ToAlign: Task-oriented Alignment for Unsupervised Domain Adaptation

Guoqiang Wei1∗ Cuiling Lan2† Wenjun Zeng2 Zhizheng Zhang2 Zhibo Chen1†

1 University of Science and Technology of China 2 Microsoft Research Asia
wgq7441@mail.ustc.edu.cn (culan,wezeng,zhizzhang@microsoft.com)
chenzhibo@ustc.edu.cn

Abstract

Unsupervised domain adaptive classification intends to improve the classification performance on unlabeled target domain. To alleviate the adverse effect of domain shift, many approaches align the source and target domains in the feature space. However, a feature is usually taken as a whole for alignment without explicitly making domain alignment proactively serve the classification task, leading to sub-optimal solution. In this paper, we propose an effective Task-oriented Alignment (ToAlign) for unsupervised domain adaptation (UDA). We study what features should be aligned across domains and propose to make the domain alignment proactively serve classification by performing feature decomposition and alignment under the guidance of the prior knowledge induced from the classification task itself. Particularly, we explicitly decompose a feature in the source domain into a task-related/discriminative feature that should be aligned, and a task-irrelevant feature that should be avoided/ignored, based on the classification meta-knowledge. Extensive experimental results on various benchmarks (e.g., Office-Home, Visda-2017, and DomainNet) under different domain adaptation settings demonstrate the effectiveness of ToAlign which helps achieve the state-of-the-art performance. The code is publicly available at https://github.com/microsoft/UDA.

1 Introduction

Convolutional Neural Networks (CNNs) have made extraordinary progress in various computer vision tasks, with image classification as a most representative one. The trained models generally perform well on the testing data which shares similar data distribution to that of the training data. However, in many practical scenarios, drastic performance degradation is observed when applying such trained models to new domains with domain shift [57], where the data distributions between the training and testing domains are different. Fine-tuning on labeled target data is a direct solution but is costly due to the requirement of target sample annotations. In contrast, unsupervised domain adaptation (UDA) requires only the labeled source data and unlabeled target data to enhance the model’s performance on the target domain, which has attracted increasing interest in both academia [3, 2, 72, 58, 25, 31] and industry [63, 27].

There has been a large spectrum of UDA methods. Supported by the theoretical analysis [3], the overwhelming majority of methods tend to align the distributions of source and target domains. A line of works [6, 67, 43, 54, 55] explicitly align the distributions based on domain discrepancy measurements, e.g., Maximum Mean Discrepancy (MMD) [6]. Another line of alignment-based UDA borrows ideas from Generative Adversarial Networks [19] and use domain adversarial training
to learn domain-aligned/invariant features, which dominate in the top performance methods. In the seminal work Domain Adversarial Neural Network (DANN) [17, 18], a domain discriminator is trained to distinguish the target features from source features while a feature extractor (generator) is trained to generate domain-invariant features to fool this discriminator. Following DANN, a plethora of variants have been proposed [58, 40, 51, 11, 50, 61, 38, 41, 14, 10, 64].

It is noteworthy that the goal of alignment in UDA is to alleviate the adverse effect of domain shift to improve the classification performance on unlabeled target data. Even though impressive progress has been made, there is a common intrinsic limitation, i.e., alignment is still not deliberately designed to dedicatedly/proactively serve the final image classification task. In many previous UDAs, as shown in Figure 1 (a), the alignment task is in parallel with the ultimate classification task. The assumption is that learning domain-invariant features (via alignment) reduces the domain gap and thus makes the image classifier trained on source readily applicable to target [3]. However, with alignment treated as a parallel task, there is a lack of mechanism to make it explicitly assist classification, where the alignment may contaminate the discriminative features for classification [28]. Previous works (e.g., CDAN [40]) exploit class information (e.g., predicted class probability) as a condition to the discriminator. MADA [42] implements class-level domain alignment by applying one discriminator per class. Their purpose is to provide additional helpful information to the discriminator [40] or perform class-level alignment [42], but they are still short of explicitly making alignment assist classification.

Some works move a step forward and investigate what features the networks should align for better adaptation. [62, 52] focus on transferable local regions, which are selected based on the uncertainty or entropy of the domain discriminator, for alignment. However, such self-induced feature selection is still not specific to the optimization of classification task; instead, it is based on the alignment task itself. There is no guarantee that alignment positively serves the classification task. Hsu et al. [23] carry out object centerness-aware alignment by aligning the center part of the objects to exclude the background distraction/noise for domain adaptive object detection. However, the feature in object center position could be task-irrelevant and thus is not suited for alignment. Moreover, regarding such centerness feature as alignment objective is somewhat ad-hoc, which is still not designed directly from the perspective of assisting classification.

We pinpoint that the selection of “right” features to achieve task-oriented alignment is important. For classification, the essence is to train the network to extract class-discriminative feature. Similarly, for UDA classification, it is also desired to assure strong discrimination of the target domain features without class label supervision. Thus, we intend to align target features to the task-discriminative source features while ignoring the task-irrelevant ones. Note that for the feature of a source sample, it contains both task/classification-discriminative and task-irrelevant information, because the network is in general not able to suppress non-discriminative feature responses (e.g., responses unrelated to
Figure 2: Conceptual comparison between (a) previous alignment and (b) our proposed task-oriented alignment. \( \{f^t\} \) and \( \{f^s\} \) denote the sets of target features and source features, respectively. (a) Previous methods take each source feature as a holistic one for alignment with target features. (b) We decompose each source feature \( f^s \) into a task-discriminative positive feature \( f^s_p \) and a task-irrelevant negative feature \( f^s_n \) and make the target features to be aligned with the positive source features \( \{f^s_p\} \) while avoiding aligning with the negative source features \( \{f^s_n\} \).

image class or those related to other tasks such as alignment) perfectly \[52, 9\]. Aligning target features with task-irrelevant source features would prevent alignment from serving classification and lead to poor adaptation. Intuitively, for example, image style that is a non-causal factor for classification can be considered as task-irrelevant information and the bias towards such factor in alignment may hurt the classification task. We demonstrate this by conducting experiments where only the source task-irrelevant features are utilized to align with target i.e., the scheme Baseline+TiAlign in Figure 3. The performance of Baseline+TiAlign (in purple) on target test set drops drastically compared to the source-only method which dose not incorporate any alignment technique. This corroborates that aligning with task-irrelevant features is even harmful to the classification on target domain.

Motivated by this, in this paper, we propose an effective UDA method named Task-oriented Alignment (ToAlign) to make the domain alignment explicitly serve classification. We achieve this by performing feature alignment guided by the meta-knowledge induced from the classification task to make the target features align with task-discriminative source features (i.e., "positive" features), to avoid the interference from task-irrelevant features (i.e., "negative" features). Figure 2 conceptually illustrates the comparison between our proposed alignment and previous one. Particularly, as illustrated in Figure 1 (b), to obtain the suitable feature from a source sample for alignment with target samples, we leverage the classification task to guide the extraction/distillation of task-related/discriminative feature \( f^p \), from original feature \( f^s \). Correspondingly, for the domain alignment task, we enforce aligning target features with the source positive features by domain adversarial training to achieve task-oriented alignment. In this way, the domain alignment will better assist the classification task.

We summarize our main contributions as follows:

- We pinpoint that the selection of "right" features to achieve task-orientated alignment is important for adaptation.
- We propose an effective UDA approach named ToAlign which enables the alignment to explicitly serve classification. We decompose a source feature into a task-relevant/discriminative one and a task-irrelevant one under the guidance of classification-meta knowledge for performing classification-oriented alignment, which explicitly guides the network what features should be aligned.

Extensive experimental results demonstrate the effectiveness of ToAlign. ToAlign is generic and can be applied to different adversarial learning based UDAs to enhance their adaption capability, which helps achieve the state-of-the-art performance with a negligible increase in training complexity and no increase in inference complexity.

2 Related Work

Unsupervised Domain Adaptation aims to transfer the knowledge from labeled source domain(s) to unlabeled target domain. Ben et al. [3] theoretically reveal that learning domain-invariant representations helps make the image classifier trained on source domain applicable to target domain. Various works learn domain-invariant features by aligning the source and target distributions measured by some metrics \[6, 67, 33, 54, 55, 45\], or by domain adversarial learning \[58, 40, 51, 11, 50, 61, 38, 69, 56, 41, 14, 10, 64, 8, 29\]. The latter is overwhelmingly popular in recent years owing to its
superiority in dealing with distribution problems [19]. Note that our proposed method is designed to enhance the capability of the widely used domain adversarial learning based approaches.

For domain adversarial learning based approach (e.g., DANN [17, 18]), in general, a domain discriminator is trained to distinguish the source domain from the target domain, meanwhile a feature extractor is trained to learn domain-invariant features. Many variants of DANN have been proposed [40, 14, 56, 10, 12, 49, 55, 5]. CDAN [40] further conditions the discriminator on the image class information conveyed in the classifier predictions. MADA [42] implements class-wise alignment with multi-discriminators. GSDA [24] performs class-, group- and domain-wise alignments simultaneously, where the three types of alignment are enforced to be consistent in their gradients for more precise alignment. HDA [12] leverages domain-specific representations as heuristics to obtain domain-invariant representations from a heuristic search perspective. CMSS [66] exploits Curriculum Learning (CL) [4] to align target samples with the dynamically selected source samples to exploit the different transferability of the source samples.

However, in these methods, the domain alignment is designed as a task in parallel with the image classification task. It does not explicitly take serving classification as its mission, where such alignment may result in loss of discriminative information. Jin et al. [28] remedy the loss of discriminative information caused by alignment via incorporating a restoration module. Wei et al. [64] pinpoint that alignment and classification are not well coordinated in optimization where they may contradict with each other. They thus propose to use meta-learning to coordinate their optimization directions.

In this paper, to make alignment explicitly serve classification, we propose a task-oriented alignment. Guided by the classification meta-knowledge, task-discriminative sub-features are selected for alignment. Different from [64], we investigate what features should be aligned to assist classification and intend to provide more interpretable alignment. We are the first to perform task-oriented alignment by decomposing source feature into task-discriminative and task-irrelevant feature, and explicitly guides the network what sub-features should be aligned. Note that Huang et al. [26] propose to decouple features into domain-invariant and domain-specific features, where the former ones are aligned for unsupervised person re-identification. [44, 7] exploit the VAE framework with several complex losses to perform the disentanglement from the perspective of domain and semantics simultaneously, and only use domain-invariant semantics for inference, leaving domain-specific but task-related information underexplored. In contrast to focusing on the domain-level, our decomposition strategy focuses on the task-level guided by image classification task, where we further enable domain alignment on the task-discriminative features to proactively serve image classification.

3 Task-Oriented Alignment for UDA

Unsupervised domain adaptation (UDA) for classification aims to train a classification model on labeled source domain image set \(X_s\) and unlabeled target domain image set \(X_t\) to obtain high classification accuracy on a target domain test set.

Most popular adversarial learning based UDAs attempt to align the features of the source and target domains to alleviate the domain gap to improve the classification performance on target domain. As
As illustrated in Figure 1, we obtain a feature map which domain (i.e., source or target) a sample belongs to, and adversarially train a feature extractor to decompose a source sample feature into a task-discriminative one that should be aligned, and a task-irrelevant one that should be ignored based on the classification meta-knowledge. Then, we perform alignment between the target features and the positive source features, which is consistent with the essence of the classification task, i.e., focusing on discriminative features.

In Sec. 3.1 to be self-contained, we briefly introduce adversarial learning based UDAs. We answer the question of what feature should be aligned to better serve classification and introduce our task-oriented feature decomposition and alignment in Sec. 3.2.

3.1 Recap of Domain Adversarial UDAs

Domain adversarial learning based UDAs typically train a domain discriminator $D$ to distinguish which domain (i.e., source or target) a sample belongs to, and adversarially train a feature extractor $G$ to fool the discriminator $D$ in order to learn domain-invariant feature representations. The network is also trained under the supervision of image classification on the labeled source samples. Particularly, $D$ is optimized to minimize the domain classification loss $L_D$ (i.e., binary cross entropy loss). Meanwhile, $G$ is optimized to maximize the domain classification loss $L_D$ and minimize the image classification loss $L_{cls}$ (i.e., cross entropy loss):

$$\argmin_D L_D, \quad \argmin_G L_{cls} - L_D,$$

To achieve adversarial training, usually, gradient reversal layer (GRL) [17, 18] which connects $G$ and $D$ is used via multiplying the gradient from $D$ by a negative constant during the back-propagation to $G$. $L_D$ is typically defined as [18, 40, 14]:

$$L_D(X_s, X_t) = -\mathbb{E}_{x_s \sim X_s} \left[ \log(D(G(x_s))) \right] - \mathbb{E}_{x_t \sim X_t} \left[ \log(1 - D(G(x_t))) \right],$$

3.2 Task-oriented Feature Decomposition and Alignment

In adversarial learning based UDAs, a feature ingested by $D$ as a holistic feature from a source or target sample, in general contains both task/classification-discriminative information and task-irrelevant information. Intuitively, aligning the task-irrelevant features would not effectively reduce the domain gap of the task-discriminative features and thus brings no obvious benefit for the classification task. Mistakenly aligning the target features with the source task-irrelevant features would hurt the discrimination power of the target features. We also experimentally confirm that in Figure 2, i.e., aligning with task-irrelevant features ($TiAlign$, line in purple) drastically reduces the classification accuracy on the target domain. Therefore, we propose to decompose a holistic feature of each source sample into a task-discriminative feature and a task-irrelevant feature to enable the task-oriented alignment with the target features.

Particularly, we softly select/re-weight (based on Grad-CAM [52]) the feature vector $f^s$ of a source sample to obtain task-discriminative feature $f^s_n$ that is discriminative for identifying the groundtruth class, which we refer to as positive feature. Correspondingly, the task-irrelevant feature $f^s_n$ can be obtained simultaneously, which we refer to as negative feature.

**Task-Oriented Feature Decomposition.** Grad-CAM [73, 52, 9] is a widely used technique to localize the most important features for classification in a convolutional neural network model. As analyzed in [73, 52, 9, 53], the gradients (w.r.t. the feature for classification) of the final predicted score corresponding to the ground-truth class convey the task-discriminative information, which identifies the relevant features to recognize the image class correctly. It is noteworthy that such task-disciminative information is, in general, highly related (but not limited) to the foreground object in the classification task. In this work, motivated by Grad-CAM, we propose to use the gradients of the predicted score corresponding to the ground-truth class as the attention weights to obtain the task-discriminative features.

As illustrated in Figure 1, we obtain a feature map $F \in \mathbb{R}^{H \times W \times M}$ (i.e., a tensor of non-negative real numbers, with height $H$, width $W$, and $M$ channels) from the final convolutional block (with ReLU
layer) of the feature extractor. After spatial-wise global average pooling (GAP), we have a feature vector \( f = \text{pool}(F) \in \mathbb{R}^M \). The logits for all classes are predicted via the classifier \( C(\cdot) \). Based on the response \( C(f) \), we can derive the gradient \( w_{cls} \in \mathbb{R}^M \) of \( y^k \) w.r.t. \( f \):

\[
w_{cls} = \frac{\partial y^k}{\partial f}.
\]

where \( y^k \) is the predicted score corresponding to the ground-truth class \( k \). As analyzed in [52][50][53], the gradient \( w_{cls} \) conveys the channel-wise importance information of feature \( f \) for classifying the sample into its ground-truth class \( k \). We draw inspiration from Grad-CAM which uses \( w_{cls} \) to modulate the feature map in channel-wise to find the classification-discriminative features. Similarly, modulated with \( w_{cls} \), we can obtain the task-discriminative (i.e., positive) feature as:

\[
f_p = w_{cls}^p \cdot f = s w_{cls} \cdot f,
\]

where \( s \) represents the Hadamard product, the attention weight vector \( w_{cls}^p = s w_{cls} \), where \( s \in \mathbb{R}_+ \) is an adaptive non-negative parameter to modulate the energy \( E(f_p) = \|f_p\|_2^2 \) of \( f_p \) such that \( E(f_p) = E(f) \):

\[
s = \sqrt{\frac{\|f\|_2^2}{\|w_{cls}\|_2^2}} = \sqrt{\frac{\sum_{m=1}^{M} f_m^2}{\sum_{m=1}^{M} (w_{cls}^p f_m)^2}},
\]

Motivated by the counterfactual analysis in [52], the task-irrelevant (i.e., negative) feature can be represented as \( f_n = -w^p_{cls} \cdot f \), where \(-w^p_{cls}\) delights the task-discriminative channels since the task-discriminative channels (with larger values in \( w^p_{cls} \)) correspond to ones with smaller values in \(-w^p_{cls}\).

To better understand and validate the discriminativeness of the positive and negative features, we visualize the spatial maps \( F \) with channels modulated by \( w_{cls} \) and \(-w_{cls}\) following [52][73]. As shown in Figure [3], the positive information is more related to the foreground objects that provide the discriminative information for the classification task, while the negative one is more in connection with the non-discriminative background regions.

**Task-oriented Domain Alignment.** As discussed above, we expect the domain alignment to explicitly serve the final classification task. Given the source task-discriminative features obtained based on the classification meta-knowledge, we can guide the target features to be aligned with the source task-discriminative features \( f \) through different domain adversarial learning based alignment methods [17][18][12]. The procedure is almost the same as that in UDAs discussed in Sec. 3.1 except that the input source feature \( f^s \) to the final domain discriminator is replaced by the positive feature \( f^s_p \) of this source sample. Thus, the domain classification loss is defined with a small modification on Eq. (2):

\[
L_D(X_s, X_t) = -\mathbb{E}_{x_s \sim X_s} [\log(D(G^p(x_s)))] - \mathbb{E}_{x_t \sim X_t} [\log(1 - D(G(x_t)))]
\]

where \( G^p(x_s) = f^s_p \) denotes the positive feature of source \( x_s \).

**Understanding from the Meta-knowledge Perspective.** To enable a better understanding of ToAlign on why it works well, here, we analyse ToAlign from the perspective of meta-learning with meta-knowledge.

In an adversarial UDA framework, the image classification task and domain alignment task can be considered to be a meta-train task \( T_{tr} \) and a meta-test task \( T_{te} \), respectively. ToAlign actually introduces knowledge communication from \( T_{tr} \) to \( T_{te} \). In the meta-training stage, we can obtain the prior/meta-knowledge \( \phi_{tr} \) of \( T_{tr} \). Without effective communication between \( T_{tr} \) and \( T_{te} \), the optimization of \( T_{te} \) may contradict that of \( T_{tr} \), considering that they have different optimization goals. To improve the knowledge communication from \( T_{tr} \) to \( T_{te} \), certain meaningful prior/meta-knowledge \( \phi_{tr} \) is helpful for a more effective \( \nabla T_{te}^{\phi_{tr}} \). A typical implementation of passing meta-knowledge from \( T_{tr} \) to \( T_{te} \) is based on gradients [39][16][54][64][33], i.e., \( \nabla T_{tr} \), which provides knowledge of \( T_{te} \). Other mechanisms e.g., leveraging the parameters regularizer in a way of weight decay, are also exploited [1][11]. In our ToAlign, instead of encoding the meta-knowledge \( \phi_{tr} \) into the gradients w.r.t. the parameters, we use \( T_{tr} \) to learn/derive attention weights for identifying \( T_{tr} \)-related sub-features in the feature space and then pass such prior/meta-knowledge \( \phi_{tr} \) to \( T_{te} \) to make meta-test task \( T_{te}^{\phi_{tr}} \) adapt its optimization based on \( \phi_{tr} \).
which span 6 domains with large domain gap: Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real (R), and Sketch (S). For MUDA, following the settings in [43, 66, 12, 33, 59], we evaluate

Table 1: Accuracy (%) of different UDAs on Office-Home with ResNet-50 as backbone. Best in bold.

4 Experiments

In this work, actually, we are motivated by the reliable human prior knowledge on what should be aligned across domains to better assist classification task for UDA (i.e., task/classification-discriminative features), while excluding the interference from task-irrelevant ones. Accordingly, in our design, we obtain the prior/meta-knowledge for identifying task-discriminative features from the classification task (meta-train) and apply it to the domain alignment task (meta-test) to achieve task-oriented alignment.

4.1 Datasets and Implementation Details

Datasets. We use two commonly used benchmark datasets (i.e., Office-Home [60] and VisDA-2017 [46]) for SUDA and a large-scale dataset DomainNet [43] for MUDA and SSDA. 1) Office-Home [60] consists of images from four different domains: Art (Ar), Clipart (Cl), Product (Pr), and Real-World (Rw). Each domain contains 65 object categories in office and home environments. Following the typical settings [14, 12, 64, 40], we evaluate methods on one-source to one-target domain adaptation, resulting in 12 adaptation cases in total. 2) VisDA-2017 [46] is a synthetic-to-real dataset for domain adaptation with over 280,000 images across 12 categories, where the source images are synthetic and the target images are real collected from MS COCO dataset [37]. 3) DomainNet [43] is a large-scale dataset containing about 600,000 images across 345 categories, which span 6 domains with large domain gap: Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real (R), and Sketch (S). For MUDA, following the settings in [43, 66, 12, 33, 59], we evaluate methods on five-sources to one-target domain adaptation, resulting in 6 MUDA cases in total. For SSDA, we take the typical protocol in [22, 48, 12], where there are 7 SSDA cases conducted on the 4 sub-domains (i.e., C, R, P and S) with 126 sub-categories selected from DomainNet. All methods are evaluated under the one-shot/three-shot setting respectively, where besides unlabeled samples, one/three sample(s) per class in the target domain are available during training.

Implementation Details. We apply our ToAlign on top of two different baseline schemes: DANNP [14] and HDAs [12]. DANNP is an improved variant of the classical adversarial learning based adaptation method DANN [17, 18], where the domain discrimination \( D \) is conditioned on the predicted class probabilities. HDAs is a state-of-the-art adversarial training based method which leverages the domain-specific representations as heuristics to obtain domain-invariant representations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Ar→Cl</th>
<th>Ar→Pr</th>
<th>Ar→Rw</th>
<th>Cl→Ar</th>
<th>Cl→Pr</th>
<th>Cl→Rw</th>
<th>Pr→Ar</th>
<th>Pr→Cl</th>
<th>Pr→Rw</th>
<th>Rw→Ar</th>
<th>Rw→Cl</th>
<th>Rw→Pr</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source-Only</td>
<td>34.9</td>
<td>50.0</td>
<td>58.0</td>
<td>37.4</td>
<td>41.9</td>
<td>46.2</td>
<td>38.5</td>
<td>31.2</td>
<td>60.4</td>
<td>53.9</td>
<td>41.2</td>
<td>59.9</td>
<td>46.1</td>
</tr>
<tr>
<td>MCD/CVPR'18</td>
<td>48.9</td>
<td>68.3</td>
<td>74.6</td>
<td>61.3</td>
<td>67.6</td>
<td>68.8</td>
<td>57.0</td>
<td>47.1</td>
<td>75.1</td>
<td>69.1</td>
<td>52.2</td>
<td>79.6</td>
<td>64.1</td>
</tr>
<tr>
<td>CDAN/NeurIPS'18</td>
<td>50.7</td>
<td>70.6</td>
<td>76.0</td>
<td>57.6</td>
<td>70.0</td>
<td>70.0</td>
<td>57.4</td>
<td>59.9</td>
<td>77.3</td>
<td>79.0</td>
<td>56.7</td>
<td>81.6</td>
<td>65.8</td>
</tr>
<tr>
<td>ALDA/AAAI'20</td>
<td>53.7</td>
<td>70.1</td>
<td>76.4</td>
<td>60.2</td>
<td>72.6</td>
<td>71.5</td>
<td>51.8</td>
<td>51.9</td>
<td>77.1</td>
<td>71.1</td>
<td>56.5</td>
<td>82.1</td>
<td>66.6</td>
</tr>
<tr>
<td>SynNet/NeurIPS'18</td>
<td>47.7</td>
<td>72.9</td>
<td>78.5</td>
<td>64.2</td>
<td>71.3</td>
<td>74.2</td>
<td>63.6</td>
<td>47.6</td>
<td>79.4</td>
<td>73.8</td>
<td>50.8</td>
<td>82.6</td>
<td>67.2</td>
</tr>
<tr>
<td>TADA/AAAF'19</td>
<td>53.1</td>
<td>72.3</td>
<td>77.2</td>
<td>59.1</td>
<td>71.2</td>
<td>72.1</td>
<td>59.7</td>
<td>53.1</td>
<td>78.4</td>
<td>72.4</td>
<td>60.0</td>
<td>82.9</td>
<td>67.6</td>
</tr>
<tr>
<td>MDS/HCI'19</td>
<td>54.9</td>
<td>73.7</td>
<td>77.8</td>
<td>60.0</td>
<td>71.4</td>
<td>71.8</td>
<td>61.2</td>
<td>53.6</td>
<td>78.1</td>
<td>72.8</td>
<td>60.2</td>
<td>82.3</td>
<td>68.1</td>
</tr>
<tr>
<td>BMN/CVPR'18</td>
<td>56.2</td>
<td>73.7</td>
<td>79.0</td>
<td>63.1</td>
<td>73.6</td>
<td>74.0</td>
<td>62.4</td>
<td>54.8</td>
<td>80.7</td>
<td>72.4</td>
<td>58.9</td>
<td>83.5</td>
<td>69.4</td>
</tr>
<tr>
<td>GSDA/CVPR'20</td>
<td>61.3</td>
<td>76.1</td>
<td>79.4</td>
<td>65.4</td>
<td>73.3</td>
<td>74.3</td>
<td>65.0</td>
<td>53.2</td>
<td>80.0</td>
<td>72.2</td>
<td>60.6</td>
<td>83.1</td>
<td>70.3</td>
</tr>
<tr>
<td>GVBN/CVPR'18</td>
<td>57.0</td>
<td>74.7</td>
<td>79.8</td>
<td>64.6</td>
<td>74.1</td>
<td>74.6</td>
<td>65.2</td>
<td>55.1</td>
<td>81.0</td>
<td>74.6</td>
<td>59.7</td>
<td>84.3</td>
<td>70.4</td>
</tr>
<tr>
<td>E-Mix(AAAI'21)</td>
<td>57.7</td>
<td>76.6</td>
<td>79.8</td>
<td>63.6</td>
<td>74.1</td>
<td>75.0</td>
<td>63.4</td>
<td>56.4</td>
<td>79.7</td>
<td>72.8</td>
<td>62.4</td>
<td>85.5</td>
<td>70.6</td>
</tr>
<tr>
<td>MetaAlign(CVPR'21)</td>
<td>59.3</td>
<td>76.0</td>
<td>80.2</td>
<td>65.7</td>
<td>74.7</td>
<td>75.1</td>
<td>65.7</td>
<td>56.5</td>
<td>81.6</td>
<td>74.1</td>
<td>61.1</td>
<td>85.2</td>
<td>71.3</td>
</tr>
<tr>
<td>DANNP</td>
<td>54.2</td>
<td>70.0</td>
<td>77.6</td>
<td>62.3</td>
<td>72.4</td>
<td>73.1</td>
<td>61.3</td>
<td>52.7</td>
<td>80.0</td>
<td>72.0</td>
<td>56.8</td>
<td>83.1</td>
<td>67.9</td>
</tr>
<tr>
<td>DANNP+ToAlign</td>
<td>56.8</td>
<td>74.8</td>
<td>79.3</td>
<td>64.0</td>
<td>73.9</td>
<td>75.3</td>
<td>63.8</td>
<td>53.7</td>
<td>81.1</td>
<td>73.1</td>
<td>58.2</td>
<td>84.0</td>
<td>69.7</td>
</tr>
<tr>
<td>HDAs/NeurIPS'20</td>
<td>56.8</td>
<td>75.2</td>
<td>79.8</td>
<td>65.1</td>
<td>73.9</td>
<td>75.2</td>
<td>66.3</td>
<td>56.7</td>
<td>81.8</td>
<td>75.4</td>
<td>59.7</td>
<td>84.7</td>
<td>70.9</td>
</tr>
<tr>
<td>HDAs+ToAlign</td>
<td>57.9</td>
<td>76.7</td>
<td>80.8</td>
<td>66.7</td>
<td>75.6</td>
<td>77.0</td>
<td>67.8</td>
<td>57.0</td>
<td>82.5</td>
<td>75.1</td>
<td>60.0</td>
<td>84.9</td>
<td>72.0</td>
</tr>
</tbody>
</table>
### 4.2 Ablation Study

**Effectiveness of ToAlign on Different Baselines.** Our proposed ToAlign is generic and applicable to different domain adversarial training based baselines, where we focus on what features to align instead of the alignment methods. The last four rows in Table 1 show the ablation comparisons on Office-Home. Our ToAlign improves the accuracy of baseline DANNP and HDA by 2.0% and 1.1% respectively. As can be seen from the results in Table 1 Table 2 Table 3 and Table 5 our ToAlign can consistently bring significant improvement over the baseline schemes under different domain adaptation settings, i.e., SUDA, MUDA and SSDA. ToAlign enables the domain alignment task to proactively serve the classification task, resulting in more effective feature alignment for image classification.

**Effectiveness of Different Ways to Obtain Positive Features.** As mentioned in Sec. [3.2] we use $w_{cls}^p s w_{cls}$ as the attention weight (which conveys the classification prior/meta-knowledge) to derive positive feature $f_p$, where $s$ is a parameter to modulate the energy of $f_p$. We study the influence of $s$ under the setting of $Rw→C1$ on Office-Home for our scheme DANNP+ToAlign and illustrate...
the results in Table 3. As discussed around Eq. 5, we can use an adaptively calculated $s$, which achieves 2% improvement over the baseline on target test data. Moreover, we can treat $s$ as a preset hyper-parameter. We found that the performance drops drastically if $s$ is too small (e.g., $s = 1$). That is because the energy of the source positive feature will get too weak when $s$ gets too small (e.g., the source feature $f$’s average energy $E(f)$ is about 800; if $s = 1$, the source positive feature’s average energy $E(f_p)$ is about 2). Then, it would be ineffective to align the target with the source positive features. When $s$ is larger than 16, the performance significantly outperforms the baseline and approaches the result of using adaptive $s$. As an optional design choice, we could transform the weight $w^{cls}$ with certain activation function $\sigma(\cdot)$ such as Sigmoid or Softmax followed by a best selected scaling factor $s$, i.e., $w^{cls}_p = s \sigma(w^{cls})$. We found the results (i.e., 69.6/69.7 for Sigmoid/Softmax) are close to that without activation function. We reckon that what is more important is the relative importance among the elements in $w^{cls}$. For simplicity, we finally take the adaptive $s$ (cf. Eq. 5) for all experiments.

4.3 Comparison with the State-of-the-arts

Single Source Unsupervised Domain Adaptation (SUDA). We incorporate our $ToAlign$ into the recent state-of-the-art UDA method $HDA$, denoted as $HDA+ToAlign$. Table 1 shows the comparisons with the previous state-of-the-art methods on Office-Home. $HDA+ToAlign$ outperforms all the previous methods and achieves the state-of-the-art performance. It is noteworthy that $HDA+ToAlign$ achieves the best adaptation results on almost all the one-source to one-target adaptation cases thanks to the effective feature alignment for classification. The results on VisDA-2017 could be found in Appendix, where $HDA+ToAlign$ outperforms $HDA$ by 0.9%.

Multi-source Unsupervised Domain Adaptation (MUDA). Table 2 shows the results on DomainNet, where all the methods take ResNet-101 as the feature extractor. We build our Baseline based on $HDA$. For simplicity, we replace the multi-class domain discriminator in the original $HDA$ by a two-class one as in [58, 17, 66]. Note that CMSS [66] selects suitable source samples for alignment while our $ToAlign$ selects task-discriminative sub-feature for each sample for task-oriented alignment. Compared with Baseline, $ToAlign$ brings about 0.9% improvement and helps to achieve the best performance on this more challenging dataset.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source-Only</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADR [18]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MME [14]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENT [12]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANN [21]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDAN [NeurIPS'18]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GVBG [CVPR'20]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HDA+ToAlign</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Accuracy (%) of different three-shot SSDA methods on DomainNet with ResNet-34 as backbone. Best in bold.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source-Only</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADR [18]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MME [14]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENT [12]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CANN [21]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDAN [NeurIPS'18]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GVBG [CVPR'20]</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HDA+ToAlign</td>
<td>73.0</td>
<td>72.0</td>
<td>71.7</td>
<td>69.3</td>
<td>64.6</td>
<td>80.8</td>
<td>76.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Accuracy (%) of different one-shot SSDA methods on DomainNet with ResNet-34 as backbone. Best in bold.
Semi-supervised Domain Adaptation (SSDA). Table 5 and Table 6 show the results on one-shot and three-shot SSDA respectively, where all the methods use ResNet-34 as backbone. To compare with previous methods, we apply ToAlign on top of HDA. HDA+ToAlign outperforms HDA by 0.6%/1.2% for one-/three-shot settings, and surpasses all previous SSDA methods.

4.4 Complexity

In Table 4, we compare the training complexity and performance of ToAlign with baseline DANNP, and DANNP+MetaAlign [64] which incorporates meta-learning to coordinate the optimization of domain alignment and image classification. In contrast, inspired by the prior knowledge of what feature should be aligned to serve classification task, we distill such meta-knowledge from classification task and explicitly pass it to alignment task for classification-oriented alignment, eschewing complex optimization. Compared with baseline, ToAlign introduces negligible additional computational cost (only 7%) and occupies almost the same GPU memory as the baseline, which is much smaller than that of DANNP+MetaAlign, which almost doubles the computational cost due to its complex meta-optimization. Thanks to our explicit design which makes domain alignment effectively serve the classification task, our ToAlign achieves superior performance to MetaAlign.

4.5 Feature Visualization

We visualize the target feature response maps \( F \) (which will be pooled to be the input of the image classifier) of the Baseline (DANNP) and ToAlign in Figure 5. Baseline sometimes focuses on the background features which are useless to the image classification task, since it aligns the holistic features without considering the discriminativeness of different channels/sub-features. Thanks to our task-oriented alignment, in ToAlign, the features with higher responses are in general related to task-discriminative features, which is more consistent with human perception. More results can be found in the Appendix.

5 Conclusion

In this paper, we study what features should be aligned across domains for more effective unsupervised domain adaptive image classification. To make the domain alignment task proactively serve the classification task, we propose an effective task-oriented alignment (ToAlign). We explicitly decompose a feature in the source domain into a task-related feature that should be aligned and a task-irrelevant one that should be ignored, under the guidance of the meta-knowledge induced from the classification task itself. Extensive experiments on various datasets demonstrate the effectiveness of our ToAlign. In our future work, we will extend ToAlign to tasks beyond image classification, e.g., object detection and segmentation.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Key Research and Development Program of China 2018AAA0101400 and NSFC under Grant U1908209, 61632001, and 62021001.
References


