南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Dual Consolidation for Pre-Trained Model-Based Domain-Incremental Learning

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye(✉), Lijun Zhang, De-Chuan Zhan
School of Artificial Intelligence, Nanjing University
National Key Laboratory for Novel Software Technology, Nanjing University
{zhoudw,caizw,yehj,zhanglj,zhandc}@lamda.nju.edu.cn

CVPR 2025

Background

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

This paper addresses the domain-incremental learning (DIL) challenge using a pre-trained Vision Transformer (ViT). To combat catastrophic forgetting when data distributions shift over time, it introduces the DUCT framework with a dual consolidation mechanism. First, historical knowledge is preserved by fusing previous model representations into the current embedding space. Second, semantic information about each class is leveraged to estimate and adjust the weights of the old classifiers so they remain effective in the new representation space. As a result, the model can learn from newly arriving domain data while retaining its ability to discriminate samples from earlier domains.

Introduction

In this paper, the task data inputs of the model are denoted as $\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^B\}$, where a single task is composed of $\mathcal{D}^b = \{\mathcal{X}_b, \mathcal{Y}_b\}$, and $\mathcal{Y}_b = \{y_i\}_{i=1}^{n_b}$ satisfies $y_i \in \mathbf{Y}$. Here, $\mathbf{Y}$ remains unchanged throughout the learning process of the model, while the data distribution varies with tasks, that is, $p(\mathcal{X}_b) \neq p(\mathcal{X}_{b'})$ for $b \neq b'$.

In addition, this paper is based on the exemplar - free setting, that is, when learning new tasks, data samples or features of historical tasks are not saved. Thus, the goal of domain incremental learning in this paper can be summarized as the following formula (where $\mathcal{H}$ is the hypothesis space and $\mathcal{I}$ is the indicator function):

$$f^* = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \mathbb{E}_{(\mathbf{x},y)\sim \mathcal{D}_t^1 \cup \cdots \cup \mathcal{D}_t^b} \mathcal{I}(y \neq f(\mathbf{x}))$$

Introduction

Considering the pre - trained model ViT selected in this paper, its structure can be decomposed into two parts: the embedding function $\phi(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^d$ and the linear classification head $W \in \mathbb{R}^{d \times \mathbf{Y}}$. The output of the model can be expressed as: $f(\mathbf{x}) = W^T \phi(\mathbf{x})$.

This paper adopts the Cosine$^{\alpha}$ classification head, and the classifier does not contain a bias term, $W = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{|\mathbf{Y}|}]$. When learning each new task, following previous work, this paper still expands a new classification head for the new task, and splices it with the old classification heads for prediction in the testing phase. The result of $(\text{argmax}_i \, \mathbf{w}_i^T \phi(\mathbf{x})) \bmod |\mathbf{Y}|$ is taken as the output of the model.

Method

two complementary consolidation stages in DUCT

1. Representation Consolidation (to resist feature-level forgetting);
2. Classifier Consolidation (to resist classifier-level forgetting).

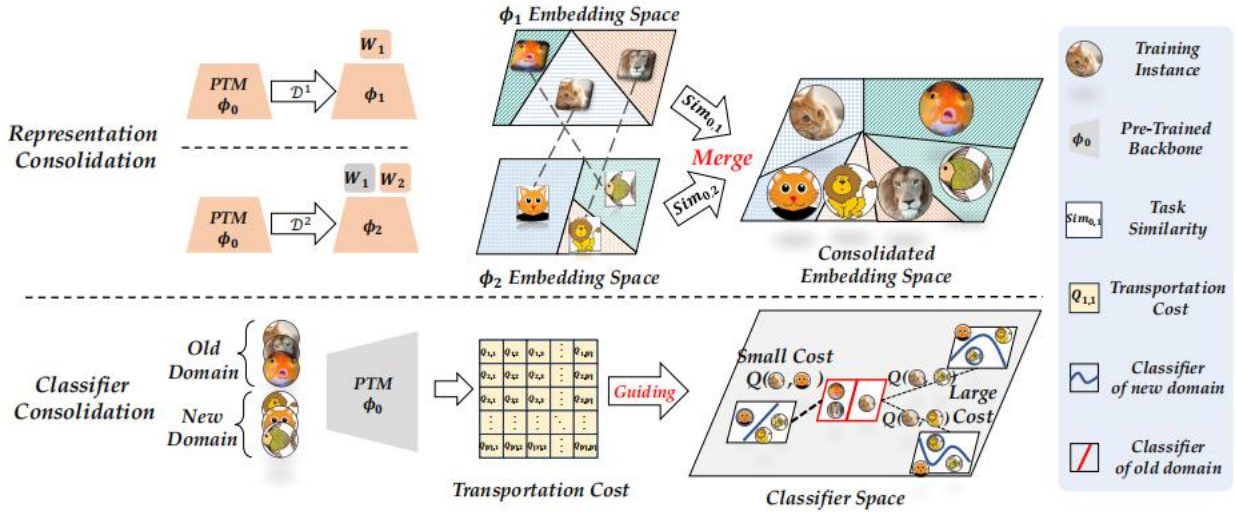


Figure 1. Illustration of DUCT. Top: Representation consolidation. We utilize the pre-trained model as initialization and optimize it for each domain, obtaining the task vectors. Afterward, we combine the pre-trained model and all seen task vectors to build the unified embedding space. Bottom: Classifier consolidation. To align the classifiers with consolidated features, we design the new classifier retraining and old classifier transport to consolidate classifiers. Class-wise semantic information is utilized in classifier transport.

Method

1、Consolidation of the Feature Space

In the context of Domain Incremental Learning (DIL), models need to balance knowledge learned from multiple domains while avoiding feature overriding and forgetting due to constant updates. To address this, a new feature update strategy is proposed to build a universal embedding space that can adapt well to data distributions from all domains. Under ideal conditions, we can train a separate model for each new domain individually, resulting in a set of domain experts:

$$\{\phi_1(\cdot), W_1\}, \ldots, \{\phi_B(\cdot), W_B\}$$

These models are optimized for domain-specific discrimination using a shared prediction model structure.

If the domain of each input sample is known, we can directly invoke the corresponding expert for inference. However, in DIL tasks, domain labels are often unavailable, making direct application of domain experts impractical.

Method

Inspired by model fusion techniques, this paper proposes a method that constructs a theoretical model for multiple domains using task vectors. This allows the learning of a powerful, general-purpose feature representation model capable of handling classification across multiple domains.

Let the weighted feature space for domain i be denoted as ϕ_i^m, where the embedding function ϕ is adjusted by a linear combination of task specific deltas:

$$\delta_{\phi_k} = \phi_i - \phi_0$$

Thus, the generalized feature space is defined as:

$$\phi_i^m = \phi_0 + \alpha_\phi \sum_{k=1}^{i} \delta_{\phi_k}$$

While this approach is intuitive and simple, it does not consider task relationships. For example, when dealing with similar tasks, stronger fusion weights should emphasize overlapping domain knowledge.

Method

To address this, a task similarity variable $\mathrm{Sim}_{0,i}$ is introduced.

One way to measure task similarity is to compute the similarity between task direction vectors. However, due to the high-dimensional nature of the weight space, direct similarity computations may lose meaning.

To avoid this, the paper measures similarity via task-specific feature distances using class centers:

$$c_p^i = \sum_{j=1}^{|\mathcal{D}^b|} \mathbb{I}(y_j = p)\phi_i(x_j) / \sum_{j=1}^{|\mathcal{D}^b|} \mathbb{I}(y_j = p)$$

We then construct class centers for both embedding spaces:

$$C^0 = [c_1^0, \ldots, c_{|Y|}^0], \quad C^i = [c_1^i, \ldots, c_{|Y|}^i]$$

Method

Pairwise similarities between these centers are averaged to get task similarity:

$$\mathbf{Sim}_{0,i} = \frac{1}{|Y|} \sum_{j=1}^{|Y|} \mathrm{sim}(\mathbf{c}_j^0, \mathbf{c}_j^i)$$

Finally, using cosine similarity to compute distances, the new generalizable embedding function becomes:

$$\phi_i^m = \phi_0 + \alpha_\phi \sum_{k=1}^{i} \mathbf{Sim}_{0,k} \delta\phi_k$$

Method

2、Consolidation of the Classification Space

Due to the changes in embedding space from fusion, the classifier may no longer align with the updated features. Thus, the classifier must also be adapted accordingly.Let W_0 be the old classifier and W_n the new classifier to be retrained. The paper proposes a two-step process for aligning classifiers.

Retraining of the New Classification Head

To adapt to the new embedding space, we retrain the new classifier W_n while freezing the embedding network ϕ_i^m:

$$\min_{W_n} \sum_{(x,y)\in\mathcal{D}^b} \ell(W_n^T \phi_i^m(x), y)$$

Method

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Migration of the Old Classification Head

Even after retraining the new classifier, the old classifier might still misalign with the new embedding space, leading to performance degradation on old tasks, i.e., catastrophic forgetting.
To solve this under the exemplar-free setting, the paper proposes an Optimal Transport (OT) based solution to adjust the old classifier to the new feature space.

Let:
μ_1: class distribution of the new classifier,
μ_2: class distribution of the old classifier,
T: transport matrix (matching distribution),
Q: cost matrix for transferring from class j to class i.

The OT objective is: $\min_T \langle T, Q \rangle \quad \text{s.t.} \quad T1 = \mu_1,\ T^T 1 = \mu_2,\ T \geq 0$

The cost matrix $Q_{i,j}$is defined by Euclidean distances between class centers in the embedding space:

$$Q_{i,j} = \|c_i^0 - c_j^i\|_2^2$$

This allows the old classifier to be adjusted to match the new embedding space and retain performance on previous tasks.
The final expression for the calibrated and fused classifier heads is given as follows:"

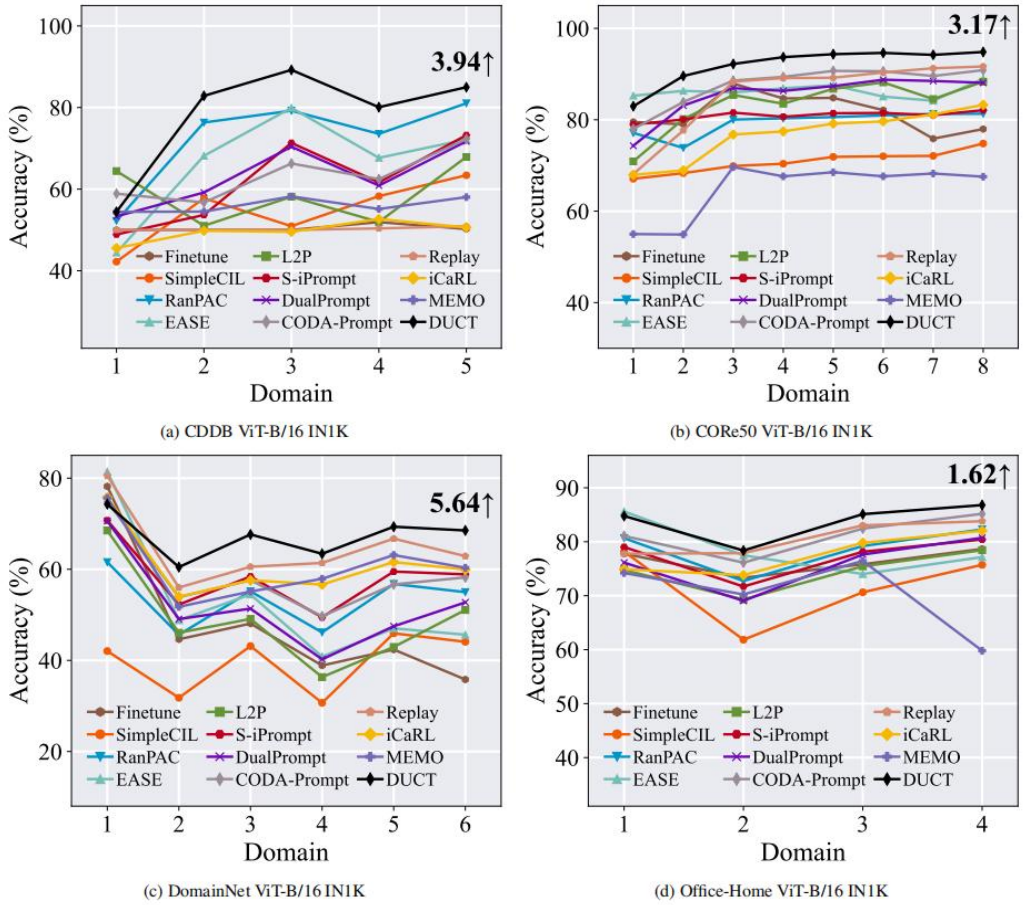
$$W_o^m = (1 - \alpha_W) W_o + \alpha_W \hat{W}_o = (1 - \alpha_W) W_o + \alpha_W W_n T$$

Experiment

Table 1. Average and last performance of different methods among five task orders. The best performance is shown in bold. All methods are implemented with ViT-B/16 IN1K. Methods with † indicate implemented with exemplars (10 per class).

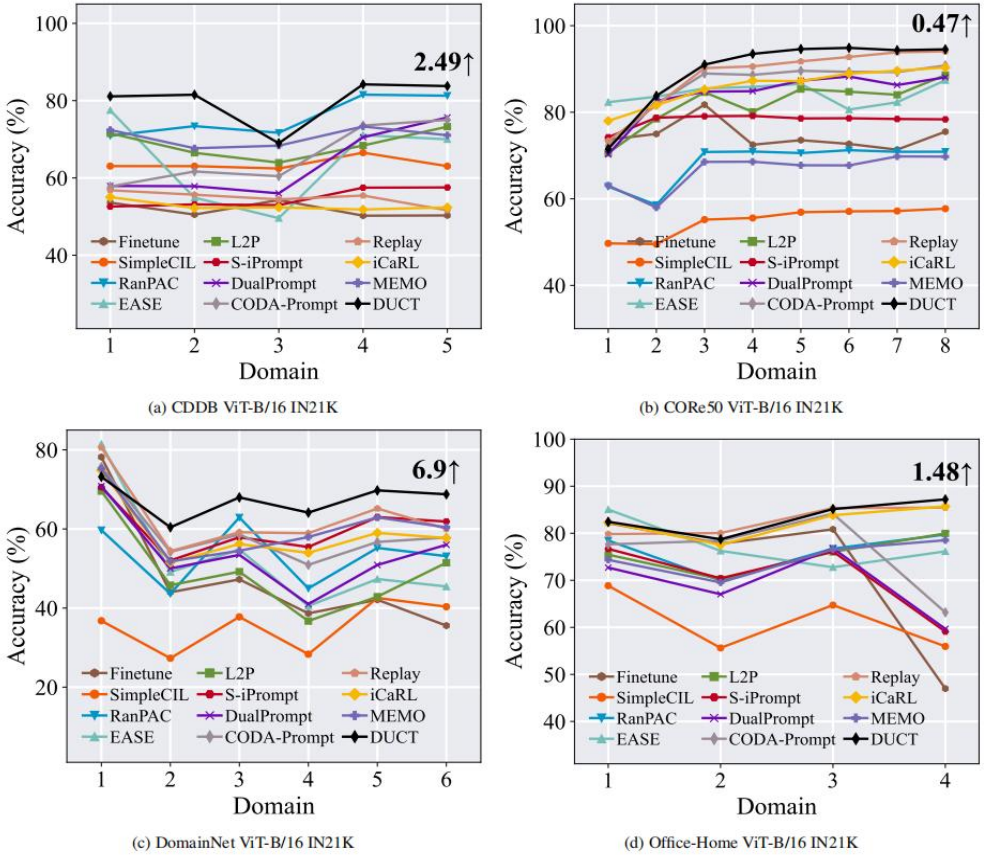
Method	Office-Home		DomainNet		CORe50		CDDB	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
Finetune	78.32$_{\pm 3.28}$	76.16$_{\pm 1.39}$	28.17$_{\pm 6.47}$	38.82$_{\pm 7.65}$	75.44$_{\pm 1.68}$	76.19$_{\pm 2.36}$	52.08$_{\pm 1.35}$	50.11$_{\pm 1.62}$
Replay† [51]	84.23$_{\pm 2.31}$	83.75$_{\pm 0.68}$	64.78$_{\pm 2.98}$	61.16$_{\pm 1.19}$	85.56$_{\pm 0.38}$	92.21$_{\pm 0.63}$	66.91$_{\pm 18.0}$	63.21$_{\pm 11.6}$
iCaRL† [52]	81.66$_{\pm 2.43}$	81.11$_{\pm 1.23}$	59.89$_{\pm 2.86}$	57.46$_{\pm 2.31}$	74.43$_{\pm 3.18}$	79.86$_{\pm 2.96}$	68.43$_{\pm 18.7}$	70.50$_{\pm 16.5}$
MEMO† [85]	71.18$_{\pm 2.76}$	63.09$_{\pm 1.80}$	61.92$_{\pm 5.39}$	58.41$_{\pm 3.20}$	64.80$_{\pm 3.16}$	68.24$_{\pm 2.41}$	60.87$_{\pm 13.4}$	58.09$_{\pm 11.7}$
SimpleCIL [88]	75.69$_{\pm 5.03}$	75.72$_{\pm 0.00}$	42.95$_{\pm 4.84}$	44.08$_{\pm 0.00}$	70.92$_{\pm 0.74}$	74.80$_{\pm 0.00}$	60.80$_{\pm 4.49}$	63.40$_{\pm 0.00}$
L2P [70]	79.72$_{\pm 4.19}$	80.03$_{\pm 1.29}$	50.45$_{\pm 4.10}$	48.72$_{\pm 2.83}$	83.57$_{\pm 0.35}$	87.87$_{\pm 0.51}$	67.33$_{\pm 5.98}$	64.45$_{\pm 5.83}$
DualPrompt [69]	80.20$_{\pm 3.81}$	80.85$_{\pm 0.14}$	52.28$_{\pm 3.35}$	50.46$_{\pm 3.17}$	84.53$_{\pm 0.89}$	87.27$_{\pm 1.06}$	68.33$_{\pm 7.52}$	71.41$_{\pm 1.24}$
CODA-Prompt [58]	84.70$_{\pm 2.94}$	85.07$_{\pm 0.34}$	59.85$_{\pm 4.49}$	59.99$_{\pm 0.88}$	87.92$_{\pm 0.41}$	91.57$_{\pm 0.69}$	69.19$_{\pm 6.11}$	74.18$_{\pm 1.33}$
EASE [86]	81.16$_{\pm 3.52}$	76.33$_{\pm 2.16}$	50.50$_{\pm 2.27}$	43.72$_{\pm 1.70}$	86.30$_{\pm 0.04}$	87.02$_{\pm 1.21}$	67.78$_{\pm 2.44}$	64.96$_{\pm 8.36}$
RanPAC [40]	82.30$_{\pm 3.34}$	82.28$_{\pm 0.00}$	55.20$_{\pm 3.93}$	54.80$_{\pm 0.36}$	79.16$_{\pm 0.55}$	81.38$_{\pm 0.12}$	78.92$_{\pm 4.85}$	80.48$_{\pm 0.68}$
S-iPrompt [68]	81.50$_{\pm 3.17}$	80.51$_{\pm 0.21}$	61.16$_{\pm 4.57}$	60.46$_{\pm 0.90}$	81.95$_{\pm 0.57}$	83.38$_{\pm 0.70}$	68.51$_{\pm 7.20}$	72.76$_{\pm 0.43}$
DUCT	**86.27**$_{\pm 2.95}$	**86.91**$_{\pm 0.06}$	**67.16**$_{\pm 3.75}$	**67.01**$_{\pm 1.35}$	**91.95**$_{\pm 0.15}$	**94.47**$_{\pm 0.33}$	**84.14**$_{\pm 4.37}$	**85.10**$_{\pm 0.52}$

Experiment



(a) CDDB ViT-B/16 IN1K (b) CORe50 ViT-B/16 IN1K (c) DomainNet ViT-B/16 IN1K (d) Office-Home ViT-B/16 IN1K

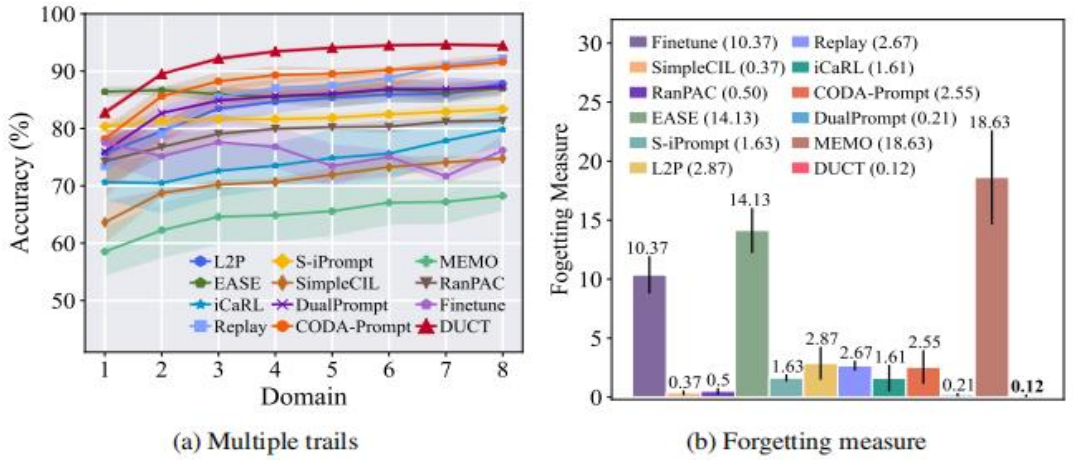
Figure 7. Incremental performance of different methods with ViT-B/16 IN1K. We report the performance gap after the last incremental stage between DUCT and the runner-up method at the end of the line.



(a) CDDB ViT-B/16 IN21K (b) CORe50 ViT-B/16 IN21K (c) DomainNet ViT-B/16 IN21K (d) Office-Home ViT-B/16 IN21K

Figure 8. Incremental performance of different methods with ViT-B/16 IN21K. We report the performance gap after the last incremental stage between DUCT and the runner-up method at the end of the line.

Experiment



(a) Multiple trails (b) Forgetting measure

Figure 3. Further analysis on multiple task orders, forgetting measure, and parameter robustness. (a): Incremental performance of different methods on CORe50 with five task orders. The shadow indicates standard deviation. (b): Forgetting measure (lower is better) of different methods on CDDB dataset among five task orders. DUCT shows the least forgetting among all methods.

Key Conclusions:

1. DUCT outperforms all baselines in final accuracy and forgetting resistance, showing strong stability.
2. It generalizes better than other prompt-based and replay-based methods across diverse DIL tasks.
3. Robust across different pretrained backbones (IN1K & IN21K), demonstrating good compatibility.

Experiment

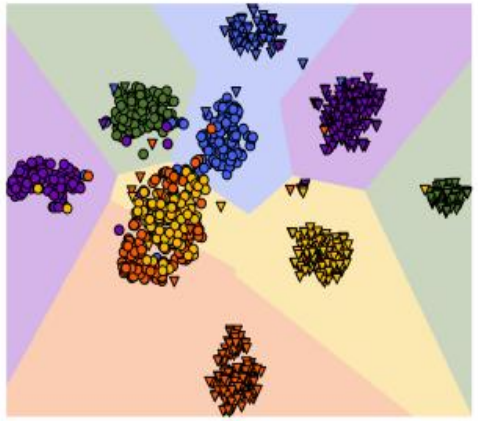


Figure 4. Before DUCT.

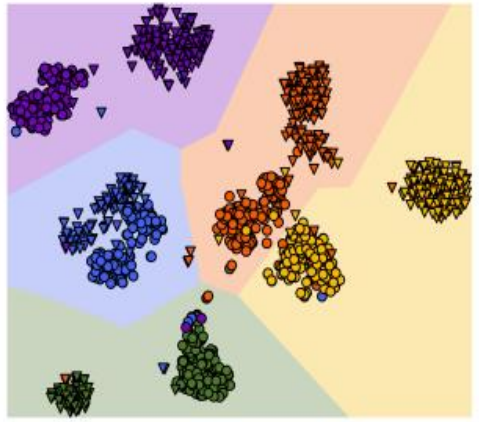


Figure 5. After DUCT.

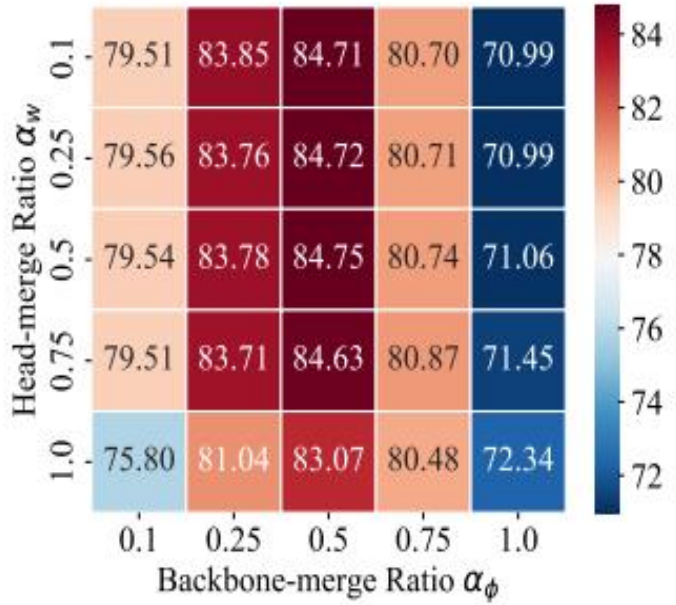
Use circles to represent samples from the first domain, and triangles for samples from the second domain. In the figure below, the left subfigure shows the class-wise distribution of samples before feature fusion, while the right subfigure illustrates the distribution after fusion.

Experiment

More detailed ablation experiments were conducted on the CDDB dataset to explore the importance of each module of DUCT. As shown in the following table, the performances of various method variants were compared:

Variations	$\bar{A}$	A_B
Baseline	66.56	63.40
Variation 1	80.36	74.17
Variation 2	81.41	76.82
Variation 3	85.42	80.31
DUCT	**87.74**	**82.35**

Table 2. Ablation study on differnt modules in DUCT.



The experiment selects from {0.1, 0.25, 0.5, 0.75, 1.0}, forming a total of 25 parameter combinations. The average performance of these parameter combinations is calculated on the Office-Home dataset. As can be seen from the figure, DUCT generally has strong robustness to parameter changes.

Thanks