Supplementary for Exploring Latent Transferability of Feature Components

Zhengshan Wang[a], Long Chen[a,*], Juan He[a], Fei-Yue Wang[b]

[a]*Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China*
[b]*State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China*

In this supplementary document, Section 1 provides all the notations in the paper. Section 2 provides visualizations of the feature distributions for the original feature, Partially Transferable Class Feature (PTCF), and Partially Transferable Domain Feature (PTDF) across 12 subtasks in OfficeHome dataset. In Section 3, we give the formula of BSP (**Disc**$_{bsp}$) [1]. Section 4 gives the reason for using Hadamard product in this paper. Section 5 delves into alternative approaches for computing transferability, exploring potential methods beyond the scope of our current study.

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

Symbol	Description
$\mathcal{D}_S$, $\mathcal{D}_T$	Distributions for source domain and target domain
$\tilde{\mathcal{D}}_S$, $\tilde{\mathcal{D}}_T$	Induced distributions for source domain and target domain
$G(\cdot)$, $C(\cdot)$, $D(\cdot)$	Feature extractor; Task classifier; Domain discriminator
$D^{cf}(\cdot,\cdot)$, $D^{df}(\cdot,\cdot)$	Domain discriminator of PTCF and PTDF
$\hat{y}_i^s$,$\hat{y}_i^t$	Prediction of i-th sample in source domain or target domain
$\hat{y}_i[c]$	Prediction corresponding to c-th category of i-th sample
x^s,x^t	A set of samples of source domain; a set of samples of target domain
x_i^s,x_i^t	i-th sample in source domain; i-th sample in target domain
f, f^s, f^t	A set of feature; a set of features of source domain; a set of features of target domain
f_i,$f_i[k]$	i-th feature, k-th value in the i-th feature
f^{cf}, f^{df}	Partially Transferable Class Feature (PTCF); Partially Transferable Domain Feature (PTDF)
$\alpha_{f_i}^c$	Gradient of the score for class c with respect to the feature f_i
$\hat{y}_{ij}^t$, $\hat{Y}_{ij}$	The raw probability and rescaled probability of the i-th sample being part of the j-th class
$T^{cf}(x_i)$, $T^{df}(x_i)$	The process of extracting PTCF and PTDF from the data x_i
$\otimes$	Multilinear map[2]

*Corresponding author.
Email address: longchen@umac.mo (Long Chen)

Table 2: $\mathcal{A}$-distance of original feature, PTCF and PTDF

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
$T^{cf}(x)$	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
$T^{df}(x)$	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

Table 3: Discriminability (**Disc**) of original feature, PTCF and PTDF 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
$T^{cf}(x^s)$	1.11	1.08	1.11	0.98	0.98	1.02	1.52	1.51	1.52	1.46	1.46	1.42
$T^{df}(x^s)$	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
$T^{cf}(x^t)$	0.80	1.05	1.08	0.84	0.91	1.04	0.80	0.70	1.03	0.92	0.82	1.24
$T^{df}(x^t)$	0.40	0.46	0.42	0.45	0.42	0.49	0.37	0.33	0.39	0.37	0.41	0.45

1. Notation Table

Table 1 contains all the notations used in the paper.

2. Visualization of Feature Distribution

In order to prove that our extracted PTCF and PTDF are more informative compared with the original features Here, we visualize the distribution of original feature, PTCF, and PTDF across all 12 subtasks in the OfficeHome.

As illustrated in Figure 1, the extracted PTCF exhibit lower noise compared to the original feature and although the PTDF 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.

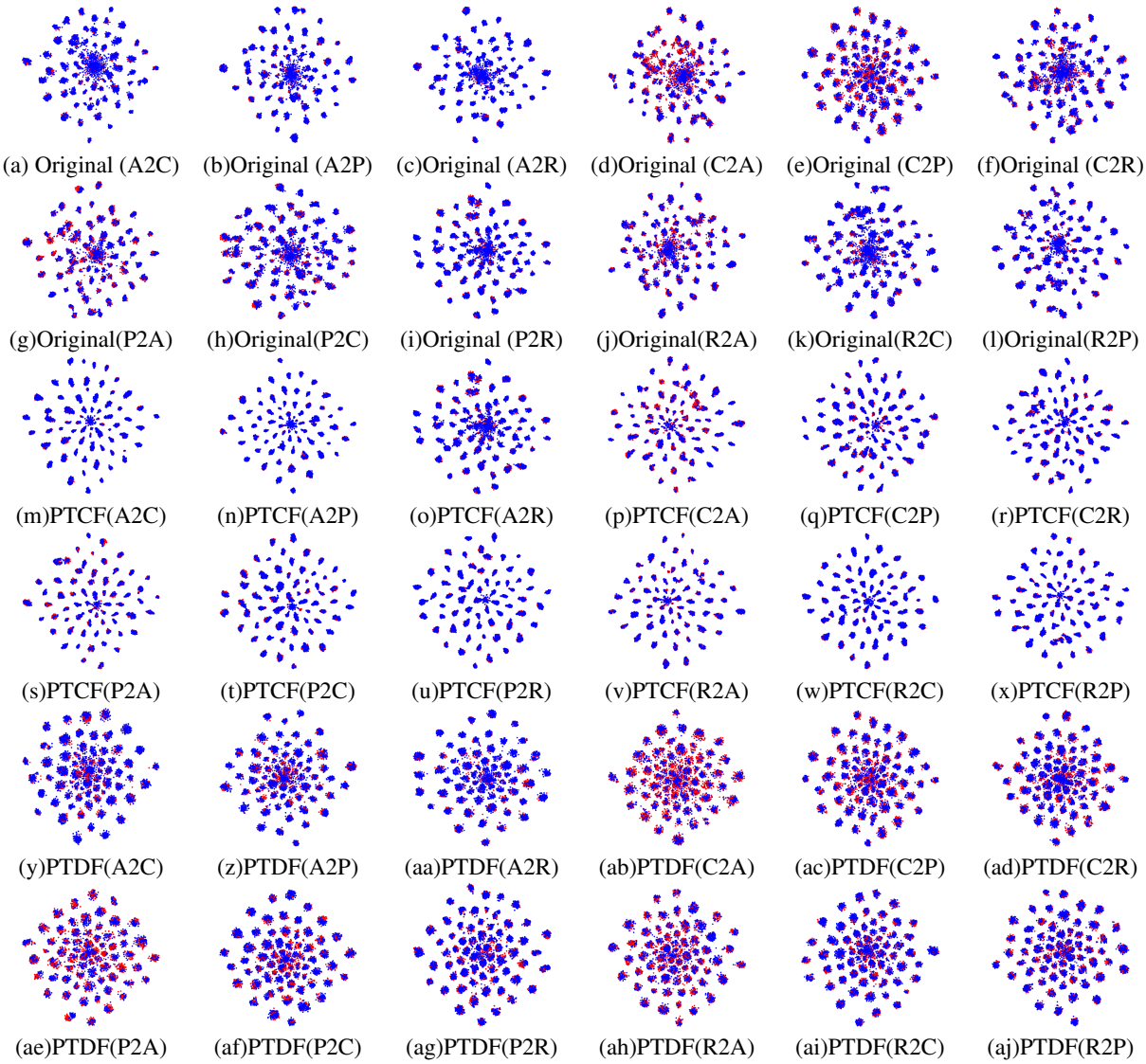

Figure 1: Visual distribution figures of the original feature, PTCF and PTDF of the source domain and the target domain under different tasks. The diagrams are arranged from left to right and from top to bottom, with the red color representing the source domain and the blue color representing the target domain. All the model are trained in the source domain only without using any domain adaptation method.

3. Calculation Formulas for $\mathbf{Disc}_{bsp}$

The calculation of $\mathbf{Disc}_{bsp}$ 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 [3] is shown as following:

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

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 $\mathbf{W}^{*} = \mathbf{U}$.

4. Hadamard product

In this paper, PTCF and PTDF are filtered out from the original features by Hadamard product, which is shown in Figure 1. The reason can be summarized as follows::

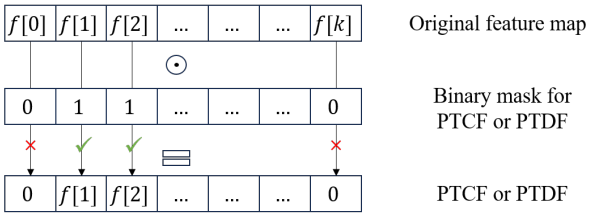

Figure 2: Hadamard product.

1. The Hadamard product is an element-wise operation, meaning it allows for the multiplication of corresponding elements of two tensors (original features and binary mask in this paper). This is particularly useful in the context of feature disentanglement as it ensures that each dimension of the original features is multiplied with the corresponding element of the binary mask, which indicates whether the feature belongs to PTCF or PTDF.
2. The Hadamard product is an element-wise operation, meaning it allows for the multiplication of corresponding elements of two tensors (original features and binary mask in this paper). This is particularly useful in the context of feature disentanglement as it ensures that each dimension of the original features is multiplied with the corresponding element of the binary mask, which indicates whether the feature belongs to PTCF or PTDF.
3. The Hadamard product is a simple operation that is easy to implement and computationally efficient, which is important for large-scale data processing and real-time application scenarios.
4. Unlike modifying numerical values of the original features, our method preserves their numerical values, avoiding the potential feature degradation problem.

5. Further discussion about Transferability

In addition to discriminabiltiy, $\mathcal{A}$-distance [4] is another metric often used to compute transferability. Here, we conduct experiments to calculate the $\mathcal{A}$-distance of original feature, PTCF and PTDF of 12 subtasks in the OfficeHome. The experimental results contradict our expectation that

PTCF should have greater transferability and a smaller $\mathcal{A}$-distance compared to PTDF, as shown in Table 2. The original features have the largest distances, followed by PTCF, while PTDF 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 PTCF is significantly larger than that of PTDF in multiple tasks, which is because PTCF couples a large amount of category information, while PTDF mainly has low information background. However, a few tasks, such as R2A and R2C, still have high **Disc** PTDF 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.

6. Implementation Details

We use the Transfer Learning-Library toolkit [5] as the framework for implementing our proposed approach. The batch normalization layer in the library [5] will be used in two domain discriminators. Table 4 shows the architecture of the task classifier and two domain discriminators. 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 temperature parameter T [6] 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 T to 3.0 and

Table 4: Architecture used for the task classifier and two domain discriminators. K is the number of classes. Both classifier and discriminator will take input from feature extractor.

Layer	Output Shape
Task Classifier	
-	Bottleneck Dimension
Linear	K
Domain Discriminator D^{df} for PTDF	
-	Bottleneck Dimension
Linear	1024
Batch Norm	1024
ReLU	1024
Linear	1024
Batch Norm	1024
ReLU	1024
Linear	1
Domain Discriminator D^{cf} for PTCF	
-	Bottleneck Dimension
Linear	1024*K
Batch Norm	1024
ReLU	1024
Linear	1024
Batch Norm	1024
ReLU	1024
Linear	1

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 1000 iterations per epoch. The remaining parameters are the same as those used for the OfficeHome dataset.

References

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

[2] M. Long, Z. Cao, J. Wang, M. I. Jordan, Conditional adversarial domain adaptation, Advances in neural information processing systems 31 (2018).

[3] Introduction to Statistical Pattern Recognition, in: Statistical Pattern Recognition, John Wiley & Sons, Ltd, 2011. Section: 1 keywords = Bayes theorem, decision theory, discriminant function, probability density functions, regression analysis, statistical pattern recognition, pages = 1–32,.

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

[5] J. Jiang, Y. Shu, J. Wang, M. Long, Transferability in deep learning: A survey, arXiv preprint arXiv:2201.05867 (2022).

[6] Y. Jin, X. Wang, M. Long, J. Wang, 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, Springer, 2020, pp. 464–480.