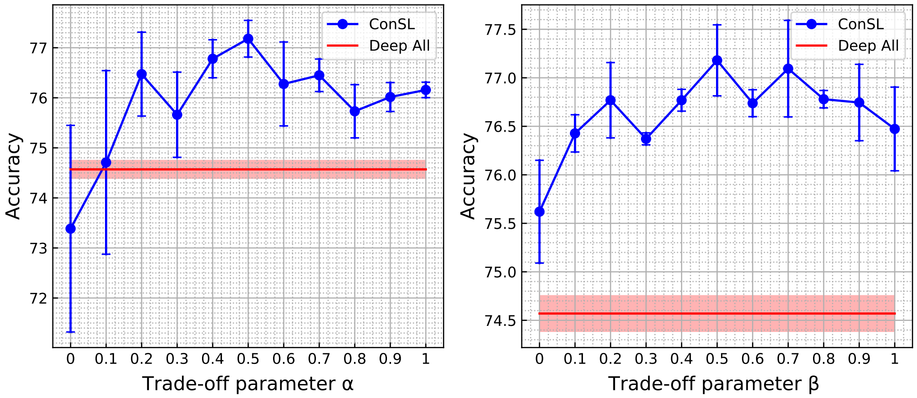

Fig. 3: The mean and standard deviation results of JCS and Deep All with regard to various trade-off parameters. The reported result is the global average over all the target domain cases. The red line represents our Deep All from Table 1.

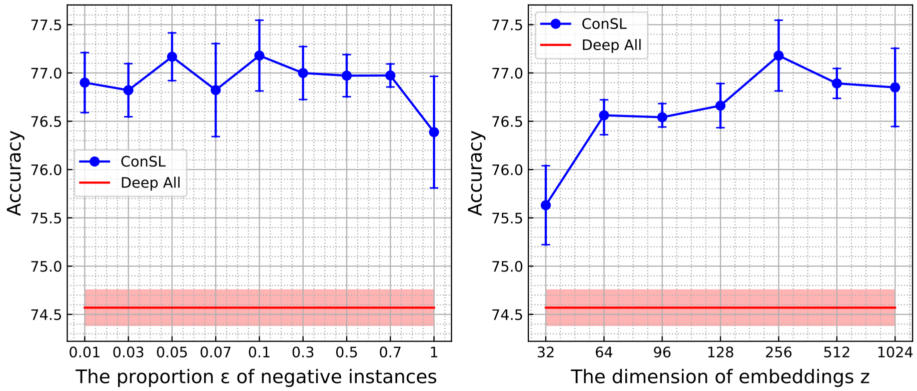

Fig. 4: The mean and standard deviation results of JCS and Deep All with regard to various negative instance size and feature embedding dimension. For details, see Fig. 3

A Hyper-parameter tuning

From Fig. 3, we find out that the trade-off parameter α is important and after it reaches 0.2, our method JCS is always better than Deep All, reach the peak at $\alpha = 0.5$. As for β, the accuracy is more stable and stays between 76 and 77.5, which means that the parameter is easy to be chosen.

B Varying the proportion of negatives and the embedding dimension

We investigate the sensitivity of the size of negative instances. Usually, the previous methods [29, 39] take the number of negative instances as hyper-parameter, so they have to adjust it for every dataset. To improve the generalization and application convenience of our model, we use the ϵ to control negative sample size, which indicates

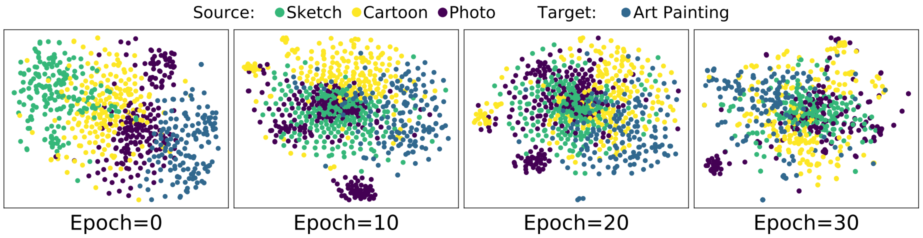

Fig. 5: Visualization of generalized features. Dots in Different color represent features of different domains. Source indicates the source datasets used in training while target dataset is only used for testing.

that the selected negatives for each instance occupy ϵ proportion of the total different-labeled instances in train set. As presented in Fig. 4 we can see that our method outperforms Deep All throughout varying the proportion of negative instances. And as the proportion moderately increases, the corresponding accuracy reaches the best at ϵ =0.1. It shows that more negatives do not always bring the accuracy promotion. When taken all qualified instances as negatives (ϵ =1), the model performance is even worst and most unstable.

Since the contrastive learning plays a significant role in our method, we then explore the dimension number of embedding used for contrast. In Fig. 4, we observe only an overall variation of 1.7 while it is still higher than Deep All. The accuracy almost maintains at the same level excluding 32 dimension and obtains a bit higher accuracy when the dimension is set to 256, implying that our model is not sensitive to embedding dimension. So it can be considered to project the feature to lower-dimension space in demand to train on larger dataset for alleviating the time and computation spaces.

C Visualization

We visualize the distributions of the features extracted by the domain-invariant feature extractor and present the results in Fig. 5, using t-SNE [4] to project them to two-dimension space. Before training when epoch=0, the features from different domains have distinctly different distributions. As the training continues, their distributions become out of order and the distribution boundary fades away as dots in different colors start to coincide. In particular, the blue dots represent the target domain which does not participate in training, gradually integrating into the group of other-color dots representing the source domains along with the training. It proves that the features from source domains cover the space spanned by any kind of target.