Improved Knowledge Transfer for Semi-supervised Domain Adaptation via Trico Training Strategy
(*Supplementary Material*)

Ba Hung Ngo, Yeon Jeong Chae, Jung Eun Kwon, Jae Hyeon Park and Sung In Cho*
Department of Multimedia Engineering, Dongguk University
{ngohung, toshinyg, kje_9912}@dgu.ac.kr {pjh0011, csi2267}@dongguk.edu

We provided the additional information and analyses, which are listed below:

- Appendix A: Notations.
- Appendixes B: We provided the ablation study in cases with and without data augmentations.
- Appendix C: We extracted the additional experimental results to show the outstanding performance on the target domain of the proposed method in various settings.
- Appendixes D: We reported the additional analyses to evaluate the effectiveness of the proposed method.
- Appendix E: We provided a short discussion to distinguish the concept among the proposed method TriCT and previous works that share several methodologies in common but have different roles in terms of three aspects: Co-training, Tri-network, and inter-and-intra-domain discrepancies.

A. Notation

This section provides the notations frequently used in this paper, as listed in Table 1.

B. Additional ablation study

We conducted additional experiments (reported in Table 2) to evaluate the effect of data augmentation. Without data augmentation, the average accuracy was dropped by 1.8%.

Two motivations for using data augmentations: addressing the problem of limited training data and enhancing the model's generalization capabilities while mitigating overfitting by introducing variations but remaining representative of the true distribution. Furthermore, we primarily employed data augmentations to promote consistency of the different models across various versions of the training data in our framework.

*Corresponding author.

Table 1: Notation used in the proposed method.

Notations	Descriptions
$\mathcal{D}_{\mathcal{S}}$	The set of source samples.
$x^i_{\mathcal{S}}$	The i-th labeled sample in the source domain.
$y^i_{\mathcal{S}}$	The ground-truth label of i-th labeled sample in the source domain.
$\mathcal{N}_{\mathcal{S}}$	The number of source samples.
$\mathcal{D}_{\mathcal{T}_l}$	The set of labeled target samples.
$x^i_{\mathcal{T}_l}$	The i-th labeled sample in the target domain.
$y^i_{\mathcal{T}_l}$	The ground-truth label of i-th labeled sample in the target domain.
$\mathcal{N}_{\mathcal{T}_l}$	The number of labeled target samples.
$\mathcal{D}_{\mathcal{T}_u}$	The set of unlabeled target samples.
$x^i_{\mathcal{T}_u}$	The i-th unlabeled sample in the target domain.
$\mathcal{N}_{\mathcal{T}_u}$	The number of unlabeled target samples.
$Aug_w(\cdot)$	The weak augmentation function.
$Aug_{str}(\cdot)$	The strong augmentation function.
$x^{i,w}_{\mathcal{S}}$	The weakly augmented version of $x^i_{\mathcal{S}}$.
$x^{i,str}_{\mathcal{S}}$	The strongly augmented version of $x^i_{\mathcal{S}}$.
$x^{i,w}_{\mathcal{T}_l}$	The weakly augmented version of $x^i_{\mathcal{T}_l}$.
$x^{i,str}_{\mathcal{T}_l}$	The strongly augmented version of $x^i_{\mathcal{T}_l}$.
$x^{i,w}_{\mathcal{T}_u}$	The weakly augmented version of $x^i_{\mathcal{T}_u}$.
$x^{i,str}_{\mathcal{T}_u}$	The strongly augmented version of $x^i_{\mathcal{T}_u}$.
E	The shared feature extractor.
F_{mlp}	The multilayer perceptron classifier.
G^{inter}_{gcn}	The inter-view GCN classifier.
G^{intra}_{gcn}	The intra-view GCN classifier.
$\mathcal{G}$	A graph.
$\mathcal{V}$	The set of nodes in a graph.
$\mathbf{v}_i$	A node $\mathbf{v}_i \in \mathcal{V}$.
$\mathcal{E}$	The set of edges in a graph.
$e_{i,j}$	An edge $e_{i,j} = (v_i, v_j) \in \mathcal{E}$ connecting nodes v_i and v_j.

Table 2: Accuracy (%) on *DomainNet* dataset in cases with and without data augmentations under 3-shot setting using ResNet-34.

Setting	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Mean
With Aug.	89.1	86.6	86.3	79.9	84.5	82.1	90.1	85.5
Without Aug.	87.4	86.1	84.9	75.8	83.0	78.8	89.8	83.7

Table 3: Accuracy (%) on *DomainNet* under 5-shot and 10-shot settings extracted by the ResNet-34 backbone network.

Method	R→C		R→P		P→C		C→S		S→P		R→S		P→R		Mean	
	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot
MME	75.5	77.1	70.4	71.9	74.0	76.3	65.0	67.0	68.2	69.7	65.5	67.8	79.9	81.2	71.2	73.0
APE	77.7	79.8	73.0	75.1	76.9	78.9	67.0	70.5	71.4	73.6	68.8	70.8	80.5	82.9	73.6	76.8
DECOTA	-	81.8	-	75.1	-	81.3	-	73.7	-	73.4	-	73.7	-	80.7	-	77.1
CLDA	80.3	81.2	76.0	77.7	77.8	80.3	71.6	74.1	74.5	77.1	72.9	74.1	84.0	85.1	76.7	78.5
CDAC	80.8	83.1	75.3	77.2	79.9	81.7	72.1	74.3	74.7	76.3	72.9	74.6	83.2	84.7	76.9	78.9
MVCL	81.1	83.0	78.2	79.2	81.7	82.7	74.7	76.0	77.2	78.1	74.6	75.9	86.3	87.0	79.1	80.3
TriCT	**90.2**	**91.4**	**87.4**	**88.7**	**87.2**	**90.6**	**80.4**	**82.0**	**86.1**	**87.0**	**84.7**	**86.6**	**91.9**	**92.6**	**86.8**	**88.4**

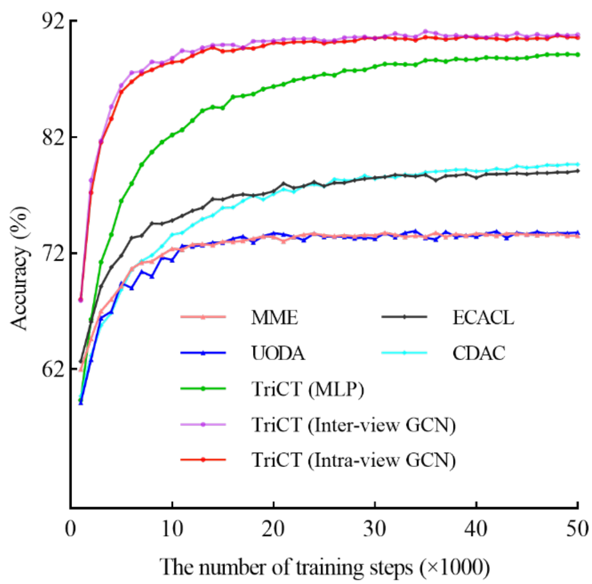

Figure 1: Convergence analysis of the domain adaptation task $R \rightarrow C$ on *DomainNet* using ResNet-34 under the 3-shot setting. We compared the convergence of the proposed method and the prior SSDA methods.

C. Additional experiments

We reported additional comparison results on 5-shot and 10-shot labeled target samples per class of the *DomainNet* dataset using ResNet-34 as the backbone network. The classification performance of various state-of-the-art SSDA methods, including MME [8], APE [3], DECOTA [11], CLDA [9], CDAC [4], MVCL [6], and the proposed method TriCT were listed in Table 3. As shown in this table, the classification performance on the target domain of the proposed method achieved the highest in all domain adaptation scenarios. The average classification accuracy of TriCT improved by 7.7% and 8.1% compared to the second-best method MVCL under 5-shot and 10-shot settings, respectively.

D. Additional analysis

D.1 Convergence analysis

We analyzed the convergence to evaluate the effectiveness of the proposed method compared to the state-of-the-art (SOTA) methods, consisting of MME [8], UODA [7], ECACL [5], and CDAC [4], as illustrated in Figure 1. This figure showed the classification accuracy of the different methods, including MME, UODA, ECACL, CDAC, and the proposed method TriCT for the domain adaptation task R→C (*Real* to *Clipart*) on *DomainNet* using ResNet-34 under the 3-shot setting. It was apparent that both inter-view GCN and intra-view GCN classifiers reached the highest classification accuracy at early steps after 20,000 training steps, and they kept a stable convergence along with the training iterations. This was because these GCN classifiers could collect the neighbor information for generalization on the target data. Similarly, MME and UODA quickly converged after 20,000 training steps. In contrast, the classification accuracy of ECACL and CDAC tended to increase after 50,000 training steps. However, it still maintained a big gap compared to the classification accuracy of our MLP classifier.

D.2 Visualization analysis

We selected the representations of 10 classes in the source and target domains of the domain adaptation task R→C on *DomainNet* using ResNet-34 under the 3-shot setting for visualization in Figure 2. The first row included the results of the target representation results extracted from the different domain adaptation methods such as MME [8], ECACL [5], CDAC [4], and TriCT. The second row consisted of the domain alignment results, which were used to evaluate the domain adaptation ability of these methods. As shown in Figures 2a, 2b, 2c, and 2d, the target representations extracted by TriCT were more discriminative compared to the other SOTA methods, which indicated that the proposed method effectively alleviates the intra-domain discrepancy. Moreover, as shown in Figures 2e, 2f, 2g, and 2h, the source and target representations extracted by TriCT showed well-align results compared to MME, UODA, ECACL, and CDAC, revealing the effectiveness of TriCT for mitigating the inter-domain discrepancy.

E. Discussion

Co-training. DECOTA [11] and MVCL [6] break SSDA into two subtasks; they then use two different models to handle the two different tasks. Finally, these models com-

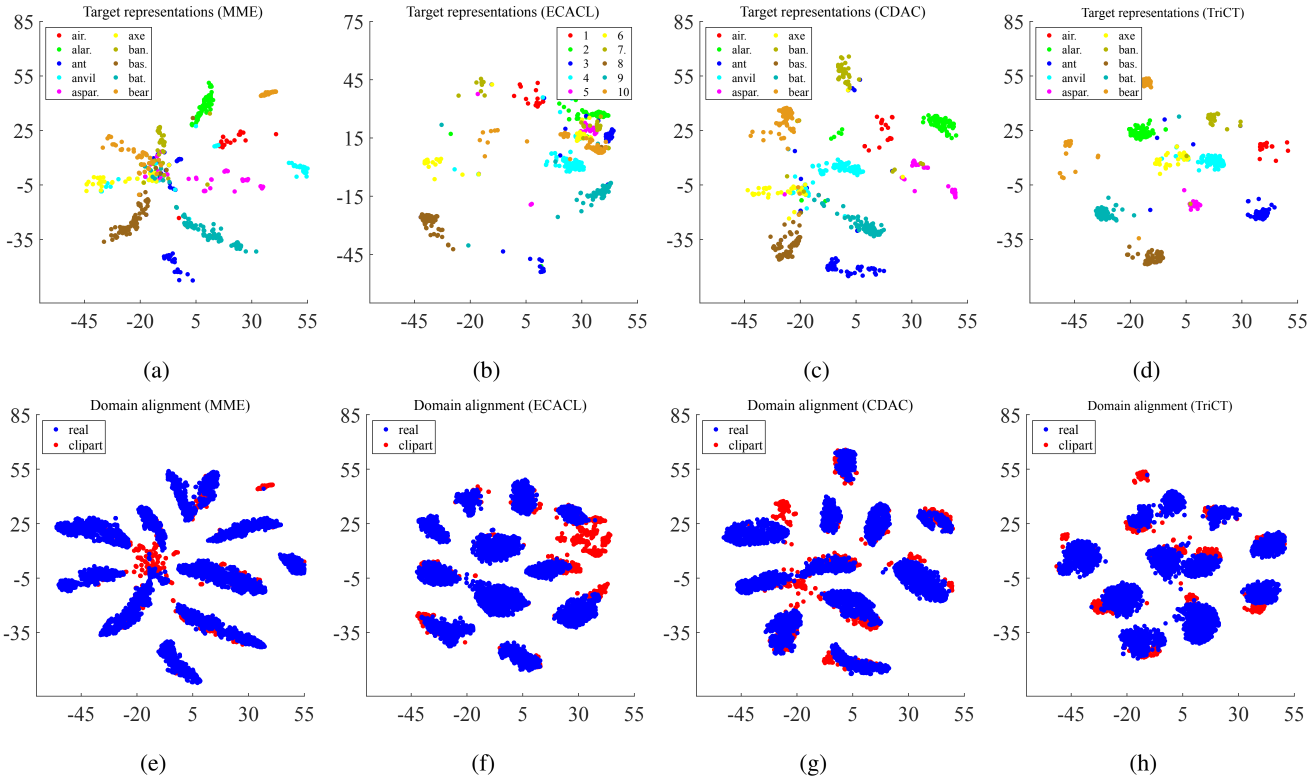

Figure 2: t-SNE [10] visualization of the extracted representations. We illustrate the representations of 10 classes in the domain adaptation task $R \rightarrow C$ on *DomainNet* using ResNet-34 under the 3-shot setting. (a), (b), (c), and (d) show the target presentations extracted by the various domain adaptation methods. (e), (f), (g), and (h) display the domain alignment results of the different domain adaptation methods. 'air.', 'alar.', 'aspar.', 'ban.', 'bas.' and 'bat.' are abbreviations of 'aircraft_carrier', 'alarm_clock', 'asparagus', 'banana', 'basket', and 'bathtub', respectively.

plemented their knowledge of each other by exchanging their generated pseudo labels via a co-training strategy. However, TriCT uses three different classifiers to establish three different co-training strategies that could effectively exploit the complementarity among these models. Figure 3 illustrates the different training strategies between DeCoTa and TriCT methods.

Tri-network. Multi-Head Co-Training [2], Tri-net [1], and TriCT have similar network architecture, including a shared feature extractor and three classifier headers. Nevertheless, these methods contain quite different training algorithms. For example, in Multi-Head Co-training, they fixedly select two classifiers as teachers to teach the third classifier working as a student model. Specifically, they minimize the consistency loss between pseudo labels generated by teacher classifiers on a weak augmentation image and predictions of the student classifier on a strong augmentation image with the same samples. In contrast, in Tri-net, a classifier can simultaneously be a teacher and a student. When it works as the teacher model, it is randomly paired with another classifier to train the remaining classifier by their generated pseudo labels. On the contrary, when it works as the student model, it learns extracted knowledge

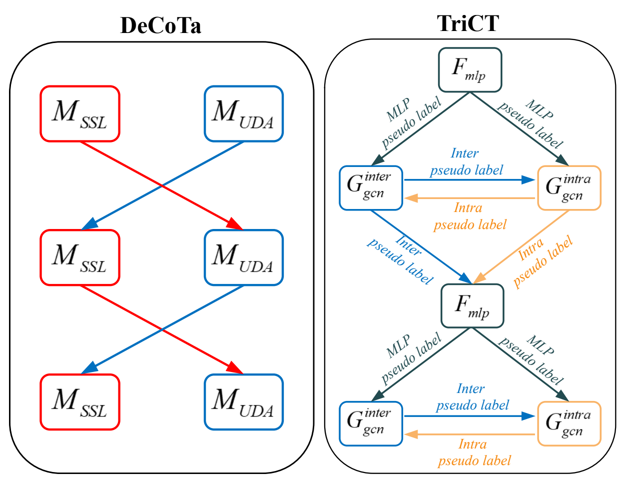

Figure 3: Overall training strategy of DeCoTa and TriCT. (a) DeCoTa optimizes the training model using co-training. (b) TriCT uses three co-training strategies to learn optimal network parameters.

information from two other models. This algorithm is conducted for all classifiers. Unlike Multi-Head Co-training and Tri-Net, TriCT introduces a novel training algorithm consisting of three co-training strategies. Each co-training scheme is proposed for a different role that exploits the correlation between two classifier models.

Inter-domain and intra-domain discrepancies. CLDA [9], and CDAC [4] also attempt to reduce inter-domain and intra-domain discrepancies simultaneously by exploiting information from unlabeled target data as much as possible. They share the same overall model architecture, including a feature extractor and a single classifier but have different training procedures. CLDA utilizes contrastive learning to achieve inter-domain and intra-domain alignment, while CDAC proposes adaptive clustering to reduce both inter-domain and intra-domain gaps. However, TriCT introduces the Trico-training algorithm with three co-training strategies to effectively handle inter-domain and intra-domain discrepancies.

Importantly, all the approaches mentioned above use the MLP classifier to design their network architectures. However, the MLP classifier can be failed to explore the data training structure because it only forces on each individual image without considering the semantic information of its neighbors; therefore, the classification performance still has room for improvement. To solve this problem, we use the GCN classifier model to aggregate extracted features; thus, the classification accuracy on the testing set is improved because the trained model can capture the training data structure. To the best of our knowledge, TriCT is the first SSDA method that achieves state-of-the-art performance with the graph-based algorithm.

References

[1] Dong-Dong Chen, Wei Wang, Wei Gao, and Zhi-Hua Zhou. Tri-net for semi-supervised deep learning. In *IJCAI*, 2018. 3

[2] Mingcai Chen, Yuntao Du, Shuwei Qian Yi Zhang, and Chongjun Wang. Semi-supervised learning with multi-head co-training. In *AAAI*, pages 6278–6286, 2022. 3

[3] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV*, pages 591–607, 2020. 2

[4] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *CVPR*, pages 2505–2514, 2021. 2, 4

[5] Kai Li, Chang Liu, Handong Zhao, Yulun Zhang, and Yun Fu. Ecacl: A holistic framework for semi-supervised domain adaptation. In *ICCV*, pages 8578–8587, 2021. 2

[6] Ba Hung Ngo, Ju Hyun Kim, Yeon Jeong Chae, and Sung In Cho. Multi-view collaborative learning for semi-supervised domain adaptation. *IEEE Access*, volume 9:166488–166501, 2021. 2

[7] Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu. Contradictory structure learning for semi-supervised domain adaptation. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 576–584, 2021. 2

[8] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, pages 8050–8058, 2019. 2

[9] Ankit Singh. Clda: Contrastive learning for semi-supervised domain adaptation. *NeurIPS*, 34:5089–5101, 2021. 2, 4

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, volume 9:770–778, 2008. 3

[11] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q. Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. In *ICCV*, pages 8906–8916, 2021. 2