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

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

Abstract. This study addresses the Domain-Class Incremental Learning problem, a realistic but challenging continual learning scenario where both the domain distribution and target classes vary across tasks. To handle these diverse tasks, pre-trained Vision-Language Models (VLMs) are introduced for their strong generalizability. However, this incurs a new problem: the knowledge encoded in the pre-trained VLMs may be disturbed when adapting to new tasks, compromising their inherent zero-shot ability. Existing methods tackle it by tuning VLMs with knowledge distillation on extra datasets, which demands heavy computation overhead. To address this problem efficiently, we propose the Distribution-aware Interference-free Knowledge Integration (DIKI) framework, retaining pre-trained knowledge of VLMs from a perspective of avoiding information interference. Specifically, we design a fully residual mechanism to infuse newly learned knowledge into a frozen backbone, while introducing minimal adverse impacts on pre-trained knowledge. Besides, this residual property enables our distribution-aware integration calibration scheme, explicitly controlling the information implantation process for test data from unseen distributions. Experiments demonstrate that our DIKI surpasses the current state-of-the-art approach using only 0.86% of the trained parameters and requiring substantially less training time. Code is available at: `https://github.com/lloongx/DIKI`.

1 Introduction

Supervised learning techniques train networks with full access to all data, which can result in a lack of flexibility when extending them to acquire knowledge from new tasks. Continual Learning (CL) has emerged as a solution, enabling ongoing model training on sequentially arriving data while retaining the learned information [8]. Conventional CL settings consider either newly introduced classes or domain distribution shifts, referred to as class incremental and domain incremental learning [74]. However, with only one type of increment considered, these existing works limit their applicability in complex real-world scenarios.

* Corresponding author

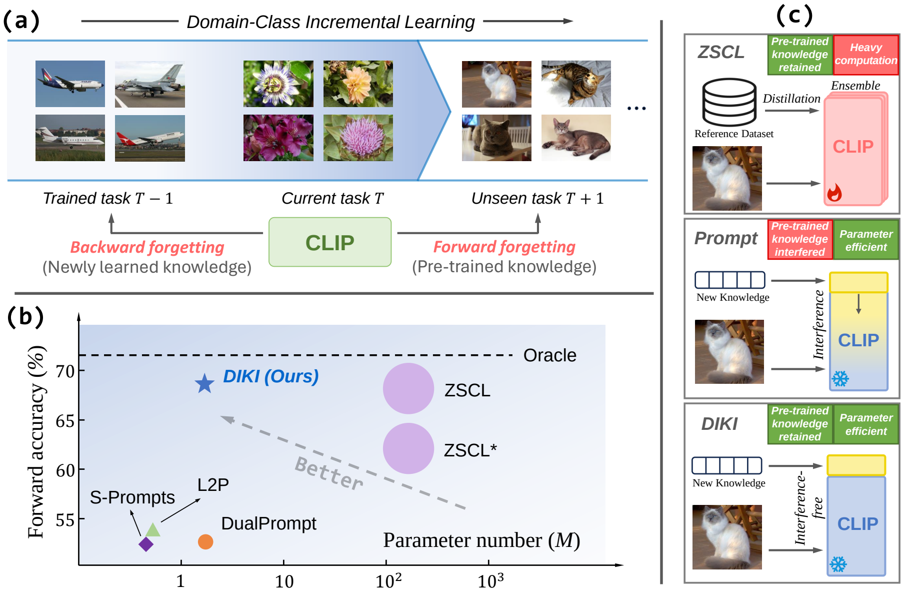


Fig. 1: (a): The domain-class incremental learning setting, where the data distribution and the classes vary across all tasks. Two kinds of forgetting exist due to the integration of pre-trained CLIP. **(b)**: The forward accuracy (i.e. zero-shot ability) and the number of trainable parameters for each method, with the size of the markers representing their computational complexity. **(c)**: Existing methods either demand heavy computation or sacrifice pre-trained knowledge. Our approach effectively retain pre-trained knowledge within a parameter-efficient framework. More details are provided in Sec. 4.1.

Consider a more challenging Domain-Class Incremental Learning (DCIL) setting, where both the domain data distribution and classes to be classified can keep varying among all tasks, as illustrated in Fig. 1(a). Vanilla image encoder-based techniques are infeasible under such circumstances due to their non-scalable classification head design [5, 23, 53–55]. Recently, the advent of contrastively trained Vision-Language Models (VLMs), such as CLIP [57], has made it possible to address this demanding but practical problem. VLMs are trained on web-scale image-text pairs and hold a powerful zero-shot generalization ability to identify nearly infinite classes, making them capable of confronting this severe task variation scenario [11, 22, 41, 61, 95].

However, the use of vision-language models introduces new challenges to incremental training. Conventional continual learning schemes aim to prevent models from forgetting previously learned knowledge, which is termed as *backward forgetting* [46]. Existing works have explored the potential of the regularization mechanism, rehearsal buffer, and architecture design to mitigate backward forgetting, achieving promising results [14, 44, 59, 64]. Nevertheless, when these approaches are applied to vision-language models, a different form of catastrophic forgetting emerges: **models tend to forget the knowledge learned**

during the pre-training phase, thus compromising their powerful zero-shot generalization capacity. This problem is termed as **forward forgetting** because it occurs when VLMs perform "forward" prediction on the unknown distributed data. Fig. 1(a) illustrates the two types of forgetting.

Recent work ZSCL [91] made an attempt to address the forward forgetting issue on CLIP. They introduced a large-scale reference dataset [9] to perform knowledge distillation and incorporated a weight ensemble scheme [81]. However, this approach requires intensive computation and external data, which could be infeasible in real-world scenarios. Meanwhile, existing VLM-based parameter-efficient continual learning methods [78–80], mostly utilizing prompt tuning mechanisms, fail to retain the pre-trained knowledge and cause zero-shot ability degradation, as shown in Fig. 1(b). We attribute this issue to **information interference**: newly introduced task-specific parameters can disturb the pre-trained knowledge. Illustrations of these methods are shown in Fig. 1(c).

To alleviate the forward forgetting problem of VLMs with a computationally and parameter-efficient approach, we introduce the **Distribution-aware Interference-free Knowledge Integration (DIKI)** framework. Specifically, we inject task-specific information into frozen VLM for each task, storing learned knowledge efficiently. (1) To maintain the pre-trained knowledge in VLMs, our knowledge integration mechanism is designed to resolve the information interference issue prevalent in existing methods. By employing our fully residual design and zero-initialization strategy, we can inject new knowledge while keeping the pre-trained knowledge untouched, introducing minimal noise to the pre-trained model compared to prompt tuning. (2) With this advantage, we further introduce a distribution-aware integration calibration mechanism, explicitly identifying the unseen distributed samples and controlling the implanted information for them, thereby enhancing the model generalization capabilities.

Our contributions are summarized in threefold:

- We introduce the parameter-efficient DIKI to retain pre-trained knowledge in VLMs under the DCIL setting. It resolves the information interference issue, mitigating the need for heavy computation and external data.
- To alleviate the forward forgetting, DIKI implants new knowledge in a fully residual manner, leaving pre-trained knowledge undisturbed. With this residual property, a distribution-aware integration calibration is incorporated to further boost performance on unseen tasks.
- Comprehensive experiments demonstrate that we achieve state-of-the-art performance with only 0.86% trained parameters and significantly less training time compared to the previous methods.

2 Related Works

Continual learning. Existing continual learning algorithms can be broadly classified into three categories [8]. *Regularization-based* methods [1,2,36,40,44,89]

introduce an extra regularization term in the loss function, consolidating previous knowledge when learning on new data. In contrast, *architecture-based* methods [14, 43, 48, 58, 87] dedicate different model parameters to each task, storing task knowledge with specific expanded network components. With memory replay technique, *rehearsal-based* methods [29,46,59,60,64] retrain current step model with stored exemplars in raw format or generated pseudo-samples with a generative model, which has been questioned for its rationality by recent work [52]. While achieving promising results, these solutions only consider one type of increment, either domain shift or new classes, along the continual training process, resulting in limited applicability in real-world scenarios. Instead, we investigate the forgetting problem under a domain-class incremental learning setting to adapt to a broader variety of situations.
Parameter-efficient fine-tuning. Fully fine-tuning a large pre-trained model is computationally expensive and requires a large-scale dataset [81]. Alternatively, parameter-efficient fine-tuning approaches only introduce a small set of parameters to rapidly adapt a pre-trained model to downstream tasks, such as LoRA [27], prompt tuning [31, 45, 67, 85] and adapters [26, 76, 77]. Due to their simple and portable design, prompt tuning techniques have attracted many applications in a variety of areas [17, 24, 33, 35, 95]. However, existing prompt learning-based methods typically prepend the learnable parameters to the original input tokens, where lies the information interference issue and eventually causes pre-trained knowledge loss during the training process.
Vision-language models. Trivial visual-only models extract features from images and then utilize a fixed head to derive final predictions, constraining their flexibility across tasks [15, 18, 19, 21, 23, 70]. Vision-Language Models (VLMs) present a solution by leveraging the interaction between image and text descriptions [30,39,57,83,84,86,88,93]. Trained on web-scale image-text pair datasets, V-L models can identify nearly infinite classes and can be easily transferred to unseen domains, holding a strong zero-shot ability. However, most previous VLMs continual learning methods [34,56,65,78,92] have not considered the zero-shot performance drop during the training process, which can cause a significant model degradation towards unseen data distributions.

3 Preliminaries

Continual learning protocol. Continual learning aims to sequentially learn different tasks without forgetting previously learned knowledge. Considering N sequentially arrived tasks $\left[\mathcal{T}^1, \mathcal{T}^2, \cdots, \mathcal{T}^N\right]$, each task $\mathcal{T}^i$ contains a dataset $D^i = \{x^i_j, y^i_j\}^{N^i}_{j=1}$, where x^i_j is an image and y^i_j is corresponding one-hot label inside current dataset, and N_i is the number of image samples. Additionally, a class name set $C^i = \{c^i_j\}^{N^i_c}_{j=1}$ is included, linking the label index to a category name used by the VLMs.

Different from previous class- and domain-incremental learning settings, this work highlights a more practical continual learning setting: Domain-Class Incremental Learning (DCIL). In this setting, domain distribution and classes to be identified keep varying among different tasks, i.e. $C^i \neq C^j$ and $\mathbb{P}(D^i) \neq \mathbb{P}(D^j)$ for $i \neq j$, where $\mathbb{P}$ represents data distribution of a task dataset.

Vision-language models. Towards the challenging DCIL setting, training a vanilla image encoder-based model, such as ResNets [23] and ViTs [13], is not practical for incrementally learning intensely shifted domains and classes. Hence, pre-trained vision-language models are introduced for their robust zero-shot transfer capabilities. CLIP [57] consists of an image encoder f and a text encoder g, which are trained to generate closely aligned feature representations for paired image-text samples. At inference time, f first encodes the input image x into a feature vector $f(x)$. Concurrently, potential class names are embedded into a template, like "a photo of $\{c\}$", and then encoded by g to form text embeddings $\{t_j\}_{j=1}^{N_c}$. The model predictions are determined by the largest similarity scores between image embedding and all text embeddings, formulated as $s_j = \langle f(x), t_j \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity.

Task-specific prompt learning. Following the success of [79, 80], a series of works [4,28,66] begin to explore the potential of parameter-efficient fine-tuning in continual learning. A common practice is learning and storing a set of lightweight prompts for each task, forming a "prompt pool" during the continual learning phase, formulated as:

$$\mathbf{P} = \{P_1, P_2, \cdots, P_N\}, \quad \text{where } P_i \in \mathbb{R}^{l \times d}, \tag{1}$$

where N is the task number, l and d are the prompt length and the feature embedding dimension.

At inference time, well-trained prompts are selected and attached to the pre-trained frozen model, restoring the learned knowledge. Assume $\boldsymbol{x_e} \in \mathbb{R}^{L \times d}$ is the feature embeddings for a transformer layer h, then we can prepend the prompts to the $\boldsymbol{x_e}$ to generate prompted inputs:

$$\boldsymbol{x_p} = \left[P_s^1; P_s^2; \cdots; P_s^l; \boldsymbol{x_e}\right] \in \mathbb{R}^{(l+L) \times d}, \tag{2}$$

where $\{P_s^i \in \mathbb{R}^d\}_{i=1}^l$ are embedding vectors of selected prompt P_s and ; represents the concatenation operation along the token length dimension. With this implanted knowledge, better image and text feature embeddings are generated, and the final classification accuracy is improved.

The prompt selection process mentioned above is implemented by query-key matching. During the continual training stage, average feature representations $\mathbf{I} = \{I^i\}_{i=1}^N$ for each task are learned by maximizing cosine similarity [79, 80] or by applying clustering algorithm [78]. When a test sample $\boldsymbol{x}$ comes, a key lookup regime is performed:

$$I_s = \arg\max_{I^i \sim \mathbf{I}} \left\langle f(\boldsymbol{x}), I^i \right\rangle. \tag{3}$$

With the most relevant key I_s, corresponding prompts P_s are selected and attached to the frozen model, performing inference process.

4 Methodology

4.1 Interference-free Knowledge Integration

Is prepending the best choice? Despite methods that prepend prompt to input tokens are widely used for their simplicity in implementation, we identified that they are suffering from issues in two folds.

Firstly, concatenating the prompts with input tokens causes them to interact during the attention process, and influences the pre-trained knowledge extraction, which will be discussed below. When the test samples are drawn from the distribution where the model learned the prompts, the adapted model can preserve relatively satisfactory results. However, once encountering samples with a distribution shift, this interference could result in model degradation and a loss of its vital zero-shot generalization ability, causing forward forgetting issues.

Besides, simply prepending prompts inevitably increases the token length across all transformer blocks, which is not desirable in many scenarios with token length constraints. In addition, its scalability is limited: a long prompt context can distract the text encoder from informative class names, resulting in poor text embedding representation.

The existence of the above issues indicates that prompt tuning-based methods do not satisfy the "residual property": we expect learned parameters should be a residual path paralleled to the frozen backbone, supplementing novel knowledge without affecting the crucial pre-trained knowledge. Therefore, we propose a *Interference-free Knowledge Integration (IKI)* scheme to inject newly learned knowledge into a pre-trained VLM with introducing minimal noise to it.

IKI mechanism. Instead of training a series of prepended prompt vectors for each task, we focus on self-attention mechanism modification following widely used parameter efficient fine-tuning methods in NLP field [27,42,71,90]. Recall the multi-head self-attention [73] mechanism conducted on input tokens $\boldsymbol{x_e} \in \mathbb{R}^{L\times d}$ in transformer layer h. For simplification, we omit the multi-head design and solely consider the one-head situation, which can be naturally extended to multi-head scenarios. Input tokens are first transformed to query Q, key K and value V matrices by linear projections:

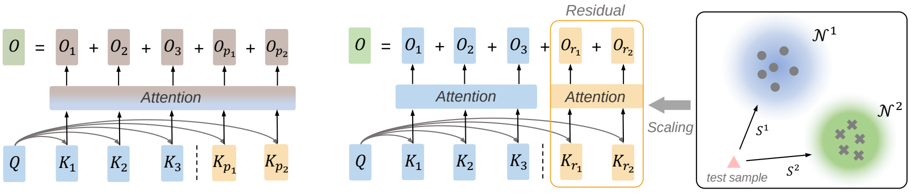
$$Q_e = \boldsymbol{x_e} W^Q + b^Q; K_e = \boldsymbol{x_e} W^K + b^K; V_e = \boldsymbol{x_e} W^V + b^V, \tag{4}$$

where $W \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$ are pre-trained parameters. Then self-attention calculation is performed to produce an output matrix via

$$O_L = \mathrm{Attn}(Q_e, K_e)V_e = \mathrm{softmax}(\frac{Q_e K_e^T}{\sqrt{d}})V_e \quad \in \mathbb{R}^{L\times d}, \tag{5}$$

where $\mathrm{softmax}(\boldsymbol{z})_i = \frac{\exp{(\boldsymbol{z_i})}}{\sum_j \exp{(\boldsymbol{z_j})}}$ can constrain the elements in attention results $\mathrm{Attn}(Q_e, K_e) \in \mathbb{R}^{L\times L}$ sum to one.

Vanilla prompt tuning methods prepend trainable prompts to input tokens, extending $\boldsymbol{x_e} \in \mathbb{R}^{L\times d}$ to $\boldsymbol{x_p} \in \mathbb{R}^{(l+L)\times d}$. Then $Q_p K_p^T \in \mathbb{R}^{(l+L)\times(l+L)}$ will be



(a) Interference in prompt tuning

(b) Distribution-aware Interference-free Knowledge Integration

Fig. 2: Illustration of the information interference issue in previous prompt tuning methods and our proposed DIKI. (a) The existing methods mix attention derived from the frozen backbone and prepended prompts, which can cause information loss and finally harm the zero-shot ability. (b) We design a zero-initialized residual attention mechanism, which injects new information with pre-trained knowledge untouched, to retain the vision-language models' zero-shot ability. Distribution-aware integration calibration is also introduced to further boost performance thanks to the residual property.

computed and passed to a softmax function. Inside softmax calculation, attention scores of input tokens and prompts interact and affect each other, leading to an inevitable loss of pre-trained knowledge, as illustrated in Fig. 2 (a).

To address this issue, we compute attention outputs for self-attention within input tokens and cross-attention between prompts and input tokens separately, as shown in Fig. 2 (b). In other words, we only train a residual attention branch, leaving the existing attention score untouched. With newly introduced keys K_r and values V_r, the output of our residual attention branch can be formulated as:

$$O_r = \text{softmax}(\frac{Q_e K_r^T}{\sqrt{d}})V_r, \text{where } K_r, V_r \in \mathbb{R}^{l \times d}. \tag{6}$$

Here the residual output $O_r \in \mathbb{R}^{L \times d}$ is derived with an orthogonal path to the original output O_L, producing no influence on the original attention process. Finally, the learned knowledge stored in O_r is implanted into output by addition. During continual training stage, we update the learnable keys K_r and values V_r instead of commonly used prompts P. Note that to keep sequence length unchanged, we didn't introduce any query parameters.

Ideally, a desirable residual block should not affect the original branch before being trained on downstream datasets, i.e. at initialization time. Widely used protocols initialize prompts with uniform or normal distribution, which injects random noise into the pre-trained VLMs even when no knowledge has been learned. Specifically, we enforce residual attention addition to be an identity function by zero-initialize the parameters V_r:

$$O = O_L + O_r^{\text{init}} = O_L + \text{softmax}(\frac{Q_e K_r^T}{\sqrt{d}})[\mathbf{0}]^{l \times d} = O_L. \tag{7}$$

Note that we only constrain values V_r^{init} to be zero at the beginning, while keeping K_r random initialized. That's because initializing both K_r and V_r to zero matrix will prevent K_r from updating by gradient flow, and make V_r degenerate to vectors with same values. We prove this in the supplementary materials.

Since zero-initialization is more like a choice rather than a technique, some studies [6, 32, 90] have adopted it across various tasks. However, these works leverage it to ensure a stable and progressive training regime, a concern that is not present in DCIL scenarios. We argue that zero-initialization is essential for our residual attention design to inject new knowledge into the pre-trained VLMs with minimal noise introduced, which is demonstrated in Sec. 5.2.

4.2 Distribution-aware Integration Calibration

Observations. At inference time, the query-key matching mechanism described in Eq. (3) is performed to retrieve appropriate learned prompts for the current test sample. This approach is tailored for conventional continual learning settings, which only considers the backward forgetting mentioned in Sec. 1. However, when confronted with data from unseen domains, this trivial matching design is enforced to assign a relatively similar task for test samples, despite there's a significant distribution gap between them.

Benefiting from the residual design of our proposed IKI, we can introduce less noise in such mismatch scenarios compared with previous methods. Nonetheless, when the discrepancy between training and testing distribution increases, it's inevitable to cause model degradation to some extent and hurt the zero-shot ability that VLMs learned during the pre-train phase.

ZSCL [91] tackles this problem via distillation. They build a reference dataset with *100k* images from ImageNet [9] to distill pre-trained knowledge from the original CLIP to the current model at every training step, explicitly performing rehearsal to avoid forgetting. This approach could be effective, but it relies on large-scale storage and high computation resources as shown in Tab. 5, making it impractical under real-world circumstances.

One intuitive solution to this issue is controlling to what extent knowledge is implanted into the model. However, previous prepending-based prompt tuning techniques have only two choices: either appending learned prompts or leaving the original CLIP model untouched. Thanks to the graceful residual property from our IKI, we obtain the ability to control this paralleled branch.

DIKI: calibrate the integration with distribution. To determine the likelihood that a test sample belongs to a learned task, we maintain a feature distribution [20,51,68,69,72,75] instead of a single key vector for every task. Here we simply apply multivariate Gaussian distribution and find it works well. Formally, we build a $\mathcal{N}^i(\boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i)$ for task i during training stage:

$$\boldsymbol{\mu}^i = \mathbb{E}_{\boldsymbol{x}_j^i \sim D^i}[f(\boldsymbol{x}_j^i)], \quad \boldsymbol{\Sigma}^i = \mathbb{E}_{\boldsymbol{x}_j^i \sim D^i}[(f(\boldsymbol{x}_j^i) - \boldsymbol{\mu}^i)^T (f(\boldsymbol{x}_j^i) - \boldsymbol{\mu}^i)], \tag{8}$$

where $f(\boldsymbol{x}_j^i)$ is the image feature extracted by frozen encoder. With these estimated distributions, the possibility of a test sample being drawn from each $\mathcal{N}^i$ can be calculated. Here we compute the logarithm of the probability density as a scoring function for input $\boldsymbol{x}$ on each learned task:

$$\begin{aligned} S^i &= \log \varphi(f(\boldsymbol{x}); \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i) \\ &= -\frac{1}{2}[(f(\boldsymbol{x}) - \boldsymbol{\mu}^i)^T (\boldsymbol{\Sigma}^i)^{-1} (f(\boldsymbol{x}) - \boldsymbol{\mu}^i) + d \log 2\pi + \log |\boldsymbol{\Sigma}^i|)], \end{aligned} \tag{9}$$

where φ is the probability density function.

Intuitively, a sample with a higher score S^i is more likely to be drawn from task i, and parameters K_r^i, V_r^i should be introduced for model prediction. Besides, we should also take into account that income sample $\boldsymbol{x}$ might come from some new distributions, which is suggested if all S^i are low. Thus we utilize the maximum score $\hat{S} = \max_{i \in [1,N]} S^i$ to weight the residual attention output:

$$O = O_L + \mathcal{M}(\hat{S}) O_r, \tag{10}$$

where $\mathcal{M}$ is a mapping function that scales the score $\hat{S}$ to the range $[0, 1]$. Here we find a simple Sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ works well here. We also conduct experiments in Sec. 5.2 to demonstrate the rationality and correctness of the calibration technique on IKI outputs.

Empowered by this distribution-aware integration calibration mechanism, the pre-trained zero-shot ability of VLMs can be retained better by assign lower weight to unfamiliar images, further resolving the forward forgetting issue.

5 Experiments

Benchmarks. To demonstrate the effectiveness of DIKI under the domain-class incremental learning setting, we conduct experiments on the recently proposed MTIL [91] benchmark. MTIL consists of 11 diverse datasets: Aircraft [47], Caltech101 [16], CIFAR100 [38], DTD [7], EuroSAT [25], Flowers [49], Food [3], MNIST [10], OxfordPet [50], StanfordCars [37], and SUN397 [82]. It's a very challenging benchmark with total of 1201 classes and severe data distribution shift across different tasks, which is infeasible for vanilla image encoder-based methods. Thus, vision-language models are necessarily included. The Order-I in original paper is applied. We also introduce the modified MTIL-FS benchmark for few-shot setting evaluation, in which only 16 samples per class of each dataset are used for training to simulate the data deficient scenario. More details can be found in the supplementary materials.

Evaluation metrics. To evaluate both backward and forward forgetting issues mentioned in Sec. 1, we adopt *Transfer*, *Avg.* and *Last* metrics from [91]. *Last* score is the model performance after all continual training, representing the degree of backward forgetting and being widely used in conventional continual learning. For forward forgetting issues, i.e. the loss of zero-shot ability, we evaluate model average accuracy on task $i+1, i+2, ..., N$ after its training on task i, denoted by *Transfer*. Lastly, *Avg.* is the average accuracy across all time steps. Detailed formulations can be found in the supplementary materials.

Table 1: *Transfer*, *Avg.*, and *Last* scores (%) of different continue learning methods on MTIL benchmark. 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.	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food	MNIST	OxfordPet	Cars	SUN397	Average
Zero-shot			24.8	92.9	68.4	43.8	47.7	71.4	85.8	59.5	89.1	65.8	62.6	64.7
Upper Bound			62.0	96.2	89.6	79.5	98.9	97.5	92.7	99.6	94.7	89.6	81.8	89.3
Transfer														
LwF [44]	✓	211 M		74.5	56.9	39.1	**51.1**	52.6	72.8	60.6	75.1	30.3	55.9	56.9
iCaRL [59]	✓	211 M		56.6	44.6	32.7	39.3	46.6	68.0	46.0	77.4	31.9	60.5	50.4
LwF-VR [12]	✓	211 M		77.1	61.0	40.5	45.3	54.4	74.6	47.9	76.7	36.3	58.6	57.2
WiSE-FT [81]	✓	211 M		73.5	55.6	35.6	41.5	47.0	68.3	53.9	69.3	26.8	51.9	52.3
ZSCL* [91]	✓	211 M		78.3	64.0	42.9	45.2	63.5	84.2	56.1	78.9	44.1	64.3	62.2
ZSCL [91]	✓	211 M		86.0	67.4	**45.4**	50.4	**69.1**	**87.6**	61.8	86.8	60.1	**66.8**	68.1
L2P† [80]	×	0.5 M		65.6	50.9	30.4	41.4	49.3	71.8	36.3	77.5	55.3	53.4	53.2
DualPmt.†[79]	×	1.8 M		56.7	51.4	28.7	33.7	45.6	70.9	59.5	77.7	49.5	50.4	52.4
S-Prompts [78]	×	0.5 M		67.3	49.4	26.4	39.7	47.1	70.2	34.3	78.9	56.7	52.2	52.2
DIKI	×	1.8 M		**92.9**	**69.0**	43.2	48.2	67.4	85.2	**63.0**	**87.9**	**63.8**	66.2	**68.7**
Avg.														
LwF [44]	✓	211 M	36.3	86.9	72.0	59.0	73.7	60.0	73.6	74.8	80.0	37.3	58.1	64.7
iCaRL [59]	✓	211 M	35.5	89.2	72.2	60.6	68.8	70.0	78.2	62.3	81.8	41.2	62.5	65.7
LwF-VR [12]	✓	211 M	29.6	87.7	74.4	59.5	72.4	63.6	77.0	66.7	81.2	43.7	60.7	65.1
WiSE-FT [81]	✓	211 M	26.7	86.5	64.3	57.1	65.7	58.7	71.1	70.5	75.8	36.9	54.6	60.7
ZSCL* [91]	✓	211 M	**50.7**	90.9	79.8	63.8	76.6	77.3	87.0	71.9	83.0	52.0	65.9	72.6
ZSCL [91]	✓	211 M	45.1	92.0	80.1	64.3	79.5	81.6	**89.6**	75.2	88.9	64.7	**68.0**	75.4
L2P† [80]	×	0.5 M	38.0	85.2	78.2	61.3	72.9	74.9	79.7	59.1	82.0	59.7	55.4	67.9
DualPmt.†[79]	×	1.8 M	37.8	84.3	78.6	60.1	71.1	73.2	79.1	73.9	82.3	55.1	52.8	68.0
S-Prompts [78]	×	0.5 M	37.5	92.5	77.5	58.2	76.4	74.1	78.8	57.9	83.0	60.8	54.4	68.3
DIKI	×	1.8 M	45.1	**95.5**	**83.1**	**64.8**	**79.9**	**83.5**	87.0	**76.2**	**89.6**	**67.0**	67.1	**76.3**
Last														
LwF [44]	✓	211 M	26.3	87.5	71.9	66.6	79.9	66.9	83.8	**99.6**	92.1	66.1	80.4	74.6
iCaRL [59]	✓	211 M	35.8	93.0	77.0	70.2	83.3	88.5	90.4	86.7	93.2	81.2	**81.9**	80.1
LwF-VR [12]	✓	211 M	20.5	89.8	72.3	67.6	85.5	73.8	85.7	**99.6**	93.1	73.3	80.9	76.6
WiSE-FT [81]	✓	211 M	27.2	90.8	68.0	68.9	86.9	74.0	87.6	**99.6**	92.6	77.8	81.3	77.7
ZSCL* [91]	✓	211 M	**46.0**	92.3	81.2	72.4	93.0	92.1	90.8	**99.6**	93.3	**86.6**	81.7	84.5
ZSCL [91]	✓	211 M	40.6	92.2	81.3	70.5	94.8	90.5	**91.9**	98.7	93.9	85.3	80.2	83.6
L2P† [80]	×	0.5 M	38.0	87.1	84.2	72.9	86.0	96.1	89.2	99.0	94.1	79.6	76.0	82.0
DualPmt.†[79]	×	1.8 M	37.8	87.1	84.6	71.8	89.2	96.3	89.1	99.1	**94.5**	79.9	76.5	82.3
S-Prompts [78]	×	0.5 M	37.5	95.1	83.7	70.2	97.5	96.5	89.0	99.1	94.0	79.5	75.8	83.4
DIKI	×	1.8 M	45.2	**95.7**	**86.3**	**72.9**	**98.0**	**97.0**	89.2	99.4	94.2	81.6	76.6	**85.1**

Comparison methods. We compare our DIKI against both full-parameter fine-tuning and parameter-efficient fine-tuning methods. For full fine-tuning, we choose ZSCL, ZSCL* [91], LwF [44], iCaRL [59], LwF-VR [12], and WiSE-FT [81] following [91]. For parameter efficient ones, L2P [80], DualPrompt [79], and S-Prompts [78] are selected for the similar task-specific parameter training procedure to our DIKI. Note that original L2P and DualPrompt are designed for ViT [13], we reproduce them on CLIP for fair comparisons. More reproduction details can be found in the supplementary materials.

Implementation details. We adopt CLIP ViT-B/16 [57] as our vision-language model for fair comparisons. Cross entropy loss and SGD optimizer with cosine learning rate scheduler is applied for all experiments, and the learning rate and batch size are set to 5 and 128, separately. Models are trained with 10 epochs

Table 2: Transfer, Avg., and Last scores (%) of different continual learning methods on 16-shot MTIL-FS benchmark. Full results can be found in the supplementary materials. Our DIKI can achieve more improvement when data is insufficient due to its non-interfered knowledge implantation scheme. † is equivalent to Tab. 1.

	Trans.	Avg.	Last	Average
Zero-shot	70.1	-	-	-
ZSCL [91]	68.3	69.3	74.0	70.5
L2P† [80]	53.9	62.3	73.3	63.2
DualPrompt† [79]	57.9	64.3	74.7	65.6
S-Prompts [78]	55.5	63.2	73.8	64.2
DIKI	**70.3**	**71.9**	**77.1**	**73.1**

Table 3: Ablation study of DIKI's components on MTIL benchmark. Our proposed modules form an integrated whole: zero-initialization only works with our residual attention design, and the calibration technique is designed on top of the residual branch. Note that our zero-initialization and calibration techniques only affect zero-shot ability, i.e. *Transfer* metric.

Prompt	ResAttn	Z-init	Calib.	Transfer	Last
✓				57.7	84.1
✓		✓		57.3	84.0
	✓			59.9	85.2
	✓	✓		63.1	85.0
	✓	✓	✓	68.7	85.1

on each task. For trainable parameters K_r and V_r, we set both the length l and training layer depth to 8 as discussed in the supplementary materials. To avoid floating point arithmetic precision problems, a small number 10^{-7} is added to diagonal elements of covariance matrix Σ^i with minor influence on final accuracy. All experiments are conducted on one NVIDIA 3090 GPU.

5.1 Main Results

Tab. 1 contains the *Transfer*, *Avg.* and *Last* scores among all methods on MTIL benchmark. "Extra data" includes memory buffers and reference datasets which used in distillation [91], and "# Param." is the number of trainable parameters. "Zero-shot" results are simply derived from leveraging the original CLIP weight on each task and perform as a comparison reference for *Transfer* metric. "Upper Bound" is calculated by applying full parameter fine-tuning technique on each separate dataset, as a guide for *Last* score.

As indicated by the bold values, our DIKI outperforms the previous state-of-the-art method [91] across all three metrics with only 0.86% trainable parameters, while alleviating the requirement for any external data. Thanks to the task-specific parameter training technique, we can memorize previous tasks' knowledge without rehearsal buffers and parameter ensemble, maintaining a high *Last* score with low computational complexity. Moreover, compared with task-specific prompt tuning methods (L2P, DualPrompt, and S-Prompts), we achieve significant improvement on *Transfer* metric, which shows that our DIKI mechanism can effectively inject new information to the frozen backbone without interfering with pre-trained knowledge.

We also conduct experiments on the 16-shot MTIL-FS benchmark. Abbreviated results are shown in Tab. 2 and the full table can be found in the supplementary materials. Since we only update a small amount of parameters, we gain more improvement over ZSCL compared to full parameter training. In addition, with minimal noise introduced, our fully residual IKI design demonstrates

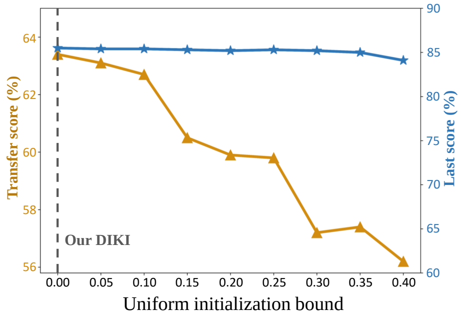


Fig. 3: Transfer and Last scores (%) with different uniform initialization bounds for residual attention parameters on MTIL benchmark. A larger initialization value will not affect the final accuracy (Last score), but could have a severe adverse impact on the model's zero-shot ability, due to the random noise introduced into the pre-trained model.

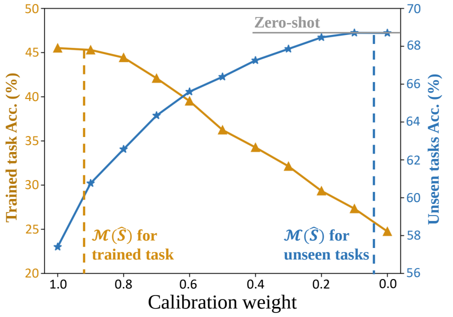


Fig. 4: Demonstration of the effect of our distribution-aware integration calibration. We evaluate the model, which is only trained on the first task of MTIL, on the trained task and unseen tasks, with **manually assigned** calibration weights. Fixed larger weights maintain high accuracy on trained task while lose zero-shot ability, and vice versa. Our DIKI tailors weight for different samples during inference time.

enhanced competitiveness when training data is deficient, compared to other interruptive prompt tuning methods.

5.2 Analysis

Ablation study. We ablate our proposed modules of DIKI on MTIL benchmark in Tab. 3. Firstly we consider *Transfer* score (i.e. zero-shot ability): from the first two rows, it can be seen that the zero-initialization mechanism is ineffective to prompt tuning methods, because they can still disturb the pre-trained knowledge by softmax function inside the attention calculation. However, with our residual attention design, the effect of zero-initialization is activated. They can work together to avoid introducing irrelevant information to the frozen backbone. Thanks to the fully residual property, distribution-aware calibration can be exploited to further boost performance by identifying unseen distributions.

Considering the *Last* metric, our interference-free mechanism stores more task knowledge because of its clear information injection process, thus enhancing the last state accuracy. However since our zero-initialization and distribution-aware calibration are designed to improve the retention of pre-trained knowledge, the addition of them does not result in an increase on *Last* score.

Effect of zero-initialized residual attention. To demonstrate the effect of our zero-initialization paradigm, we conduct experiments for different distributed initialization strategies on the MTIL benchmark, as shown in Fig. 3. Following previous common practice [94, 95], we choose uniform distribution with different bounds to initialize our trainable K_r and V_r in Eq. (6). Results show that with

Table 4: Results of CIL task on the 10-split CIFAR-100 dataset. We replace the prepending way in previous prompt-based CIL methods with our IKI strategy.

Method	Avg. Acc (↑)	Forgetting (↓)
L2P [80]	83.86±0.28	7.35±0.38
+ IKI	84.61±0.20	7.28±0.31
DualPrompt [79]	86.51±0.33	5.16±0.09
+ IKI	**88.77±0.25**	4.38±0.13
CODA-P [66]	86.25±0.74	5.02±0.41
+ IKI	87.17±0.35	**3.95±0.11**

Table 5: Training costs comparisons. "GPU Mem." denotes the training requirement, and "# Ref img" is the number of extra images used in the training stage except the continual training set. **-** means no extra data needed. We achieve higher performance with lower training costs.

Method	# Param.	Time	GPU Mem.	# Ref img
ZSCL [80]	211 M	11.3 h	96 GB	100k
DIKI	1.8 M	2.3 h	24 GB	**-**

different initialization values, the model can achieve constant final performance after being trained on all tasks (*Last* score keeps invariant). However, as the initialization bound increases, model's zero-shot ability degenerates due to the noise introduced by random initialization (*Transfer* score is decreasing).

Effect of distribution aware calibration. To demonstrate our calibration technique, we conduct experiments with manually set calibration weights. Specifically, we train the model exclusively on the first task of MTIL (Aircraft [47] dataset) and test it on all tasks, including trained and unseen datasets. Here we replace $\mathcal{M}(\hat{S})$ in Eq. (10) with fixed values, as shown in Fig. 4. When the weight is set to 1.0, which means full use of newly learned knowledge, the trained task accuracy is maximized while the vital zero-shot ability is interfered with. Conversely, as weight decreases, the zero-shot capability returns, while trained task accuracy decreases due to the reduced incorporation of new knowledge.

Our distribution-aware attention calibration tailors appropriate weights for different inference samples by the distribution modeling, allocating higher/lower weights to samples from learned/unseen domains. It alleviates the need to select a "balance point" which compromises overall performance.

Effect of IKI on CIL. To validate the universality of the proposed IKI, we evaluate it on the conventional Class Incremental Learning (CIL) task. Specifically, IKI is integrated into existing prompt-based CIL methods, serving as a replacement for their original prepending mechanisms. Experiments are conducted on the 10-split CIFAR-100 dataset following the common protocol [79,80], as shown in Tab. 4. IKI explicitly formulates a knowledge injection process, thus boosting the average accuracy by achieving superior performance on each task. For the forgetting metric, result of L2P [79] remains comparable due to the absence of shared information across tasks. Conversely, for methods with shared prompts (DualPrompt [80] and CODA-P [66]), our non-interference attention mechanism facilitates the knowledge shareability and alleviates the forgetting problem.

Training cost analysis. We compare the computational requirement of our DIKI and previous state-of-the-art method ZSCL [80] in Tab. 5. Benefiting from our parameter efficient framework, the training process of DIKI only lasts 2.3 hours on a single GPU, while ZSCL requires 4 GPUs, nearly half a day for training, and extra 100k images to perform distillation. With a much faster

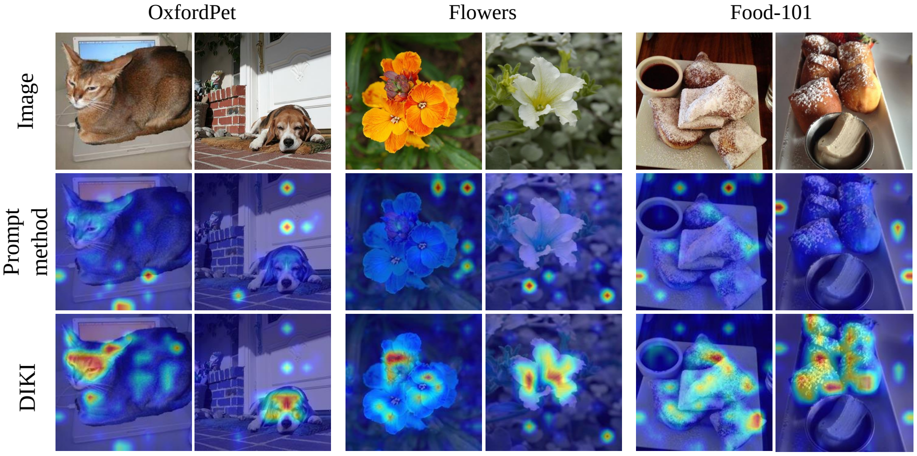


Fig. 5: Heatmap visualization comparisons. We employ Grad-CAM [62] to evaluate the model, which only has been trained on Aircraft [47], across unseen datasets OxfordPet [50], Flowers [49] and Food-101 [3]. It demonstrates that the commonly used prompt-based methods introduce noise into the model, thus resulting in forward forgetting issue and model degradation. Our DIKI implants new knowledge in a fully residual manner, optimizing the retention of pre-trained knowledge.

model adaptation speed, our method can be more effective and adoptable in tackling real-world continual learning problems.

Qualitative visualization results. We implement Grad-CAM [62] on the attention maps of the CLIP visual encoder, following the practice used in [63], as depicted in Fig. 5. Specifically, we load the model which is only trained on the first dataset Aircraft [47] of MTIL benchmark, and test it on several subsequent unseen datasets. We observe that the vanilla prompting way (employed by L2P, Dualprompt, and S-Prompts) interferes with pre-trained knowledge and undermines the zero-shot ability. However, with utilizing our DIKI, the generalization ability acquired during pre-training is preserved.

6 Conclusions

This study introduced Distribution-aware Interference-free Knowledge Integration (DIKI) mechanism for domain-class incremental learning. DIKI preserves the pre-trained knowledge of VLMs while effectively implanting new task information, without heavy computation and external data. DIKI infuses new knowledge into a frozen backbone in a fully residual manner, effectively mitigating the forward forgetting issue. A distribution-aware integration calibration technique is also integrated, which controls the information injection for data from unseen distributions. Experiments show that DIKI surpasses the previous SOTA method with only 0.86% trainable parameters.

References

1. Ahn, H., Cha, S., Lee, D., Moon, T.: Uncertainty-based continual learning with adaptive regularization. Advances in neural information processing systems **32** (2019)
2. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European conference on computer vision (ECCV). pp. 139–154 (2018)
3. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13. pp. 446–461. Springer (2014)
4. Bowman, B., Achille, A., Zancato, L., Trager, M., Perera, P., Paolini, G., Soatto, S.: a-la-carte prompt tuning (apt): Combining distinct data via composable prompting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14984–14993 (2023)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
6. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. Advances in Neural Information Processing Systems **35**, 16664–16678 (2022)
7. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3606–3613 (2014)
8. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE transactions on pattern analysis and machine intelligence **44**(7), 3366–3385 (2021)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
10. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine **29**(6), 141–142 (2012)
11. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
12. 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)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
14. Douillard, A., Ramé, A., Couairon, G., Cord, M.: Dytox: Transformers for continual learning with dynamic token expansion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9285–9295 (2022)
15. Fang, C., He, C., Xiao, F., Zhang, Y., Tang, L., Zhang, Y., Li, K., Li, X.: Real-world image dehazing with coherence-based label generator and cooperative unfolding network. arXiv preprint arXiv:2406.07966 (2024)

16. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 conference on computer vision and pattern recognition workshop. pp. 178–178. IEEE (2004)
17. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: Clip-adapter: Better vision-language models with feature adapters. International Journal of Computer Vision pp. 1–15 (2023)
18. He, C., Fang, C., Zhang, Y., Li, K., Tang, L., You, C., Xiao, F., Guo, Z., Li, X.: Reti-diff: Illumination degradation image restoration with retinex-based latent diffusion model. arXiv preprint arXiv:2311.11638 (2023)
19. He, C., Li, K., Zhang, Y., Tang, L., Zhang, Y., Guo, Z., Li, X.: Camouflaged object detection with feature decomposition and edge reconstruction. In: CVPR. pp. 22046–22055 (2023)
20. He, C., Li, K., Zhang, Y., Xu, G., Tang, L.: Weakly-supervised concealed object segmentation with sam-based pseudo labeling and multi-scale feature grouping. NeurIPS (2024)
21. He, C., Li, K., Zhang, Y., Zhang, Y., Guo, Z., Li, X.: Strategic preys make acute predators: Enhancing camouflaged object detectors by generating camouflaged objects. In: ICLR (2024)
22. He, C., Shen, Y., Fang, C., Xiao, F., Tang, L., Zhang, Y., Zuo, W., Guo, Z., Li, X.: Diffusion models in low-level vision: A survey. arXiv preprint arXiv:2406.11138 (2024)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
24. Hegde, D., Valanarasu, J.M.J., Patel, V.: Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2028–2038 (2023)
25. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **12**(7), 2217–2226 (2019)
26. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning. pp. 2790–2799. PMLR (2019)
27. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
28. Hu, Z., Lyu, J., Gao, D., Vasconcelos, N.: Pop: Prompt of prompts for continual learning. arXiv preprint arXiv:2306.08200 (2023)
29. Isele, D., Cosgun, A.: Selective experience replay for lifelong learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
30. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021)
31. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European Conference on Computer Vision. pp. 709–727. Springer (2022)

32. Jie, S., Deng, Z.H.: Fact: Factor-tuning for lightweight adaptation on vision transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1060–1068 (2023)
33. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: European Conference on Computer Vision. pp. 105–124. Springer (2022)
34. Khan, M.G.Z.A., Naeem, M.F., Van Gool, L., Stricker, D., Tombari, F., Afzal, M.Z.: Introducing language guidance in prompt-based continual learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11463–11473 (2023)
35. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19113–19122 (2023)
36. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
37. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013)
38. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
39. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9579–9589 (2024)
40. Lai, X., Tian, Z., Jiang, L., Liu, S., Zhao, H., Wang, L., Jia, J.: Semi-supervised semantic segmentation with directional context-aware consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1205–1214 (2021)
41. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning. pp. 12888–12900. PMLR (2022)
42. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
43. Li, X., Zhou, Y., Wu, T., Socher, R., Xiong, C.: Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In: International Conference on Machine Learning. pp. 3925–3934. PMLR (2019)
44. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
45. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9), 1–35 (2023)
46. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. Advances in neural information processing systems **30** (2017)
47. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
48. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7765–7773 (2018)

49. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian conference on computer vision, graphics & image processing. pp. 722–729. IEEE (2008)
50. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3498–3505. IEEE (2012)
51. Peng, B., Tian, Z., Wu, X., Wang, C., Liu, S., Su, J., Jia, J.: Hierarchical dense correlation distillation for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23641–23651 (2023)
52. Prabhu, A., Torr, P.H., Dokania, P.K.: Gdumb: A simple approach that questions our progress in continual learning. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 524–540. Springer (2020)
53. Pu, Y., Han, Y., Wang, Y., Feng, J., Deng, C., Huang, G.: Fine-grained recognition with learnable semantic data augmentation. IEEE Transactions on Image Processing (2024)
54. Pu, Y., Liang, W., Hao, Y., Yuan, Y., Yang, Y., Zhang, C., Hu, H., Huang, G.: Rank-detr for high quality object detection. Advances in Neural Information Processing Systems **36** (2024)
55. Pu, Y., Wang, Y., Xia, Z., Han, Y., Wang, Y., Gan, W., Wang, Z., Song, S., Huang, G.: Adaptive rotated convolution for rotated object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6589–6600 (2023)
56. Qian, Z., Wang, X., Duan, X., Qin, P., Li, Y., Zhu, W.: Decouple before interact: Multi-modal prompt learning for continual visual question answering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2953–2962 (2023)
57. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
58. Rao, D., Visin, F., Rusu, A., Pascanu, R., Teh, Y.W., Hadsell, R.: Continual unsupervised representation learning. Advances in neural information processing systems **32** (2019)
59. 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)
60. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., Wayne, G.: Experience replay for continual learning. Advances in Neural Information Processing Systems **32** (2019)
61. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
62. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
63. Shen, S., Li, L.H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.W., Yao, Z., Keutzer, K.: How much can clip benefit vision-and-language tasks? arXiv preprint arXiv:2107.06383 (2021)

64. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. Advances in neural information processing systems **30** (2017)
65. Smith, J.S., Cascante-Bonilla, P., Arbelle, A., Kim, D., Panda, R., Cox, D., Yang, D., Kira, Z., Feris, R., Karlinsky, L.: Construct-vl: Data-free continual structured vl concepts learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14994–15004 (2023)
66. 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)
67. Sohn, K., Chang, H., Lezama, J., Polania, L., Zhang, H., Hao, Y., Essa, I., Jiang, L.: Visual prompt tuning for generative transfer learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19840–19851 (2023)
68. Tang, L., Li, K., He, C., Zhang, Y., Li, X.: Consistency regularization for generalizable source-free domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4323–4333 (2023)
69. Tang, L., Li, K., He, C., Zhang, Y., Li, X.: Source-free domain adaptive fundus image segmentation with class-balanced mean teacher. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 684–694. Springer (2023)
70. Tian, Z., Lai, X., Jiang, L., Liu, S., Shu, M., Zhao, H., Jia, J.: Generalized few-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11563–11572 (2022)
71. Tian, Z., Shu, M., Lyu, P., Li, R., Zhou, C., Shen, X., Jia, J.: Learning shape-aware embedding for scene text detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4234–4243 (2019)
72. Tian, Z., Zhao, H., Shu, M., Yang, Z., Li, R., Jia, J.: Prior guided feature enrichment network for few-shot segmentation. IEEE transactions on pattern analysis and machine intelligence **44**(2), 1050–1065 (2020)
73. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
74. Van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. arXiv preprint arXiv:1904.07734 (2019)
75. Wang, J., Ma, Y., Guo, J., Xiao, Y., Huang, G., Li, X.: Cove: Unleashing the diffusion feature correspondence for consistent video editing. arXiv preprint arXiv:2406.08850 (2024)
76. Wang, J., Pu, Y., Han, Y., Guo, J., Wang, Y., Li, X., Huang, G.: Gra: Detecting oriented objects through group-wise rotating and attention. arXiv preprint arXiv:2403.11127 (2024)
77. Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., ji, J., Cao, G., Jiang, D., Zhou, M.: K-adapter: Infusing knowledge into pre-trained models with adapters (2020)
78. 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)
79. 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)

80. 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)
81. 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)
82. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 3485–3492. IEEE (2010)
83. Yang, J., Ding, R., Brown, E., Qi, X., Xie, S.: V-irl: Grounding virtual intelligence in real life. arXiv preprint arXiv:2402.03310 (2024)
84. Yang, S., Tian, Z., Jiang, L., Jia, J.: Unified language-driven zero-shot domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23407–23415 (2024)
85. Yang, S., Wu, J., Liu, J., Li, X., Zhang, Q., Pan, M., Gan, Y., Chen, Z., Zhang, S.: Exploring sparse visual prompt for domain adaptive dense prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 16334–16342 (2024)
86. Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., Xu, C.: Filip: Fine-grained interactive language-image pre-training. arXiv preprint arXiv:2111.07783 (2021)
87. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv preprint arXiv:1708.01547 (2017)
88. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: Lit: Zero-shot transfer with locked-image text tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18123–18133 (2022)
89. Zhang, J., Zhang, J., Ghosh, S., Li, D., Tasci, S., Heck, L., Zhang, H., Kuo, C.C.J.: Class-incremental learning via deep model consolidation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1131–1140 (2020)
90. Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., Qiao, Y.: Llama-adapter: Efficient fine-tuning of language models with zero-init attention. arXiv preprint arXiv:2303.16199 (2023)
91. 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)
92. Zhou, D.W., Zhang, Y., Ning, J., Ye, H.J., Zhan, D.C., Liu, Z.: Learning without forgetting for vision-language models. arXiv preprint arXiv:2305.19270 (2023)
93. Zhou, H., Yang, R., Zhang, Y., Duan, H., Huang, Y., Hu, R., Li, X., Zheng, Y.: Unihead: unifying multi-perception for detection heads. IEEE Transactions on Neural Networks and Learning Systems (2024)
94. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16816–16825 (2022)
95. 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)