Exploring Latent Transferability of Feature Components

Anonymous Authors[1]

Abstract

Feature disentanglement techniques have been extensively employed to extract transferable features from non-transferable court parts. However, given the intricate interplay among high-dimensional features, the separated "non-transferable" features may still partially informative in most cases. Suppressing or disregarding them, as commonly employed in previous methods, may inadvertently overlook the inherent transferability. In this work, two concepts: Class and Domain Partial Transferable Feature (CPTF and DPTF), along with a novel feature disentanglement method are introduced. We systematically Explore their Latent Transferability (ELT) and dynamically utilize them based on their relevance. Since we don't rely on seeking a faultless feature disentanglement technique to thoroughly peel off the non-transferable features, as we believe it is challenging both theoretically and practically. Instead, we adopt a two-stage strategy consisting of separation and dynamic learning. Our approach is more practical and applicable to diverse and complex application scenarios. Extensive experimental results have proved its efficiency, even without more additional tricks. The code is available at https://github.com/njtjmc/ELT.

1. Introduction

Deep learning models exhibit considerable advantages when trained on large labeled datasets. However, their performances tend to deteriorate significantly when evaluated on an unseen domain. To address this issue, Transfer Learning (TL) has been developed extensively in recent years. Unsupervised Domain Adaptation (UDA) (Wang & Deng, 2018; Wilson & Cook, 2020; Ganin et al., 2016; Long et al., 2018; Na et al., 2021), as a subfield of TL, is dedicated to scenarios where the labeled training data and unlabeled testing data demonstrate disparate distributions, yet their underlying tasks remain consistent.

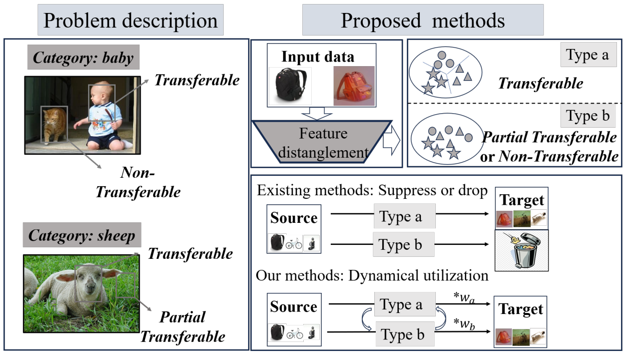


Figure 1. A description of motivation and the proposed method. The left side argues that the background may contain partially transferable information. For example, sheep and grass contain a certain relevance. The right is the innovation of our method. Existing methods predominantly suppress or even discard type b information. On the contrary, our approach achieves a dynamic balance by seamlessly integrating both types of elements.

In most UDA studies (Na et al., 2021; Chen et al., 2022; Rangwani et al., 2022), knowledge are transferred when ignoring the potential presence of detrimental information underlying them, which may negatively impact the performance. For example, the class feature or domain invariant feature is often seen as transferable, while the domain specific feature is often seen as nontransferable. Aligning indiscriminately poses the risk of erroneous pairing between them and causes the negative transfer problem.

To alleviate this problem, feature disentanglement technique has been widely used in recent UDA methods (Gao et al., 2022; Kong et al., 2022; Li et al., 2022; Deng et al., 2022; Wei et al., 2021). For example, some researches (Liu et al., 2018; Chang et al., 2019b; Wu et al., 2021) utilize GANs (Goodfellow et al., 2014) or VAEs (Kingma & Welling, 2013), while others (Bousmalis et al., 2016; Zhou et al., 2023) employ disentanglement loss functions to better extract diverse elements underlying the data. These methods all aim to separate transferable features from non-transferable components and address the misalignment issue by prioritizing transferable knowledge and suppressing non-

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

transferable counterparts. However, the intricate interplay between various latent elements creates complex coupling relationships, and a complete separation between them is highly challenging (Liu et al., 2019). As shown on the left side of Figure 1, the background may also contain partially transferable information like the sheep more likely appears in grass. We believe that in most existing feature disentanglement methods, the extracted domain-specific features may still retain partial transferability due to the complex coupling relationship between semantic and structural information in natural objects. Therefore, there is a risk of losing transferability and damaging the model's performance when transferring domain invariant features and suppressing domain-specific features.

In this work, we divide the original chaotic features into Class Partial Transferable Feature (CPTF) and Domain Partial Transferable Feature (DPTF). Unlike previous methods that categorize features as strictly transferable or non-transferable, we believe that both CPTF and DPTF contain partially useful information, and this assumption can better correspond to reality. Moreover, our work further encompasses an in-depth Exploration of the Latent Relevance (ELT) between them. Specifically, we propose a novel evaluation index for measuring transferability and dynamically integrate CPTF and DPTF into the subsequent model based on their transferability. The advantage of ELT over existing methods lies in the fact that it refrains from pursuing the complete elimination of non-transferable knowledge. Instead, ELT adopts a two-stage strategy consisting of separation and dynamic learning. This approach alleviates the need for the flawless disentanglement strategy and effectively provides an elevated level of flexibility. Moreover, this dynamic evaluation also makes the proposed method efficient to avoid the common negative transfer problem. The idea of this paper is shown in Figure 1. We make three significant contributions, which are outlined as follows:

- The roles of diverse feature elements in the transfer process are synthesized and two concepts: CPTF and DPTF are introduced along with their associations to the domain invariant and domain specific feature.

- We propose an efficient explicit disentangling method named ELT, which can dynamically incorporate the disentangled features into the knowledge transfer process by exploring the latent transferability and relevance between them.

- With the feature disentanglement and dynamic evaluation, ELT can achieve excellent performance even without using more additional tricks. The experimental results demonstrate that ELT can achieve significant improvements on multiple datasets.

2. Related Work

Unsupervised Domain Adaptation Unsupervised Domain Adaptation (UDA) enables models to learn representations of labeled data in the source domain, facilitating effective adaptation to unseen data in the target domain. One prominent approach of UDA is based on Domain Adversarial Training (DAT) (Ganin et al., 2016). DAT introduces an additional domain discriminator to distinguish between source domain and target domain. By training the model to deceive the discriminator, the domain-invariant representations that facilitate knowledge transfer are acquired. Subsequent studies have made significant progress by introducing innovations such as the class information based discriminator (Long et al., 2018), specialized batch normalization layers (Chang et al., 2019a), and the utilization of enhanced discrepancy measures (Zhang et al., 2019; Chen et al., 2022), among other techniques. To better utilize the target domain information, other researches utilize self-training or pseudo-labels technique. For example, MCC (Jin et al., 2020) minimizes pairwise class confusion on the target domain. Due to space limitations, we cannot introduce all works (Liang et al., 2021; Wang & Deng, 2018; Wilson & Cook, 2020; Ganin et al., 2016; Long et al., 2018; Na et al., 2021; Zhang et al., 2019; Chen et al., 2022; Rangwani et al., 2022) in detail. These studies continuously contribute to the UDA research and inject new vitality into the field.

Disentanglement Learning Disentanglement learning decomposes chaotic features into distinct components for better understanding underlying factors and has been extensively studied in UDA, especially in DAT-based methods. For example, DSN (Bousmalis et al., 2016) extracts image representations divided into two subspaces: one exclusive to each domain and another shared between domains. FFDI (Wang et al., 2022) disentangles features into high-frequency and low-frequency components. ToAlign (Wei et al., 2021) explicitly converts features to task-related features under the guidance of the prior knowledge induced from the classification task. However, the majority of these works (Bousmalis et al., 2016; Wang et al., 2022; Wei et al., 2021; Xie et al., 2022; Tian et al., 2022) tend to emphasize the domain invariant features and neglecting domain specific features. Consequently, there is a potential risk of losing valuable information embedded within the data. Furthermore, in these methods, feature disentanglement often involves adding extra model parameters, which can increase the complexity of the model.

3. Method

3.1. Recap of Preliminary Knowledge

Define source domain set and target domain set as $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s samples, $\{(x_i^t)\}_{i=1}^{n_t}$ with n_t samples.

Here, x_i^s and x_i^t represent labeled samples drawn from the source domain $\mathcal{D}_S$ and unlabeled samples drawn from the target domain $\mathcal{D}_T$, respectively. y_i^s denotes the ground-truth labels, which cover K classes. The deep recognition model can be decomposed into a feature extractor $G(\cdot)$ that extracts features $f \in \mathbb{R}^d$ from input data, i.e., $f = G(x)$, and a task classifier $C(\cdot)$ that generates corresponding predictions $\hat{y} \in \mathbb{R}^k$, i.e., $\hat{y} = C(f)$. An explanation of all characters utilized in this paper can be found in the supplementary file. The goal of UDA is to train the feature extractor G and task classifier C using the data from the source domain, and the trained model can perform well in the target domain. To achieve it, DAT-based UDA models minimize distribution discrepancy by adversarial neural networks (Goodfellow et al., 2014; Ganin et al., 2016) and learn transferable features. Their objective functions can be summarized as the following:

$$\mathcal{L}_{cls}^{dat}\left(x^s, y^s\right) = \mathbb{E}_{\left(x_i^s, y_i^s\right)\sim\mathcal{D}_S}\mathcal{L}_{ce}\left(C\left(G\left(x_i^s\right)\right), y_i^s\right) \tag{1}$$

$$\begin{aligned}\mathcal{L}_{adv}^{dat}\left(x^s, x^t\right) =& \mathbb{E}_{G\left(x_i^s\right)\sim\tilde{\mathcal{D}}_S}\log\left[D\left(G\left(x_i^s\right)\right)\right]\\ &+ \mathbb{E}_{G\left(x_i^t\right)\sim\tilde{\mathcal{D}}_T}\log\left[1 - D\left(G\left(x_i^t\right)\right)\right]\end{aligned} \tag{2}$$

where $\tilde{\mathcal{D}}_S$ and $\tilde{\mathcal{D}}_T$ denote the induced feature distribution of $\mathcal{D}_S$ and $\mathcal{D}_T$, respectively, and $\mathcal{L}_{ce}(\cdot,\cdot)$ is the cross-entropy loss function. The feature extractor G serves as a generator, with the objective of producing the transferable feature that confuse the domain discriminator. On the contrary, the domain discriminator attempts to distinguish the domain of the extracted features.

3.2. Motivation and Analysis

Motivation Re-clarification. In this section, we perform an in-depth analysis to identify the key factors influencing the effectiveness of knowledge transfer. Firstly, the raw chaotic data can be decomposed into the class-exclusive feature (f_{ce}), domain-exclusive feature (f_{de}), mixed feature (f_m), and noise (f_n).

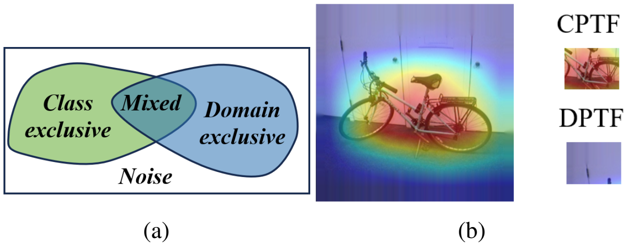


Figure 2. (a) Simplified venn diagram illustrating the relationship between different feature components. (b) Visualization of gradient-oriented decomposition for DPTF and CPTF.

The class-exclusive feature represents inherent object characteristics that remain consistent across domains. For example, the rule of "a human typically having two legs and one head" applies in most cases. The domain-exclusive feature primarily arises from disparities in temporal factors, spatial locations, or variations in data collection methodologies. For instance, oil paintings and color paintings have noticeable differences in terms of color palettes and texture. Moreover, partial domain-related features can also contribute to the identity of the category. For example, the sheep category is more commonly found in grass rather than in the ocean. The concept of mixed feature is employed to illustrate it. The mixed feature varies depending on the fluctuation of image semantic structure information. It can be closer to the class-exclusive feature in some samples or more similar to the domain-exclusive features in other samples. Finally, the noise primarily originates from random and uncontrollable factors, such as Gaussian noise resulting from random fluctuations or image distortions caused by camera defects during image capture. Their relationships are shown in Figure 2a, we employ two concepts: Class Partial Transferable Feature (CPTF: f_{cp}) to represent the collective representation of class-exclusive and mixed features (green part in the figure), and Domain Partial Transferable Feature (DPTF: f_{dp}) to depict the collective representation of domain-exclusive and mixed features (blue part in the figure). In contrast to conventional approaches that rigidly categorize features as transferable or non-transferable components, we argue that both CPTF and DPTF contain partially exploitable information. Given the complex interdependencies present within mixed information, this proposal is more aligned with real-world scenarios.

Feature Disentanglement Learning. Existing feature disentanglement methods often rely on GANs or VAE, which will greatly increase the complexity of the model. Building upon Grad-CAM (Selvaraju et al., 2017; Wei et al., 2021), we design a new feature disentanglement approach to derive CPTF and DPTF based on the classification meta-knowledge. In prior work (Wei et al., 2021), gradients are directly multiplied with original features to extract task-related/discriminative features that should be aligned, but this operation may exist the following two drawbacks. Firstly, the unidirectional filtering of task-related features can potentially compromise the richness of the original features. Secondly, it overlooks the influence of negative gradients. Such oversight may be harmful, particularly when there are prominent misleading objects in an image. We will explain it in the supplementary file. To address these limitations, we design two masks based on gradients' signs, enabling binary distinction of features. Unlike modifying numerical values of original features, our method prioritizes the preservation of their numerical values, mitigating the potential concern of feature degradation.

Specifically, the feature map $f \in \mathbb{R}^d$ generated by feature extractor is a tensor composed of d non-negative numbers.

The k-th value within the tensor is denoted with $f[k]$. The gradient of the score for class c with respect to the feature map is represented by $\alpha_f^c = \frac{\partial \hat{y}[c]}{\partial f} \in \mathbb{R}^d$. Specifically, $\alpha_f^c[k]$ represents the k-th value within the vector α_f^c. A positive value of $\alpha_f^c[k]$ indicates that the feature $f[k]$ positively contributes to the class c, while the negative value illustrate it belongs to others. Therefore, it can be used to make binary masks to filter ideal features from the original chaotic feature. Consequently, we calculate DPTF and CPTF as depicted in Eq.(3).

$$f_{dp} = \text{ReLU}(f \odot \mathbf{sgn}(-\alpha_f^{c^*}))\; c^* = \underset{1 \le c \le K}{\text{argmax}}(\hat{y}[c]), \tag{3a}$$

$$f_{cp} = \text{ReLU}(f \odot \mathbf{sgn}(\alpha_f^{c^*}))\quad c^* = \underset{1 \le c \le K}{\text{argmax}}(\hat{y}[c]), \tag{3b}$$

Where c^* corresponds to the category with the largest predicted value, and **sgn** is the sign (positive:1 or negative:-1) of the gradient and is utilized as the indicator vector here. The $\odot$ here is the Hadamard product, which denotes element-wise multiplication at corresponding positions. The symbol ReLU (Glorot et al., 2011) aims to ensure the output is non-negative. The result is visualized in Figure 2b: CPTF primarily corresponds to the foreground, while DPTF predominantly represents the background. This observation aligns with our expectations.

This work utilizes two independent domain discriminators, the one denoted as D_{dp} is for aligning DPTF and the another denoted as D_{cp} is for aligning CPTF. Among them, DPTF is directly input to D_{dp} (Ganin et al., 2016). In contrast, the multilinear conditioning information (Long et al., 2018) is integrated into CPTF, considering the strong correlation between its feature distribution and associated label distribution and the resulting information is then input to D_{cp}. The calculation formats of $D_{dp}(f, \hat{y})$ and $D_{cp}(f, \hat{y})$ are shown in Eq.(4a) and Eq.(4b), respectively. Here, the $\otimes$ is the multilinear map (Long et al., 2018).

$$\begin{aligned} D_{dp}(f, \hat{y}) &= D_{dp}(f_{dp}) \qquad c^* = \underset{1 \le c \le K}{\text{argmax}}(\hat{y}[c]) \\ &= D_{dp}(\text{ReLU}(f \odot (1 - \mathbf{sgn}(\frac{\partial \hat{y}[c^*]}{\partial f})))), \end{aligned} \tag{4a}$$

$$\begin{aligned} D_{cp}(f, \hat{y}) &= D_{cp}(\hat{y} \otimes f_{cp}) \qquad c^* = \underset{1 \le c \le K}{\text{argmax}}(\hat{y}[c]) \\ &= D_{cp}(\hat{y} \otimes \text{ReLU}(f \odot \mathbf{sgn}(\frac{\partial \hat{y}[c^*]}{\partial f}))). \end{aligned} \tag{4b}$$

3.3. Dynamic Utilization of CPTF and DPTF

This work uses two dynamic weights w_{dp} and w_{cp} to balance the alignment degree between CPTF and DPTF, and the adversarial loss is defined in Eq.(5).

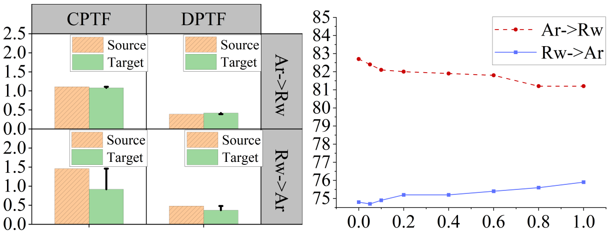
$$\mathcal{L}_{adv}(x^s, x^t) = w_{dp}\mathcal{L}_{dp}(x^s, x^t) + w_{cp}\mathcal{L}_{cp}(x^s, x^t), \tag{5a}$$

$$\begin{aligned} &\mathcal{L}_{(dp\,\mathbf{or}\,cp)}(x^s, x^t) = \\ &\quad \mathbb{E}_{G(x_i^s) \sim \tilde{\mathcal{D}}_S}\left[\log(D_{(dp\,\mathbf{or}\,cp)}(G(x_i^s), \hat{y}_i^s))\right] - \\ &\quad \mathbb{E}_{G(x_i^t) \sim \tilde{\mathcal{D}}_T}\left[\log(1 - D_{(dp\,\mathbf{or}\,cp)}(G(x_i^t), \hat{y}_i^t))\right]. \end{aligned} \tag{5b}$$

Compared with w_{cp}, an excessive small w_{dp} may result in potential loss of transferable information, while an overly large w_{dp} could introduce negative transfer problems, and we need to achieve a meticulous equilibrium between them. The goal of UDA is to transfer useful knowledge from the source domain to the target domain. This means that only when the knowledge from the source domain is more informative and reliable than the target domain, it should be transferred. Otherwise, the negative transfer problem may occur. Thus, the first step is to evaluate the informativeness of features extracted from the source and target domain. Then, a distance of them can be used to quantify the degree of transferability. Inspired from BSP (Chen et al., 2019), we can approximate the informativeness of features using their discriminability. However, the format for discriminability used in BSP, denoted as $\mathbf{Disc}_{bsp}$, exhibits a deficiency in stability. We have used experiments to prove it in the supplementary file. In this study, we present a simplified formula, as shown in (6).

$$\mathbf{Disc}(f) = \frac{\sum_j^K n_j \left(\boldsymbol{\mu}_j(f) - \boldsymbol{\mu}(f)\right)^\top \left(\boldsymbol{\mu}_j(f) - \boldsymbol{\mu}(f)\right)}{\sum_j^K \sum_{\mathbf{f} \in \mathcal{F}_j(f)} \left(\mathbf{f} - \boldsymbol{\mu}_j(f)\right)^\top \left(\mathbf{f} - \boldsymbol{\mu}_j(f)\right)}, \tag{6}$$

Where $\boldsymbol{\mu}_j(f)$ and $\boldsymbol{\mu}(f)$ are the centers of feature vectors for class j and all classes, respectively. n_j represents the number of samples in class j. $\mathcal{F}_j(f)$ denotes the set of features belonging to the class j. The numerator corresponds to the between-class scatter value, and the denominator represents the within-class scatter value. In contrast to $\mathbf{Disc}_{bsp}$, $\mathbf{Disc}$ circumvents the necessity of intricate matrix inversions, thereby enhancing scalability and leading to expedited computational speed.



(a) (b)

Figure 3. Test results of Ar → Rw and Rw → Ar tasks in Office-Home. (a)**Disc** of original feature, CPTF and DPTF in the source and target domains. (b) Performance curves of two tasks under different λ.

To further explore the relationship between $\mathbf{Disc}_{bsp}$ and transferability, supposing $\lambda = \frac{w_{cp}}{w_{dp}}$, we give the $\mathbf{Disc}$ of

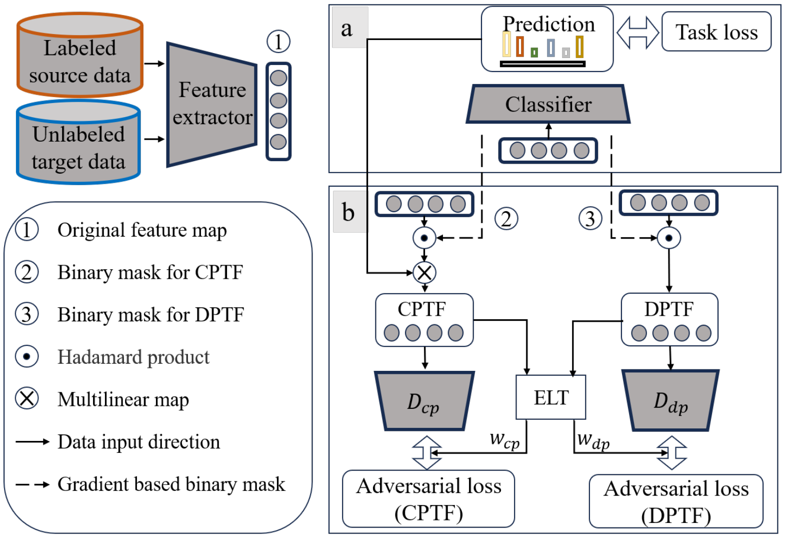


Figure 4. An overview of the ELT's model structure. The labeled source data and unlabeled target data are input to the feature extractor, then we can get the original feature map f. a shows the training process of the task classifier, the task classifier takes the feature map as input and generates the prediction $\hat{y}$. The classifier also outputs $\alpha_f^{c^*}$ for the highest predicted class c^*. b illustrates the process of dynamic feature alignment. On the left side, the feature map f undergoes a Hadamard product operation with the binary mask for CPTF. It then undergoes a multilinear operation (Ganin et al., 2016) with the prediction. The resulting CPTF is input to the Domain discriminator D_{cp} for alignment. On the right side, the feature map undergoes a Hadamard product operation with the binary mask for DPTF, resulting in the DPTF. The DPTF is then input to the domain discriminator D_{dp}. Simultaneously, the CPTF and DPTF are input to the ELT module for dynamic evaluation. The ELT module outputs the weights w_{cp} and w_{dp} to control the alignment balance of CPTF and DPTF. Notably, this figure does not include ReLU operation and the reverse gradient layer in the domain discriminator and CPTF or DPTF.

the original features, CPTF and DPTF of source and target domains under different subtasks and the corresponding performance under different λ in Figure 3. Notably, as the value of λ increases, the extent to which DPTF is transferred relative to CPTF also increases.

As shown in Figure 3a, we observe that CPTF has a higher **Disc** in the source domain compared to the target domain for both tasks. However, in the A2R task, DPTF exhibits a lower **Disc** in the source domain compared to the target domain, whereas in the R2A task, DPTF shows a higher **Disc** in the source domain compared to the target domain. This means unlike CDTF, which consistently exhibits stable transferability, DPTF's transferability varies across different scenarios.

$$\mathbf{Gap}(f_{cp}^s, f_{cp}^t) = \max\left(\frac{\mathbf{Disc}(f_{cp}^s) - \mathbf{Disc}(f_{cp}^t)}{\mathbf{Disc}(f_{cp}^s) + \mathbf{Disc}(f_{cp}^t)}, 0\right), \quad (7a)$$

$$\mathbf{Gap}(f_{dp}^s, f_{dp}^t) = \max\left(\frac{\mathbf{Disc}(f_{dp}^s) - \mathbf{Disc}(f_{dp}^t)}{\mathbf{Disc}(f_{dp}^s) + \mathbf{Disc}(f_{dp}^t)}, 0\right), \quad (7b)$$

$$w_{cp} = \gamma \frac{\mathbf{Gap}(f_{cp}^s, f_{cp}^t)}{\mathbf{Gap}(f_{dp}^s, f_{dp}^t) + \mathbf{Gap}(f_{cp}^s, f_{cp}^t) + \varepsilon}, \quad (7c)$$

$$w_{dp} = \gamma \frac{\mathbf{Gap}(f_{dp}^s, f_{dp}^t)}{\mathbf{Gap}(f_{dp}^s, f_{dp}^t) + \mathbf{Gap}(f_{cp}^s, f_{cp}^t) + \varepsilon}. \quad (7d)$$

This conclusion is supported by the results in Figure 3b, where smaller λ values lead to improved performance in the A2R task, while larger λ values result in higher performance in the R2A task. To be concluded, we can quantify transferability of CPTF and DPTF by utilizing the **Disc** distance of them between the source and target domain. Therefore, to achieve a dynamic balance between the alignment degree of CPTF and DPTF, w_{cp} and w_{dp} are calculated through Eq. (7). The γ is a hyperparameter which are set to 1 default and ε is a minimum value to avoid Nan.

Under this dynamic evaluation, the alignment process of DPTF and CPTF will be affected by their **Disc** status. Moreover, if the **Disc** of DPTF or CPTF of the target domain is larger than that of the source domain, the alignment process of them will be completely stopped. We name our method as Exploring Latent Transferability (ELT) of Feature Components. ELT is simple and effective, as it considers both the relationship between the source and target domains and the relationship between DPTF and CPTF. With the feature disentanglement and dynamic evaluation, ELT can achieve excellent performance on several datasets.

3.4. Adversarial Learning

Figure 4 provides an overview of the ELT model structure. It consists of a feature extractor G based on a pretrained ResNet, a classifier C comprising a fully connected layer and two domain discriminators w_{cp}, w_{dp} . In this manner, ELT can be trained using a min-max game framework:

$$\min_{G} \max_{C} \mathcal{L}_{adv}\left(x^{s}, x^{t}\right). \tag{8}$$

To further regularize the training process, we introduce the Minimum Class Confusion (MCC) term proposed by (Jin et al., 2020). This regularization term, denoted as $\mathcal{L}_{mcc}(x^t)$, is incorporated into the overall training objective. In summary, the total loss function utilized for optimizing the classification model can be expressed as follows:

$$\min_{C,G} \left\{\mathcal{L}_{cls}^{dat}\left(x^{s}, y^{s}\right) + \max_{C} \mathcal{L}_{adv}\left(x^{s}, x^{t}\right) + \mathcal{L}_{mcc}(x^{t})\right\}. \tag{9}$$

4. Experiments

4.1. Setup

Datasets: OfficeHome (Venkateswara et al., 2017), **DomainNet** (Peng et al., 2019), and **VisDA2017** (Peng et al., 2017) are used for conducting comparative experiments and evaluating the performance of ELT. The OfficeHome dataset includes 15,500 images divided into 65 classes, covering four domains: Art (Ar), Clipart (Cl), Product (Pr), and Real World (Rw). The DomainNet dataset contains a vast collection of 0.6 million images, distributed among 345 classes across six domains: infograph (inf), clipart (clp), painting (pnt), sketch (skt), real (rel), and quickdraw (qdr). The VisDA-2017 dataset is designed for sim-to-real domain adaptation and consists of approximately 280,000 images across 12 classes.

Implementation Details. We use the Transfer Learning-Library toolkit (Jiang et al., 2022; 2020) as the framework for implementing our proposed approach. For the experiments conducted on the OfficeHome and DomainNet, a pretrained ResNet-50 backbone is used and a pretrained ResNet-101 backbone is used in the experiments performed on the VisDA-2017. All the aforementioned backbone architectures are initialized with weights pretrained on the ImageNet dataset (Deng et al., 2009). In all experiments, we adopt a batch size of 32 and set the learning rate to 0.01.

4.2. Comparison Results

OfficeHome: As shown in Table 1, ELT achieve the best performance in subtasks of P→C and R→C. The average accuracy of all subtasks are 73.2%, it surpasses the accuracy of BiMem, DALN+MCC and AdD by 1.7%, 0.6% and 0.5%.

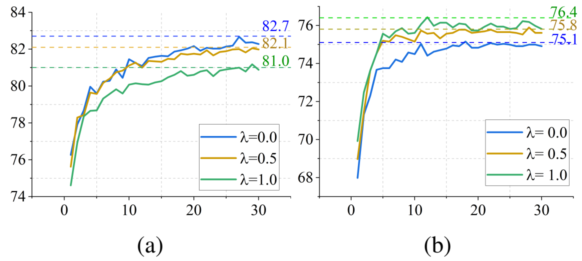
DomainNet: Table 2 shows the results of DomainNet dataset. ELT achieves the best performance with 31.6% average accuracy and outperforms the SoTA methods with identical backbones, surpasses the accuracy of MCD, CCM, SCDA by 11.1%, 3.1% and 1.4%.

VisDA-2017: Table 3 report the mean of the accuracy across classes on the VisDA-2017, and Table 4 is the overall accuracy of the dataset. On the average evaluation metric, ELT achieves a performance of 85.4%, and on the overall evaluation metric, ELT achieved a performance of 82.8%. ELT achieves the best performance compared to all existing methods.

4.3. Insight Analysis

The parameters w_{cp} and w_{dp} will be dynamic adjusted during training. In this section, we will empirically validate three key facts: (1) The substantial contribution of DPTF in transferring knowledge across domains. (2) The adaptability of ELT, allowing for dynamic adjustments during model training. (3) The effectiveness of this dynamic adjustment mechanism in extracting transferable information from DPTF.

Unignorable Contribution of DPTF. Our method innovatively pays attention to DPTF, which often be ignored in prior methods. To validate this assumption that DPTF may also retain partial useful information, we test the performance of the model under various λ values (the λ here are fixed and will not update during training). We conduct experiments on two tasks within the OfficeHome: A2R and R2A. Figure 5a and 5b show that smaller λ values lead to better performance in the A2R task, while larger λ values result in better performance in the R2A task. These experiments suggest that the DPTF extracted in the former task is detrimental, whereas in the latter task it is beneficial. This instability demonstrates that fixed λ are unable to adapt to diverse scenarios, underscoring the need and rationale for the dynamic evaluation method proposed in this paper.



(a) (b)

Figure 5. (a) The performance under different fixed λ in A2R task. (b) The performance under different fixed λ in R2A task.

Analysis of Dynamic Weights. Instead of a fixed λ, we propose a dynamic approach that seeks to achieve a harmonious equilibrium between CPTF and DPTF. To demonstrate its effectiveness, we plot the training curve for the Ar → Cl (A2C) task in Figure 6a. Significantly, during the initial phases of training, specifically when the number of training epochs is less than 10, the parameter w_{dp} exhibits a tendency to assume high values. This finding highlights

Table 1. Classification accuracy (%) on OfficeHome (ResNet50). The best performance is marked in bold.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
DANN (Ganin et al., 2016)	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN (Long et al., 2018)	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
MCD (Yang et al., 2022b)	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
MCC (Jin et al., 2020)	55.1	75.2	79.5	63.3	73.2	75.8	66.1	52.1	76.9	73.8	58.4	83.6	69.4
SDAT (Jin et al., 2020)	57.9	77.4	81.5	66.5	76.2	76.5	63.9	56.5	82.6	75.0	62.9	85.4	72.2
DALN (Chen et al., 2022)	57.8	**79.9**	82.0	66.3	76.2	77.2	66.7	55.5	81.3	73.5	60.4	85.3	71.8
DALN+MCC (Chen et al., 2022)	58.1	79.6	**83.7**	67.7	77.9	78.7	66.8	56.0	81.9	73.9	60.9	**86.1**	72.6
BiMem (Chen et al., 2022)	54.5	78.8	81.4	66.7	78.7	**79.6**	65.9	53.6	82.3	73.6	57.8	84.9	71.5
CDAN+MCC+f (Wang et al., 2023)	58.6	77.0	81.9	66.9	77.6	77.0	65.5	57.5	82.5	**76.0**	64.2	85.6	72.5
AaD (Yang et al., 2022a)	**59.3**	79.3	82.1	**68.9**	**79.8**	79.5	67.2	57.4	**83.1**	72.1	58.5	85.4	72.7
CMA (Wang et al., 2024)	58.1	78.7	80.0	68.2	77.2	78.5	**68.0**	54.2	80.4	73.5	57.8	83.4	71.5
ELT (ours)	59.2	78.9	83.1	67.3	78.5	78.5	65.4	**59.1**	82.4	75.4	**66.1**	85.7	**73.2**

Table 2. Classification accuracy (%) on DomainNet (ResNet50). The best performance is marked in bold.

MCD (Yang et al., 2022b)	clp	inf	pnt	qdr	rel	skt	Avg	CDAN (Long et al., 2018)	clp	inf	pnt	qdr	rel	skt	Avg	BNM (Cui et al., 2020)	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	15.4	25.5	3.3	44.6	31.2	24.0	clp	-	13.5	28.3	9.3	43.8	30.2	25.0	clp	-	12.1	33.1	6.2	50.8	40.2	28.5
inf	24.1	-	24.0	1.6	35.2	19.7	20.9	inf	18.9	-	21.4	1.9	36.3	21.3	20.0	inf	26.6	-	28.5	2.4	38.5	18.1	22.8
pnt	31.1	14.8	-	1.7	48.1	22.8	23.7	pnt	29.6	14.4	-	4.1	45.2	27.4	24.2	pnt	39.9	12.2	-	3.4	54.5	36.2	29.2
qdr	8.5	2.1	4.6	-	7.9	7.1	6.0	qdr	11.8	1.2	4.0	-	9.4	9.5	7.2	qdr	17.8	1.0	3.6	-	9.2	8.3	8.0
rel	39.4	17.8	41.2	1.5	-	25.2	25.0	rel	36.4	18.3	40.9	3.4	-	24.6	24.7	rel	48.6	13.2	49.7	3.6	-	33.9	29.8
skt	37.3	12.6	27.2	4.1	34.5	-	23.1	skt	38.2	14.7	33.9	7.0	36.6	-	26.1	skt	54.9	12.8	42.3	5.4	51.3	-	33.3
Avg	28.1	12.5	24.5	2.4	34.1	21.2	20.5	Avg	27.0	12.4	25.7	5.1	34.3	22.6	21.2	Avg	37.6	10.3	31.4	4.2	40.9	27.3	25.3
SCDA (Li et al., 2021)	clp	inf	pnt	qdr	rel	skt	Avg	CCM (Yu et al., 2023)	clp	inf	pnt	qdr	rel	skt	Avg	ELT (ours)	clp	inf	pnt	qdr	rel	skt	Avg
clp	-	18.6	39.3	5.1	55.0	44.1	32.4	clp	-	17.8	38.8	9.7	55.8	43.1	33.0	clp	-	**21.5**	40.4	**12.8**	**55.8**	**46.0**	**35.3**
inf	29.6	-	34.0	1.4	46.3	25.4	27.3	inf	28.2	-	32.4	4.0	43.6	24.7	26.6	inf	**35.3**	-	**34.4**	**5.5**	**47.7**	**27.6**	**30.1**
pnt	44.1	19.0	-	2.6	**56.2**	**42.0**	32.8	pnt	40.8	17.8	-	**5.6**	53.7	36.9	31.0	pnt	**45.6**	**21.5**	-	**5.6**	**56.2**	39.8	**33.8**
qdr	**30.0**	**4.9**	**15.0**	-	**25.4**	**19.8**	**19.0**	qdr	22.9	4.1	8.7	-	15.3	13.1	12.8	qdr	23.7	4.2	9.9	-	16.6	16.4	14.2
rel	54.0	22.5	51.9	2.3	-	42.5	34.6	rel	51.6	20.4	49.0	4.5	-	38.6	32.8	rel	**55.3**	**26.1**	**53.1**	**6.7**	-	**43.0**	**36.8**
skt	55.6	18.5	44.7	6.4	53.2	-	35.7	skt	53.3	16.2	42.4	10.3	52.6	-	35.0	skt	**58.9**	**22.0**	**46.8**	**13.7**	**54.9**	-	**39.3**
Avg	42.6	16.7	**37.0**	3.6	**47.2**	**34.8**	30.3	Avg	39.4	15.3	34.3	6.8	44.2	31.3	28.5	Avg	**43.8**	**19.1**	36.9	**8.8**	46.2	34.6	**31.6**

an interesting characteristic of the dynamic weighting factor, indicating its propensity to prioritize DPTF in the early stages of the training process. Subsequently, as the training process advances, we observe a gradual decrease in w_{dp}. This diminishing trend serves to effectively mitigate the potential risks associated with over-alignment problem. By dynamically adjusting the two weighting factors, our proposed approach ensures a balanced integration of CPTF and DPTF, promoting robust and discriminative representations throughout the training process.

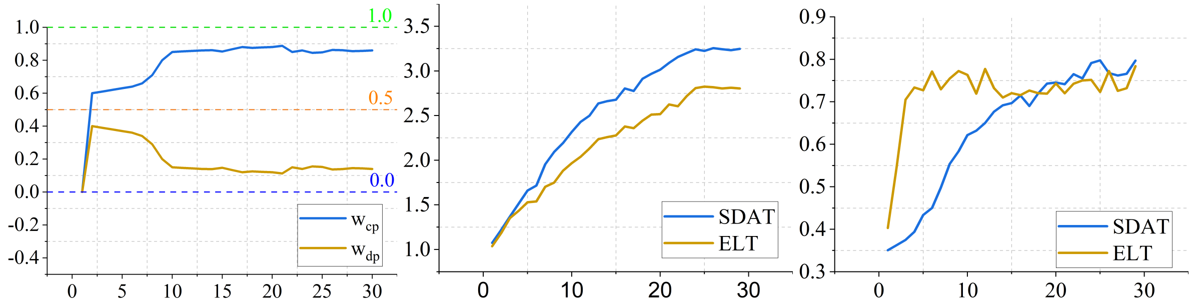
Efficient Utilization of DPTF. To substantiate the efficacy of ELT in facilitating the utilization of DPTF, we plot the curves of $\mathbf{Disc}$ of CPTF and DPTF during training. The comparison algorithm is chosen with SDAT, which is a representative UDA method of direct alignment (Rangwani et al., 2022). Figure 6b presents the discriminability analysis of CPTF, it shows the CPTF extracted by ELT exhibit relatively lower $\mathbf{Disc}$ in comparison to SDAT. Nevertheless, as illustrated in Figure 6c, the DPTF extracted by ELT demonstrate considerably higher $\mathbf{Disc}$ in comparison to the SDAT. It serves as a compelling testament to the effectiveness of our ELT in effectively leveraging and harnessing the discriminative power latent within DPTF. Furthermore, it is worth noting that the $\mathbf{Disc}$ of DPTF attains its highest point at approximately step 4-10 and subsequently exhibits a gradual decline before eventually stabilizing at a specific value. This observation provides empirical evidence supporting the effectiveness of our method in preventing the over-alignment problem.

These experiments provide evidence that ELT can address the limitations of existing approaches that only focus on CPTF and overlook the potential of DPTF. Additionally, our method effectively reduces the risk of excessive alignment of DPTF, avoiding the potential negative transfer problem.

t-SNE Visualization. To further illustrate the superiority of the ELT, we provide distributions visualization of the CPTF and DPTF. We use t-SNE (van der Maaten & Hinton, 2008) to visualize the feature representations obtained from SDAT and ELT in Figure 7. Both the SDAT and ELT methods demonstrate significant alignment performance in CPTF. However, an important distinction between ELT and SDAT lies in their treatment of DPTF. While the SDAT primarily concentrates on achieving alignment of CPTF, ELT also places emphasis on effectively aligning DPTF. The experiment demonstrate that by designing a dynamic evaluation, ELT can ensure a more comprehensive and robust alignment of both CPTF and DPTF, thereby enhancing the overall performance and adaptability of the model in domain

Table 3. Mean classification accuracy (%) on Visda2017 dataset (ResNet101). The best performance is marked in bold.

	plane	bcycl	bus	car	horse	knife	mcyle	persn	plant	sktb	train	truck	mean
DANN (Ganin et al., 2016)	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
CDAN (Long et al., 2018)	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
MCD (Yang et al., 2022b)	87.0	60.9	**83.7**	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
MCC (Jin et al., 2020)	88.1	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
SDAT (Jin et al., 2020)	95.8	85.5	76.9	69.0	**93.5**	97.4	88.5	78.2	93.1	**91.6**	86.3	55.3	84.3
DALN (Chen et al., 2022)	96.0	86.3	74.3	50.0	92.4	94.7	83.5	76.4	91.0	87.2	88.4	47.4	80.6
DALN+MCC (Chen et al., 2022)	**96.1**	82.7	76.8	71.4	92.5	96.8	88.2	81.3	92.2	88.7	84.1	53.7	83.7
SUDA (Zhang et al., 2022)	88.3	79.3	66.2	64.7	87.4	80.1	85.9	78.3	86.3	87.5	78.8	**74.5**	79.8
CCM (Yu et al., 2023)	95.0	**87.2**	83.1	**76.6**	92.1	96.9	88.6	**84.3**	**95.3**	85.9	**88.9**	45.5	85.0
ELT (ours)	96.0	86.3	80.0	74.5	**93.5**	**98.1**	**90.2**	79.4	93.9	**91.6**	84.3	55.7	**85.4**



(a) Training curve of w_{cp} and w_{dp} (b) Training curve of CPTF's **Disc** (c) Training curve of DPTF's **Disc**

Figure 6. Analysis of dynamic weights.

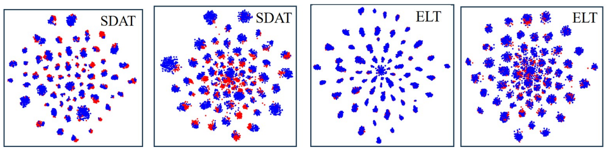
Table 4. Classification accuracy (%) on Visda2017 (ResNet101). The best performance is marked in bold.

Method	Synthetic->Real
DANN (Ganin et al., 2016)	57.4
MCD (Yang et al., 2022b)	71.4
CDAN (Long et al., 2018)	73.7
SDAT (Jin et al., 2020)	81.2
ELT (ours)	**82.8**

adaptation scenarios.

5. Conclusion

In this work, we introduce two new concepts called CPTF and DPTF, along with an efficient disentangling method. Subsequently, we present a novel method to evaluate their transferability based on the discriminability distance across domains. Finally, we dynamically incorporate varied latent elements into the knowledge transfer process. The proposed model is named as ELT, since ELT can automatically adjust the importance of internal feature transfer, making it a more practical and applicable UDA method suitable for diverse and complex application scenarios. ELT can help the model focus on useful information better and can alleviate the negative transfer problem simultaneously. Experiments have proven it can achieve ideal results in multiple data sets even without the other tricks. We demonstrate the transferability evaluation method proposed in this paper is not only limited to this work alone, but also can be extended to more works in the future, such as multi-source domain transfer problem.



(a) CPTF (b) DPTF (c) CPTF (d) DPTF

Figure 7. t-SNE visualizations of feature distributions of SDAT and ELT on the task Ar → Cl of OfficeHome. Blue and red points represent source and target features, respectively.

References

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.

Chang, W.-G., You, T., Seo, S., Kwak, S., and Han, B. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 7354–7362, 2019a.

Chang, W.-L., Wang, H.-P., Peng, W.-H., and Chiu, W.-C. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1909, 2019b.

Chen, L., Chen, H., Wei, Z., Jin, X., Tan, X., Jin, Y., and Chen, E. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7181–7190, 2022.

Chen, X., Wang, S., Long, M., and Wang, J. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1081–1090. PMLR, 09–15 Jun 2019.

Cui, S., Wang, S., Zhuo, J., Li, L., Huang, Q., and Tian, Q. 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, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Deng, W., Zhao, L., Liao, Q., Guo, D., Kuang, G., Hu, D., Pietikäinen, M., and Liu, L. Informative feature disentanglement for unsupervised domain adaptation. *IEEE Transactions on Multimedia*, 24:2407–2421, 2022. doi: 10.1109/TMM.2021.3080516.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Gao, Y., Chen, P., Gao, Y., Wang, J., Pan, Y., and Ma, A. J. Hierarchical feature disentangling network for universal domain adaptation. *Pattern Recognition*, 127:108616, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2022.108616.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In Gordon, G., Dunson, D., and Dudík, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jiang, J., Chen, B., Fu, B., and Long, M. Transfer-learning-library. `https://github.com/thuml/Transfer-Learning-Library`, 2020.

Jiang, J., Shu, Y., Wang, J., and Long, M. Transferability in deep learning: A survey. *arXiv preprint arXiv:2201.05867*, 2022.

Jin, Y., Wang, X., Long, M., and Wang, J. Minimum class confusion for versatile domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pp. 464–480. Springer, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kong, L., Xie, S., Yao, W., Zheng, Y., Chen, G., Stojanov, P., Akinwande, V., and Zhang, K. Partial disentanglement for domain adaptation. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11455–11472. PMLR, 17–23 Jul 2022.

Li, S., Xie, M., Lv, F., Liu, C. H., Liang, J., Qin, C., and Li, W. Semantic concentration for domain adaptation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9082–9091, 2021. doi: 10.1109/ICCV48922.2021.00897.

Li, Y., Chang, Y., Gao, Y., Yu, C., and Yan, L. Physically Disentangled Intra- and Inter-Domain Adaptation for Varicolored Haze Removal. pp. 5841–5850, 2022.

Liang, J., Hu, D., and Feng, J. Domain Adaptation With Auxiliary Target Domain-Oriented Classifier. pp. 16632–16642, 2021.

Liu, Y., Tian, X., Li, Y., Xiong, Z., and Wu, F. Compact Feature Learning for Multi-Domain Image Classification. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7186–7194, June 2019.

Liu, Y.-C., Yeh, Y.-Y., Fu, T.-C., Wang, S.-D., Chiu, W.-C., and Wang, Y.-C. F. Detach and adapt: Learning cross-domain disentangled deep representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8867–8876, 2018.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Na, J., Jung, H., Chang, H. J., and Hwang, W. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1094–1103, 2021.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment Matching for Multi-Source Domain Adaptation. pp. 1406–1415, 2019.

Rangwani, H., Aithal, S. K., Mishra, M., Jain, A., and Radhakrishnan, V. B. A closer look at smoothness in domain adversarial training. In *International Conference on Machine Learning*, pp. 18378–18399. PMLR, 2022.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Tian, Q., Zhu, Y., Sun, H., Chen, S., and Yin, H. Unsupervised domain adaptation through dynamically aligning both the feature and label spaces. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12): 8562–8573, 2022. doi: 10.1109/TCSVT.2022.3192135.

van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86): 2579–2605, 2008.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5385–5394, 2017.

Wang, H., Zheng, L., Zhao, H., Li, S., and Li, X. Unsupervised domain adaptation with class-aware memory alignment. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2024. doi: 10.1109/TNNLS.2023.3238063.

Wang, J., Du, R., Chang, D., Liang, K., and Ma, Z. Domain generalization via frequency-domain-based feature disentanglement and interaction. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 4821–4829, 2022.

Wang, M. and Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

Wang, Z., Chen, L., and Wang, F.-Y. Fuzzy inference attention module for unsupervised domain adaptation. *IEEE Transactions on Fuzzy Systems*, pp. 1–13, 2023. doi: 10.1109/TFUZZ.2023.3332751.

Wei, G., Lan, C., Zeng, W., Zhang, Z., and Chen, Z. Toalign: task-oriented alignment for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:13834–13846, 2021.

Wilson, G. and Cook, D. J. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Wu, A., Han, Y., Zhu, L., and Yang, Y. Instance-invariant domain adaptive object detection via progressive disentanglement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4178–4193, 2021.

Xie, Q., Li, Y., He, N., Ning, M., Ma, K., Wang, G., Lian, Y., and Zheng, Y. Unsupervised Domain Adaptation for Medical Image Segmentation by Disentanglement Learning and Self-Training. *IEEE Transactions on Medical Imaging*, pp. 1–1, 2022. ISSN 1558-254X. doi: 10.1109/TMI.2022.3192303.

Yang, S., Wang, Y., Wang, K., van de Weijer, J., and Jui, S. Local prediction aggregation: A frustratingly easy source-free domain adaptation method. *arXiv preprint arXiv:2205.04183*, 2022a.

Yang, Y., Kim, T., and Wang, G. Multiple classifiers based adversarial training for unsupervised domain adaptation. In *2022 19th Conference on Robots and Vision (CRV)*, pp. 40–47, 2022b.

Yu, Z., Li, J., Zhu, L., Lu, K., and Shen, H. T. Classification certainty maximization for unsupervised domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(8):4232–4243, 2023. doi: 10.1109/TCSVT.2023.3242614.

Zhang, J., Huang, J., Tian, Z., and Lu, S. Spectral unsupervised domain adaptation for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9829–9840, 2022.

Zhang, Y., Liu, T., Long, M., and Jordan, M. Bridging theory and algorithm for domain adaptation. In *International conference on machine learning*, pp. 7404–7413. PMLR, 2019.

Zhou, Q., Gu, Q., Pang, J., Lu, X., and Ma, L. Self-adversarial disentangling for specific domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8954–8968, 2023. doi: 10.1109/TPAMI.2023.3238727.