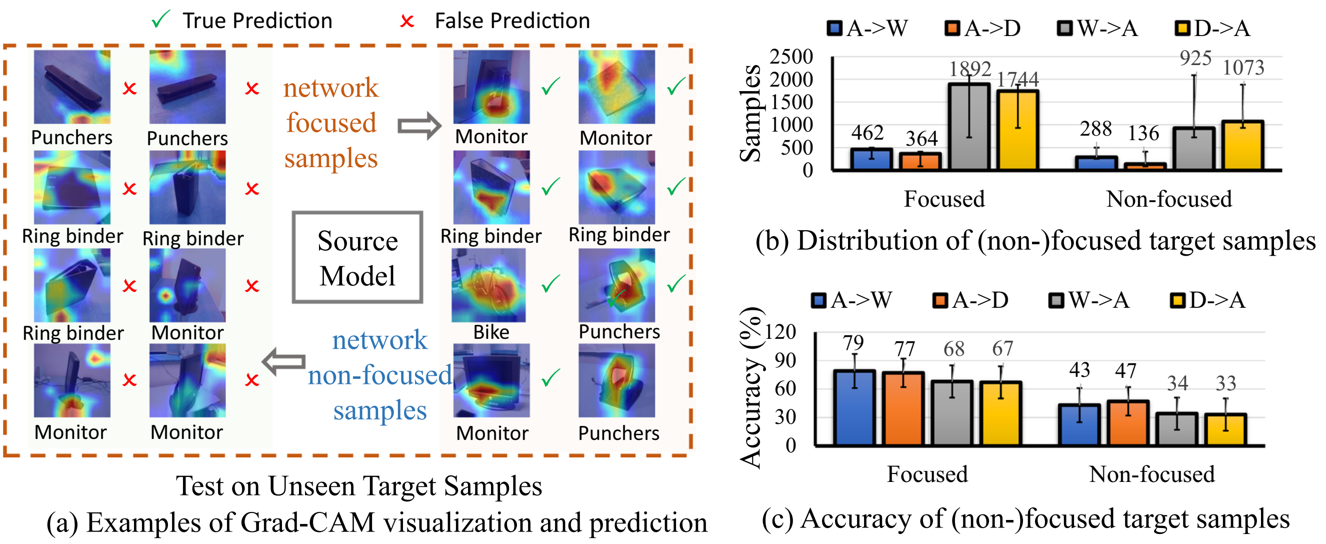
Inspired by the above observation, we propose TransDA for SDFA by equipping a convolutional model with a Transformer [17] module, which can effectively encourage the network to focus on the objects and thus improve the performance on target samples. Specifically, by injecting the Transformer after the last convolutional layer of ResNet-50, we can leverage its long-range dependence to force the model to pay more attention to the objects. In this way, the model can produce more informative representation captured on the objects, which can significantly improve the generalization ability of the source model. In addition, during the target adaptation, we propose a self-supervised knowledge distillation on generated pseudo-labels, further leading the Transformer to learn to focus on the objects of target samples. We evaluate our TransDA on three domain adaptation tasks, including closed-set, partial-set [18], and open-set [19] domain adaptation. Extensive results demonstrate that TransDA is competitive with state-of-the-art methods on all tasks.

To summarize, this work provides the following three contributions:

- We reveal for the first time the importance of network attention for SFDA, through an in-depth empirical study. This provides a new perspective for improving the generalization ability of the source model.
- We propose a Transformer-based network for SFDA, which can effectively lead the model to pay attention to the objects and thus significantly increase the model generalization ability. To our knowledge, we are the first to propose a Transformer for solving the domain adaptation task.
- We introduce a novel self-supervised knowledge distillation approach to further help the Transformer to focus on target objects.

2 Related work

Traditional domain adaptation Domain adaptation aims to improve moedel's performance on the target domain by using a labeled source domain that belongs to a different distribution. With the recent advancement in deep convolutional neural networks, a number of methods have been proposed for unsupervised domain adaptation. One common solution in the previous works is to guide the model to learn a domain-invariant representation by minimizing the domain discrepancy [5, 7, 20, 21]. For example, CAN [5] optimizes the network by considering the discrepancy between the intra- and inter-class domains. In a similar way, ETD [7] employs an enhanced transport distance to reflect the discriminative information. Different with learning a domain-invariant representation, several methods have focused on the feature discrimination ability during domain adaptation. They introduce different normalization or penalization strategies to boost the feature discrimination ability on the target domain, such as batch normalization [22], batch spectral penalization [23], batch nuclear-norm maximization [24], and transferable normalization [25]. Another branch of



(a) Examples of Grad-CAM visualization and prediction

(b) Distribution of (non-)focused target samples

(c) Accuracy of (non-)focused target samples

Fig. 1 Evaluation of attention for the source model on the Office-31 dataset. (a) Examples of Grad-CAM visualization and prediction results in A→W. (b) The distribution and (c) accuracy of the source model on focused and non-focused samples in A→W , A→D, W→A and D→A

methods exploit a generative adversarial network to address the domain confusion [2]. BDG [8] and GVB-GD [26] use bi-directional generation to construct domain-invariant representations. Despite the large success of the above methods, they typically require access to source data when learning from the target domain. This may lead to personal information leakage and thus violate data privacy requirements.

Source-free domain adaptation To alleviate the issue of data privacy, recent works [9, 13, 14] have turned their attention to the problem of source-free domain adaptation, where only the pretrained source model and the target samples are provided during the target adaptation stage. C3-GAN [9] introduces a collaborative class conditional generative adversarial net to produce target-style training samples. Clustering-based regularization is also used to obtain more discriminative features in the target domain. Meanwhile, SHOT [13] freezes the classifier module and learns the target-specific feature extractor with information maximization and self-supervised pseudo-labeling, which can align the target feature distribution with the source domain. DECISION [14] extends SHOT to a multi-source setting by learning different weights for each source model. Different from the above methods, in this paper, we attempt to improve the generalization ability of the source model by injecting a Transformer module into the network. Our approach can be readily incorporated into most existing frameworks to boost the adaptation accuracy.

Vision transformers The Transformer was first proposed by [27] for machine translation and has been used to establish state-of-the-art results in many natural language processing tasks. Recently, the Vision Transformer (ViT) [17] achieved state-of-the-art results on the image classification task by directly applying Transformers with global self-attention on full-sized images. Since then, Transformer-based approaches have been shown to be efficient in many computer vision tasks, including object detection [28, 29], image segmentation [30], video inpainting [31], video classification [32], pose estimation [33], object re-identification [34] and depth estimation [35]. Different from these approaches, in this paper, we adopt a Transformer-based network to address the source-free domain adaptation task. To this end, we propose a generic yet straightforward TransDA framework for domain adaptation to encourage the model to focus on the object regions, leading to improved domain generalization.

3 Method

3.1 Problem definition

In traditional domain adaptation (DA), the models are trained on an unlabeled target domain $X_t = \{x_t^i\}_{i=1}^M$ and a source domain $X_s = \{x_s^j\}_{j=1}^N$ along with corresponding labels $Y_s = \{y_s^j\}_{j=1}^N$, where y_s^j belongs to the set of K classes. The distributions of the source and target data are denoted as $x_s \sim p_{data}(x_s)$ and $x_t \sim p_{data}(x_t)$, respectively, where $p_{data}(x_t) \neq p_{data}(x_s)$. The goal is to learn a target network ϕ_t using the labeled source data and unlabeled target data that can accurately recognize the target samples.

In SFDA, (1) the labeled source data are only used to pretrain a source model ϕ_s, and (2) the target network ϕ_t is learned with the pretrained source model ϕ_s and the unlabeled target data X_t.

3.2 Overview

The overall framework of the proposed method, which consists of (1) source training and (2) target adaptation, is shown in Fig. 2. First, we train the source model ϕ_s with samples from the source domain using cross-entropy loss. The source model ϕ_s is composed of two modules: the feature extractor $f_s{:}\mathcal{X}\rightarrow\mathcal{R}^d$ and the classifier $g_s{:}\mathcal{R}^d\rightarrow\mathcal{R}^K$, where d indicates the dimension of the feature extractor output and K refers to the number of categories in the source data. Therefore, we have $\phi_s{=}f_s{\circ}g_s$. Different from existing methods that use a convolutional neural network (CNN) as the feature extractor (e.g., ResNet), we propose injecting a Transformer module [17] after the last CNN layer. With the help of the innate self-attention mechanisms injected in Transformer, the model is encouraged to capture foreground regions instead of background factors. As a result, the feature extractor can pay more attention to the objects and thus produce a more robust representation.

Second, we aim to learn a network ϕ_t in the target domain, given the pretrained ϕ_s. Specifically, we maintain a teacher model ϕ_t^{Tea} and a student model ϕ_t^{Stu}, which are both initialized by the parameters of ϕ_s. Following [13], we fix the classifiers and only update the feature extractors. The updating strategies for the feature extractors are different. We update f_t^{Stu} with the gradients produced by the target adaptation losses. f_t^{Tea} is updated with an exponential moving average of the parameters of f_t^{Stu}. The teacher model ϕ_t^{Tea} is used to produce a hard pseudo-label $\hat{y}_t$ and soft pseudo-label $\bar{y}_t$ for calculating the self-labeling loss and knowledge distillation loss, respectively. In addition, the information maximization loss is computed with the outputs of ϕ_t^{Stu}. The above three losses are used to update f_t^{Stu}, where the information maximization and self-labeling losses are employed to align the cross-domain feature distributions and the knowledge distillation loss is designed to force the model to focus on objects.

3.3 Model pretraining on the source with a transformer

In this stage, we aim to learn a source model with the labeled source data. Generally, given a network ϕ_s initialized on ImageNet, we train it with the label smoothing cross-entropy loss [36]:

$$\mathcal{L}_{ce} = -\mathbb{E}_{x_s \in X_s} \sum_{k=1}^{K} \hat{y}_k \log \sigma(\phi_s(x_s)), \tag{1}$$

where $\hat{y}_k{=}(1-\alpha)\ y_k{+}\alpha/K$. α is the smoothing factor and is empirically set to 0.1. $\sigma(.)$ is the softmax operation.

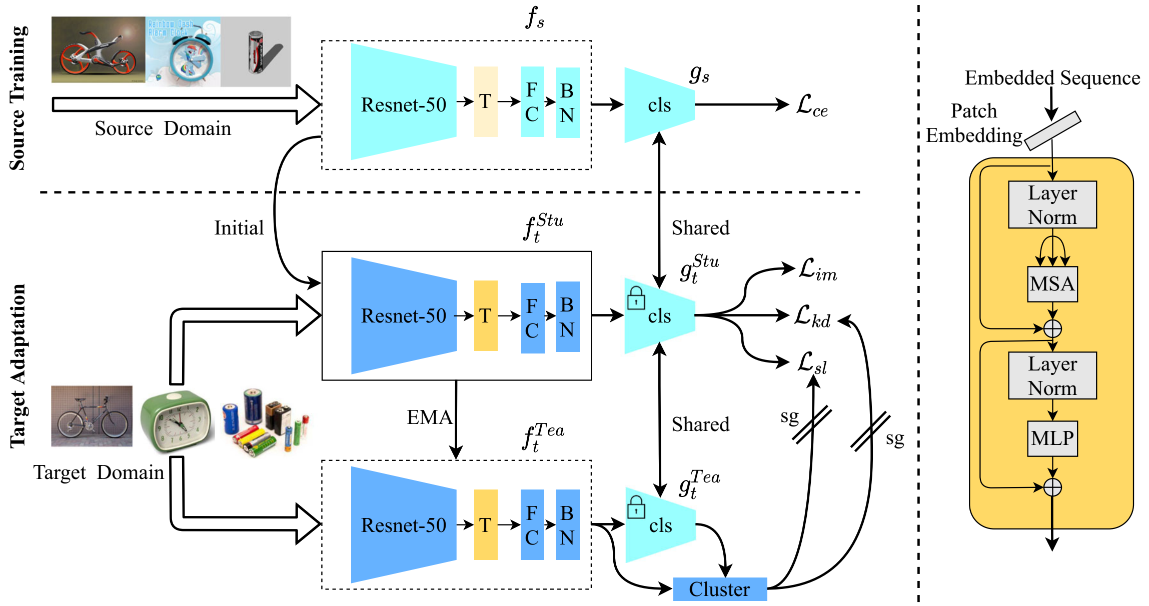


Fig. 2 Overview of the proposed method. The model includes a feature extractor f and a classifier g. We inject a Transformer module (structure is shown on the right) into f to obtain a representation with improved generalization capability. First, we train the model with the labeled source data. Then, we create a teacher model (Tea) and a student (Stu), cloned from the source model, for target adaptation. The teacher model is used to produce pseudo-labels for computing the self-labeling loss ($\mathcal{L}_{sl}$) and knowledge distillation loss ($\mathcal{L}_{kd}$). We also calculate the information maximization loss ($\mathcal{L}_{im}$) based on the outputs of the student model: *sg*: stop gradient, *EMA*: exponential moving average, *FC*: fully connected layer, *BN*: batch normalization layer

In SFDA, the target adaptation stage greatly relies on the target outputs obtained by the source model. Hence, it is important to learn a robust source model, to counteract with domain bias. As shown in Fig. 1, the model can effectively classify the target samples if it can localize the object regions. This observation is reasonable because the model learns to capture the common object patterns instead of domain-specific information (e.g., background, color distribution) if it can always focus on the objects. Therefore, one solution for improving the generalization ability is forcing the model to focus on objects during training. Existing approaches typically select a CNN model as the feature extractor of ϕ_s. However, due to the intrinsic locality of the convolution operation (i.e., small receptive fields), the CNN model prefers to capture local information, which may lead it to overfit on domain-specific information and thus fail to focus on objects, especially when encountering a large domain shift.

To address the drawback of CNNs in SFDA, in this paper, we propose injecting a Transformer module after the convolutional network [17]. By doing so, we can leverage the property of the Transformer to capture long-range dependencies and explicitly encourage the model to pay more attention to the objects. This enables us to reduce the impact of domain-specific information and produce more robust and transferable representations.

Specifically, as shown in the right part of Fig. 2 we construct the feature extractor by injecting the Transformer after the last convolutional layer of ResNet-50. Since the input of the Transformer should be a sequence, we first reshape the output of the ResNet-50 backbone $F \in \mathcal{R}^{h\times w\times \hat{d}}$ to $\hat{F} \in \mathcal{R}^{u\times \hat{d}}$, where h, w, and $\hat{d}$ indicate the height, width, and dimension of F, respectively. u is the product of h and w. Then, $\hat{F}$ is regarded as the input sequence with patch embeddings for the Transformer.

In the first layer of the Transformer, we map the dimension of the patch embeddings $\hat{F}$ from $\hat{d}$ to $\bar{d}$ with a linear projection layer, producing $Z_0 \in \mathcal{R}^{u\times \bar{d}}$. Then, Z_0 is fed into L Transformer layers, which include multi-headed self-attention (MSA) and multi-layer perceptron (MLP) blocks.

Given the feature Z_{l-1} obtained from the $l-1$-th Transformer layer, MSA (Z_{l-1}) is defined as:

$$\begin{aligned}\mathrm{MSA}(Z_{l-1}) \;=\; & Z_{l-1} + \mathrm{cont}(\mathrm{AH}_1(\mathrm{LN}(Z_{l-1}));\\ & \mathrm{AH}_2(\mathrm{LN}(Z_{l-1})); \cdots ; \mathrm{AH}_m(\mathrm{LN}(Z_{l-1}))) \times W, \end{aligned} \tag{2}$$

where AH represents a self-attention head [27], cont$(\cdot)$) is the concentration operation, LN$(\cdot)$ is the layer normalization, and m indicates the number of self-attention heads. $W \in \mathbb{R}^{m\cdot \check{d}\times \bar{d}}$ are the learnable weight matrices, where $\check{d}$ is the output dimension of each AH.

The output of MSA is then transformed into an MLP block with a residual skip, formulated as:

$$Z_l = \mathrm{MLP}(\mathrm{LN}(\mathrm{MSA}(Z_{l-1}))) + \mathrm{MSA}(Z_{l-1}). \tag{3}$$

Given the output $Z_l \in \mathcal{R}^{u\times \bar{d}}$ of the Transformer, we obtain the global feature by average pooling, which is fed into the following layers, including one FC layer, one BN layer, and the classifier g_s.

3.4 Self-training on target with transformer

Information maximization In the target adaptation stage, we are given a pretrained source model ϕ_s and the unlabeled target domain. We first initialize the target model ϕ_t with the parameters ϕ_s. Following [13], we fix the classifier g_t to maintain the class distribution information of the source domain and update the feature extractor f_t using the information maximization (IM) loss [37]. This enables us to reduce the feature distribution gap between the source and target domains. The IM loss consists of a conditional entropy term and a diversity term:

$$\mathcal{L}_{im} = -\mathbb{E}_{x_t\in X_t} \sum_{k=1}^{K} \sigma(\phi_t(x_t)) \log \sigma(\phi_t(x_t)) + \sum_{k=1}^{K} \bar{p}_k \log \bar{p}_k, \tag{4}$$

where $\bar{p} = \mathbb{E}_{x_t\in X_t}[\sigma(\phi_t(x_t))]$ is the mean of the softmax outputs for the current batch.

Self-labeling Although the IM loss can make the predictions on the target domain more confident and globally diverse, it is inevitably affected by the noise generated by incorrect label matching. To address this issue, one solution is to utilize a self-labeling strategy to further constrain the model. In this paper, we use self-supervised clustering [9, 13, 14] to generate pseudo-labels for target samples. Specifically, we first compute the centroid for each class in the target domain similar to weighted k-mean clustering,

$$\mu_k^{(0)} = \frac{\sum_{x_t\in X_t} \sigma(\phi_t(x_t)) f_t(x_t)}{\sum_{x_t\in X_t} \sigma(\phi_t(x_t))}. \tag{5}$$

Then, the initial pseudo-labels are generated by the nearest centroid classifier:

$$\hat{y}_t = \arg\min_k 1 - \frac{f_t(x_t)\cdot \mu_k^{(0)}}{||f_t(x_t)||_2 ||\mu_k^{(0)}||_2}, \tag{6}$$

where $||*||_2$ denotes the $L2$-norm. Finally, the class centroids and pseudo-labels are updated as follows:

$$\begin{aligned}\mu_k^{(1)} &= \frac{\sum_{x_t\in X_t} \xi(\hat{y}_t = k) f_t(x_t)}{\sum_{x_t\in X_t} \xi(\hat{y}_t)},\\ \hat{y}_t &= \arg\min_k 1 - \frac{f_t(x_t)\cdot \mu_k^{(1)}}{||f_t(x_t)||_2 ||\mu_k^{(1)}||_2},\end{aligned} \tag{7}$$

where $\xi(*)$ is an indicator that produces 1 when the argument is true. Although the pseudo-labels and centroids can be updated by (7) multiple times, we find that one round of updating is sufficient. Given the generated pseudo-labels, the loss function for self-labeling is calculated using the cross-entropy loss, formulated by:

$$\mathcal{L}_{sl} = -\mathbb{E}_{x_t \in X_t} \sum_{k=1}^{K} \xi(\hat{y}_t = k) \log \sigma(\phi_t(x_t)). \tag{8}$$

Self-knowledge distillation Recall that we aim to encourage the network to focus on the objects to produce more robust feature representations. Although we inject a Transformer module into the model to achieve this goal, we hope to further improve object attention ability by learning with the target samples. The above loss functions ($\mathcal{L}_{im}$ and $\mathcal{L}_{sl}$) are designed to align the feature distribution of the domain, but do not explicitly consider the attention constraint. Therefore, they cannot further improve object attention ability of the Transformer. DINO [38] showed that learning with a self-knowledge distillation strategy can lead the Transformer to capture more semantic information, i.e., pay more attention to objects. Inspired by this observation, we propose adopting the self-knowledge distillation strategy to force the model to turn more attention to objects in the target samples.

Specifically, we employ a teacher model ϕ_t^{Tea} and a student model ϕ_t^{Stu} to implement self-knowledge distillation. We use the teacher model ϕ_t^{Tea} to generate pseudo-labels and optimize the parameters of the student model ϕ_t^{Stu} with training losses. Hence, (5), (6), and (7) are reformulated by replacing f_t and g_t with f_t^{Tea} and g_t^{Tea}, respectively. Similarly, $\mathcal{L}_{im}$ and $\mathcal{L}_{sl}$ are re-formulated by replacing ϕ_t with ϕ_t^{Stu}.

For the self-knowledge distillation, we generate the soft pseudo-labels by

$$\bar{y}_t = \frac{\exp(\delta(f_t^{Tea}(x_t), \mu_k^{(1)})/\tau)}{\sum_{k=1}^{K} \exp(\delta(f_t^{Tea}(x_t), \mu_k^{(1)})/\tau}, \tag{9}$$

where $\delta(a, b)$ indicates the cosine distance between a and b. Then, our knowledge distillation loss is formulated as:

$$\mathcal{L}_{kd} = -\mathbb{E}_{x_t \in X_t} \sum_{k=1}^{K} \bar{y}_t \log \phi_t^{Stu}(x_t). \tag{10}$$

Table 1 Accuracy (%) on Office-31 for closed-set domain adaptation (ResNet-50)

Method	Source-free	A→D	A→W	D→W	W→D	D→A	W→A	Avg
ETD [7]	×	88.0	92.1	100.0	100.0	71.0	67.8	86.2
Hou *et al.* [43]	✓	89.9	91.8	98.7	99.9	73.9	72.0	87.7
BDG [8]	×	93.6	93.6	99.0	100.0	73.2	72.0	88.5
CDAN+BSP [23]	×	93.0	93.3	98.7	100.0	73.6	72.6	88.5
CDAN+BNM [24]	×	92.9	92.8	98.8	100.0	73.5	73.8	88.6
BDCA [44]	×	93.8	94.0	99.0	100.0	73.5	73.0	88.9
f-DAL [45]	×	93.8	95.4	98.8	100.0	74.9	74.2	89.5
CDAN+TransNorm [25]	×	94.0	95.7	98.7	100.0	73.4	74.2	89.3
ILA-DA [46]	×	93.4	95.7	99.3	100.0	72.1	75.4	89.3
NRC [47]	✓	96.0	90.8	99.0	100.0	75.3	75.0	89.4
GVB-GD [26]	×	96.1	93.8	98.8	100.0	74.9	72.8	89.4
GSDA [48]	×	94.8	95.7	99.1	100.0	73.5	74.9	89.7
SHOT [13]	✓	94.0	90.1	98.4	99.9	74.7	74.3	88.6
3C-GAN [9]	✓	92.7	93.7	98.5	99.8	75.3	77.8	89.6
HCL[49]	✓	94.7	92.5	98.2	100.0	75.9	77.7	89.8
D-MCD [50]	✓	94.1	93.5	98.8	100.0	76.4	76.4	89.9
SCDA [51]	×	95.2	94.2	98.7	99.8	75.7	76.2	90.0
A^{2}Net [52]	✓	94.5	94.0	99.2	100.0	76.7	76.1	90.1
RSDA [53]	×	95.2	95.3	99.3	100.0	75.5	76.0	90.2
CAN [5]	×	95.0	94.5	99.1	99.8	78.0	77.0	90.6
TSA [54]	×	95.4	96.0	98.7	100.0	76.7	76.8	90.6
TransDA (Ours)	✓	97.2	95.0	99.3	99.6	73.7	79.3	90.7

Table 2 Accuracy (%) on Office-Home for closed-set domain adaptation (ResNet-50)

Method	Source-free	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ETD [7]	×	51.3	71.9	85.7	57.6	69.2	73.7	57.8	51.2	79.3	70.2	57.5	82.1	67.3
CKB [55]	×	54.7	74.4	77.1	63.7	72.2	71.8	64.1	51.7	78.4	73.1	58.0	82.4	68.5
BDG [8]	×	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
CDAN+BNM [24]	×	56.2	73.7	79.0	63.1	73.6	74.0	62.4	54.8	80.7	72.4	58.9	83.5	69.4
CDAN+TransNorm [25]	×	56.3	74.2	79.0	63.9	73.5	73.1	62.3	55.2	80.3	73.5	58.4	83.3	69.4
BDCA [44]	×	51.8	73.9	80.7	66.3	71.8	74.2	65.3	51.8	81.1	74.7	58.5	84.6	69.6
f-DAL [45]	×	56.7	77.0	81.1	63.1	72.2	75.9	64.5	54.4	81.0	72.3	58.4	83.7	70.0
GVB-GD [26]	×	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
GSDA [48]	×	61.3	76.1	79.4	65.4	73.3	74.3	65.0	53.2	80.0	72.2	60.6	83.1	70.3
SCDA [51]	×	57.5	76.9	80.3	65.7	74.9	74.5	65.5	53.6	79.8	74.5	59.6	83.7	70.5
TCM [56]	×	58.6	74.4	79.6	64.5	74.0	75.1	64.6	56.2	80.9	74.6	60.7	84.7	70.7
RSDA+DANN [53]	×	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
TSA [54]	×	57.6	75.8	80.7	64.3	76.3	75.1	66.7	55.7	81.2	75.7	61.9	83.8	71.2
FGDA+MDD [57]	×	57.1	77.5	81.0	68.4	77.2	75.9	65.8	55.8	81.0	74.3	60.5	83.6	71.5
SHOT [13]	✓	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
NRC [47]	✓	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
D-MCD [50]	✓	59.4	78.9	80.2	67.2	79.3	78.6	65.3	55.6	82.2	73.3	62.8	83.9	72.2
FixBi [58]	×	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
A^{2}Net [52]	✓	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.8
TransDA (Ours)	✓	67.5	83.3	85.9	74.0	83.8	84.4	77.0	68.0	87.0	80.5	69.9	90.0	79.3

Table 3 Accuracy (%) on Digits for closed-set domain adaptation

Method	Source-free	S→M	U→M	M→U	avg
CDAN+E [59]	×	89.2	98.0	95.6	94.3
CyCADA [40]	×	90.4	96.5	95.6	94.2
KL [60]	×	98.2	97.3	92.5	96.0
SWD [6]	×	98.9	97.1	98.1	98.0
SHOT [13]	✓	98.9	98.4	98.0	98.4
TransDA (Ours)	✓	99.1	98.7	98.3	98.7

S: SVHN, M:MNIST, U: USPS

In summary, the final objective of self-training on the target domain is given by

$$\mathcal{L}_{tgt} = \mathcal{L}_{im} + \alpha \mathcal{L}_{sl} + \beta \mathcal{L}_{kd}, \tag{11}$$

where α and β are the weights of self-labeling loss and self-knowledge distillation loss, respectively. We update ϕ_t^{Stu} with $\mathcal{L}_{tgt}$. Additionally, we update ϕ_t^{Tea} with an exponential moving average of the parameters of ϕ_t^{Stu}. Note that the classifiers of ϕ_t^{Tea} and ϕ_t^{Stu} are both fixed during training.

4 Experiments

4.1 Experimental setup

Datasets We conduct experiments on three datasets, including Office-31, Office-Home, VisDA [39] and Digits [40]. Office-31 includes 4,652 images and 31 categories from three domains, *i.e.*, Amazon (A), Webcam (W), and DSLR (D). Office-Home consists of around 15,500 images from 65 categories. It is composed of four domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Rw). VisDA contains 152K synthetic images (regarded as the source domain) and 55K real object images (regarded as the target domain), which are divided into 12 shared classes. Digits contains three subsets: SVHN (S), MNIST (M), and USPS (U). Following the evaluation protocol of CyCADA [40], the three directions, USPS to MNIST (U → M), MNIST to USPS (M → U), and SVHN to MNIST (S → M), are chosen.

Evaluation settings We evaluate the proposed method on three DA settings, including closed-set DA, partial-set DA [41], and open-set DA [42]. Closed-set DA is a standard setting that assumes that the source and target domains share the same class set. Partial-set DA assumes that the target domain belongs to a subclass set of the source domain. In contrast, open-set DA assumes that the target domain includes unknown classes that are absent in the source domain. For closed-set DA, we evaluate our method on all

Table 4 Accuracy (%) on VisDA for closed-set domain adaptation (ResNet-50)

Method	Source-free	airplane	bicycle	bus	car	horse	knife	motorcycle	person	plant	skateboard	train	truck	Avg
KL [60]	×	–	–	–	–	–	–	–	–	–	–	–	–	70.6
MDD [61]	×	–	–	–	–	–	–	–	–	–	–	–	–	74.6
RSDA [53]	×	–	–	–	–	–	–	–	–	–	–	–	–	75.8
GSDA [48]	×	93.1	67.8	83.1	83.4	94.7	93.4	93.4	79.5	93.0	88.8	83.4	36.7	81.5
Hou et al. [43]	✓	94.3	79.0	84.9	63.6	92.6	92.0	88.4	79.1	92.2	79.8	87.6	43.0	81.4
SHOT [13]†	✓	94.5	85.7	77.3	52.2	91.6	15.7	82.6	80.3	87.8	88.0	85.1	58.8	75.0
TSA [54]	×	–	–	–	–	–	–	–	–	–	–	–	–	82.0
TransDA (Ours)	✓	97.2	91.1	81.0	57.5	95.3	93.3	82.7	67.2	92.0	91.8	92.5	54.7	83.0
SWD [6]∗	×	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
3C-GAN [9]∗	✓	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
CGDM [67]∗	×	93.4	82.7	73.2	68.4	92.9	94.5	88.7	82.1	93.4	82.5	86.8	49.2	82.3
STAR [68]∗	×	95.0	84.0	84.6	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
SHOT [13]∗	✓	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

∗ indicates the methods that use ResNet-101 as the backbone. † indicates reproduction using the official code

Table 5 Accuracy (%) on Office-Home for partial-set and open-set domain adaptation (ResNet-50)

Partial-set DA	Source-free	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	P→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
IWAN [69]	×	53.9	54.5	78.1	61.3	48.0	63.3	54.2	52.0	81.3	76.5	56.8	82.9	63.6
SAN [18]	×	44.4	68.7	74.6	67.5	65.0	77.8	59.8	44.7	80.1	72.2	50.2	78.7	65.3
ETN [41]	×	59.2	77.0	79.5	62.9	65.7	75.0	68.3	55.4	84.4	75.7	57.7	84.5	70.5
SAFN [62]	×	58.9	76.3	81.4	70.4	73.0	77.8	72.4	55.3	80.4	75.8	60.4	79.9	71.8
DCC [64]	×	59.0	84.4	83.4	67.8	72.7	79.8	68.4	53.2	83.7	75.8	59.0	88.3	73.0
BA^{3}US [66]	×	60.6	83.2	88.4	71.8	72.8	83.4	75.5	61.6	86.5	79.3	62.8	86.1	76.0
SHOT [13]	✓	64.8	85.2	92.7	76.3	77.6	88.8	79.7	64.3	89.5	80.6	66.4	85.8	79.3
AR+LS [65]	×	65.67	87.36	89.62	79.25	75.01	86.97	80.81	65.79	90.61	80.81	65.25	86.12	79.4
TransDA (Ours)	✓	73.0	79.5	90.9	72.0	83.4	86.0	81.1	71.0	86.9	87.8	74.9	89.2	81.3
Open-set DA	Source-free	Ar→Cl	Ar→Pr	Ar→	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
DCC [64]	×	52.9	67.4	80.6	49.8	66.6	67.0	59.5	52.8	64.0	56.0	76.9	62.7	64.2
STA [42]	×	58.1	53.1	54.4	71.6	69.3	81.9	63.4	65.2	74.9	85.0	75.8	80.8	69.5
TIM [63]	×	60.1	54.2	56.2	70.9	70.0	78.6	64.0	66.1	74.9	83.2	75.7	81.3	69.6
SHOT [13]	✓	64.5	80.4	84.7	63.1	75.4	81.2	65.3	59.3	83.3	69.6	64.6	82.3	72.8
TransDA (Ours)	✓	71.9	79.1	84.3	71.4	77.1	82.0	68.2	67.5	83.1	76.0	73.0	87.6	76.8

three datasets. For partial-set and open-set DA, we evaluate our method on Office-Home. Following [13], for partial-set DA, we choose 25 classes for the target domain, while all 65 classes are used for the source domain. For open-set DA, we select 25 classes as the shared classes while the other classes make up the unknown class in the target domain.

Implementation details We use a ResNet-50 pretrained on ImageNet as the feature extractor backbone. Moreover, the Transformer [17] layers are injected after the backbone, followed by a bottleneck layer with batch normalization and a task-specific classifier layer. Different from [13, 14], we adopt a teacher-student structure for target adaptation. We use stochastic gradient descent(SGD) with momentum 0.9 and weight decay 10^{-3} to update the network. The learning rates are set to 10^{-3} for the backbone and Transformer layers and set to 10^{-2} for the bottleneck and classifier layers. For source training, we train the model over 100, 50, 30 and 10 epochs for Office-31, Office-Home, Digits and VisDA, respectively. For target adaptation, the number of epochs is set to 15 for all settings. We set α and β in (11) to 0.3 and 1, respectively, which yields consistently high performance across all settings. The batch size is set to 64, and the size of the input image is reshaped to 224×224.

User study To evaluate the importance of the model attention, we conduct a user study based on Amazon Mechanical Turk. Specifically, given the Grad-CAM samples generated by a model, we invite 120 participants to label the samples with "focused / non-focused" according to whether the heat map is localized on the object. Each sample is annotated by all participants and the final label of it is the one selected by the most participants. Given the labels of attention, we then compute the true and false predictions based on the ground-truth class labels for focused samples and non-focused samples, respectively.

4.2 Comparison with state-of-the-art methods

We first compare the proposed TransDA with state-of-the-art methods under closed-set, partial-set, and open-set DA.

- For closed-set DA, the compared methods include: ETD [7], BDG [8], BSP [23], BNM [24], TransNorm [25], GVB-GD [26], GSDA [48], CAN [5], SHOT [13], GSDA [48], RSDA [53], MDD [61], f-DAL [45], ILA-DA [46], CKB [55], and 3C-GAN [9].
- For partial-set DA and open-set DA, we compare with ETN [41], SAFN [62], SHOT [13], TIM [63], STA [42], DCC [64], AR+LS [65], and BA^{3}US [66]. In these methods, only SHOT [13] , 3C-GAN [9], and Hou et al. [43] are designed for source-free domain adaptation.

Table 6 Ablation study on the Transformer, teacher-student structure (EMA) and self-knowledge distillation (KD)

Method	Office-31	Office-Home	VisDA
Source Only	78.6	65.7	46.7
+ Transformer	80.8	67.6	48.0
Baseline	88.6	71.8	75.0
+ Transformer	90.0	78.8	81.0
+ Transformer + EMA	90.2	78.7	81.2
+ Transformer + EMA + KD	90.7	79.3	83.0

Results are evaluated for the closed-set DA under Office-31, Office-Home, and VisDA

Results on Closed-Set DA We report the results on Office-31, Office-Home, Digits and VisDA in Tables 1, 2, 3 and 4, respectively. We can make the following five observations. (1) Our TransDA outperforms all compared methods on all datasets, yielding state-of-the-art accuracies for closed-set DA. (2) When using the same backbone, our TransDA surpasses the source-free method (SHOT [13]) by a large margin. Specifically, when using ResNet-50 as the backbone, TransDA outperforms SHOT [13] by 2.1%, 7.5%, 0.3% and 8.0% on Office-31, Office-Home, and VisDA, respectively. This verifies the effectiveness of the proposed TransDA for source-free DA. (3) Although existing methods already obtain very high performance on Digits, our method still achieves better results on all directions. (4) On VisDA, TransDA with ResNet-50 can produce competitive results compared to the methods that use ResNet-101 as the backbone, further demonstrating the superiority of the proposed TransDA . (5) For VisDA, our method produces lower performance on car and motorcycle classes. One possible reason is that these two classes include many samples and the diversity of them are richer than other classes. In our method, we generate one prototype for each class. However, using only one prototype to represent these diverse classes may not be enough and thus leads to lower performance on them.

Results on Partial-Set and Open-Set DA To verify the versatility of TransDA , we evaluate it on two more challenging tasks, i.e., partial-set DA [18] and open-set DA [42]. For fair comparison, we follow the protocols proposed by [18] and [42] to set the evaluation settings for partial-set DA and open-set DA, respectively. Specifically, we first rank the classes in alphabetical order and select the first 25 classes as the shared classes. For partial-set DA, the target domain includes the samples of the shared 25 classes while the source domain includes the samples of all 65 classes. For open-set DA, the source domain consists of the samples of the shared 25 classes while the target domain includes the samples of all 65 classes where the non-overlapped 40 classes are regarded unknown classes.

The results on Office-Home are reported in Table 5. The advantage of TransDA is similar to that for closed-set DA. That is, TransDA clearly outperforms the compared methods on both settings. Specifically, TransDA is 2.0% and 4.0% better than SHOT [13] on partial-set DA and open-set DA, respectively. This demonstrates that our Transformer-based structure is effective under various domain adaptation settings.

4.3 Ablation study

Accuracy comparison In Table 6, we study the effectiveness of the proposed Transformer structure and self-knowledge distillation. We first explain the components in Table 6: *Source Only* indicates the pretrained source model; *Baseline* denotes further training the model on the target data with the information maximization and self-labeling losses; *+Transformer* means injecting the Transformer module into the model; *+EMA* refers to using the teacher-student structure; and *+KD* indicates using the self-knowledge distillation loss. From Table 6, we can draw the following conclusions. (1) Injecting the Transformer into the network

Table 7 Accuracy (%) on Office-Home for comparison to different attention modules

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	P→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
Baseline	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
w/ SE [70]	57.3	78.4	81.8	68.3	78.5	78.4	67.7	55.1	82.5	73.6	59.0	84.7	72.1
w/ Non-local [71]	61.7	76.2	78.6	67.7	76.6	77.2	70.4	62.2	79.5	73.6	63.9	82.3	72.5
w/ Transformer	67.5	83.3	85.9	74.0	83.8	84.4	77.0	68.0	87.0	80.5	69.9	90.0	79.3

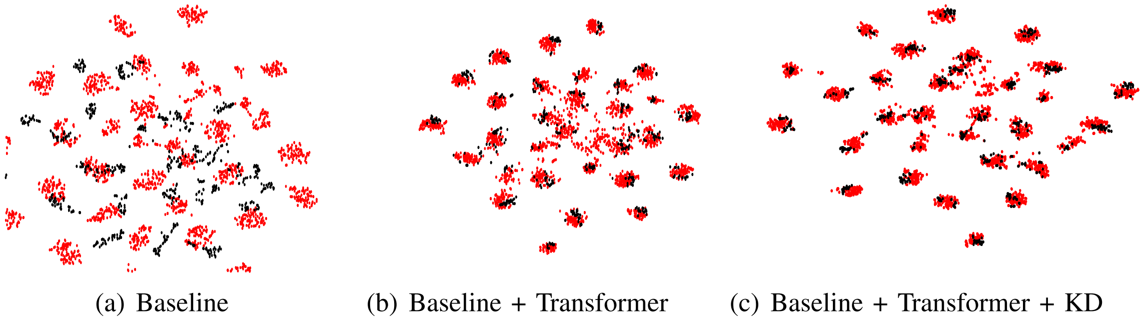


Fig. 3 t-SNE visualization for different methods on Office-31 (A→W). We use the outputs of the feature extractor as the features. Red/black denote the source/target domains. Best viewed in color

can consistently improve the results, regardless of the model as learned on the source data or the target data. Specifically, when using the Transformer, the accuracy of *Baseline* improves from 72.1% to 78.8% on Office-Home. This demonstrates the effectiveness of the Transformer in domain adaptation. (2) Adding the knowledge distillation loss can further boost the performance, verifying the advantage of the self-knowledge distillation. (3) Applying the teacher-student structure fails to produce clear improvements, indicating that the gains of +*KD* are mainly obtained by knowledge distillation rather than by generating pseudo-labels with the teacher model.

Comparisons of different attention modules In Table 7, we compare different attention modules, including the SE module [70], non-local module [71] and Transformer. We can observe that the SE module and non-local modules fail to improve the average accuracy significantly, while the Transform obtains a 7.5% improvement, demonstrating the effect of Transformer. According to Table 6, adding transformer and EMA leads to amount of computing consumption. However, taking computing consumption and performance improvement, our TransDA holds the obvious advantage among the different attention modules.

t-SNE visualization In Fig. 3, we show the t-SNE of features for different methods. We find that adding the Transformer and knowledge distillation can (1) make the intra-class samples more compact and (2) reduce the distances between source and target domain clusters. These findings reveal that TransDA can encourage the model to be more robust to intra-class variations and can decrease the cross-domain distribution gap.

Visualization of Grad-cam and statistics study for focused and non-focused samples In Fig. 4, we compare the Grad-CAM visualizations for different variants of our method. We obtain the following findings. (1) When adding the Transformer into the network, the red regions on the objects

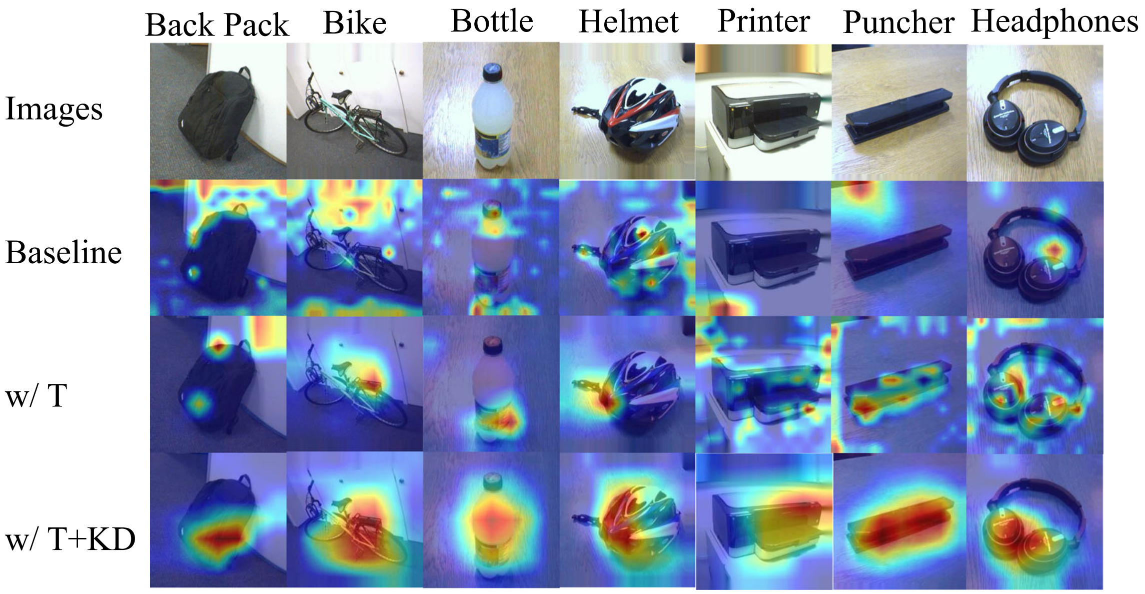


Fig. 4 Visualization of Grad-CAM for different methods. The results are evaluated on Office-31 (A→W)

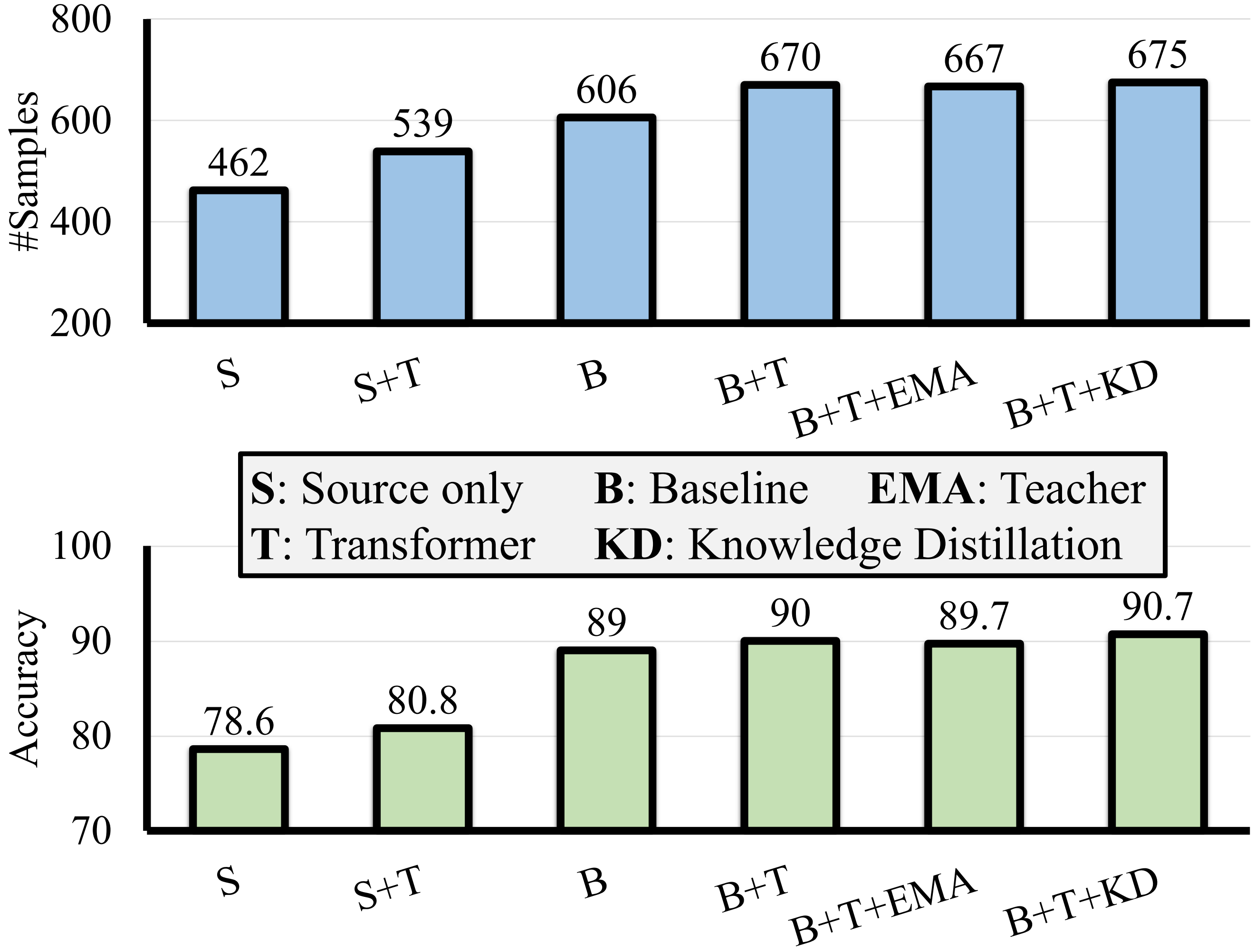


Fig. 5 Visualization of statistics studies for different methods on Office-31 (A→W)

increase, indicating that the network is encouraged to pay more attentions to the objects. (2) When training the model with the knowledge distillation loss, the attention ability of the network is further improved. In addition, we use Amazon Mechanical Turk to estimate the "focused / non-focused" samples for different methods (see Fig. 5). We can observe that the focused samples and accuracy increase when adding the Transformer and the knowledge distillation loss. The above observations verify that 1) the proposed Transformer structure and knowledge distillation loss can effectively encourage the network to focus on objects, and 2) improving the attention ability of the network can consistently improve domain adaptation accuracy.

Discussion Indeed, our method with transformer backbone and teacher-student structure increases the computation cost. However, this increment is largely lower compared to the costs of persistently keeping and using source domains during training. In practice, the dataset commonly requires high saving cost, *e.g.*, the size of VisDA [39] dataset is larger than 7 GB. However, the size of our backbone model is less than 100 MB, which is largely lower than most of the source domains. This indicates that using source-free constraint can significantly reduce the saving and loading costs, especially when using large-scale source domains. Therefore, we believe it is appropriate to designing a source-free method to achieve performance improvement at the cost of using more complicated models.

5 Conclusion

In this paper, we proposed a generic yet straightforward representation learning framework, named TransDA, for source-free domain adaptation (SFDA). Specifically, by employing a Transformer module and learning the model with the self-knowledge distillation loss, the network is encouraged to pay more attention to the objects in an image. Experiments on closed-set, partial-set, and open-set DA confirm the effectiveness of the proposed TransDA.

Importantly, this work reveals that the attention ability of a network is highly related to its adaptation accuracy. We hope these findings will provide a new perspective for designing domain adaptation algorithms in the future. In the future work, we would like to study two reasonable and promising directions to improve our method. First, we will investigate more style-based augmentation technologies during the source-training and self-training processes, which can encourage the model be more robust to domain shifts. Second, we attempt to design flexible clustering methods that can generate robust prototypes to better handle the intra-class variance.

Data Availability The datasets generated and analysed during this study are available in the Github repository: https://github.com/ygjwd12345/TransDA

Declarations

Conflict of Interests We would like to note that in the manuscript entitled "Self-Training Transformer for Source-Free Domain Adaptation", no conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication.

References

1. Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. In: International conference on machine learning, PMLR, pp 97–105
2. Tzeng E, Hoffman J, Saenko K, Darrell T (2017) Adversarial discriminative domain adaptation. In: roceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7167–7176
3. Zhang Y, David P, Gong B (2017) Curriculum domain adaptation for semantic segmentation of urban scenes. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2020–2030
4. Long M, Zhu H, Wang J, Jordan MI (2016) Unsupervised domain adaptation with residual transfer networks. In: Advances in neural information processing systems, vol 29
5. Kang G, Jiang L, Yang Y, Hauptmann AG (2019) Contrastive adaptation network for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4893–4902
6. Lee C-Y, Batra T, Baig MH, Ulbricht D (2019) Sliced wasserstein discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10285–10295
7. Li M, Zhai Y-M, Luo Y-W, Ge P-F, Ren C-X (2020) Enhanced transport distance for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13936–13944
8. Yang G, Xia H, Ding M, Ding Z (2020) Bi-directional generation for unsupervised domain adaptation. In: Proceedings of the Proceedings of the AAAI conference on artificial intelligence conference on artificial intelligence. vol 34, no 04, pp 6615–6622
9. Li R, Jiao Q, Cao W, Wong H-S, Wu S (2020) Model adaptation: unsupervised domain adaptation without source data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9641–9650
10. Sankaranarayanan S, Balaji Y, Castillo CD, Chellappa R (2018) Generate to adapt: aligning domains using generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8503–8512
11. Saito K, Watanabe K, Ushiku Y, Harada T (2018) Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3723–3732
12. Kurmi VK, Kumar S, Namboodiri VP (2019) Attending to discriminative certainty for domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 491–500
13. Liang J, Hu D, Feng J (2020) Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International conference on machine learning, PMLR, pp 6028–6039
14. Ahmed SM, Raychaudhuri DS, Paul S, Oymak S, Roy-Chowdhury AK (2021) Unsupervised multi-source domain adaptation without access to source data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10103–10112
15. Qiao F, Zhao L, Peng X (2020) Learning to learn single domain generalization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12556–12565
16. Qiao F, Peng X (2021) Uncertainty-guided model generalization to unseen domains. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6790–6800
17. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S et al (2021) An image is worth 16x16 words: transformers for image recognition at scale. In: International conference on learning representations
18. Cao Z, Long M, Wang J, Jordan MI (2018) Partial transfer learning with selective adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2724–2732
19. Panareda Busto P, Gall J (2017) Open set domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 754–763
20. Hu C, He S, Wang Y (2021) A classification method to detect faults in a rotating machinery based on kernelled support tensor machine and multilinear principal component analysis. Appl Intell 51(4):2609–2621
21. Hu C, Wang Y, Gu J (2020) Cross-domain intelligent fault classification of bearings based on tensor-aligned invariant subspace learning and two-dimensional convolutional neural networks, vol 209
22. Carlucci FM, Porzi L, Caputo B, Ricci E, Bulo SR (2020) Multidial: domain alignment layers for (multisource) unsupervised domain adaptation. IEEE Trans Pattern Anal Mach Intell 43(12):4441–4452
23. Chen X, Wang S, Long M, Wang J (2019) Transferability vs. discriminability: batch spectral penalization for adversarial domain adaptation. In: International conference on machine learning, PMLR, pp 1081–1090
24. Cui S, Wang S, Zhuo J, Li L, Huang Q, Tian Q (2020) Towards discriminability and diversity: batch nuclear-norm maximization under label insufficient situations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3941–3950
25. Wang X, Jin Y, Long M, Wang J, Jordan MI (2019) Transferable normalization: towards improving transferability of deep neural networks. Advances in Neural Information Processing Systems, vol. 32
26. Cui S, Wang S, Zhuo J, Su C, Huang Q, Tian Q (2020) Gradually vanishing bridge for adversarial domain adaptation. In:

Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, pp 12455–12464

27. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need, Advances in Neural Information Processing Systems, vol. 30
28. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: European Conference on Computer Vision, 213–229
29. Zhu X, Su W, Lu L, Li B, Wang X, Dai J (2021) Deformable detr: deformable transformers for end-to-end object detection. In: International Conference on Learning Representations
30. Zheng S, Lu J, Zhao H, Zhu X, Luo Z, Wang Y, Fu Y, Feng J, Xiang T, Torr PH et al (2021) Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6881–6890
31. Zeng Y, Fu J, Chao H (2020) Learning joint spatial-temporal transformations for video inpainting. In: European Conference on Computer Vision, pp 528–543
32. Neimark D, Bar O, Zohar M, Asselmann D (2021) Video transformer network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 3163–3172
33. Huang L, Tan J, Liu J, Yuan J (2020) Hand-transformer: non-autoregressive structured modeling for 3d hand pose estimation. In: European Conference on Computer Vision, pp 17–33
34. He S, Luo H, Wang P, Wang F, Li H, Jiang W (2021) Transreid: transformer-based object re-identification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 15013–15022
35. Yang G, Tang H, Ding M, Sebe N, Ricci E (2021) Transformer-based attention networks for continuous pixel-wise prediction. In: Proceedings of the IEEE/CVF International Conference on Computer vision, pp 16269–16279
36. Müller R, Kornblith S, Hinton G (2019) When does label smoothing help? in Advances in neural information processing systems, vol 32
37. Hu W, Miyato T, Tokui S, Matsumoto E, Sugiyama M (2017) Learning discrete representations via information maximizing self-augmented training. In: International Conference on Machine Learning, pp 1558–1567
38. Caron M, Touvron H, Misra I, Jégou H., Mairal J, Bojanowski P, Joulin A (2021) Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision
39. Peng X, Usman B, Kaushik N, Hoffman J, Wang D, Saenko K (2017) Visda: the visual domain adaptation challenge, arXiv
40. Hoffman J, Tzeng E, Park T, Zhu J.-Y., Isola P, Saenko K, Efros A, Darrell T (2018) Cycada: Cycle-consistent adversarial domain adaptation. In: International conference on machine learning, Pmlr, pp 1989–1998
41. Cao Z, You K, Long M, Wang J, Yang Q (2019) Learning to transfer examples for partial domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2985–2994
42. Liu H, Cao Z, Long M, Wang J, Yang Q (2019) Separate to adapt: open set domain adaptation via progressive separation
43. Hou Y, Zheng L (2021) Visualizing adapted knowledge in domain transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 13824–13833
44. Yang G, Ding M, Zhang Y (2022) Bi-directional class-wise adversaries for unsupervised domain adaptation. Appl Intell 52(4):3623–3639
45. Acuna D, Zhang G, Law MT, Fidler S (2021) F-domain adversarial learning: theory and algorithms. In: International Conference on Machine Learning, PMLR, pp 66–75
46. Sharma A, Kalluri T, Chandraker M (2021) Instance level affinity-based transfer for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5361–5371
47. Yang S, van de Weijer J, Herranz L, Jui S et al (2021) Exploiting the intrinsic neighborhood structure for source-free domain adaptation. Adv Neural Inf Process Syst 34:29393–29405
48. Hu L, Kan M, Shan S, Chen X (2020) Unsupervised domain adaptation with hierarchical gradient synchronization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4043–4052
49. Huang J, Guan D, Xiao A, Lu S (2021) Model adaptation: historical contrastive learning for unsupervised domain adaptation without source data. Adv Neural Inf Process Syst 34:3635–3649
50. Chu T, Liu Y, Deng J, Li W, Duan L (2022) Denoised maximum classifier discrepancy for sourcefree unsupervised domain adaptation. In: Proceedings of the AAAI conference on artificial intelligence, vol 2
51. Li S, Xie M, Lv F, Liu CH, Liang J, Qin C, Li W (2021) Semantic concentration for domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9102–9111
52. Xia H, Zhao H, Ding Z (2021) Adaptive adversarial network for source-free domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9010–9019
53. Gu X, Sun J, Xu Z (2020) Spherical space domain adaptation with robust pseudo-label loss. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9101–9110
54. Li S, Xie M, Gong K, Liu CH, Wang Y, Li W (2021) Transferable semantic augmentation for domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11516–11525
55. Luo Y.-W., Ren C.-X. (2021) Conditional bures metric for domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 13989–13998
56. Yue Z, Sun Q, Hua X-S, Zhang H (2021) Transporting causal mechanisms for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8599–8608
57. Gao Z, Zhang S, Huang K, Wang Q, Zhong C (2021) Gradient distribution alignment certificates better adversarial domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 8937–8946
58. Na J, Jung H, Chang HJ, Hwang W (2021) Fixbi: bridging domain spaces for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1094–1103
59. Long M, Cao Z, Wang J, Jordan MI (2018) Conditional adversarial domain adaptation. Advances in Neural Information Processing Systems, 31
60. Nguyen AT, Tran T, Gal Y, Torr PH, Baydin AG (2022) Kl guided domain adaptation. In: International conference on learning representations
61. Zhang Y, Liu T, Long M, Jordan M (2019) Bridging theory and algorithm for domain adaptation. In: International Conference on Machine Learning, PMLR, pp 7404–7413
62. Xu R, Li G, Yang J, Lin L (2019) Larger norm more transferable: an adaptive feature norm approach for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 1426–1435
63. Kundu JN, Venkat N, Revanur A, Babu RV et al (2020) Towards inheritable models for open-set domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12376–12385
64. Li G, Kang G, Zhu Y, Wei Y, Yang Y (2021) Domain consensus clustering for universal domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9757–9766

65. Gu X, Yu X, Sun J, Xu Z et al (2021) Advances in neural information processing systems. Adversarial Reweighting for Partial Domain Adaptation 34:14860–14872
66. Liang J, Wang Y, Hu D, He R, Feng J (2020) A balanced and uncertainty-aware approach for partial domain adaptation. In: European Conference on Computer Vision, pp 123–140
67. Du Z, Li J, Su H, Zhu L, Lu K (2021) Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3937–3946
68. Lu Z, Yang Y, Zhu X, Liu C, Song Y-Z, Xiang T (2020) Stochastic classifiers for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9111–9120
69. Zhang J, Ding Z, Li W, Ogunbona P (2018) Importance weighted adversarial nets for partial domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8156–8164
70. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7132–7141
71. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7794–7803

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Guanglei Yang received a B.S. degree in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2016. He is currently pursuing a Ph.D. degree at the School of Instrumentation Science and Engineering, Harbin Institute of Technology (HIT), Harbin, China. He has been working at the University of Trento as a visiting student since 2020. His research interests include domain adaptation and pixel-level prediction.

Zhun Zhong received the Ph.D. Degree in Computer Science and Technology from Xiamen University, China, in 2019. He was also a joint Ph.D. student at University of Technology Sydney. He is currently a Fellowship Researcher in University of Trento. His research interests include person re-identification and domain adaptation.

Mingli Ding received the B.S., M.S. and Ph.D. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 1996, 1997 and 2001, respectively. He worked as a visiting scholar in France from 2009 to 2010. Currently, he is a professor in the School of Instrumentation Science and Engineering at Harbin Institute of Technology. Prof. Ding's research interests are intelligence tests and information processing, automation test technology, computer vision, and machine learning. He has published over 40 papers in peer-reviewed journals and conferences.

Nicu Sebe is Professor in the University of Trento, Italy, where he is leading the research in the areas of multimedia analysis and human behavior understanding. He was the General Co-Chair of the IEEE FG 2008 and ACM Multimedia 2013. He was a program chair of ACM Multimedia 2011 and 2007, ECCV 2016, ICCV 2017 and ICPR 2020. He is a general chair of ACM Multimedia 2022 and a program chair of ECCV 2024. He is a fellow of IAPR.

Elisa Ricci received a Ph.D. degree from the University of Perugia in 2008. She is an associate professor at the University of Trento and a researcher at Fondazione Bruno Kessler. She has since been a postdoctoral researcher at Idiap, Martigny, and Fondazione Bruno Kessler, Trento. She was also a visiting researcher at the University of Bristol. Her research interests are in the areas of computer vision and machine learning. She is a member of the IEEE.