Supplementary material for "Collaborative Learning with Disentangled Features for Zero-shot Domain Adaptation"

Won Young Jhoo, Jae-Pil Heo
Sungkyunkwan University

{jhoowy, jaepilheo}@skku.edu

Algorithm 1: Learning procedure of our method

```
Input  : labeled datasets $X_s^{ir}, X_s^r, X_t^{ir}$
Output: trained feature extractors $G_c, G_d$; trained
        classifiers $C_r, C_{ir}$; trained refine module $R$
for number of training iterations do
    Sample mini-batch from $X_s^{ir}, X_s^r, X_t^{ir}$;
    // Domain Disentanglement
    Update $D$ by Eq.3;
    Update $C_r, C_{ir}$ by Eq.4, 5;
    Update $G_c, G_d$ by Eq.6, 7;
    // Task Disentanglement
    Freeze others and update $C_r, C_{ir}$ by Eq.9, 10;
    Freeze others and update $G_d$ by Eq.8;
    // Collaborative Learning
    Update $G_c, G_d, C_r, C_{ir}, R$ by Eq.13;
end
```

1. Details of learning procedure

In this section, we describe a detailed learning procedure of the proposed framework. As shown in algorithm 1, the feature disentanglement and collaborative learning are performed at every iteration.

Domain disentanglement During the domain disentanglement stage, we update all networks except refine module via cross-entropy and adversarial losses. This can be implemented by adding a gradient reversal layer between G_c and D, and back-propagate all losses from Eq. 1, 2, 4, 5.

Task disentanglement In the task disentanglement stage, the feature extractor G_d and the classifiers C_r, C_{ir} are updated by different loss functions so we must freeze other networks and update each component separately.

Collaborative learning The collaborative learning stage is the same as general classification network training procedures that use a gradient descent method, and the only difference is the network architecture.

ToI	D_M		D_F		D_E		# params
IrT	D_F	D_E	D_M	D_E	D_M	D_F	
Add	86.9	96.5	66.2	72.1	84.3	67.7	0
FC^1	68.0	93.5	54.8	64.3	74.9	64.8	2.66M
FC^2	88.5	97.7	61.4	74.0	84.4	72.8	78.7M
Ours	**93.3**	**97.9**	**67.7**	**76.3**	**86.4**	**74.1**	0.5M

Table 1: Comparison with different refine module architectures in the domain (G $\rightarrow$ C) setting. Add: add two features, FC^1: single FC layer with $128 \times 3 \times 3$ feature size, FC^2: FC layer with the original feature size.

2. Ablation studies

Refine module There could be various alternative methods for the collaborative learning stage that fuses two disentangled feature representations. For example, the proposed refine module can be replaced by fully connected layers or a simple binary operation. We tested three types of refine modules in some X-NIST domain adaptation experiment settings to find the best architecture for the refine module.

We first used a single fully connected layer as a refine module to maximize the expression capability of the module, but it requires too many parameters to recover the original feature size that is $128 \times 7 \times 7$ in our implementation for X-NIST experiments. Then we changed the feature size to $128 \times 3 \times 3$ to reduce the computational costs and optimization difficulty, but it also drops the overall performances significantly.

Then we tried the simplest fusing operation as a refine module, which adds the two disentangled feature representations. It is very easy to implement and has neglectable computational costs, but cannot effectively fuse the disentangled feature representations compared to the proposed attention mechanism.

ToI / IrT	D_M		D_F		D_E		Avg
	D_F	D_E	D_M	D_E	D_M	D_F	
No Refine	68.6	96.6	57.3	73.3	81.9	71.7	74.9
$\lambda_r = 1$	61.9	97.4	57.2	72.9	85.5	76.8	75.3
$\lambda_r = 2$	91.8	97.6	58.4	75.7	84.6	**82.7**	81.8
$\lambda_r = 3$	**93.3**	97.9	**67.7**	**76.3**	**86.4**	74.1	**82.6**
$\lambda_r = 4$	88.1	**98.0**	65.5	73.7	86.6	69.9	80.3
$\lambda_r = 5$	85.4	97.9	67.0	72.5	84.4	70.9	79.7

Table 2: Comparison with different λ_r in the domain (G $\rightarrow$ C) setting.

Domain	ToI / IrT	D_M		D_F		D_E	
		D_F	D_E	D_M	D_E	D_M	D_F
G $\rightarrow$ C	RS	92.8	97.5	**69.7**	73.3	**87.8**	**79.4**
	LM	**93.3**	**97.9**	67.7	**76.3**	86.4	74.1
G $\rightarrow$ E	RS	89.9	97.1	64.8	67.9	85.4	81.6
	LM	**92.9**	**98.9**	**65.0**	**74.4**	**91.1**	**82.9**

Table 3: Ablation study for label matching. RS: random sampling, LM: label matching sampling.

The reason why we choose the transformer-based architecture [1] is that it can effectively capture the spatial attention from disentangled feature representations with a reasonable number of parameters, and Table 1 shows the classification accuracy comparison against the alternative implementations.

Hyper-parameter As mentioned in Section 3.2 of the paper, we use hyper-parameter λ_r in Eq. 13 to balance the losses of the collaborative learning stage and disentanglement stage, and Table 2 shows the comparison with different values of λ_r. Improper λ_r can lead the classifiers focus only on f_c or f_r, and setting $\lambda_r = 1$ shows similar performances with when the collaborative learning is not applied. We choose $\lambda_r = 3$ as it shows the best result overall.

Label Matching We matched the labels of each mini-batch as described in Section 4.2 of the paper. This label matching sampling method helps to find the domain shift in some domain adaptation settings, and we reported the effectiveness of it in Table 3. While it does not increases the performance in every experiment, but we decide to use the label matching sampling as it shows better performances than random sampling in overall settings.

3. Dataset description

We randomly split ToI and IrT classes in the Office-Home experiments. We report the ToI classes of each split in this section.

Split 0 ”File Cabinet”, ”ToothBrush”, ”Pen”, ”Flowers”, ”Batteries”, ”Backpack”, ”Sneakers”, ”Computer”, ”Toys”, ”Oven”

Split 1 ”Glasses”, ”Flipflops”, ”Monitor”, ”Hammer”, ”Radio”, ”Sink”, ”Ruler”, ”Shelf”, ”Eraser”, ”Curtains”

Split 2 ”Calendar”, ”Screwdriver”, ”Marker”, ”Candles”, ”Mop”, ”Fork”, ”Bike”, ”Folder”, ”Spoon”, ”Bottle”

Split 3 ”Keyboard”, ”Exit Sign”, ”Fan”, ”Knives”, ”TV”, ”Clipboards”, ”Refrigerator”, ”Bed”, ”Speaker”, ”Telephone”

Split 4 ”Notebook”, ”Drill”, ”Laptop”, ”Scissors”, ”Mug”, ”Lamp Shade”, ”Couch”, ”Helmet”, ”Mouse”, ”Postit Notes”

Split 5 ”Soda”, ”Desk Lamp”, ”Paper Clip”, ”Trash Can”, ”Chair”, ”Alarm Clock”, ”Webcam”, ”Table”, ”Calculator”, ”Kettle”

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2