Supplementary for Exploring Latent Transferability of Feature Components

Anonymous Authors[1]

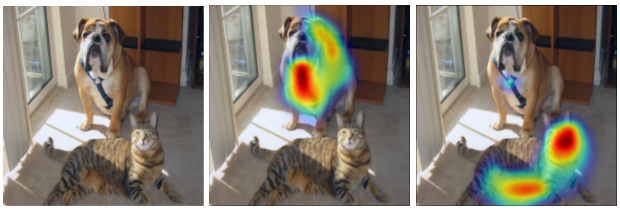
In this supplementary document, Section 1 provides all the notations in the paper. Section 2 elucidates the limitations of direct multiplying the gradient with the original feature. Section 3 provides visualizations of the feature distributions for the original feature, Class Partial Transferable Feature (CPTF), and Domain Partial Transferable Feature (DPTF) across 12 subtasks in OfficeHome dataset. Section 4 illustrates the strength of the separate alignment method compared to the direct alignment method. In Section 5, we compare our proposed discriminability ($\mathbf{Disc}$) with discriminability in BSP ($\mathbf{Disc}_{bsp}$) (Chen et al., 2019) to demonstrate the stability and efficacy of ours. Section 6 delves into alternative approaches for computing transferability, exploring potential methods beyond the scope of our current study. Section 7 presents a detail description of the experimental setup and configuration used in our study.

1. Notation Table

Table 1 contains all the notations used in the paper.

2. Impact of Negative Gradients

ToAlign (Wei et al., 2021) extracts the task-related feature by multiplying the gradient with the original feature directly, and the task-related feature is considered to be transferable. However, it may exist the following methods: Firstly, this rough approach can cause the model to overlook transferable information in the background. Additionally, it is also prone to the influence of negative gradients. The first drawback has been explained in detail in the paper, and the second one will be illustrated here. Supposing an image contains both "cat" and "dog" objects, as shown in Figure 1a. If the image is predicted as a "cat", then these features related to the "dog" regions would generate strong negative gradients, as shown in Figure 1b. Conversely, if the image is predicted as a "dog", these features associated with the "cat" regions would yield strong negative gradients, as shown in Figure 1c.



(a) Original Image (b) Cat Counterfactual (c) Dog Counterfactual

Figure 1. Analysis of the impact of the negative gradient

In this case, if we multiply the original feature and the gradient directly, these features that correspond to strong negative gradients will also be strengthened (Because its value will also be reinforced). Therefore, directly multiplying gradients with original features carries an inherent risk of unintentionally amplifying incorrect task-related features. Although this phenomenon may not be prominent in publicly available datasets characterized by high data purity, it remains a critical factor that cannot be overlooked when considering practical applications.

3. Visualization of Feature Distribution

In order to prove that our extracted CPTF and DPTF are more informative compared with the original features Here, we visualize the distribution of original feature, CPTF, and DPTF across all 12 subtasks in the OfficeHome. As illustrated in Figure 2, the extracted CPTF exhibit lower noise compared to the original feature and although the DPTF contain a huge amount of noise, they still retain some useful information, as the class boundary are still vaguely present. This phenomenon not only demonstrates the reliability of our feature disentanglement method, but also supports another point in our paper: background also possess partially transferable information.

4. Comparison between Separate and Direct Alignment

We made a schematic diagram to illustrate the difference between separate and direct alignments. As shown in the left of Figure 3, if features from the source and target domain are aligned directly, CPTF can be mistakenly matched with DPTF. However, the strategy of disentangling first and then

[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.

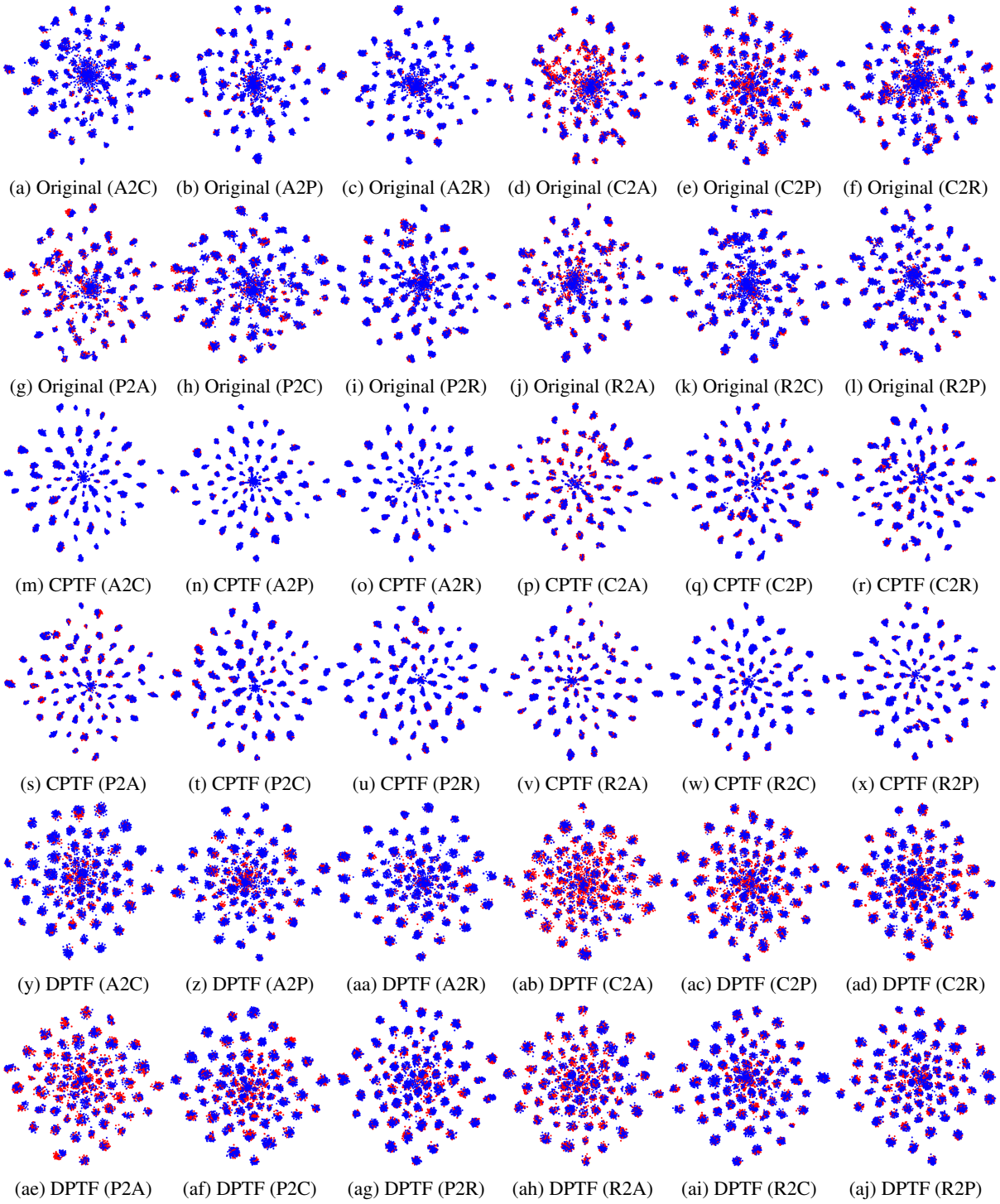


Figure 2. Visual distribution figures of the original feature, CPTF and DPTF of the source domain and the target domain under different tasks. The diagrams are arranged from left to right and from top to bottom, with the red color representing the source domain and the blue color representing the target domain. All the model are trained in the source domain only without using any domain adaptation method.

Table 1. The notations used in the paper and the corresponding meaning.

Symbol	Description
$\mathcal{D}_{\mathcal{S}}$, $\mathcal{D}_{\mathcal{T}}$	Distributions for source domain and target domain
$\tilde{\mathcal{D}}_S$, $\tilde{\mathcal{D}}_T$	Induced distributions for source domain, target domain
$G(.;\theta_G)$, $C(.;\theta_C)$, $D(.;\theta_D)$	Feature extractor; Task-specific classifier; Domain discriminator
$\hat{y}_i^s$, $\hat{y}[c]$	Prediction of i-th sample in source domain; Prediction corresponding to c-th category
x_i^s,x_i^t	i-th sample in source domain; i-th sample in target domain
f^s, f^t,$f[k]$	Feature of source domain, Feature of target domain, k-th value in the feature
f_{cp}, f_{dp}	CPTF, DPTF,
α_f^c	Gradient of the score for class c with respect to the feature

Table 2. $\mathcal{A}$-distance of original feature, CPTF and DPTF

Task	Ar2Cl	Ar2Pr	Ar2Rw	Cl2Ar	Cl2Pr	Cl2Rw	Pr2Ar	Pr2Cl	Pr2Rw	Rw2Ar	Rw2Cl	Rw2Pr	Avg
f	1.47	1.14	0.53	1.39	1.19	1.26	1.39	1.28	0.85	0.93	1.35	0.72	1.13
f_{cp}	1.09	1.02	0.68	0.97	0.93	1.03	1.16	1.23	0.74	0.76	1.03	0.58	0.94
f_{dp}	1.07	0.99	0.64	0.85	0.84	0.87	1.23	1.04	0.75	0.71	0.87	0.50	0.86

aligning separately dynamically can effectively circumvent the occurrence of such mismatching problems, as shown in the right of Figure 3.

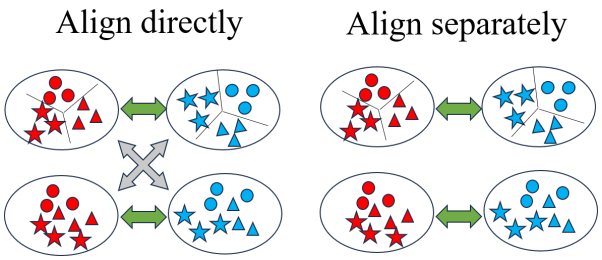


Figure 3. Comparison between directly alignment and separately alignment. The upper part with distinct boundaries is CPTF, and the lower part is DPTF. Red represent the source domain, blue represent the target domain

5. Comparison Between Disc and Disc_{bsp}

In this section, we will demonstrate the superior stability of our proposed **Disc** compared to $\textbf{Disc}_{bsp}$ (Chen et al., 2019). The calculation of $\textbf{Disc}_{bsp}$ involves determining the between-class variance $\mathbf{S}_{bsp,b}$ and the within-class variance $\mathbf{S}_{bsp,w}$:

$$\mathbf{S}_{bsp,b}(f) = \sum_{j=1}^{K} n_j \left(\boldsymbol{\mu}_j(f) - \boldsymbol{\mu}(f)\right)\left(\boldsymbol{\mu}_j(f) - \boldsymbol{\mu}(f)\right)^{\top} \tag{1a}$$

$$\mathbf{S}_{bsp,w}(f) = \sum_{j=1}^{K} \sum_{\mathbf{f}\in\mathcal{F}_j(f)} \left(\mathbf{f} - \boldsymbol{\mu}_j(f)\right)\left(\mathbf{f} - \boldsymbol{\mu}_j(f)\right)^{\top} \tag{1b}$$

Where K is the number of classes, n_j represents the number of samples in class j, $\mathcal{F}_j(f)$ denotes the set of features belonging to the class j. $\boldsymbol{\mu}_j(f)$ and $\boldsymbol{\mu}(f)$ are the centers of the feature vectors in class j and in all classes respectively. $\mathbf{S}_{bsp,b}$ and $\mathbf{S}_{bsp,w}$ represent the total variance of the feature vectors across different classes and in the same class, respectively. Then, the discriminability criterion based on LDA (noa, 2011) is shown as following:

$$\arg\max_{\mathbf{W}} J(\mathbf{W}) = \frac{\operatorname{tr}\left(\mathbf{W}^{\top}\mathbf{S}_{bsp,b}\mathbf{W}\right)}{\operatorname{tr}\left(\mathbf{W}^{\top}\mathbf{S}_{bsp,w}\mathbf{W}\right)} \tag{2}$$

The optimal solution to the above optimization problem is the K (set to the number of classes) the largest singular values of $\mathbf{S}_{bsp,w}^{-1}\mathbf{S}_{bsp,b}$, found by the Singular Value Decomposition (SVD).

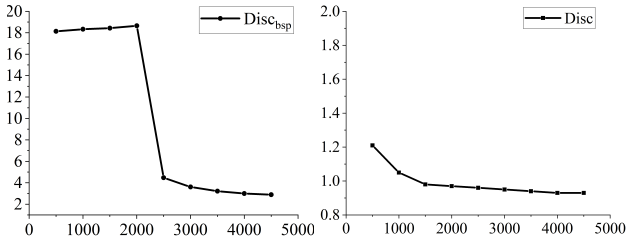
$$\mathbf{S}_{bsp,w}^{-1}\mathbf{S}_{bsp,b} = \mathbf{U}\sum\mathbf{V}^{\top} \tag{3}$$

The optimal solution is $\mathbf{W}^* = \mathbf{U}$.

However, we have observed substantial variability in the $\textbf{Disc}_{bsp}$, particularly when the length of the input matrix varies. To clarify it, we conduct experiments to compare the results of $\textbf{Disc}_{bsp}$ and **Disc** for the same input matrix. Since the value of $\textbf{Disc}_{bsp}$ is too large when the length of the input matrix is small, we chose the logarithmic line chart to plot its results. As shown on the left of Figure 4, the value of $\textbf{Disc}_{bsp}$ changes drastically when the length of the input matrix is small, especially when it is below the dimension of the input matrix (2048 in the experiment). The right side represents the value of **Disc**, which is more stable compared to $\textbf{Disc}_{bsp}$. Even with a large variation in the length of the input matrix, the value of **Disc** changes only slightly. In contrast to $\textbf{Disc}_{bsp}$, **Disc** circumvents the necessity of intricate matrix inversions, thereby enhancing scalability and leading to expedited computational speed.

Table 3. Discriminability (**Disc**) of original feature, CPTF and DPTF in the source and target domains

Task	Ar2Cl	Ar2Pr	Ar2Rw	Cl2Ar	Cl2Pr	Cl2Rw	Pr2Ar	Pr2Cl	Pr2Rw	Rw2Ar	Rw2Cl	Rw2Pr
f^s	0.68	0.66	0.67	0.58	0.58	0.62	1.00	0.99	0.99	0.94	0.95	0.91
f^s_{cp}	1.11	1.08	1.11	0.98	0.98	1.02	1.52	1.51	1.52	1.46	1.46	1.42
f^s_{dp}	0.39	0.37	0.39	0.39	0.39	0.43	0.46	0.47	0.47	0.48	0.48	0.44
f^t	0.34	0.52	0.58	0.31	0.42	0.49	0.33	0.30	0.54	0.45	0.35	0.70
f^t_{cp}	0.80	1.05	1.08	0.84	0.91	1.04	0.80	0.70	1.03	0.92	0.82	1.24
f^t_{dp}	0.40	0.46	0.42	0.45	0.42	0.49	0.37	0.33	0.39	0.37	0.41	0.45



(a) $\mathbf{Disc}_{bsp}$ (b) **Disc**

Figure 4. Comparison between $\mathbf{Disc}_{bsp}$ and **Disc**. The test dataset is OfficeHome, the source domain is Rw and the target domain is Pr. The input matrix is in the format $\mathbb{R}^{n\times 2048}$, where n is the length of the input matrix and is also the horizontal axis. The vertical axis of (a) is the logarithmic value of $\mathbf{Disc}_{bsp}$, while the vertical axis of (b) is the value of **Disc**.

Table 4. Architecture used for the task-specific classifier and the Domain discriminator. K is the number of classes. Both classifier and discriminator will take input from feature generator.

Layer	Output Shape
Task-specific Classifier	
-	Bottleneck Dimension
Linear	K
Domain Discriminator	
-	Bottleneck Dimension
Linear	1024
Batch Norm	1024
ReLU	1024
Linear	1024
Batch Norm	1024
ReLU	1024
Linear	1

6. Further discussion about Transferability

In addition to discriminabiltiy, $\mathcal{A}$-distance (Ben-David et al., 2010) is another metric often used to compute transferability. Here, we conduct experiments to calculate the $\mathcal{A}$-distance of original feature, CPTF and DPTF of 12 subtasks in the OfficeHome. The experimental results contradict our expectation that CPTF should have greater transferability and a smaller $\mathcal{A}$-distance compared to DPTF, as shown in Table 2. The original features have the largest distances, followed by CPTF, while DPTF has the smallest distances. This shows that using $\mathcal{A}$-distance to measure transferability is not reliable, because it may be affected by the numerical strength of the features.

Moreover, to provide a more comprehensive analysis of the **Disc** in this paper, we also give the **Disc** of 12 subtasks in the OfficeHome. The results are shown in Table 3. A larger **Disc** value of feature in the source domain indicates more reliable information, while a smaller **Disc** value of feature in the target domain suggests a greater need for knowledge from the source domain. It can be found that the **Disc** of CPTF is significantly larger than that of DPTF in multiple tasks, which is because CPTF couples a large amount of category information, while DPTF mainly has low information background. However, a few tasks, such as R2A and R2C, still have high **Disc** DPTF in source domain, which indicates that there is some information available for them to be exploited. The experiments demonstrate that compared with $\mathcal{A}$-distance, the proposed **Disc** for measuring transferability is more stable.

7. Experimental Details

7.1. Architecture OF Domain Discriminator

We use the batch normalization layer in domain discriminator, which was done in the library (Jiang et al., 2022). Table 4 shows the architecture of the feature classifier and domain discriminator.

7.2. Settings

In all experiments, we train the ETL using the smooth domain adversarial training strategy (Rangwani et al., 2022). Since ELT incorporates MCC (Jin et al., 2020) as an additional loss function, it also includes a temperature parameter to control the strength of uncertainty reweighting. OfficeHome: For the OfficeHome dataset, we train ELT for 30 epochs, with 1000 iterations per epoch, using a batch size of 32 and a learning rate of 0.01. We set the momentum parameter to 0.9 and use a weight decay of 0.001. The tem-

Table 5. Parameter Tuning of γ

	OfficeHome	DomainNet	Visda2017
γ=0.5	72.8	31.0	84.8
γ=1.0	73.2	31.2	85.4
γ=1.5	72.9	31.5	85.2
γ=2.0	72.5	31.6	84.6

perature parameter (Jin et al., 2020) is set to 2.5, and the bottleneck dimension for the features is set to 2048.

VisDA-2017: For the VisDA-2017 experiments, we utilize a ResNet-101 backbone initialized with ImageNet weights. Center Crop augmentation is applied during training. Both algorithms use a bottleneck dimension of 256. Additionally, we set the temperature parameter to 3.0 and the learning rate to 0.002.

DomainNet: For the DomainNet experiments, we employ a ResNet-50 backbone initialized with ImageNet weights. We conduct all experiments for 30 epochs, with 2500 iterations per epoch. The remaining parameters are the same as those used for the OfficeHome dataset.

7.3. Parameter Tuning

The γ parameter regulates the trade-off between classification and transfer tasks. We evaluate the performance of ELT with different parameter values on three datasets. Specifically, we set γ to 0.5, 1, 1.5, and 2. For the OfficeHome and Visda2017 datasets, we measured the classification accuracy across all subtasks. As for the Visda2017 dataset, we calculated the mean classification accuracy. Table 5 shows that $\gamma = 1$ is the best parameter for OfficeHome and Visda2017 datasets, and $\gamma = 2$ is the best parameter for DomainNet.

References

Introduction to Statistical Pattern Recognition. In *Statistical Pattern Recognition*. John Wiley & Sons, Ltd, 2011. ISBN 978-1-119-95295-4. doi: 10.1002/9781119952954.ch1. Section: 1 keywords = Bayes theorem, decision theory, discriminant function, probability density functions, regression analysis, statistical pattern recognition, pages = 1–32,.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F. C., and Vaughan, J. W. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010. URL https://api.semanticscholar.org/CorpusID:8577357.

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.

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.

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.

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.