Supplementary Material: Mutual Learning of Joint and Separate Domain Alignments for Multi-Source Domain Adaptation

Yuanyuan Xu[1,2] Meina Kan[1,2] Shiguang Shan[1,2,3] Xilin Chen[1,2]
[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Peng Cheng Laboratory, Shenzhen, China
yuanyuan.xu@vipl.ict.ac.cn {kanmeina,sgshan,xlchen}@ict.ac.cn

In supplementary material, we present more experimental analyses and show detailed model architectures used in this paper. Specifically, Sec.1 and Sec.2 conduct complementarity analyses at the levels of sample and category on DomainNet dataset, respectively. Sec.3 shows the model architectures when training on DomainNet, Office-31, and Digits-five datasets.

1. Complementarity Analysis at the Level of Sample

For the level of sample, as shown in Fig.1 in this supplementary material, the proportion of complementary test samples to the total test samples is calculated for the two branches on DomainNet. Specifically, 'joint' represents the proportion of test samples that the joint alignment branch predicts correctly but the separate alignment branch makes mistakes, while 'separate' means the ratio of test samples in the opposite case. As can be seen from Fig.1, for each branch, there is a fair number of samples it can predict correctly while the other branch can't. This indicates that the two branches are complementary in their advantages of classifying different samples, so it is reasonable to design an algorithm to combine their advantages.

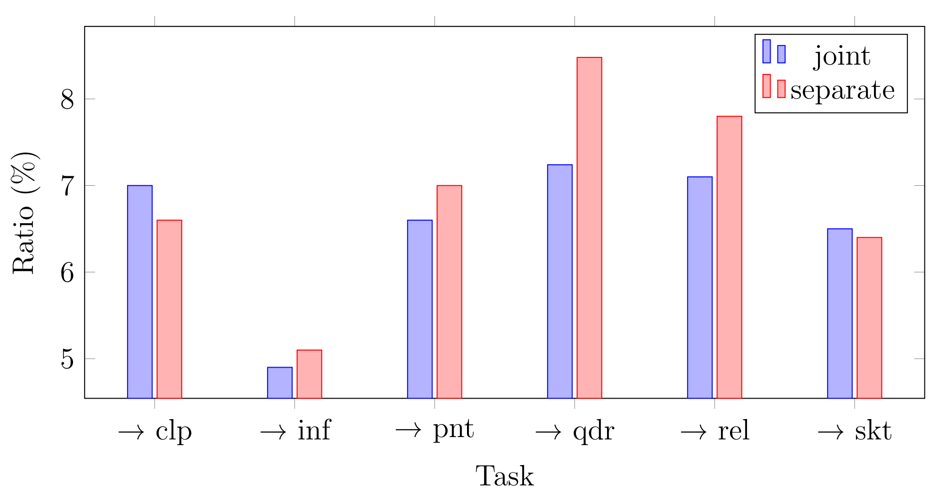
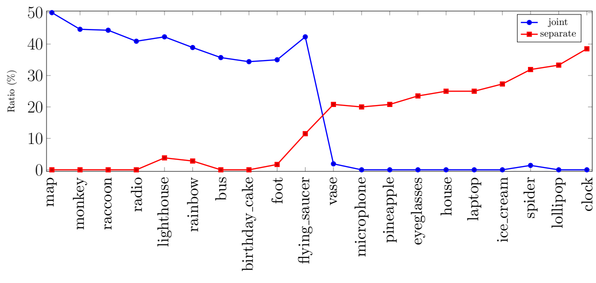


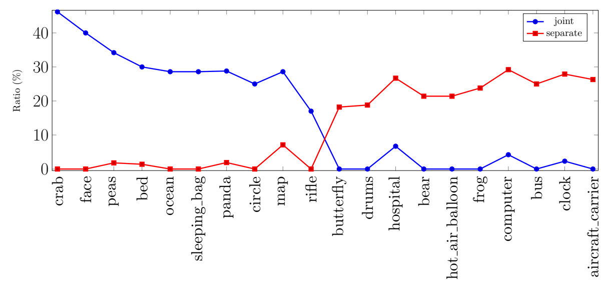
Figure 1: Complementarity analysis of the joint and separate alignment branches at the level of sample on DomainNet.

2. Complementarity Analysis at the Level of Category

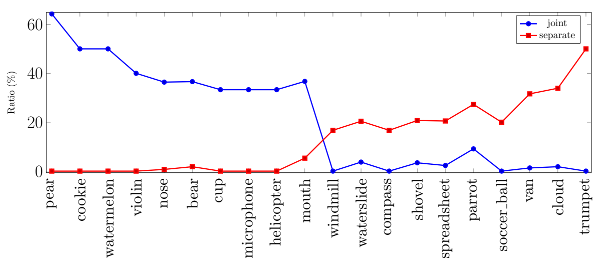
For the level of category, as shown in Fig.2 in this supplementary material, the ratio of complementary test samples is calculated for each category on DomainNet. Specifically, the numerator is the test samples of a specific category that can be correctly predicted by only one branch, and the denominator is the total test samples of that category. Similarly, 'joint' and 'separate' mean that the joint and the separate branches are correct, respectively. For each task, we select 20 categories with obvious predicting complementarities between the two branches. As shown in Fig.2a, on the task of '$\rightarrow$ clp', the joint alignment branch outperforms the separate alignment branch for categories such as 'map' and 'monkey', while the separate alignment branch significantly performs better for those categories such as 'clock' and 'lollipop'. Similar phenomena can be observed for other tasks. These show that different branches are good at predicting different categories, so the performance will improve when combining both of them.



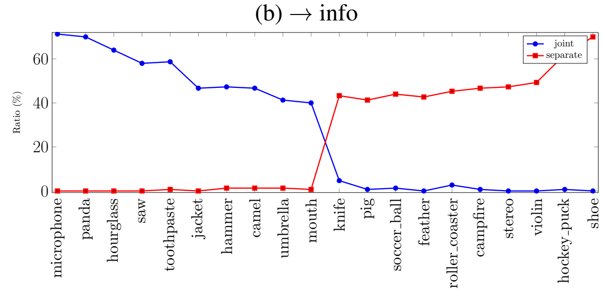
(a) $\rightarrow$ clp



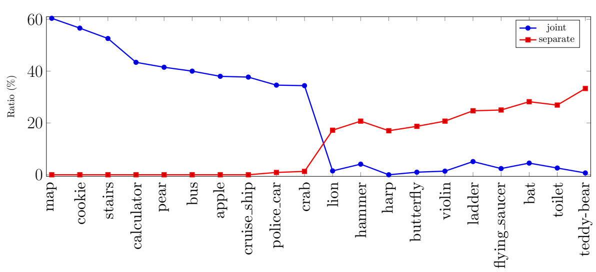
(b) $\rightarrow$ info



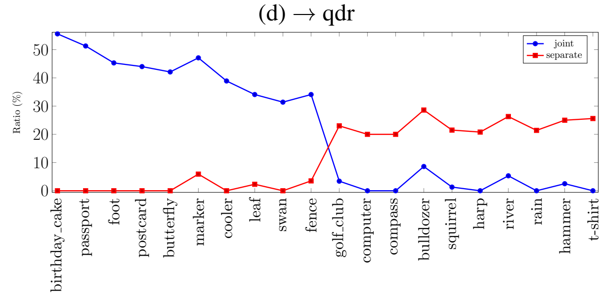
(c) $\rightarrow$ pnt



(d) $\rightarrow$ qdr



(e) $\rightarrow$ rel



(f) $\rightarrow$ skt

Figure 2: Complementarity analysis of the joint and separate alignment branches at the level of category on DomainNet.

3. Model Architecture

Tab.1, Tab.2, and Tab.3 show the model architectures when training on DomainNet, Office-31, and Digits-five, respectively, where 'FC(N)' represents a linear layer with N output nodes. Take Tab.1 for DomainNet as an example, the feature extractors include a shared ResNet-101 backbone followed by six branch-specific layers, one for the joint alignment branch and five for the separate alignment branch. In addition, one discriminator and one classifier are displayed. Note that there is a total of six classifiers with the same structure and so as the discriminators.

Table 1: Network Architecture for DomainNet

Feature Extractors						**Discriminator**	**Classifier**
ResNet-101 (backbone)						FC(1024), ReLU, Dropout	FC(345)
FC(256),BN	FC(256), BN	FC(256), BN	FC(256), BN	FC(256), BN	FC(256), BN	FC(1024), ReLU, Dropout	
						FC(1)	

Table 2: Network Architecture for Office-31

Feature Extractors			Discriminator	Classifier
ResNet-50 (backbone)			FC(1024), ReLU, Dropout	FC(31)
FC(256), BN	FC(256), BN	FC(256), BN	FC(1024), ReLU, Dropout	
			FC(1)	

Table 3: Network Architecture for Digits-five

Feature Extractors					Discriminator	Classifier
LeNet (backbone)					FC(1024), ReLU, Dropout	FC(10)
FC(256), BN	FC(256), BN	FC(256), BN	FC(256), BN	FC(256), BN	FC(1024), ReLU, Dropout	
					FC(1)	