

Supplementary Materials: Importance-aware Shared Parameter Subspace Learning for Domain Incremental Learning

Anonymous Authors

A ADDITIONAL DETAILS

A.1 Algorithm Pseudo-code

The training procedure for our method is outlined in Algorithm 1. We introduce the decomposed parameter matrices ΔW_i into the fixed pretrained vision transformer W_0 for each domain i . The decomposed parameter matrices consist of the subspaces A^c shared by all M domains, the subspaces A_i^s specific to each domain i , and their corresponding coefficient matrix B_i . At each training step, we derive the importance score ρ_i for these subspaces based on their gradients and utilize the derived importance score ρ_i to adaptively weight the domain-shared parameter subspaces and domain-specific parameter subspaces. Then, we obtain the feature for each sample by utilizing the feature extractor constructed by combining the pretrained model W_0 with the importance-aware parameter matrices ΔW_i . Subsequently, we leverage these features to calculate the per-sample cross-entropy loss. Next, we calculate the cross-domain contrastive loss on current and historical domain-specific parameter subspaces, as to amplify the distinctions between them. Moreover, we calculate the orthogonality penalty loss between the domain-shared subspaces A^c and domain-specific subspaces A_i^s , as to minimize the interference between them. The aforementioned three losses are aggregated and utilized as the overall loss function for optimizing the domain-specific parameter subspaces A_i^s and the coefficient matrix B_i . As for the domain-shared subspaces, we conduct momentum update on them to smoothly and incrementally capture shared information across different domains while mitigating forgetting.

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 Ablation Study

To further validate the effectiveness of the momentum update strategy on domain-shared subspaces, we design several variants and conduct experiments on three datasets. The results are reported in Table IV. "Ours-w.o.- A_i^s " denotes our method without exploiting the domain-specific subspaces A_i^s . "Ours-w.o.-mom-w.o.- A_i^s " refers to our method that replaces the momentum update strategy with gradient updates on the domain-shared subspaces and does not utilize the domain-specific subspaces A_i^s . "Ours-w.o.- A_i^s " outperforms "Ours-w.o.-mom-w.o.- A_i^s ", demonstrating that the application of the momentum update strategy on domain-shared subspaces effectively mitigates forgetting and contributes to an improvement in performance. "Ours-w.o.-mom" indicates that our method utilizes gradient updates on both the domain-specific and domain-shared subspaces without employing the momentum update strategy. As presented in Table IV, our method performs better than the three aforementioned variants, illustrating the benefits of leveraging momentum update on domain-shared subspaces and effectively combining them with domain-specific subspaces.

Table IV: Ablation study of momentum update on domain-shared subspaces across three datasets.

Method	Cddb-Hard	DomainNet	CORE50
Ours-w.o.- A_i^s	66.95±0.32	41.02±0.09	84.53±0.49
Ours-w.o.-mom-w.o.- A_i^s	62.92±0.40	38.45±0.31	82.61±0.47
Ours-w.o.-mom	88.71±0.51	66.56±0.39	89.76±0.56
Ours	90.10±0.38	67.80±0.11	91.07±0.52

Table V: Impact of different momentum coefficients (η) in the momentum update strategy across three datasets.

Dataset	η	0.999	0.9999	0.99999
Cddb-Hard	3domains	90.19±0.33	91.03±0.35	89.25±0.28
	5domains	89.35±0.33	90.10±0.38	88.74±0.29
DomainNet	3domains	62.41±0.20	63.14±0.07	62.96±0.10
	6domains	62.97±0.21	67.80±0.11	65.37±0.19
Core50	4domains	89.38±0.40	89.79±0.43	89.22±0.39
	8domains	89.76±0.43	91.07±0.52	90.52±0.44

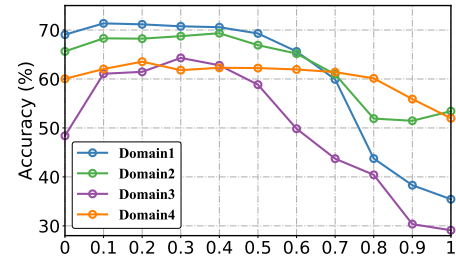


Figure VI: Empirical study on the importance of domain-shared information for different domains. Different importance proportions of the domain-shared subspaces are manually applied to four domain data from the DomainNet dataset.

Momentum coefficient η of the momentum update strategy plays an important role in smoothly accumulation of knowledge in the current domain while mitigating the risk of forgetting the knowledge acquired from previous domains. We conduct experiments with different momentum coefficients (η) on three datasets. We test two kinds of different numbers of domains for each dataset: the first three domains and all five domains for Cddb-Hard, the first three domains and all six domains for DomainNet, and the first four domains and all eight domains for CORE50. The average results are reported in Table V. We can observe that using a relatively small

Algorithm 1: The training procedure of our network

Input: M training domains $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_M\}$, pretrained vision transformer backbone W_0 , epochs E , hyperparameter α and β ;

```

1 for each domain  $i$  from 1 to  $M$  do
2   Attach the decomposed parameter matrices  $\Delta W_i$ , consisting of the domain-shared parameter subspaces  $A^c$ , domain-specific
   parameter subspaces  $A_i^s$  and their corresponding coefficient matrix  $B_i$ , to the pretrained transformer  $W_0$  as the feature extractor;
3   Initialize the dynamic importance weight  $\rho_i(1)$  to 0.5;
4   for each epoch  $e$  from 1 to  $E$  do
5     for each step  $t$  in epoch  $e$  do
6       Conduct importance-conditioned subspace enhancement with the derived importance weight  $\rho_i$  as
        $\Delta W_i(t) = B_i(t)(\rho_i(t)A_i^s(t) + (1 - \rho_i(t))A^c(t))$ ;
7       Obtain the sample feature with the feature extractor  $W_0 + \Delta W_i(t)$  and calculate the per sample cross entropy loss  $\mathcal{L}$ ;
8       Calculate the cross-domain contrastive loss  $\mathcal{L}_c$  with current and historical domain-specific parameter subspaces;
9       Calculate the orthogonality penalty loss  $\mathcal{L}_o$  between the domain-shared subspaces  $A^c$  and domain-specific subspaces  $A_i^s$ ;
10      Calculate the final objective function  $\mathcal{L}_{final} = \mathcal{L} + \alpha\mathcal{L}_c + \beta\mathcal{L}_o$ ;
11      Update the domain-specific parameter subspaces  $A_i^s$  and coefficients matrix  $B_i^s$  with the final objective  $\mathcal{L}_{final}$ ;
12      Conduct momentum update on the shared subspaces  $A^c$  to smoothly accumulate knowledge of the current domain  $i$ ;
13      Dynamically derive the importance weight  $\rho_i(t+1)$  for these parameter subspaces based on the gradients of these
       subspaces at current step  $t$ ;
14    end
15  end
16 end

```

Output: The learned network.

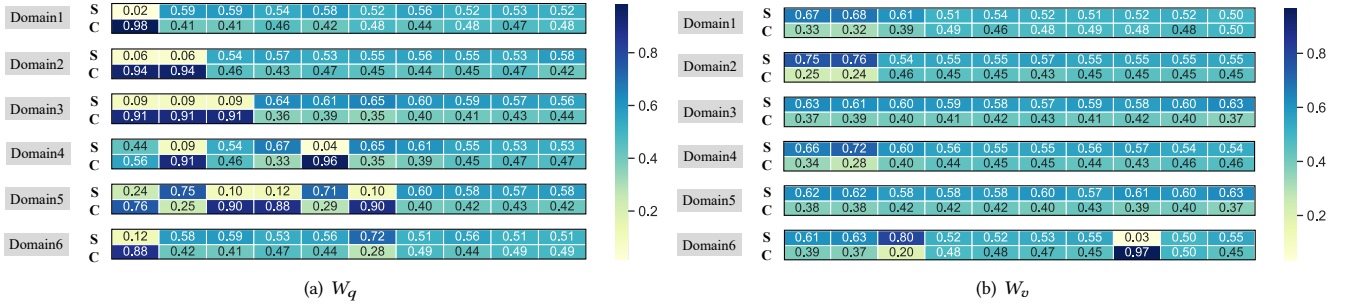


Figure VII: Heatmap of the importance of domain-specific subspaces (denoted as S) and the importance of domain-shared subspaces (denoted as C) of six domains from the DomainNet dataset.

value (e.g., 0.999) can lead to the integration of excessive unstable new knowledge when incrementally learning new domains at each step. On the other hand, employing a significantly larger value (e.g., 0.99999) can result in the shared subspaces being unable to effectively absorb new knowledge. In all of our experiments in the paper, we set the momentum coefficient η to 0.9999. Our experimental results in Table V demonstrate that this value yields the best performance.

B.2 Analysis of Importance Weighting

B.2.1 Effect of Different Importance Weighting. We additionally conduct experiments on the DomainNet dataset to illustrate that domain-shared parameter subspaces hold varying degrees of importance to different domains. We utilize the first cross entropy loss term \mathcal{L} to acquire domain-shared subspaces and domain-specific subspaces in a momentum update manner. After that, we

manually introduce varying proportions of importance on domain-shared subspaces and domain-specific subspaces, ranging from $\{0, 0.1, 0.2, \dots, 1\}$. The results are shown in Figure VI. We obtain similar observations to those described in the main paper's experiments on the CDDb-Hard dataset(Section 4.2.3) as: (1) There is a notable variation in the importance of domain-shared information across each domain. (2) The utilization of domain-shared information enhances performance for each domain in the DomainNet dataset (When the proportion is set to zero, it means that we do not use domain-shared information). (3) The optimal proportion of domain-shared information significantly varies across different domains. We can draw the conclusion that dynamically assigning an importance weight to the domain-shared subspaces benefits the final performance of DIL.

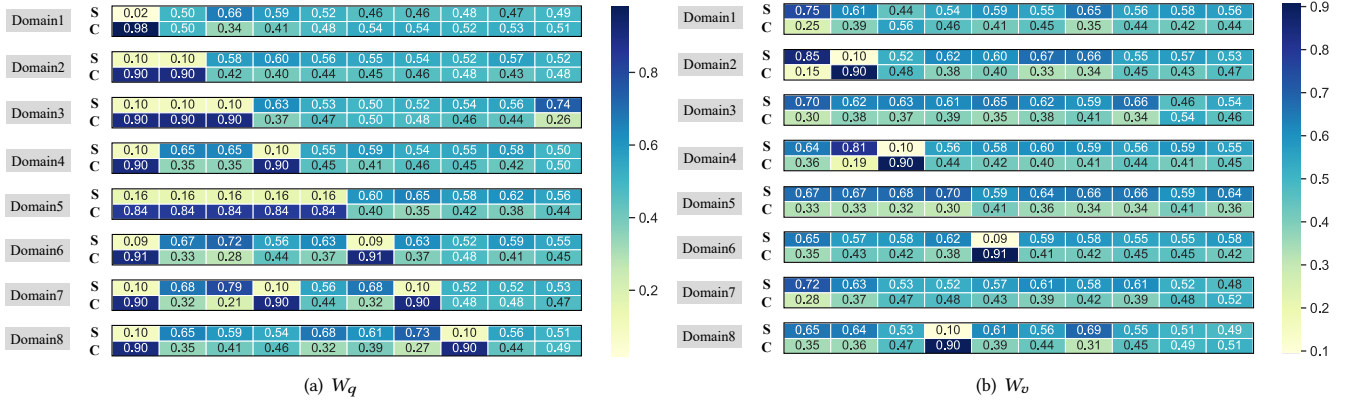


Figure VIII: Heatmap of the importance of domain-specific subspaces (denoted as S) and the importance of domain-shared subspaces (denoted as C) of eight domains from the Core50 dataset.

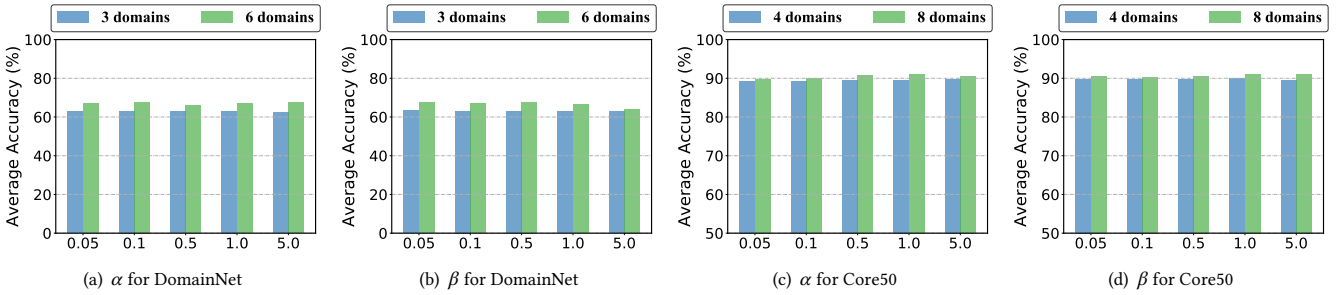


Figure IX: Parameter sensitivity analysis of the hyperparameter in the objective function described in Eq.(21) of the main paper on the DomainNet and Core50 datasets.

B.2.2 Visualization of the Obtained Importance. We additionally visualize the obtained importance of domain-shared and domain-specific subspaces by our method on the DomainNet and Core50 dataset. During the incremental training process of six domains on DomainNet and eight domains on Core50, we attach the decomposed subspaces to the first 10 transformer blocks. Specifically, we introduce the decomposed subspaces into the widely used query and value projection matrix (denoted as W_q and W_v) for the selected transformer layer, following [1]. Once the training is completed, we obtain the dynamical importance scores for different layers of all different domains. As shown in Figure VII and Figure VIII, we can clearly observe that there indeed exists varying importance degrees of domain-shared and domain-specific subspaces across different domains.

B.3 Impact for Hyper-parameters

We conduct the parameter sensitivity analysis on the hyperparameters α and β introduced in the final objective function (Eq.(21)) of the main paper. This analysis is additionally performed on the DomainNet and Core50 datasets. Recall that α is hyper-parameter of the cross-domain contrastive constraints, and β is the trade-off parameter involved in enforcing orthogonality on the domain-shared

and domain-specific subspaces. For evaluation, we keep all other hyperparameters to be fixed except for the one being tested. For two dataset, we analyse α and β varying from $\{0.05, 0.1, 0.5, 1, 5\}$ respectively, and report the results in Figure IX. we can see the performance of our method is relatively stable in a relatively wide range.

REFERENCES

- [1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.