Mind the Interference: Retaining Pre-trained Knowledge in Parameter Efficient Continual Learning of Vision-Language Models - Supplementary Materials -

Longxiang Tang[1], Zhuotao Tian[4], Kai Li[5], Chunming He[1], Hantao Zhou[1], Hengshuang Zhao[6], Xiu Li[1⋆], and Jiaya Jia[2,3]

[1]Tsinghua University [2]SmartMore [3]CUHK
[4]HIT(SZ) [5]Meta Reality Labs [6]HKU

A Proof of the Initialization Strategy in Eq. (7)

In Eq. (7), we stated that only values V_r^{init} should be initialized to zero, while keys K_r need to be random at the beginning. We argue that initializing both K_r and V_r to zero will result in K_r remaining zero throughout the whole training process, and cause V_r to degenerate into a matrix where all vectors are identical. Here, we provide a brief proof.

Recall the self-attention process for a single query vector $\boldsymbol{q} \in \mathbb{R}^d$, where d is the model embedding dimension. Note that in this proof, subscripts m and i denote the corresponding vectors of a matrix, while n and j denote subscripts for the individual elements within a vector.

We first derive the attention vector $\boldsymbol{z}$ with

$$\boldsymbol{z}_m = \frac{\sum_{j=1}^{d} \boldsymbol{q}_j \boldsymbol{K}_{m,j}}{\sqrt{d}}, \quad \boldsymbol{z} \in \mathbb{R}^l \tag{A.1}$$

where $\boldsymbol{K} \in \mathbb{R}^{l \times d}$ is the trainable key matrix, and the subscript represents taking the corresponding element. Then a softmax function will be applied to get normalized attention score $\boldsymbol{a}$:

$$\boldsymbol{a}_m = \frac{e^{\boldsymbol{z}_m}}{\sum_{j=1}^{l} e^{\boldsymbol{z}_j}}, \quad \boldsymbol{a} \in \mathbb{R}^l \tag{A.2}$$

Finally, the layer output vector $\boldsymbol{o}$ of the input query $\boldsymbol{q}$ can be derived with

$$\boldsymbol{o}_n = \sum_{i=1}^{l} \boldsymbol{a}_i \boldsymbol{V}_{i,n}, \quad \boldsymbol{o} \in \mathbb{R}^d \tag{A.3}$$

where $\boldsymbol{V} \in \mathbb{R}^{l \times d}$ is the trainable value matrix.

Now we prove our statement with these preliminaries.

⋆ Corresponding author

(1) First we discuss the situation that both key $\boldsymbol{K}$ and value $\boldsymbol{V}$ matrices are initialized to zero. Here we omit the multi-layer design of transformers and focus on one single attention layer. Assume that we have ground truth for output vector $\boldsymbol{o}$, and then we can get the training loss $\mathcal{L}$. Then we can calculate the derivative of $\mathcal{L}$ with respect to $\boldsymbol{K}$ and $\boldsymbol{V}$.

(i) The first training step.

Here we use the parenthesized superscript the denote the parameter after the corresponding training step. Before the first training step, we have

$$\boldsymbol{K}^{(0)} = \boldsymbol{V}^{(0)} = [\boldsymbol{0}]^{l\times d} \tag{A.4}$$

For the derivative of $\mathcal{L}$ with respect to $\boldsymbol{K}^{(0)}_{m,n}$, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{K}^{(0)}_{m,n}} &= \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}^{(0)}_j} \frac{\partial \boldsymbol{o}^{(0)}_j}{\partial \boldsymbol{K}^{(0)}_{m,n}} \\ &= \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}^{(0)}_j} \left(\sum_{i=1}^{l} \frac{\partial \boldsymbol{o}^{(0)}_j}{\partial \boldsymbol{a}^{(0)}_i} \frac{\partial \boldsymbol{a}^{(0)}_i}{\partial \boldsymbol{K}^{(0)}_{m,n}} \right) \\ &= \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}^{(0)}_j} \left(\sum_{i=1}^{l} \frac{\partial \boldsymbol{o}^{(0)}_j}{\partial \boldsymbol{a}^{(0)}_i} \frac{\partial \boldsymbol{a}^{(0)}_i}{\partial \boldsymbol{z}^{(0)}_m} \frac{\partial \boldsymbol{z}^{(0)}_m}{\partial \boldsymbol{K}^{(0)}_{m,n}} \right)\end{aligned} \tag{A.5}$$

Based on Eqs. (A.1) and (A.3), we know

$$\frac{\partial \boldsymbol{o}^{(0)}_j}{\partial \boldsymbol{a}^{(0)}_i} = \boldsymbol{V}^{(0)}_{i,j}, \quad \frac{\partial \boldsymbol{z}^{(0)}_m}{\partial \boldsymbol{K}^{(0)}_{m,n}} = \boldsymbol{q}_n \tag{A.6}$$

and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{o}^{(0)}_j}$ is an arbitrary value.

Then we discuss the value of $\frac{\partial \boldsymbol{a}_i}{\partial \boldsymbol{z}_m}$. For softmax function, it's easy to prove that:

$$\frac{\partial \boldsymbol{a}_s}{\partial \boldsymbol{z}_t} = \begin{cases} \boldsymbol{a}_s(1-\boldsymbol{a}_s) & , s = t \\ -\boldsymbol{a}_s \boldsymbol{a}_t & , s \neq t \end{cases} \tag{A.7}$$

With Eqs. (A.5) to (A.7), we can get the final derivative value:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{K}^{(0)}_{m,n}} &= \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}^{(0)}_j} \left(\frac{\partial \boldsymbol{o}^{(0)}_j}{\partial \boldsymbol{a}^{(0)}_m} \frac{\partial \boldsymbol{a}^{(0)}_m}{\partial \boldsymbol{z}^{(0)}_m} \frac{\partial \boldsymbol{z}^{(0)}_m}{\partial \boldsymbol{K}^{(0)}_{m,n}} + \sum_{i=1, i\neq m}^{l} \frac{\partial \boldsymbol{o}^{(0)}_j}{\partial \boldsymbol{a}^{(0)}_i} \frac{\partial \boldsymbol{a}^{(0)}_i}{\partial \boldsymbol{z}^{(0)}_m} \frac{\partial \boldsymbol{z}^{(0)}_m}{\partial \boldsymbol{K}^{(0)}_{m,n}} \right) \\ &= \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}^{(0)}_j} \left(\boldsymbol{V}^{(0)}_{m,j} \boldsymbol{a}^{(0)}_m (1-\boldsymbol{a}^{(0)}_m) \boldsymbol{q}_n + \sum_{i=1, i\neq m}^{l} \boldsymbol{V}^{(0)}_{i,j} (-\boldsymbol{a}^{(0)}_i \boldsymbol{a}^{(0)}_j) \boldsymbol{q}_n \right)\end{aligned} \tag{A.8}$$

With Eq. (A.4), it's easy to get:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{K}^{(0)}_{m,n}} = 0 \tag{A.9}$$

thus after the first parameter update, we get

$$\boldsymbol{K}^{(1)} = [\boldsymbol{0}]^{l\times d} \tag{A.10}$$

For the derivative of $\mathcal{L}$ w.r.t. $\boldsymbol{V}_{m,n}$, we have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_{m,n}^{(0)}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_n^{(0)}} \frac{\partial \boldsymbol{o}_n^{(0)}}{\partial \boldsymbol{V}_{m,n}^{(0)}} \tag{A.11}$$

With Eqs. (A.1), (A.2) and (A.4), we can get

$$\frac{\partial \boldsymbol{o}_n}{\partial \boldsymbol{V}_{m,n}^{(0)}} = \boldsymbol{a}_m^{(0)} = \frac{e^0}{\sum_{j=1}^{l} e^0} = \frac{1}{l} \tag{A.12}$$

and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_n}$ is an arbitrary value.

So we have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_{m,n}^{(0)}} = \frac{1}{l} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_n^{(0)}} \tag{A.13}$$

We can find that The value of $\frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_{m,n}^{(0)}}$ is independent of m, which means the gradients of all vectors in $\boldsymbol{V}$ are the same, formulated as

$$\boldsymbol{V}^{(1)} = \begin{pmatrix} \boldsymbol{e}^{(1)} \\ \vdots \\ \boldsymbol{e}^{(1)} \end{pmatrix} \in \mathbb{R}^{l\times d} \tag{A.14}$$

where $\boldsymbol{e}^{(1)}$ is arbitrary vector.

(*ii*) The subsequent training steps.

After the first training step, we get new parameter values according to Eqs. (A.10) and (A.14). Consider the second training step. Substituting Eqs. (A.10) and (A.14) into Eq. (A.8), we get:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{K}_{m,n}^{(1)}} = \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_j^{(1)}} \left(\boldsymbol{V}_{m,j}^{(1)} \boldsymbol{a}_m^{(1)} (1 - \boldsymbol{a}_m^{(1)}) \boldsymbol{q}_n + \sum_{i=1, i\neq m}^{l} \boldsymbol{V}_{i,j}^{(1)} (-\boldsymbol{a}_i^{(1)} \boldsymbol{a}_j^{(1)}) \boldsymbol{q}_n \right) \tag{A.15}$$

Since for all i, $\boldsymbol{V}_{i,j}^{(1)}$ are the same and $\boldsymbol{a}_i^{(1)} = \frac{1}{l}$, we can simplify Eq. (A.15) to

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{K}_{m,n}^{(1)}} &= \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_j^{(1)}} \boldsymbol{V}_{m,j}^{(1)} \boldsymbol{q}_n \left(\boldsymbol{a}_m^{(1)} (1 - \boldsymbol{a}_m^{(1)}) + \sum_{i=1, i\neq m}^{l} (-\boldsymbol{a}_i^{(1)} \boldsymbol{a}_j^{(1)}) \right) \\ &= \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_j^{(1)}} \boldsymbol{V}_{m,j}^{(1)} \boldsymbol{q}_n \left(\frac{l-1}{l^2} - \frac{l-1}{l^2} \right) \\ &= 0 \end{aligned} \tag{A.16}$$

So the $\boldsymbol{K}^{(2)}$ is still zero matrix:

$$\boldsymbol{K}^{(2)} = [\boldsymbol{0}]^{l \times d} \tag{A.17}$$

Since $\boldsymbol{a}^{(1)} = \boldsymbol{a}^{(0)} = \frac{1}{l}$, the derivative of $\mathcal{L}$ w.r.t. $\boldsymbol{V}_{m,n}^{(1)}$ becomes

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_{m,n}^{(1)}} = \frac{1}{l} \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}_n^{(1)}} \tag{A.18}$$

which is still independent of m, thus we have

$$\boldsymbol{V}^{(2)} = \begin{pmatrix} \boldsymbol{e}^{(2)} \\ \vdots \\ \boldsymbol{e}^{(2)} \end{pmatrix} \in \mathbb{R}^{l \times d} \tag{A.19}$$

where $\boldsymbol{e}^{(2)}$ is arbitrary vector.

It's easy to find that $\boldsymbol{K}^{(2)}, \boldsymbol{V}^{(2)}$ share the same properties as $\boldsymbol{K}^{(1)}, \boldsymbol{V}^{(1)}$, thus $\boldsymbol{K}$ remains zero throughout the subsequent training process, and $\boldsymbol{V}$ is degenerated into a matrix where all vectors are identical.

(2) We then discuss the scenario that key $\boldsymbol{K}$ is randomly initialized and value $\boldsymbol{V}$ is zero-initialized. It's obvious that Eqs. (A.12) to (A.14) no longer valid, resulting in $\boldsymbol{V}^{(1)}$ becoming an arbitrary matrix. After that, all subsequent $\boldsymbol{K}^{(i)}$ and $\boldsymbol{V}^{(i)}$ can be correctly trained.

B Effect of Cross-task Knowledge-sharing Strategies

As we discussed in the "Limitations and future work" section, recent literature has demonstrated the effectiveness of sharing knowledge across tasks in the class incremental learning setting, where the domain gap between tasks is relatively small. Here we implement two notable methods, G-Prompt from DualPrompt [6] and Attention-based Prompting from CODA-Prompt [4], into our DIKI framework, and test them under the challenging DCIL protocol, as shown in Tab. 1. G-Prompt is a set of shared prompts that are trained and utilized by all tasks, and the Attention-based Prompting mechanism weights prompts from different tasks based on key-similarity matching results, which can be naturally replaced by our distribution scores $\{S^i\}$ in Eq. (9). Results show that the integration of cross-task knowledge-sharing strategies leads to a decrease on *Last* metric, while *Transfer* scores remain comparable. It indicates that this degradation is caused by backward forgetting due to the sharing of task-specific knowledge. This observation underscores the need for further research into effective knowledge-sharing mechanisms specifically tailored for the DCIL setting.

C Algorithm Procedure

We elaborate on the training and test process of our proposed DIKI in Algorithms 1 and 2. We train separate K_r^i and V_r^i and maintain corresponding $\boldsymbol{\mu}^i$

Table 1: Results of our DIKI with cross-task knowledge-sharing strategies on MTIL benchmark. The G-Prompt [6] and the Attention-based Prompting (AbP) mechanism [4] are reproduced and integrated into our DIKI. Both two strategies don't work under the DCIL setting due to the severe domain gap between tasks.

	Transfer	Avg.	Last	Average
DIKI	**68.7**	**76.3**	**85.1**	**76.7**
+ G-Prompt	67.7	74.0	81.9	74.5
+ AbP	66.5	72.6	74.3	71.1

and $\boldsymbol{\Sigma}^i$ for each task during the training phase. At test time, the $\boldsymbol{\mu}^i$ and $\boldsymbol{\Sigma}^i$ are used to identify the task information for the test sample, and K_r^i and V_r^i are injected into the frozen backbone to reach better performance.

D Details about Reproduction

L2P [7] We reproduce L2P on CLIP by simply prompting both the visual and text encoders. In the original L2P paper, the updated prompts are selected by a key-matching mechanism during the training stage, and the diversity of prompt-selection is guaranteed by a frequency-based weight technique. However, in their official code repository[1], they mask specific prompts for different tasks. We follow the implementation of their official code.

At test time, we only select the top-1 prompt because, under the domain-class incremental learning setting, it is challenging to extract domain-invariant knowledge, and most of the learned knowledge is non-shareable. Adopting the original setting (i.e., top-5) would significantly degrade performance.

Regarding other hyper-parameters, the prompt length is set at 32, the learning rate is set to 0.05, and the weight of the key match loss is set at 5. The remaining training settings are the same as those in our DIKI.

DualPrompt [6] Similar to L2P, we simply prompt both the visual and text encoders to adapt DualPrompt to CLIP. Following the original paper, the prefix tuning is applied. DualPrompt separate prompts into G(eneral)-Prompt and E(xpert)-Prompt. However, similar to our discussion in Sec. B, we find that the G-Prompt will cause a significant performance drop. This is because the knowledge learned in different tasks is mostly non-shareable, different from class-incremental settings. So we remove the G-Prompt to prevent degradation.

Regarding other hyper-parameters, the prompt length and depth are both set at 8, the learning rate is set to 5, and the weight of the key match loss is set at 0.1. The remaining training settings are the same as those in our DIKI.

S-Prompts [5] S-Prompts is originally proposed on CLIP model, we simply adopt it on MTIL benchmark. Toward hyper-parameters, the prompt length is set at 32, the learning rate is set to 0.05. The remaining training settings are the same as those in our DIKI.

[1] `https://github.com/google-research/l2p`

Algorithm 1 Training process of DIKI.

Input: Training datasets $D^i = \{x_j^i, y_j^i\}_{j=1}^{N^i}$ with class names $C^i = \{c_j^i\}_{j=1}^{N_c^i}$ for each task, pre-trained image encoder f and text encoder g, learning rate η, batch size N_{bs}, max iterations $I_{\max}$.

for $i = 1, \cdots, N$ **do**
  $B_{\text{feat}} = \{\}$
  for $j = 1, \cdots, N^i$ **do**
    Calculate image feature $f(x_j^i)$
    Append $f(x_j^i)$ to B_{feat}
  end for
  Calculate $\boldsymbol{\mu}^i$ and $\boldsymbol{\Sigma}^i$ with B_{feat} ▷ Eq. (8)
  Initialize K_r^i with uniform distribution and V_r^i with zero ▷ Eq. (7)
  for $iter = 1, \cdots, I_{\max}$ **do**
    Fetch mini-batch samples $\{x_j^i, y_j^i\}_{j=1}^{N_{bs}}$ from D^i
    Insert K_r^i and V_r^i to f and g, get f' and g' ▷ Eq. (6)
    Calculate image features $\{f'(x_j^i)\}_{j=1}^{N_{bs}}$
    Calculate class name text embeddings $\{g'(c_j^i)\}_{j=1}^{N_c^i}$
    Compute cosine similarities between them $s_{j,k} = \langle f'(x_j^i), g'(c_k^i) \rangle$
    Get final predictions with softmax $p_{j,k} = \frac{\exp(s_{j,k})}{\sum_{k'}^{I} \exp(s_{j,k'})}$
    Calculate Cross-Entropy loss $\mathcal{L} = \text{CELoss}(p, y^i)$
    Update $K_r^i = K_r^i - \eta \nabla_{K_r^i} \mathcal{L}$
    Update $V_r^i = V_r^i - \eta \nabla_{V_r^i} \mathcal{L}$
  end for
end for

Algorithm 2 Test process of DIKI.

Input: Test dataset $D_t = \{x_j\}_{j=1}^{N_t}$ with class names $C^i = \{c_j^i\}_{j=1}^{N_c^i}$, pre-trained image encoder f and text encoder g, currently trained parameters $\{K_r^i, V_r^i\}_{i=1}^{N_{\text{cur}}}$, distribution parameters $\{\boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i\}_{i=1}^{N_{\text{cur}}}$.

for x in D_t **do**
  Calculate image feature $f(x)$
  Compute the logarithm of the probability density $\{S^i\}_{i=1}^{N_{\text{cur}}}$ ▷ Eq. (9)
  Get max value $\hat{S}$ and corresponding index s
  Calculate the integration calibration weight $\mathcal{M}(\hat{S})$ ▷ Eq. (10)
  Insert K_r^s and V_r^s to f and g, get f' and g' ▷ Eq. (6)
  Calculate image feature $f'(x)$ with calibration weight ▷ Eq. (10)
  Calculate text embeddings $\{g'(c_j^s)\}_{j=1}^{N_c^s}$ with calibration weight ▷ Eq. (10)
  Compute cosine similarities between them $s_j = \langle f'(x), g'(c_j^s) \rangle$
  Compute predictions with softmax $p_j = \frac{\exp(s_j)}{\sum_k \exp(s_k)}$ and get classification results
end for

E Details about MTIL Benchmark

E.1 Datasets

Authors introduced two different dataset orders in the original paper [7]. The first, Order-I, follows an alphabetical sequence: Aircraft, Caltech101, CIFAR100, DTD, EuroSAT, Flowers, Food, MNIST, OxfordPet, StanfordCars, SUN397. The second, Order-II, is arranged randomly: StanfordCars, Food, MNIST, OxfordPet, Flowers, SUN397, Aircraft, Caltech101, DTD, EuroSAT, CIFAR100. Order-I is adopted for results presented in Tab. 1 of the manuscript, and experiments were also conducted on Order-II, as indicated in Tabs. 9 to 12.

For our modified MTIL-FS benchmark in a few-shot setting, we only use 16 samples per class for model training. We exclude EuroSAT, MNIST, and OxfordPet due to their severely insufficient training samples caused by their small number of classes. More discussion on this can be found in Sec. G. To maintain reproducibility, we adopt the data splits from the official repository of CoOp [10], which is widely used by many CLIP-based few-shot learning works. Since CIFAR100 is not included by CoOp, we generate its training set by random selection with a random seed 42.

E.2 Metrics

Here we formulate the *Transfer*, *Avg.* and *Last* metrics.

Assume that $p_j^{(i)}$ is the model's accuracy on task j after being trained on task i, then the Transfer, Avg and Last metrics for task j can be calculated as:

$$\begin{aligned} \text{Transfer}_j &= \frac{1}{j-1}\sum_{i=1}^{j-1} p_j^{(i)}, \quad j = 2, 3, \cdots, N \\ \text{Avg}_j &= \frac{1}{N}\sum_{i=1}^{N} p_j^{(i)}, \quad j = 1, 2, \cdots, N \\ \text{Last}_j &= p_j^{(N)}, \quad j = 1, 2, \cdots, N \end{aligned} \tag{E.1}$$

where N is the number of tasks. It's clear that *Transfer* metric can indicate the zero-shot capability while *Last* metric shows the extent of backward forgetting.

F Additional Results

Tab. 2 shows the results of *Transfer*, *Avg.*, and *Last* metrics on MTIL benchmark with Order-II, and Tab. 3 provides full results of different continue learning methods on our modified 16-shot MTIL-FS benchmark. Our DIKI shows consistent improvements compared to previous methods.

For the selection of hyper-parameters, we perform a search on the structure parameters of our introduced K_r and V_r in our IKI. Length denotes the vector number l in Eq. (6), and depth indicates the number of layers implemented,

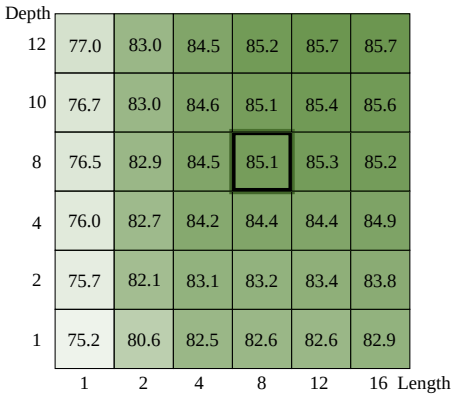

Fig. 1: *Last* score (%) with different IKI structure hyper-parameters. Setting highlighted in bold was chosen in our all experiments.

starting from the input layer. Given that our distribution-aware attention scaling scheme ensures minimal variation in the *Transfer* metric across different hyper-parameters, we focus on demonstrating the Last scores with varying parameters, as depicted in Fig. 1. Generally, an increase in the number of trainable parameters correlates with improved model accuracy. But we observe diminishing returns when depth and length exceed 8, thus we select a configuration of (8, 8) for all our experiments.

We also record the task assignment results during the test phase, as shown in Tab. 4. When the model is only trained on task i and earlier tasks, the task assignment results for samples from unseen tasks $i+1, \cdots, N$ are always incorrect. Thus we omit the meaningless upper triangular area and only consider the rest part. Results demonstrate that our task assignment on learned tasks holds high accuracy. Note that the misassignment of samples from unseen tasks is also resolved by our distribution-aware integration calibration.

Additionally, Tabs. 5 to 8 shows per training step accuracies of different methods on MTIL benchmark with Order-I, Tabs. 9 to 12 shows that results with Order-II, and Tabs. 13 to 17 shows per training step accuracies of different methods on MTIL-FS benchmark.

G Limitations and Future Directions

Our DIKI follows a task-specific tuning paradigm, where the training on different tasks is independent. Although some recent CIL research works have verified the effect of sharing knowledge across tasks [4, 6], we find these solutions are impractical within the DCIL context, as shown in Sec. B. We attribute this to the significant domain gap among DCIL datasets, which hinders the shareability of knowledge from different tasks. Future works could explore suitable knowledge-sharing strategies tailored to the DCIL problems.

Because of the use of parameter-efficient fine-tuning techniques, we could achieve high performance with significantly fewer trainable parameters. However, the knowledge learned by such a small number of parameters is definitely less than that obtained through full-parameter fine-tuning, as evident from the per-step accuracies table. For example, if we compare Tab. 5 and the Tab. 11 from the ZSCL paper [9], it's easy to find that our DIKI achieves lower accuracy when the model is tested immediately after training on the some datasets compared to ZSCL. The reason for our higher final performance is that DIKI can precisely memorize previously trained knowledge, while ZSCL suffers from backward forgetting issues. One future direction is to find a parameter-efficient fine-tuning method that can store more information to mitigate the gap.

In our modified MTIL-FS benchmark, we exclude three datasets for their lack of classes. This is because task-specific prompt learning methods (L2P [7], DualPrompt [6], S-Prompts [5], DIKI) can't obtain robust task identities with such limited training samples, leading to test performance degradation due to the inaccurate task assignments. This indicates a prevalent challenge associated with task-specific prompt learning methods: their heavy dependence on accurate task assignments. ZSCL [9] leverages knowledge distillation from large-scale reference datasets to alleviate the need for the task assignment process, which requires extensive computation and storage resources. Future works can tackle this issue by developing more robust task identification techniques or introducing task assignment-free prompt learning methods.

Table 2: *Transfer*, *Avg.*, and *Last* scores (%) of different continue learning methods on MTIL benchmark with Order-II. Metric "transfer" represents the model zero-shot ability retention after being trained on each task. † means we reproduce the original methods on vision-language models.

	Extra data	# Param.	Cars	Food	MNIST	OxfordPet	Flowers	SUN397	Aircraft	Caltech101	DTD	EuroSAT	CIFAR100	Average
Zero-shot			65.8	85.8	59.5	89.1	71.4	62.6	24.8	92.9	43.8	47.7	68.4	64.7
Upper Bound			89.6	92.7	99.6	94.7	97.5	81.8	62.0	96.2	79.5	98.9	89.6	89.3
Transfer														
LwF [2]	✓	211 M		87.8	58.5	71.9	46.6	57.3	12.8	81.4	34.5	34.5	46.8	53.2
iCaRL [3]	✓	211 M		86.1	51.8	67.6	50.4	57.9	11.0	72.3	31.2	32.7	48.1	50.9
LwF-VR [1]	✓	211 M		88.2	57.0	71.4	50.0	58.0	13.0	82.0	34.4	29.3	47.6	53.1
WiSE-FT [8]	✓	211 M		87.2	57.6	67.0	45.0	54.0	12.9	78.6	35.5	28.4	44.3	51.0
ZSCL* [9]	✓	211 M		**88.8**	56.7	75.5	58.8	62.5	16.1	87.0	42.0	44.0	66.5	59.8
ZSCL [9]	✓	211 M		88.3	57.5	84.7	68.1	**64.8**	21.1	88.2	**45.3**	**55.2**	**68.2**	64.2
L2P† [7]	×	0.5 M		70.6	30.7	78.3	42.8	38.3	17.4	75.3	27.4	23.1	20.7	42.5
DualPmt.† [6]	×	1.8 M		79.9	46.9	85.2	51.3	45.1	9.3	82.7	29.9	42.9	47.2	52.1
S-Prompts [5]	×	0.5 M		59.8	46.2	67.7	47.5	43.8	13.5	76.8	31.4	22.6	43.5	45.3
DIKI	×	1.8 M		85.8	**59.8**	**89.1**	**71.8**	62.6	**24.3**	**93.3**	42.7	46.8	67.8	**64.4**
Avg.														
LwF [2]	✓	211 M	49.0	77.0	**92.1**	85.9	66.5	67.2	20.9	84.7	44.6	45.5	50.5	62.2
iCaRL [3]	✓	211 M	52.0	75.9	77.4	74.6	58.4	59.3	11.7	79.6	42.1	43.2	51.7	56.9
LwF-VR [1]	✓	211 M	44.9	75.8	91.8	85.3	63.5	67.6	16.9	84.9	44.0	40.6	51.3	60.6
WiSE-FT [8]	✓	211 M	52.6	79.3	91.9	83.9	63.4	65.2	23.3	83.7	45.4	40.0	48.2	61.5
ZSCL* [9]	✓	211 M	72.0	89.8	91.7	87.9	78.8	71.5	**35.1**	89.0	51.4	53.9	68.5	71.8
ZSCL [9]	✓	211 M	81.7	**91.3**	91.1	91.0	82.9	**72.5**	33.6	89.7	**53.3**	**62.8**	**69.9**	**74.5**
L2P† [7]	×	0.5 M	80.1	87.4	86.7	89.6	76.8	59.1	27.7	79.5	39.9	34.6	26.5	62.5
DualPmt.† [6]	×	1.8 M	78.6	88.4	89.7	91.7	80.0	62.4	23.2	85.0	41.3	51.6	50.7	67.5
S-Prompts [5]	×	0.5 M	79.2	86.5	89.5	87.0	78.2	61.5	25.5	83.6	41.9	36.3	47.2	65.1
DIKI	×	1.8 M	**81.9**	88.9	**92.1**	**92.8**	**87.7**	70.3	34.3	**94.2**	51.5	56.1	69.5	**74.5**
Last														
LwF [2]	✓	211 M	34.6	69.6	99.3	88.7	61.1	72.5	32.5	88.1	65.6	90.9	87.9	71.9
iCaRL [3]	✓	211 M	46.0	81.5	91.3	82.8	66.5	72.2	16.3	91.6	68.1	83.2	87.8	71.6
LwF-VR [1]	✓	211 M	27.4	61.2	99.4	86.3	60.6	70.7	23.4	88.0	61.3	84.3	88.1	68.3
WiSE-FT [8]	✓	211 M	35.6	76.9	**99.5**	89.1	62.1	71.8	27.8	90.8	67.0	85.6	87.6	72.2
ZSCL* [7]	✓	211 M	63.5	89.6	99.2	92.4	84.5	**78.3**	**55.2**	92.4	74.6	97.4	**88.6**	83.3
ZSCL [7]	✓	211 M	78.2	**91.1**	97.6	92.5	87.4	78.2	45.0	92.3	72.7	96.2	86.3	83.4
L2P† [7]	×	0.5 M	80.1	89.1	99.1	93.8	96.2	76.5	40.1	86.9	73.5	86.3	84.2	82.3
DualPmt.† [6]	×	1.8 M	78.6	89.3	99.2	94.1	96.5	76.8	39.8	89.0	71.6	90.7	84.9	82.8
S-Prompts [5]	×	0.5 M	79.2	89.1	99.1	**94.3**	95.8	76.3	39.9	95.5	70.1	97.6	84.4	83.8
DIKI	×	1.8 M	**81.9**	89.2	99.4	**94.3**	**96.8**	76.7	46.3	**95.9**	**74.8**	**98.3**	86.6	**85.5**

Table 3: Full results of different continue learning methods on 16-shot MTIL-FS benchmark. † means we reproduce the original methods on vision-language models.

	Extra data	# Param.	Aircraft	Caltech101	CIFAR100	DTD	Flowers	Food	Cars	SUN397	Average
Zero-shot			24.8	92.9	68.4	43.8	71.4	85.8	65.8	62.6	64.4
Upper Bound			62.0	96.2	89.6	79.5	97.5	92.7	89.6	81.8	86.1
Transfer											
ZSCL [9]	✓	211 M		87.3	67.7	**45.4**	67.8	**86.6**	59.7	63.4	68.3
L2P† [7]	×	0.5 M		66.7	54.3	30.6	47.3	71.5	54.6	52.4	53.9
DualPmt.† [6]	×	1.8 M		78.8	64.4	32.0	51.7	77.5	49.4	51.3	57.9
S-Prompts [5]	×	0.5 M		70.3	52.7	31.5	54.8	74.0	55.4	50.0	55.5
DIKI	×	1.8 M		**92.7**	**68.8**	44.1	**70.0**	86.2	**65.1**	**65.5**	**70.3**
Avg.											
ZSCL [9]	✓	211 M	33.5	90.5	74.7	**58.5**	79.7	**87.7**	64.8	64.8	69.3
L2P† [7]	×	0.5 M	30.2	84.5	70.1	51.9	69.6	77.1	60.0	55.2	62.3
DualPmt.† [6]	×	1.8 M	36.5	89.5	72.5	52.7	72.3	80.8	56.1	54.2	64.3
S-Prompts [5]	×	0.5 M	30.6	86.8	70.0	51.7	74.3	78.5	60.7	53.0	63.2
DIKI	×	1.8 M	**41.3**	**95.3**	**76.5**	**58.5**	**82.2**	86.4	**68.2**	**66.6**	**71.9**
Last											
ZSCL [7]	✓	211 M	27.7	90.9	74.4	64.7	90.2	**89.2**	**80.6**	74.6	74.0
L2P† [7]	×	0.5 M	30.2	87.1	75.4	64.7	91.9	86.4	76.1	**74.7**	73.3
DualPmt.† [6]	×	1.8 M	36.5	91.0	75.1	65.1	92.9	86.2	76.2	74.2	74.7
S-Prompts [5]	×	0.5 M	30.6	89.2	75.8	63.8	93.9	86.2	76.7	73.9	73.8
DIKI	×	1.8 M	**41.3**	**95.6**	**79.0**	**67.3**	**94.4**	86.8	77.6	74.4	**77.1**

Table 4: Task assignment accuracy (%) for test data. Each row represents the assignment accuracy on every dataset of the model trained after the corresponding task.

	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food	MNIST	OxfordPet	Cars	SUN397
Aircraft	100.0	-	-	-	-	-	-	-	-	-	-
Caltech101	99.0	99.8	-	-	-	-	-	-	-	-	-
CIFAR100	99.0	99.8	99.6	-	-	-	-	-	-	-	-
DTD	99.0	99.6	99.6	97.5	-	-	-	-	-	-	-
EuroSAT	99.0	99.6	99.6	97.5	99.4	-	-	-	-	-	-
Flowers	99.0	99.1	99.6	97.0	99.4	97.7	-	-	-	-	-
Food	99.0	98.9	99.6	95.9	99.4	97.7	99.6	-	-	-	-
MNIST	99.0	98.9	99.6	95.9	99.4	97.7	99.6	99.6	-	-	-
OxfordPet	99.0	98.4	99.6	95.9	99.4	97.7	99.6	99.6	96.8	-	-
Cars	99.0	98.3	99.6	95.9	99.4	97.7	99.6	99.6	96.8	99.7	-
SUN397	98.3	95.5	99.5	94.3	99.3	97.7	99.0	99.6	96.0	99.1	99.3

Table 5: Accuracy (%) of our DIKI on the MTIL benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food	MNIST	OxfordPet	Cars	SUN397	
Transfer		92.9	69.0	43.2	48.2	67.4	85.2	63.0	87.9	63.8	66.2	68.7
Aircraft	45.2	92.9	68.4	43.9	47.7	71.3	85.8	59.8	89.2	65.8	62.4	
Caltech101	45.1	95.7	69.5	42.9	49.0	66.4	85.8	50.3	87.7	63.5	66.7	
CIFAR100	45.1	95.7	86.3	42.9	47.4	66.4	85.8	66.1	87.7	63.5	66.7	
DTD	45.1	95.7	86.3	72.9	48.7	66.3	84.5	66.1	87.7	63.5	66.6	
EuroSAT	45.1	95.7	86.3	72.9	98.0	66.3	84.5	66.1	87.7	63.5	66.6	
Flowers	45.1	95.7	86.3	72.9	98.0	97.0	84.5	66.1	87.7	63.5	66.6	
Food	45.1	95.7	86.3	72.9	98.0	97.0	89.2	66.1	87.7	63.5	66.6	
MNIST	45.1	95.7	86.3	72.9	98.0	97.0	89.2	99.4	87.7	63.5	66.6	
OxfordPet	45.1	95.8	86.3	72.9	98.0	97.0	89.2	99.4	94.2	63.5	66.6	
Cars	45.1	95.8	86.3	72.9	98.0	97.0	89.2	99.4	94.2	81.5	66.6	
SUN397	45.2	95.7	86.3	72.9	98.0	97.0	89.2	99.4	94.2	81.6	76.6	85.1
Avg.	45.1	95.5	83.1	64.8	79.9	83.5	87.0	76.2	89.6	67.0	67.1	76.3

Table 6: Accuracy (%) of L2P on the MTIL benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food	MNIST	OxfordPet	Cars	SUN397	
Transfer		65.6	50.9	30.4	41.4	49.3	71.8	36.3	77.5	55.3	53.4	53.2
Aircraft	38.0	65.6	40.7	16.6	26.4	22.1	43.9	39.9	54.8	57.8	41.8	
Caltech101	38.0	87.1	61.1	37.3	43.7	56.4	77.1	47.4	80.7	55.0	54.2	
CIFAR100	38.0	87.1	84.2	37.3	47.6	56.4	77.1	33.4	80.7	55.0	54.2	
DTD	38.0	87.1	84.2	72.9	47.6	55.8	77.6	33.4	80.7	55.0	54.9	
EuroSAT	38.0	87.1	84.2	72.9	97.4	55.8	77.6	33.4	80.7	55.0	54.7	
Flowers	38.0	87.1	84.2	72.9	97.4	96.1	77.6	33.4	80.7	55.0	54.6	
Food	38.0	87.1	84.2	72.9	97.4	96.1	89.2	33.4	80.7	55.0	54.6	
MNIST	38.0	87.1	84.2	72.9	86.0	96.1	89.2	99.0	80.7	55.0	54.6	
OxfordPet	38.0	87.1	84.2	72.9	86.0	96.1	89.2	99.0	94.1	55.0	54.9	
Cars	38.0	87.1	84.2	72.9	86.0	96.1	89.2	99.0	94.1	79.6	54.9	
SUN397	38.0	87.1	84.2	72.9	86.0	96.1	89.2	99.0	94.1	79.6	76.0	82.0
Avg.	38.0	85.2	78.2	61.3	72.9	74.9	79.7	59.1	82.0	59.7	55.4	67.9

Table 7: Accuracy (%) of DualPrompt on the MTIL benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food	MNIST	OxfordPet	Cars	SUN397	
Transfer		56.7	51.4	28.7	33.7	45.6	70.9	59.5	77.7	49.5	50.4	52.4
Aircraft	37.8	56.7	35.8	18.5	30.6	31.6	52.5	45.0	61.6	46.7	20.1	
Caltech101	37.8	87.1	67.0	33.9	53.7	52.9	73.7	48.0	80.0	49.9	54.4	
CIFAR100	37.8	87.1	84.6	33.9	25.2	52.9	73.7	64.7	80.0	49.9	54.4	
DTD	37.8	87.1	84.6	71.8	25.2	45.3	75.1	64.7	80.0	49.9	53.3	
EuroSAT	37.8	87.1	84.6	71.8	97.0	45.3	75.1	64.7	80.0	49.9	53.3	
Flowers	37.8	87.1	84.6	71.8	97.0	96.3	75.1	64.7	80.0	49.9	53.8	
Food	37.8	87.1	84.6	71.8	97.0	96.3	89.1	64.7	80.0	49.9	53.7	
MNIST	37.8	87.1	84.6	71.8	89.2	96.3	89.1	99.1	80.0	49.9	53.7	
OxfordPet	37.8	87.1	84.6	71.8	89.2	96.3	89.1	99.1	94.5	49.9	53.8	
Cars	37.8	87.1	84.6	71.8	89.2	96.3	89.1	99.1	94.5	79.9	53.4	
SUN397	37.8	87.1	84.6	71.8	89.2	96.3	89.1	99.1	94.5	79.9	76.5	82.3
Avg.	37.8	84.3	78.6	60.1	71.1	73.2	79.1	73.9	82.3	55.1	52.8	68.0

Table 8: Accuracy (%) of S-Prompts on the MTIL benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food	MNIST	OxfordPet	Cars	SUN397	
Transfer		67.3	49.4	26.4	39.7	47.1	70.2	34.3	78.9	56.7	52.2	52.2
Aircraft	37.5	67.3	40.1	12.8	23.5	15.3	41.1	37.5	47.7	57.9	37.9	
Caltech101	37.5	95.0	58.8	33.2	36.5	56.5	77.3	39.1	83.4	56.5	54.5	
CIFAR100	37.5	95.0	83.7	33.2	49.3	56.5	77.3	32.7	83.4	56.5	54.5	
DTD	37.5	95.0	83.7	70.2	49.3	53.5	75.2	32.7	83.4	56.5	53.7	
EuroSAT	37.5	95.0	83.7	70.2	97.5	53.5	75.2	32.7	83.4	56.5	53.7	
Flowers	37.5	95.0	83.7	70.2	97.5	96.5	75.2	32.7	83.4	56.5	53.7	
Food	37.5	95.0	83.7	70.2	97.5	96.5	89.0	32.7	83.4	56.5	53.7	
MNIST	37.5	95.0	83.7	70.2	97.5	96.5	89.0	99.1	83.4	56.5	53.7	
OxfordPet	37.5	95.0	83.7	70.2	97.5	96.5	89.0	99.1	94.0	56.5	53.7	
Cars	37.5	95.0	83.7	70.2	97.5	96.5	89.0	99.1	94.0	79.5	53.7	
SUN397	37.5	95.0	83.7	70.2	97.5	96.5	89.0	99.1	94.0	79.5	75.8	83.4
Avg.	37.5	92.5	77.5	58.2	76.4	74.1	78.8	57.9	83.0	60.8	54.4	68.3

Table 9: Accuracy (%) of our DIKI on the MTIL benchmark with order-II. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Cars	Food	MNIST	OxfordPet	Flowers	SUN397	Aircraft	Caltech101	DTD	EuroSAT	CIFAR100	
Transfer		85.8	59.8	89.1	71.8	62.6	24.3	93.3	42.7	46.8	67.8	64.4
Cars	81.9	85.8	59.7	89.2	71.5	62.6	24.9	92.9	44.0	47.6	68.4	
Food	81.9	89.2	59.9	89.1	71.9	62.7	24.9	93.1	43.8	47.7	68.5	
MNIST	81.9	89.2	99.3	89.1	71.9	62.7	24.9	93.1	43.8	47.7	68.5	
OxfordPet	81.9	89.2	99.3	94.2	71.9	62.6	24.9	93.0	43.8	47.7	68.4	
Flowers	81.9	89.2	99.3	94.2	96.7	62.6	24.9	93.0	44.0	47.7	68.4	
SUN397	81.9	89.2	99.3	94.2	96.8	76.7	21.2	94.0	40.8	44.6	67.4	
Aircraft	81.9	89.2	99.3	94.2	96.8	76.7	46.3	94.0	40.8	44.6	67.4	
Caltech101	81.9	89.2	99.3	94.3	96.8	76.7	46.3	95.9	40.7	44.7	67.2	
DTD	81.9	89.2	99.3	94.3	96.8	76.7	46.3	95.9	74.8	48.4	66.8	
EuroSAT	81.9	89.2	99.3	94.3	96.8	76.7	46.3	95.9	74.8	98.2	66.8	
CIFAR100	81.9	89.2	99.4	94.3	96.8	76.7	46.3	95.9	74.8	98.3	86.6	85.5
Avg.	81.9	88.9	92.1	92.8	87.7	70.3	34.3	94.2	51.5	56.1	69.5	74.5

Table 10: Accuracy (%) of L2P on the MTIL benchmark with order-II. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Cars	Food	MNIST	OxfordPet	Flowers	SUN397	Aircraft	Caltech101	DTD	EuroSAT	CIFAR100	
Transfer		70.6	30.7	78.3	42.8	38.3	17.4	75.3	27.4	23.1	20.7	42.5
Cars	80.1	70.6	41.1	67.6	42.1	44.6	17.5	79.0	27.8	24.3	51.8	
Food	80.1	89.1	20.3	83.7	56.9	50.1	17.5	84.7	28.9	25.1	52.0	
MNIST	80.1	89.1	99.1	83.7	56.9	29.8	17.5	44.2	14.4	12.7	12.9	
OxfordPet	80.1	89.1	99.1	93.8	15.2	30.0	17.5	69.4	14.4	12.7	12.9	
Flowers	80.1	89.1	99.1	93.8	96.2	37.1	17.5	77.8	27.8	12.7	12.9	
SUN397	80.1	89.1	99.1	93.8	96.2	76.5	16.8	89.7	35.9	29.8	12.9	
Aircraft	80.1	89.1	99.1	93.8	96.2	76.5	40.1	82.3	35.9	29.8	12.9	
Caltech101	80.1	89.1	99.1	93.8	96.2	76.5	40.1	86.9	33.8	29.8	12.9	
DTD	80.1	89.1	99.1	93.8	96.2	76.5	40.1	86.9	73.5	30.8	12.9	
EuroSAT	80.1	89.1	99.1	93.8	96.2	76.5	40.1	86.9	73.5	86.3	12.9	
CIFAR100	80.1	89.1	99.1	93.8	96.2	76.5	40.1	86.9	73.5	86.3	84.2	82.3
Avg.	80.1	87.4	86.7	89.6	76.8	59.1	27.7	79.5	39.9	34.6	26.5	62.5

Table 11: Accuracy (%) of DualPrompt on the MTIL benchmark with order-II. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Cars	Food	MNIST	OxfordPet	Flowers	SUN397	Aircraft	Caltech101	DTD	EuroSAT	CIFAR100	
Transfer		79.9	46.9	85.2	51.3	45.1	9.3	82.7	29.9	42.9	47.2	52.1
Cars	78.6	79.9	47.7	82.8	50.1	48.7	9.3	84.2	29.4	49.7	61.7	
Food	78.6	89.3	46.2	86.5	53.4	54.5	9.3	87.6	28.7	51.5	64.2	
MNIST	78.6	89.3	99.2	86.5	53.4	42.4	9.3	80.4	23.9	28.6	43.3	
OxfordPet	78.6	89.3	99.2	94.1	48.4	38.4	9.3	76.5	23.9	28.6	43.3	
Flowers	78.6	89.3	99.2	94.1	96.5	41.4	9.3	76.3	26.8	28.6	43.3	
SUN397	78.6	89.3	99.2	94.1	96.5	76.8	9.3	90.2	35.9	50.0	43.3	
Aircraft	78.6	89.3	99.2	94.1	96.5	76.8	39.8	83.8	35.9	50.0	43.3	
Caltech101	78.6	89.3	99.2	94.1	96.5	76.8	39.8	89.0	34.5	50.0	43.3	
DTD	78.6	89.3	99.2	94.1	96.5	76.8	39.8	89.0	71.6	49.4	43.3	
EuroSAT	78.6	89.3	99.2	94.1	96.5	76.8	39.8	89.0	71.6	90.7	43.3	
CIFAR100	78.6	89.3	99.2	94.1	96.5	76.8	39.8	89.0	71.6	90.7	84.9	82.8
Avg.	78.6	88.4	89.7	91.7	80.0	62.4	23.2	85.0	41.3	51.6	50.7	67.5

Table 12: Accuracy (%) of S-Prompts on the MTIL benchmark with order-II. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Cars	Food	MNIST	OxfordPet	Flowers	SUN397	Aircraft	Caltech101	DTD	EuroSAT	CIFAR100	
Transfer		59.8	46.2	67.7	47.5	43.8	13.5	76.8	31.4	22.6	43.5	45.3
Cars	79.2	59.8	60.1	55.0	26.9	38.0	13.4	70.3	27.5	14.3	39.7	
Food	79.2	89.1	32.3	74.0	56.1	47.2	13.4	76.6	27.7	18.1	53.5	
MNIST	79.2	89.1	99.1	74.0	56.1	46.8	13.4	72.6	30.5	18.7	42.7	
OxfordPet	79.2	89.1	99.1	94.3	50.9	44.3	13.4	66.2	31.4	18.7	42.7	
Flowers	79.2	89.1	99.1	94.3	95.8	42.5	13.4	77.8	27.7	18.7	42.7	
SUN397	79.2	89.1	99.1	94.3	95.8	76.3	13.9	91.3	35.5	29.4	42.7	
Aircraft	79.2	89.1	99.1	94.3	95.8	76.3	39.9	83.0	35.5	29.4	42.7	
Caltech101	79.2	89.1	99.1	94.3	95.8	76.3	39.9	95.5	35.2	29.4	42.7	
DTD	79.2	89.1	99.1	94.3	95.8	76.3	39.9	95.5	70.1	27.1	42.7	
EuroSAT	79.2	89.1	99.1	94.3	95.8	76.3	39.9	95.5	70.1	97.6	42.7	
CIFAR100	79.2	89.1	99.1	94.3	95.8	76.3	39.9	95.5	70.1	97.6	84.4	83.8
Avg.	79.2	86.5	89.5	87.0	78.2	61.5	25.5	83.6	41.9	36.3	47.2	65.1

Table 13: Accuracy (%) of our DIKI on the MTIL-FS benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	Flowers	Food	Cars	SUN397	
Transfer		92.7	68.8	44.1	70.0	86.2	65.1	65.5	70.3
Aircraft	41.4	92.7	68.4	43.9	71.3	85.8	65.8	62.5	
Caltech101	41.3	95.7	69.2	44.2	69.5	86.3	64.9	66.0	
CIFAR100	41.3	95.7	79.0	44.2	69.5	86.3	64.9	66.0	
DTD	41.3	95.7	79.0	67.1	69.5	86.3	64.9	66.0	
Flowers	41.3	95.7	79.0	67.1	94.5	86.3	64.9	66.0	
Food	41.3	95.7	79.0	67.1	94.5	86.8	64.9	66.0	
Cars	41.3	95.7	79.0	67.1	94.5	86.8	77.5	66.0	
SUN397	41.3	95.6	79.0	67.3	94.4	86.8	77.6	74.4	77.1
Avg.	41.3	95.3	76.5	58.5	82.2	86.4	68.2	66.6	71.9

Table 14: Accuracy (%) of L2P on the MTIL-FS benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	Flowers	Food	Cars	SUN397	
Transfer		66.7	54.3	30.6	47.3	71.5	54.6	52.4	53.9
Aircraft	30.1	66.7	44.3	23.0	32.4	47.8	49.9	32.9	
Caltech101	30.1	87.1	64.2	34.5	53.5	77.6	55.6	56.5	
CIFAR100	30.1	87.1	75.3	34.5	53.5	77.6	55.6	56.5	
DTD	30.1	87.1	75.3	64.7	49.7	77.3	55.6	55.6	
Flowers	30.1	87.1	75.3	64.7	91.9	77.3	55.6	55.0	
Food	30.1	87.1	75.3	64.7	91.9	86.4	55.6	55.0	
Cars	30.1	87.1	75.3	64.7	91.9	86.4	76.2	55.5	
SUN397	30.1	87.1	75.3	64.7	91.9	86.4	76.2	74.7	73.3
Avg.	30.2	84.5	70.1	51.9	69.6	77.1	60.0	55.2	62.3

Table 15: Accuracy (%) of DualPrompt on the MTIL-FS benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	Flowers	Food	Cars	SUN397	
Transfer		78.8	64.4	32.0	51.7	77.5	49.4	51.3	57.9
Aircraft	36.5	78.8	61.5	28.4	51.6	79.4	57.5	52.2	
Caltech101	36.5	91.0	67.4	33.8	51.5	75.0	47.8	51.5	
CIFAR100	36.5	91.0	75.1	33.8	51.5	75.0	47.8	51.5	
DTD	36.5	91.0	75.1	65.1	52.2	79.2	47.8	51.1	
Flowers	36.5	91.0	75.1	65.1	92.9	79.2	47.8	51.1	
Food	36.5	91.0	75.1	65.1	92.9	86.2	47.8	51.1	
Cars	36.5	91.0	75.1	65.1	92.9	86.2	76.2	50.7	
SUN397	36.5	91.0	75.1	65.1	92.9	86.2	76.2	74.2	74.7
Avg.	36.5	89.5	72.5	52.7	72.3	80.8	56.1	54.2	64.3

Table 16: Accuracy (%) of S-Prompts on the MTIL-FS benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	Flowers	Food	Cars	SUN397	
Transfer		70.3	52.7	31.5	54.8	74.0	55.4	50.0	55.5
Aircraft	30.6	70.3	44.5	24.5	46.2	72.6	53.7	32.3	
Caltech101	30.6	89.2	60.8	35.0	60.2	75.7	55.7	53.8	
CIFAR100	30.6	89.2	75.8	35.0	60.2	75.7	55.7	53.8	
DTD	30.6	89.2	75.8	63.8	52.7	72.8	55.7	52.4	
Flowers	30.6	89.2	75.8	63.8	93.9	72.8	55.7	52.4	
Food	30.6	89.2	75.8	63.8	93.9	86.2	55.7	52.4	
Cars	30.6	89.2	75.8	63.8	93.9	86.2	76.7	52.4	
SUN397	30.6	89.2	75.8	63.8	93.9	86.2	76.7	73.9	73.8
Avg.	30.6	86.8	70.0	51.7	74.3	78.5	60.7	53.0	63.2

Table 17: Accuracy (%) of ZSCL on the MTIL-FS benchmark with order-I. Each row represents the performance on every dataset of the model trained after the corresponding task. Transfer, Avg., and Last metrics are shown in color.

	Aircraft	Caltech101	CIFAR100	DTD	Flowers	Food	Cars	SUN397	
Transfer		87.3	67.7	45.4	67.8	86.6	59.7	63.4	68.3
Aircraft	41.0	87.3	67.8	45.4	68.6	88.5	63.2	64.1	
Caltech101	38.5	91.5	67.7	45.0	65.4	85.9	59.6	62.9	
CIFAR100	37.1	91.4	79.5	45.7	68.6	87.3	60.0	64.7	
DTD	36.0	91.2	78.6	68.6	68.5	86.4	59.3	62.9	
Flowers	32.1	91.1	77.3	67.5	93.8	85.1	58.3	63.1	
Food	30.0	90.9	76.8	66.5	91.7	90.0	57.7	62.8	
Cars	25.7	90.2	75.4	64.6	90.8	89.2	80.1	63.3	
SUN397	27.7	90.9	74.4	64.7	90.2	89.2	80.6	74.6	74.0
Avg.	33.5	90.5	74.7	58.5	79.7	87.7	64.8	64.8	69.3

References

1. Ding, Y., Liu, L., Tian, C., Yang, J., Ding, H.: Don't stop learning: Towards continual learning for the clip model. arXiv preprint arXiv:2207.09248 (2022)
2. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
3. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
4. Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11909–11919 (2023)
5. Wang, Y., Huang, Z., Hong, X.: S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. Advances in Neural Information Processing Systems **35**, 5682–5695 (2022)
6. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: European Conference on Computer Vision. pp. 631–648. Springer (2022)
7. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 139–149 (2022)
8. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Lopes, R.G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al.: Robust fine-tuning of zero-shot models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7959–7971 (2022)
9. Zheng, Z., Ma, M., Wang, K., Qin, Z., Yue, X., You, Y.: Preventing zero-shot transfer degradation in continual learning of vision-language models. arXiv preprint arXiv:2303.06628 (2023)
10. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022)