Multiple Queries with Multiple Keys: A Precise Prompt Matching Paradigm for Prompt-based Continual Learning

Dunwei Tu
tudunwei@smail.nju.edu.cn
National Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Huiyu Yi
211300010@smail.nju.edu.cn
National Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Yuchi Wang
yuchi.wang@smail.nju.edu.cn
National Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Baile Xu*
xubaile@nju.edu.cn
National Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Jian Zhao
jianzhao@nju.edu.cn
School of Electronic Science and Engineering, Nanjing University
Nanjing, China

Furao Shen
frshen@nju.edu.cn
National Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Abstract

Continual learning requires machine learning models to continuously acquire new knowledge in dynamic environments while avoiding the forgetting of previous knowledge. Prompt-based continual learning methods effectively address the issue of catastrophic forgetting through prompt expansion and selection. However, existing approaches often suffer from low accuracy in prompt selection, which can result in the model receiving biased knowledge and making biased predictions. To address this issue, we propose the Multiple Queries with Multiple Keys (MQMK) prompt matching paradigm for precise prompt selection. The goal of MQMK is to select the prompts whose training data distribution most closely matches that of the test sample. Specifically, Multiple Queries enable precise breadth search by introducing task-specific knowledge, while Multiple Keys perform deep search by representing the feature distribution of training samples at a fine-grained level. Each query is designed to perform local matching with a designated task to reduce interference across queries. Experiments show that MQMK enhances the prompt matching rate by over 30% in challenging scenarios and achieves state-of-the-art performance on three widely adopted continual learning benchmarks. The code is available at https://github.com/DunweiTu/MQMK.

CCS Concepts

• **Computing methodologies** → *Online learning settings*; **Supervised learning by classification**; **Transfer learning**; **Lifelong machine learning**.

*Baile Xu is the corresponding author.

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755195

Keywords

Continual Learning, Prompt Tuning, Prompt Selection

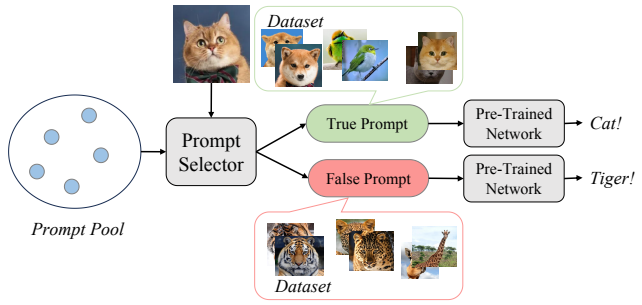
ACM Reference Format:
Dunwei Tu, Huiyu Yi, Yuchi Wang, Baile Xu, Jian Zhao, and Furao Shen. 2025. Multiple Queries with Multiple Keys: A Precise Prompt Matching Paradigm for Prompt-based Continual Learning. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3755195

1 Introduction

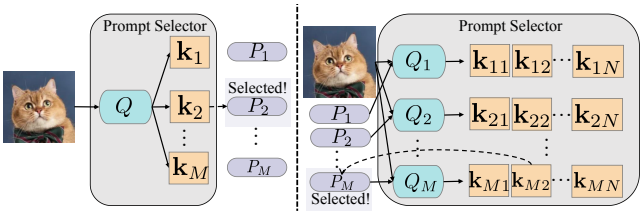
Neural networks have become one of the most important models in the field of machine learning [10, 19, 34]. In real-world scenarios, data may not always be fully accessible from the start, and the dynamic nature of the world continuously introduces new data. This requires neural networks to possess the ability for continual learning [1, 4, 49] to maintain knowledge updates and adapt to new environments. Additionally, humans do not forget old knowledge when learning new knowledge, making continual learning a key technology for building artificial intelligence that mirrors human learning.

In the context of continual learning, the data across different tasks is often non-independent and identically distributed (non-i.i.d.). Non-i.i.d. data requires the model to continuously adjust its parameters based on new data to establish decision boundaries more suited to the current data. However, this adaptation can lead to forgetting of previously learned knowledge. Some studies suggest that unless certain measures are taken, the neural network's ability to recognize previously encountered classes inevitably deteriorates during learning of new tasks, a phenomenon known as catastrophic forgetting [25, 26, 29].

Early continual learning methods design sustainable learning from aspects such as parameters [16, 22], data [33, 37], gradients [23, 28], features [24, 31], and architectures [41, 46], training a neural network from scratch, which has yielded promising results. Recent continual learning algorithms [27, 39, 45, 48, 49] have started



(a) The motivation for selecting the appropriate prompt.



(b) Left: Prior work. Right: Our work.

Figure 1: (a) Prompts inconsistent with test samples may introduce bias. (b) Prior prompt selector uses a single query and task-level keys; our method adopts task-level breadth queries and class-level depth keys, involving prompts in the selection process.

to focus on using pre-trained models to leverage their strong generalization ability for adapting to dynamic downstream tasks.

Prompt-based methods [7, 9, 13, 18, 38, 44, 45, 47], as emerging pre-trained model–based approaches, counter forgetting by cleverly reusing the generalization knowledge of pre-trained models. These methods utilize a frozen backbone network, such as Vision Transformer (ViT) [6] pre-trained on large-scale datasets like ImageNet [35], and adapt to continual learning tasks through Visual Prompt Tuning (VPT) [12] or Prefix Tuning [21]. A major advantage is that task-specific knowledge is stored in prompts, and as tasks increase, prompts are expanded to form a prompt pool covering all learned knowledge.

Existing methods [44, 45] use a query-key matching mechanism for prompt selection: a pre-trained ViT without prompts as the query function, and task-specific learnable parameters as the key. However, the matching is often imprecise. A prompt trained on a distribution inconsistent with test samples may fail to assist inference (fig. 1a). Results show that improving the matching rate effectively boosts model performance (see section 3.3).

To improve the matching rate and consistency between prompt and test samples, we propose the Multiple Queries with Multiple Keys (MQMK) paradigm. Unlike the current Single Query–Single Key (SQSK) paradigm [38, 44, 45], our method builds a query pool—each query corresponding to a task—for task-level breadth search (fig. 1b). Additionally, we extend task-level keys to Multiple Keys within each task for deep search. Experiments show performance is best when keys are at class level. Due to knowledge discrepancies among queries, we introduce a local matching mechanism to mitigate mutual interference. To address query overhead under resource constraints, we propose MQMK-Efficient Inference (MQMK-EI), which uses a single enhanced query to match multiple keys. By horizontally expanding the query-key matching mechanism, MQMK achieves up to 32.82% improvement in matching rate and delivers SOTA performance across three recognized continual learning datasets. Our results show that, beyond improving prompt quality, directly enhancing the matching mechanism can effectively improve performance.

The contributions of this paper are summarized as follows:

(1) We highlight the importance of query-key matching rate in prompt-based methods, define current approaches as SQSK, and introduce the MQMK paradigm, which achieves 32.82% improvement and SOTA performance.
(2) We identify that SQSK lacks task-specific knowledge in the query, leading to low precision, and that the prompt itself is not involved in prompt selection. To address this, we extend Single Query to Multiple Queries for task-level breadth search.
(3) After introducing Multiple Queries, class-level feature gaps become more pronounced. A single key cannot fully represent training distributions. Therefore, we extend Single Key to Multiple Keys for deep search and find class-level keys perform best. Finally, we propose a local matching mechanism to further enhance matching precision.

2 Related Work

Continual Learning. Early continual learning methods can generally be classified into five categories: regularization-based methods [16, 22], replay-based methods [33, 37], optimization-based methods [23, 28], representation-based methods [24, 31], and architecture-based methods [41, 46]. Regularization-based methods address the forgetting problem by adding explicit regularization terms to balance the old and new tasks. Replay-based methods design strategies to retain important old samples in order to preserve the model's previous knowledge. Optimization-based methods aim to make the gradients of new and old tasks as independent as possible, avoiding interference in order to prevent forgetting. Representation-based methods enhance the compatibility with new knowledge through meta-learning [8, 32] and self-supervised learning [3]. Architecture-based methods adapt to dynamic task objectives through dynamic networks.

Prompt-based Continual Learning. Prompt-based methods are a type of continuous learning method based on pre-trained models. They leverage task-specific knowledge from the prompt and the generalized knowledge from the pre-trained model, resulting in superior performance. L2P [45] and DualPrompt [44] introduce Visual Prompt Tuning (VPT) into continual learning, where prompt vectors help mitigate catastrophic forgetting by providing task-specific conditioning while keeping the backbone frozen. CODA-P [38] introduces a decomposed attention-based prompt querying method, where prompts are combined with varying weights. ESN [43] addresses stage interference and performance imbalance by using stage-isolated classifiers, energy normalization, and voting-based inference. EvoPrompt [18] addresses prompt selection mismatches and adaptive prompting challenges by using a dynamic, evolving prompt memory system that integrates reference and working prompts through optimal transport and bias adjustment.

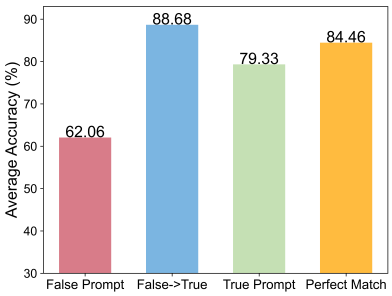


Figure 2: The average accuracy for four scenarios: when SQSK selects False Prompt and True Prompt, when samples initially with False Prompt are manually replaced with True Prompt, and when all samples use True Prompt (Perfect Match).

HiDe-Prompt [42] optimizes intra-task prediction, task identity inference, and task-adaptive prediction, using task-specific prompts and contrastive regularization to overcome sub-optimality in self-supervised pretraining. CPrompt [9] employs a random prompt selection training approach to address the issue of prompt inconsistency between training and testing phases.

3 Preliminaries

3.1 Problem Setting

In our problem setting, we feed the model with a sequence of dataset $\{\mathscr{D}_t\}_{t=1}^{T}$, where $\mathscr{D}_t$ is the dataset of the task t and T is the number of all tasks. $\mathscr{D}_t = \{(\mathbf{x}_i, y_i)\}$ contains pairs of the sample $\mathbf{x}_i$ and its corresponding label y_i. Each task has a distinct label space, meaning that $\mathscr{Y}^t \cap \mathscr{Y}^{t'} = \emptyset$ for any $t \neq t'$, where $\mathscr{Y}^t$ represents the label space of task t. Moreover, once the model transitions to the next task, it cannot access any of the previous datasets. When there are test samples for inference, the model needs to make predictions based on the label spaces of all the tasks it has encountered. We do not provide the task ID of the sample to the model, making this a more challenging class-incremental continual learning [2] setup.

3.2 Model Tuning and Prompt Pool

Following previous prompt-based continual learning works [9, 18, 38, 44, 45], we use ViT as the backbone $f = f_r \circ f_e$ and a single linear layer W as the classification head, where f_e is the embedding layer and f_r is the Multiple Self-Attention (MSA) layers. The MSA layers can be written as:

$$\mathrm{MSA}\left(\mathbf{h}_Q, \mathbf{h}_K, \mathbf{h}_V\right) = \mathrm{Concat}\left(\mathrm{h}_1, \ldots, \mathrm{h}_m\right) W^O \tag{1}$$

$$\text{where } \mathrm{h}_i = \mathrm{Attention}\left(\mathbf{h}_Q W_i^Q, \mathbf{h}_K W_i^K, \mathbf{h}_V W_i^V\right), \tag{2}$$

where $\mathbf{h}_Q$, $\mathbf{h}_K$, and $\mathbf{h}_V$ are the input query, key, and value, respectively. W^O, W_i^Q, W_i^K, and W_i^V are projection matrices. m is the number of heads, and i is the index of the layer. The function $\mathrm{Attention}(\cdot,\cdot,\cdot)$ denotes the self-attention mechanism. Given an image input $\mathbf{x}$, we first divide it into patches $\mathbf{x}_p \in \mathbb{R}^{L\times(S^2\times C)}$, where L is the token length, S is patch size and C is the number of channels. And then pass them through f_e to obtain corresponding embeddings $\mathbf{x}_e \in \mathbb{R}^{L\times D}$, where D is embedding (token) dimension (768). Next, the embeddings $\mathbf{x}_e$ and the prompt P are fed into f_r for feature extraction to obtain the output $f_r(P; \mathbf{x}_e) \in \mathbb{R}^{L\times D}$. Specifically, P is split into $P^K, P^V \in \mathbb{R}^{L_P/2\times D}$, which are prepended to $\mathbf{h}_K$ and $\mathbf{h}_V$, respectively which can written by $\mathrm{MSA}\left(\mathbf{h}_Q, [P^K; \mathbf{h}_K], [P^V; \mathbf{h}_V]\right)$. For simplicity, the layer indices of P, $\mathbf{h}_K$, and $\mathbf{h}_V$ are omitted. The first token of the output, $f_r(P; \mathbf{x}_e)[0] \in \mathbb{R}^D$, which serves as the [class] token, is passed through the classification head $W \in \mathbb{R}^{D\times|\mathscr{Y}^1\cup\mathscr{Y}^2\cup\cdots\cup\mathscr{Y}^T|}$ to obtain the prediction, where $|\mathscr{Y}^1 \cup \mathscr{Y}^2 \cup \cdots \cup \mathscr{Y}^T|$ represents the total number of classes across all tasks.

To address catastrophic forgetting, L2P and DualPrompt design a paradigm where a prompt pool is used to adapt to multiple tasks. Specifically, as tasks expand, the expert prompts (e-prompts) specific to each task and the general prompts (g-prompts) [44] shared across tasks together form the prompt pool $\mathbf{P} = \{P_g, P_1, P_2, \cdots, P_M\}$, where $P_g \in \mathbb{R}^{L_g\times D}$ is the g-prompt, $P_t \in \mathbb{R}^{L_e\times D}$ is the e-prompt of t-th task, M is the pool size, L_g is the length of g-prompt and L_e is the length of e-prompt. To retrieve prompts, they set a learnable parameter $\mathbf{k}_t \in \mathbb{R}^D$ for the t-th e-prompt and form a prompt pool consisting of key-value pairs, $\mathbf{P} = \{P_g, (\mathbf{k}_1, P_1), (\mathbf{k}_2, P_2), \cdots, (\mathbf{k}_M, P_M)\}$. The features extracted by the ViT without prompts, $f_r(\mathbf{x}_e)[0] \in \mathbb{R}^D$, are used as the query Q, and the prompt is selected by:

$$I = \underset{t\subseteq[1,M]}{\mathrm{argmax}} \cos\langle Q, \mathbf{k}_t\rangle, \tag{3}$$

where I represents the index of the selected prompt P_I and $\cos\langle\cdot,\cdot\rangle$ denotes cosine similarity function. Finally, the ViT, along with P_I and P_g, is used to classify the samples. Since each sample has only one Q and each P corresponds to one $\mathbf{k}$, we define this query paradigm as the Single Query-Single Key (SQSK) paradigm.

3.3 Prompt Matching Rate

We define the true prompt as a prompt that was trained on the task to which the sample belongs, i.e., $I = t$ for a sample $(\mathbf{x}_i, y_i) \in \mathscr{D}_t$, which means the sample's distribution aligns with the distribution of the prompt's training samples. Our experiments show that the matching rate under the SQSK paradigm is only 45.15% in 10-task Split ImageNet-R [11]. As shown in fig. 2, if the samples initially with False Prompt are manually replaced with True Prompt, the performance can be improved by 26.62% (from 62.06% to 88.68%) with the help of the task ID label. This indicates the significant impact that prompt selection has on performance. Interestingly, the accuracy of the samples initially with False Prompt, after being replaced with True Prompt, is even higher than that of the samples initially with the True Prompt. This phenomenon may be attributed to the fact that the tasks associated with samples selected by the False Prompt are more challenging to identify, whereas category prediction within those tasks is relatively easier. By replacing the False Prompt with the True Prompt, implicit task information is introduced, which leads to improved accuracy—even surpassing that of the samples originally selected by the True Prompt. Perfect matching refers to an idealized setting where all samples are matched with their corresponding True Prompt. This setting can be viewed as the performance upper bound achievable by prompt selection methods. For more detailed settings and explanations of this experiment, please refer to section 1 of supplementary materials.

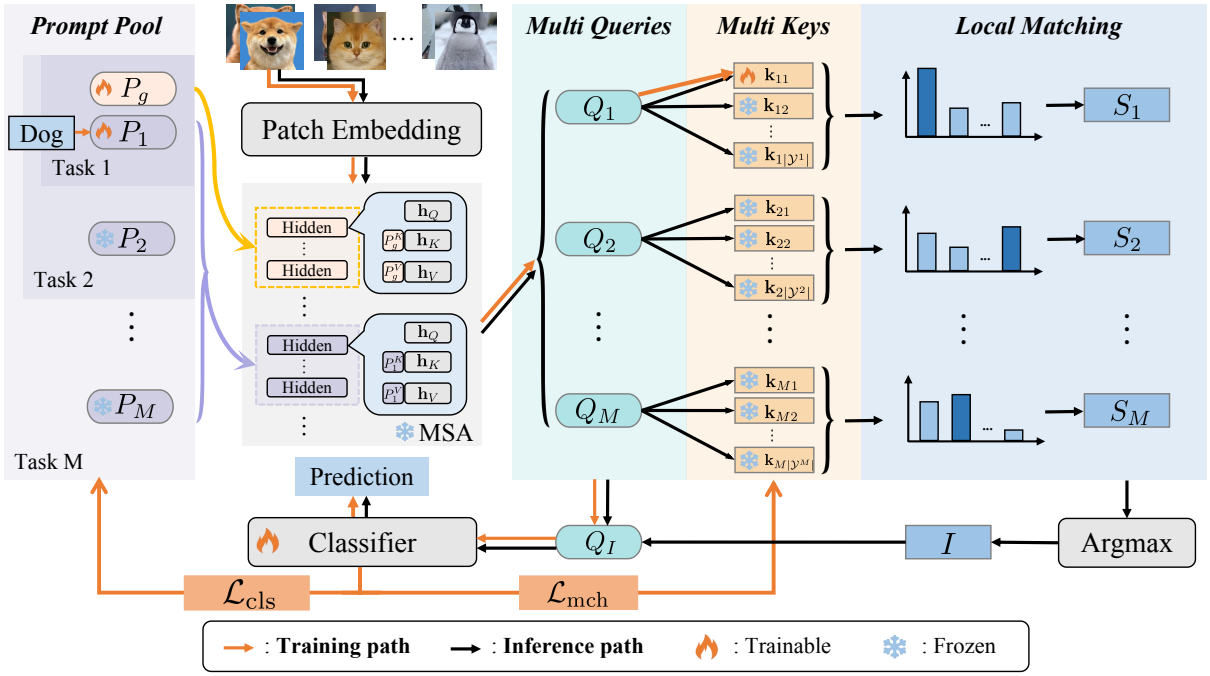


Figure 3: Overall pipeline of MQMK. Training: Select the true prompt based on the label information and input it into hidden states to obtain the corresponding query for classification. This process updates the prompt, the locally matched key and the classifier. Inference: Feed all prompts into hidden states to generate the query pool and select the most consistent query for classification by locally matching the task-level queries with the corresponding class-level keys.

4 Method

Our method uses the same model and tuning approach as SQSK, with the innovation of our method lying in the design of keys and queries, as well as their learning and matching mechanism. The overall pipeline of our framework is illustrated in fig. 3.

4.1 The Design Goal

We aim to ensure that the test sample follows to the distribution of the training samples used for prompt optimization, as this will enable the prompt to effectively handle the test sample. To achieve this goal, we need to address the following three questions:

(1) How can we extract sample features for similarity judgment?
(2) How can we represent the feature distribution of the samples used for prompt training?
(3) How can we effectively establish a matching mechanism between the features of test samples and the distribution of training samples?

For question 1, we employ a ViT with prompts as the query function. A ViT with prompts not only incorporates ViT's generalized knowledge but also task-specific fine-grained knowledge, enabling more precise feature extraction for queries. For question 2, we introduce class-level learnable parameters as keys. Since samples within a class tend to be similar, class-level keys can represent the feature distribution of a task's samples better. For question 3, to reduce the interference introduced by queries generated from different prompts, we employ a local matching strategy between each query and the task-specific keys. With queries, keys and the matching strategy in place, the model can evaluate the similarity between the query and keys, selecting the most consistent prompt for the sample to make the final prediction.

4.2 Local Matching Mechanism

We set $|\mathcal{Y}^t|$ different learnable keys $\mathbf{k}_t$ for P_t, where $|\mathcal{Y}^t|$ denotes the number of categories in task t. The set of keys $\mathbf{K}_t = \{\mathbf{k}_{t1}, \mathbf{k}_{t2}, \cdots, \mathbf{k}_{t|\mathcal{Y}^t|}\}$ collectively performs retrieval for P_t, where $\mathbf{k}_{tj}$ denotes the key for class j in task t. The keys and prompts together form a key-value pair pool: $\mathbf{P} = \{P_g, (\mathbf{K}_1, P_1), (\mathbf{K}_2, P_2), \cdots, (\mathbf{K}_M, P_M)\}$. We refer to these keys, where a task contains multiple keys for deep search, as MK. MK was originally introduced by CPrompt [9], but in fact, our MK differs fundamentally from the MK in CPrompt. This distinction will be explained in detail in section 4.3.

Given a sample $\mathbf{x}$, we use the t-th e-prompt and g-prompt in combination with ViT for feature extraction to obtain $f_r(P_g; P_t; \mathbf{x}_e)[0]$, which serves as the t-th query $Q_t \in \mathbb{R}^D$. The queries generated by using different e-prompts form a query pool $\mathbf{Q} = \{Q_1, Q_2, \cdots, Q_M\}$. We refer to task-level queries for breadth search as MQ. Next, Q_t is locally matched with $\mathbf{k}_{tj}$ by calculating the cosine similarity $\cos\langle Q_t, \mathbf{k}_{tj}\rangle$. Since different queries encode knowledge from different tasks, enforcing local matching between each query and its corresponding task allows for more effective utilization of task-specific information and helps mitigate cross-query interference. At this point, each prompt has matching scores for all the categories within the task. We need to select the top-K categories with the

highest score within a task for aggregation by:

$$S^t = \max_{\{c_m\}_{m=1}^K \subseteq [1,|\mathcal{Y}^t|]} \sum_{m=1}^{K} \cos\langle Q_t, \mathbf{k}_{tc_m} \rangle, \tag{4}$$

where S^t is the aggregated matching score between the sample and the t-th prompt. Finally, we select the highest matching score, and get the corresponding index as follows:

$$I = \underset{t \subseteq [1,M]}{\operatorname{argmax}}\, S^t. \tag{5}$$

Finally, the model can make predictions according to the selected query Q_I and the linear classifier.

4.3 Optimization Objective

The overall optimization objective consists of two components: (1) performing classification conditioned on the selected prompt, and (2) selecting prompts whose distribution aligns closely with that of the input sample. Based on the g-prompt and e-prompt, the model is optimized by minimizing the cross-entropy loss between the predicted output and the ground truth label by:

$$\mathcal{L}_{\text{cls}} = \text{CE}(W^T f_r(P_g; P_t; \mathbf{x}_e)[0], y), \tag{6}$$

where $\text{CE}(\cdot,\cdot)$ is cross-entropy function, and t is the task ID. During the training process, the task ID can be directly used to select the e-prompt.

Since the queries are already guided by the cross-entropy loss in the classification objective, the matching objective focuses solely on aligning the keys with the queries. Hence, the loss for query-key matching can be written as:

$$\mathcal{L}_{\text{mch}} = \sum_{i=1}^{M} \mathbb{I}(i=t) \sum_{j=1}^{|\mathcal{Y}^i|} (1 - \cos\langle Q_i, \mathbf{k}_{ij} \rangle) \mathbb{I}(j=y), \tag{7}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The overall loss function can be written as:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{mch}}. \tag{8}$$

Our MK learns by aligning with the MQ that has already been dispersed through cross-entropy. In contrast, the MK in CPrompt learns by dispersing itself through cross-entropy, since SQ is fixed and not dispersed. Our MK is a component specifically adapted for MQ.

As shown in eq. (7), keys corresponding to other categories and queries corresponding to other prompt are not involved in the loss calculation. This means that during training, only the designated query is needed, indicating that the MQ mechanism not only avoids introducing additional computational overhead, but actually requires one fewer query without a prompt compared to SQSK.

4.4 Efficient Inference of MQMK

MQ integrates knowledge from all tasks for precise querying but comes with a higher inference cost. We aim to design a method that approximates MQ's query accuracy while achieving SQ's query speed, seeking a trade-off between accuracy and speed. To ensure a speed comparable to SQ, we maintain a non-expanding single-query architecture. However, to incorporate knowledge from all tasks within the query, we need to enhance the query itself. Here, we design a simple yet effective knowledge fusion method to enhance the query by integrating fused knowledge of all tasks. Specifically, we first perform knowledge fusion by averaging all existing e-prompts in the prompt pool and then obtain the enhanced query using the fused prompt and the g-prompt by:

$$Q^+ = f_r(P_g; \bar{P}; \mathbf{x}_e)[0], \tag{9}$$

where $Q^+ \in \mathbb{R}^D$ is the enhanced query and $\bar{P} = \frac{1}{t}\sum_{i=1}^{t} P_i$ is the fused prompt. Finally, we replace Q^t in MQMK with Q^+ and select prompts following the same approach as in eq. (4). The above describes the Efficient Inference method of MQMK (MQMK-EI). MQMK-EI is an approximation of MQMK designed to accelerate inference. During training, MQMK is more accurate and more efficient. Therefore, MQMK-EI follows the same training process as MQMK, as described in section 4.3.

4.5 The Perspective of Prompt Ensemble

MQ can be regarded as a prompt ensemble [20, 36] query for multiple tasks in continual learning. Unlike prompt ensemble methods that learn different perspectives of knowledge within a single task, the prompt pool in continual learning stores distinct knowledge across tasks. In the context of continual learning, directly combining or voting over the logits from different prompts may lead to suboptimal performance. MQMK employs a multi-query mechanism in which each query is locally aligned with a specific task. This design helps reduce cross-task interference and knowledge entanglement, making it more suitable for continual learning scenarios characterized by large task discrepancies.

Similar to prompt ensemble methods, we do not need to forward M times through the ViT backbone to obtain the query pool. Instead, we replicate a sample M times, concatenate it with different prompts, and process them in a single batch for parallel inference. If computational resources are highly limited (e.g., the batch size is restricted to 1), MQMK-EI is more efficient. If computational resources are abundant, the computation time of MQMK, when parallelized, is reduced to half compared to the two-stage serial computations of MQMK-EI and SQSK. The MQMK series is training-efficient, achieving approximately a 1/3 speedup compared to the two-stage query process of SQ-based methods. This is a theoretical analysis, and the practical results, the parameters analysis and computational costs during the training phase, are discussed in section 3 of supplementary materials.

5 Experiments

5.1 Evaluation Benchmarks

Dataset. We shuffle the classes and perform task splitting on three widely used visual datasets for prompt-based continual learning: CIFAR-100 [17], ImageNet-R [11], and DomainNet [30], in order to align with the problem setup and conduct comprehensive experiments. We divide CIFAR-100 and DomainNet into 10 tasks, while ImageNet is divided into three cases: 5 tasks, 10 tasks, and 20 tasks. The detailed dataset introduction is in section 8 in supplementary materials. The data distributions of DomainNet and ImageNet-R differ significantly from the ImageNet dataset used for model pre-training. This requires the model to continually learn new knowledge from these datasets, rather than relying on the knowledge learned from the pre-training dataset.

Tasks	5		10		20	
Method	$A_T(\uparrow)$	$F_T(\downarrow)$	$A_T(\uparrow)$	$F_T(\downarrow)$	$A_T(\uparrow)$	$F_T(\downarrow)$
joint train	79.27	-	79.27	-	79.27	-
L2P++*	70.83 ± 0.58	3.36 ± 0.18	69.29 ± 0.73	2.03 ± 0.19	65.89 ± 1.30	1.24 ± 0.14
Deep L2P++*	73.93 ± 0.37	2.69 ± 0.10	71.66 ± 0.64	1.78 ± 0.16	68.42 ± 1.20	1.12 ± 0.13
DualPrompt*	73.05 ± 0.50	2.64 ± 0.17	71.32 ± 0.62	1.71 ± 0.24	67.87 ± 1.39	1.07 ± 0.14
ESN‡	73.42 ± 0.40	3.79 ± 0.55	75.11 ± 0.36	5.68 ± 0.77	70.57 ± 0.62	6.84 ± 0.36
CODA-P*	76.51 ± 0.38	2.99 ± 0.19	75.45 ± 0.56	1.64 ± 0.10	72.37 ± 1.19	**0.96 ± 0.15**
EvoPrompt†	77.16 ± 0.18	9.89 ± 0.30	76.83 ± 0.08	2.78 ± 0.06	74.41 ± 0.23	2.56 ± 0.22
CPrompt‡	78.42 ± 0.14	5.15 ± 0.58	77.14 ± 0.11	5.97 ± 0.68	74.79 ± 0.28	7.34 ± 0.65
MQMK	**79.61 ± 0.27**	1.59 ± 0.26	**78.36 ± 0.35**	2.24 ± 0.20	**76.10 ± 0.17**	3.33 ± 0.26
MQMK-EI	78.50 ± 0.21	**1.24 ± 0.18**	78.00 ± 0.12	**1.20 ± 0.25**	75.82 ± 0.18	1.87 ± 0.20

Table 1: Results (%) on Split ImageNet-R under 5-task, 10-task and 20-task settings. Best results are marked in bold. Second-best results are underlined. All our results are over 5 trials. ∗: Results from [38]. ‡: Results from [9]. †: Results from [18].

Method	$A_T(\uparrow)$	$F_T(\downarrow)$
joint train	91.79	-
L2P++*	82.50 ± 1.10	1.75 ± 0.42
Deep L2P++*	84.30 ± 1.03	**1.53 ± 0.40**
DualPrompt*	83.05 ± 1.16	1.72 ± 0.40
ESN‡	86.42± 0.80	6.08± 0.48
CODA-P*	86.25 ± 0.74	1.67 ± 0.26
EvoPrompt†	87.97 ± 0.30	2.60 ± 0.42
CPrompt‡	87.82 ± 0.21	5.06 ± 0.50
MQMK	91.73 ± 0.18	2.67 ± 0.17
MQMK-EI	**92.00 ± 0.29**	1.58 ± 0.18

Table 2: Results (%) on Split CIFAR-100 under 10-task setting. Best results are marked in bold. Second-best results are underlined. All our results are over 5 trials. ∗: Results from [38]. ‡: Results from [9]. †: Results from [18].

Method	$A_T(\uparrow)$	$F_T(\downarrow)$
joint train	89.15	-
L2P	81.17 ± 0.83	8.98 ± 1.25
DualPrompt	81.70 ± 0.78	8.04 ± 0.31
ESN	79.22 ± 2.04	10.62± 2.12
CODA-P	80.04 ± 0.79	10.16 ± 0.35
CPrompt	82.97 ± 0.34	7.45 ± 0.93
MQMK	**85.62 ± 0.33**	5.51 ± 0.22
MQMK-EI	84.54 ± 0.36	**3.33 ± 0.40**

Table 3: Results (%) on Split DomainNet under 10-task setting. Best results are marked in bold. Second-best results are underlined. All our results are over 5 trials. Except for MQMK, all the other results come from [9].

Evaluation Metrics. Average accuracy [14, 22, 46] and forgetting rate [2, 23] are the two core metrics we use. A_T is the average accuracy on tasks 1 to T after the model has learned task T, and can be computed by:

$$A_T = \frac{1}{T}\sum_{t=1}^{T} \text{Accuracy}(t, T), \tag{10}$$

where $\text{Accuracy}(t, T)$ represents the accuracy on task t after learning task T. F_T is the average accuracy drop across all tasks, and can be computed by:

$$F_T = \frac{1}{T}\sum_{t=1}^{T} \left(\text{Accuracy}\,(t, t) - \text{Accuracy}\,(t, T)\right). \tag{11}$$

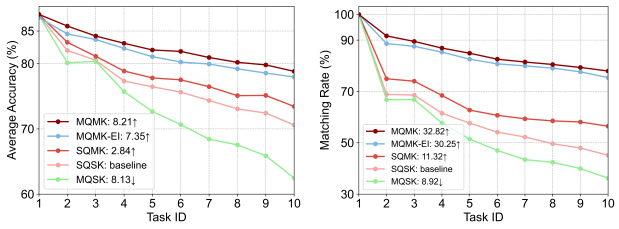
Average accuracy directly reflects the overall performance of the model, while forgetting rate indicates the trend of performance change. Therefore, average accuracy is relatively more important.

Implementation Details. ViT-B/16 [6] pre-trained on ImageNet-21K [5] and fine-tuned on ImageNet-1K [35] is the backbone we use, so S is set to 16. G-prompt is used in the first two layers of MSA, with a length of 5. The depth and length of e-prompt are discussed in section 5.5. In all experiments, M is set equal to T, meaning that only one prompt is learned for each task. Each prompt selects only the category with the highest similarity for aggregation which implies that K is set to 1. For both SQSK and MQMK, a learning rate of 0.005, a batch size of 64, and the Adam [15] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are used in all experiments.

Competitors and Joint Training. We compare our method with SOTA prompt-based continual learning models, including L2P [45], DualPrompt [44], ESN [43], CODA-P [38], EvoPrompt [18] and CPrompt [9]. To ensure a fair comparison, we use SQSK as a baseline in some experiments. SQSK and MQMK have identical prompts, with the only difference being the matching mechanism. SQSK can be seen as a version of DualPrompt with adjusted hyperparameters, and it has nearly the same performance as DualPrompt reported in tables 1 to 3. Additionally, we jointly train a model using data from all tasks with linear prototypes and prompt tuning as a reference for the upper bound.

5.2 Performance Comparison

As shown in table 1, in the three task settings on ImageNet-R, MQMK achieves SOTA performance. In the 5-task and 10-task splits, MQMK shows no significant performance gap compared to joint training. For the small task split, the continual learning performance with MQMK no longer shows a significant gap compared to traditional joint training with all data. This indicates that the



(a) Average Accuracy. **(b) Matching Rate.**

Figure 4: Ablation experiments on the 10-task Split ImageNet-R. The average accuracy and matching rate change with the learning process.

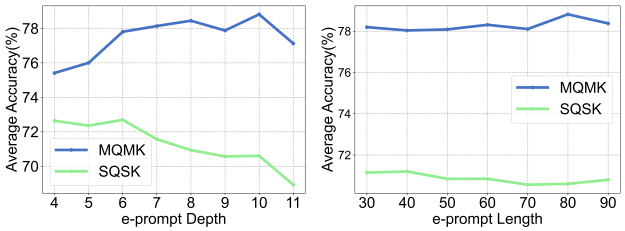
Number	A_T (↑)	Matching Rate (↑)
1	62.47	36.23
4	71.20	57.82
10	75.17	68.77
20	78.82	77.97

Table 4: A_T (%) and matching rate (%) under keys of different granularities on 10-task Split Imagenet-R.

performance on ImageNet-R with the 5-task and 10-task splits is relatively close to its oracle. However, on the 20-task split, MQMK still shows a noticeable performance gap compared to joint training. A possible explanation is that as the number of tasks increases, the limited data per task becomes inadequate for training high-quality e-prompts, ultimately resulting in a degradation of overall performance. Additionally, compared to the high-performing CPrompt and EvoPrompt, MQMK has a clear advantage in terms of forgetting rate. This indicates that MQMK not only achieves higher final performance, but also experiences smaller performance degradation. CODA-P has a low forgetting rate, yet its average accuracy is also relatively low. This suggests that CODA-P exhibits overall stability in performance, but its final performance is average.

On CIFAR-100, **MQMK also achieves performance close to that of joint training, improving by 3.91% compared to CPrompt**, while maintaining a low forgetting rate. On DomainNet, MQMK shows a 2.65% performance improvement over CPrompt, along with the lowest forgetting rate. Split-DomainNet is an class-imbalanced dataset, where MQMK achieves 90% accuracy on some tasks, while on tasks with small sample sizes, the accuracy drops to only 60%. The issues of small samples and class imbalance may be the reasons for the poor performance of all continual learning algorithms.

Overall, MQMK-EI slightly lags behind MQMK but still achieves SOTA performance. On CIFAR-100 and the 20-task split of ImageNet-R, MQMK-EI achieves performance that is very close to MQMK. This is because CIFAR-100 is a relatively simple dataset with consistent image styles, where knowledge fusion among different prompts introduces minimal loss or interference. In the 20-task split of ImageNet-R, the limited number of samples per task makes it difficult to train high-quality e-prompts. In such cases, knowledge fusion helps to enhance prompt quality to a certain extent.



(a) Different depth. **(b) Different length.**

Figure 5: Average accuracy of MQMK and SQSK on the 10-task Split ImageNet-R under different depths and lengths of the e-prompt. (a) The length is fixed at 80, and the depth varies. (b) The depth is fixed at 10, and the length varies.

5.3 Ablation Study

The components of our method include Multiple Queries (MQ) and Multiple Keys (MK). The corresponding components in existing methods are Single Query (SQ) and Single Key (SK). We discuss the average accuracy and matching rate under four combinations: MQMK, MQSK, SQMK, and SQSK, as well as the inference-efficient variant MQMK-EI, as illustrated in fig. 4. Using SQSK as the baseline and adding MK, A_T improves by 2.84% and the matching rate increases by 11.32%. This demonstrates the effectiveness of setting class-level keys for each prompt. By adding MQ, A_T decreases by 8.13% and the matching rate drops by 8.92%. This is because, under the supervision of cross-entropy, the features of samples from the same class are more clustered, while the features of samples from different classes are more dispersed. The features of multiple classes in a task have already been dispersed, and in this case, SK can no longer represent the feature distribution of all training samples in a task. **When MQ is combined with MK, A_T increases by 8.21% and the matching rate improves by 32.82%.** This demonstrates that incorporating task-related knowledge significantly enhances retrieval accuracy. In fact, in query-key matching, the improvement of query quality and key quality are complementary. Enhancing both simultaneously has a significant impact on matching rate. MQMK-EI achieves performance comparable to MQMK while incurring a lower query cost.

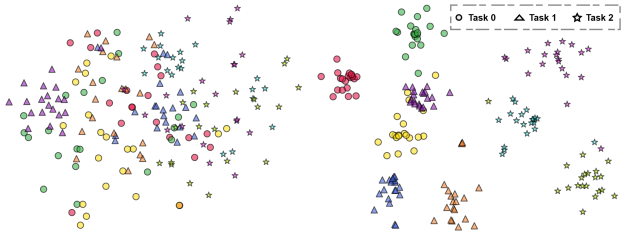
5.4 Setting Fine-grained Keys

MQSK leads to a performance decline, whereas MQMK improves performance. This highlights the critical importance of learning reliable keys for distribution representation. Based on MQ, we gradually increase the number of keys from 1 to the number of classes in each task (20 class per task in 10-task Split Imagenet-R), and we report their performance in table 4. We divide each task into several class groups based on the number of keys, with each group containing multiple classes. From this perspective, both SK and MK are special cases of N-keys.

It can be observed that **as the number of keys increases, both the matching rate and accuracy improve**, and the optimal performance is achieved when the key reaches the class level. Since the smallest granularity of our labels is at the class level, when the

Method	DomainNet	ImageNet-R	CIFAR-100
L2P	72.31	42.82	48.73
ESN	72.84	70.33	78.50
CPrompt	75.36	64.00	69.37
SQSK	70.55	45.15	49.14
MQMK	82.24	77.97	82.28
MQMK-EI	76.67	75.40	86.17

Table 5: Matching rate (%) of different methods on three 10-task split benchmarks.



(a) SQ. **(b) Selected Queries in MQ.**

Figure 6: t-SNE [40] visualization of the Queries. The samples come from three classes of three tasks in the 5-task Split Imagenet-R. The shapes represent the sample tasks, and the colors represent the sample categories.

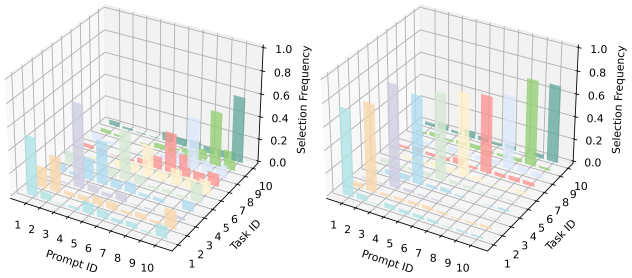
number of keys exceeds the number of classes, it becomes difficult to provide appropriate supervisory signals for the keys.

5.5 Going Deeper and Longer

As shown in fig. 5, the performance of the MQMK method improves as the e-prompt depth and length increase, reaching its optimal performance of 78.82% when the e-prompt depth is 10 and length is 80. In contrast, the performance of the SQSK method decreases as the prompts go deeper and longer. A possible reason is that increasing the depth and length of the prompt can improve its quality. The higher matching rate of MQMK allows it to effectively utilize the higher-quality prompt, thereby enhancing performance. In contrast, SQSK does not benefit in the same way. This suggests that **an accurate query paradigm can allow the prompt to go deeper and longer, thereby improving model performance**.

5.6 Visualization of Queries

We present visualizations of the queries selected by MQ, as well as queries in SQ, as shown in fig. 6. It can be observed that the queries selected by MQ exhibit stronger intra-class cohesion and inter-class separation, indicating that MQ queries are precise and the introduction of prompts is necessary. On the contrary, due to SQ's reliance solely on ViT's generalization ability and the lack of task-specific knowledge, the queries of SQ are highly dispersed, resulting in almost failed clustering. Additionally, although the queries show a tendency to cluster by task, the smallest granularity of clustering is at the class level, which indicates the necessity of setting class-level keys.



(a) SQSK. **(b) MQMK.**

Figure 7: The 3D visualization of specific prompt selection process of MQMK and SQSK on 10-task Split Imagenet-R.

6 Matching Rate Across Methods

As shown in table 5, MQMK improves the matching rate by over 30% compared to SQSK on the challenging CIFAR-100 and ImageNet-R. Due to the significant differences in data styles across different domains, SQSK can achieve high performance on DomainNet. To gain a deeper understanding of the prompt selection mechanism, we visualize the specific prompt selection process of MQMK and SQSK, generating two 3D bar charts, as shown in fig. 7. The results indicate that MQMK exhibits a significantly higher probability of selecting the true prompt compared to SQSK. MQMK outperforms MQMK-EI on datasets with large style discrepancies, such as ImageNet-R and DomainNet, whereas on more consistent datasets like CIFAR-100, MQMK-EI, with its knowledge fusion enhanced queries, achieves higher matching rates. Both experiments demonstrate the effectiveness of our approach in improving the matching rate.

CODA-P and EvoPrompt use soft, weighted prompt selection, where correctness does not exist. The matching objective of ESN is task-specific classifiers, which bears similarity to prompt selection. According to ablation studies, MQMK is identical to other variants such as SQSK in all aspects except for the prompt strategy. In vertical comparisons with other methods, MQMK still demonstrates a clear advantage in terms of matching rate.

7 Conclusion

To address the issue of low prompt selection accuracy in prompt-based continual learning, we propose the Multiple Queries with Multiple Keys prompt local matching paradigm. Multiple Queries achieve precise feature extraction by incorporating task-specific knowledge. Multiple Keys, through fine-grained learnable keys, better represent the feature distribution of the training samples. They complement each other, significantly improving the matching rate and performance through local matching mechanism. An accurate parameter selection strategy can bring significant performance improvements to existing continual learning methods based on parameter expansion and selection. Although precise querying incurs a certain query cost, MQMK-EI maintains high query accuracy while effectively reducing the query cost.

Acknowledgments

This work was partially supported by the STI 2030-Major Projects of China (Grant No. 2021ZD0201300), the National Natural Science Foundation of China (Grant Nos. 62276127, 62495094), and the Fundamental Research Funds for the Central Universities (Grant No. 2024300394). The authors gratefully acknowledge these supports.

References

[1] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. 2021. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks* 135 (2021), 38–54.

[2] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2018. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420* (2018).

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[4] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence* 44, 7 (2021), 3366–3385.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[6] Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[7] Yu Feng, Zhen Tian, Yifan Zhu, Zongfu Han, Haoran Luo, Guangwei Zhang, and Meina Song. 2024. CP-Prompt: Composition-Based Cross-modal Prompting for Domain-Incremental Continual Learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 2729–2738.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[9] Zhanxin Gao, Jun Cen, and Xiaobin Chang. 2024. Consistent Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 28463–28473.

[10] Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks* (2012), 37–45.

[11] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*. 8340–8349.

[12] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *European Conference on Computer Vision*. Springer, 709–727.

[13] Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. 2023. Generating instance-level prompts for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11847–11857.

[14] Dongwan Kim and Bohyung Han. 2023. On the stability-plasticity dilemma of class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20196–20204.

[15] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[17] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[18] Muhammad Rifki Kurniawan, Xiang Song, Zhiheng Ma, Yuhang He, Yihong Gong, Yang Qi, and Xing Wei. 2024. Evolving Parameterized Prompt Memory for Continual Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 13301–13309.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[20] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[21] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021).

[22] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.

[23] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017).

[24] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. 2021. Representational continuity for unsupervised continual learning. *arXiv preprint arXiv:2110.06976* (2021).

[25] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review* 102, 3 (1995), 419.

[26] Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Vol. 24. Elsevier, 109–165.

[27] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. 2024. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems* 36 (2024).

[28] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. 2020. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems* 33 (2020), 7308–7320.

[29] Cuong V Nguyen, Alessandro Achille, Michael Lam, Tal Hassner, Vijay Mahadevan, and Stefano Soatto. 2019. Toward understanding catastrophic forgetting in continual learning. *arXiv preprint arXiv:1908.01091* (2019).

[30] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1406–1415.

[31] Quang Pham, Chenghao Liu, and Steven Hoi. 2021. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems* 34 (2021), 16131–16144.

[32] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *International conference on learning representations*.

[33] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.

[34] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature* 323, 6088 (1986), 533–536.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115 (2015), 211–252.

[36] Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676* (2020).

[37] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems* 30 (2017).

[38] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. 2023. 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*. 11909–11919.

[39] Dunwei Tu, Huiyu Yi, Tieyi Zhang, Ruotong Li, Furao Shen, and Jian Zhao. 2025. Embedding Space Allocation with Angle-Norm Joint Classifiers for few-shot class-incremental learning. *Neural Networks* (2025), 107608.

[40] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[41] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. 2022. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*. Springer, 398–414.

[42] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. 2023. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *Advances in Neural Information Processing Systems* 36 (2023), 69054–69076.

[43] Yabin Wang, Zhiheng Ma, Zhiwu Huang, Yaowei Wang, Zhou Su, and Xiaopeng Hong. 2023. Isolation and impartial aggregation: A paradigm of incremental learning without interference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10209–10217.

[44] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. 2022. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*. Springer, 631–648.

[45] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 139–149.

[46] Shipeng Yan, Jiangwei Xie, and Xuming He. 2021. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3014–3023.

[47] Chengyi Yang, Wentao Liu, Shisong Chen, Jiayin Qi, and Aimin Zhou. 2024. Generating Prompts in Latent Space for Rehearsal-free Continual Learning. In

Proceedings of the 32nd ACM International Conference on Multimedia. 8913–8922.

[48] Huiyu Yi. 2024. Few-Shot Class-Incremental Learning with Class Centers and Contrastive Learning for Incremental Vehicle Recognition. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[49] Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. 2024. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386* (2024).