LEARNING TO PROMPT FOR CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The mainstream learning paradigm behind continual learning has been to adapt the model parameters to non-stationary data distributions, where catastrophic forgetting is the central challenge. This work explores a new paradigm for continual learning – learning to dynamically *prompt* the model to learn tasks sequentially under different task transitions. Specifically, our method, Learning to Prompt for Continual Learning (L2P), prepends a subset of learnable parameters (called *Prompts*) from a larger set (called *Prompt Pool*) to the input embeddings. The training objective is designed to dynamically select and update prompts from the prompt pool to learn tasks sequentially given a pretrained backbone model. Under our new framework, instead of mitigating catastrophic forgetting via adapting large model parameters as in the previous continual learning paradigm, we tackle the problem of learning better small prompt parameters. In this framework, the prompt pool explicitly manages task-invariant and task-specific knowledge while maintaining model plasticity. The proposed L2P outperforms previous work in terms of forgetting on all datasets, including rehearsal-based methods on certain benchmarks, with privacy benefits from not requiring access to the data of previous tasks. Moreover, when L2P is additionally equipped with a rehearsal buffer, it matches the performance of training all tasks together, which is often regarded as an upper bound in continual learning. Source code will be released.

1 INTRODUCTION

Contrary to ordinary supervised learning that trains on independent and identically distributed (i.i.d.) data, continual learning tackles the problem of training a single model on non-stationary data distributions where different classification tasks are presented sequentially. Mainstream continual learning methods (Parisi et al., 2019; Mai et al., 2021) follow a natural learning paradigm: adapting the entire model continually as the data distribution shifts. However, since the model only has access to the data in an individual phase of the learning cycle, it is prone to overfit on the currently available data and suffers from performance deterioration on the previously trained data. This is commonly known as *catastrophic forgetting* (McCloskey & Cohen, 1989).

In addition to the catastrophic forgetting problem, other challenges in continual learning have recently been receiving increasing attention (Hadsell et al., 2020): (1) *knowledge transfer*: the model should be able to transfer knowledge between tasks by identifying shared knowledge among tasks; (2) *model plasticity*: the model should be able to keep learning new tasks effectively by capturing task-specific knowledge; and (3) *task-agnosticity*: it is desirable that a continual learning algorithm can handle the case where distribution shifts gradually without clear task boundaries.

On the other hand, prompt-based learning, or prompting, has recently achieved great success in the field of natural language processing (NLP) as a new transfer learning technique (Liu et al., 2021). Prompting techniques design model inputs with textual *prompt* tokens containing additional task-specific information, such that the pretrained language model can process parameterized inputs in order to perform prompt-specific prediction. Several methods (Lester et al., 2021; Shin et al., 2020; Li & Liang, 2021) further make prompts learnable to allow the overall backbone model to extract task-specific information automatically. Intuitively, prompt-based learning reformulates learning downstream tasks from directly adapting model weights to designing prompts that enable the model perform tasks conditionally. A prompt encodes task-specific knowledge and has the ability to utilize pre-trained frozen models more effectively than ordinary fine-tuning (Lester et al., 2021; Raffel et al., 2020). Inspired by these recent advances in prompt learning, we revisit continual learning

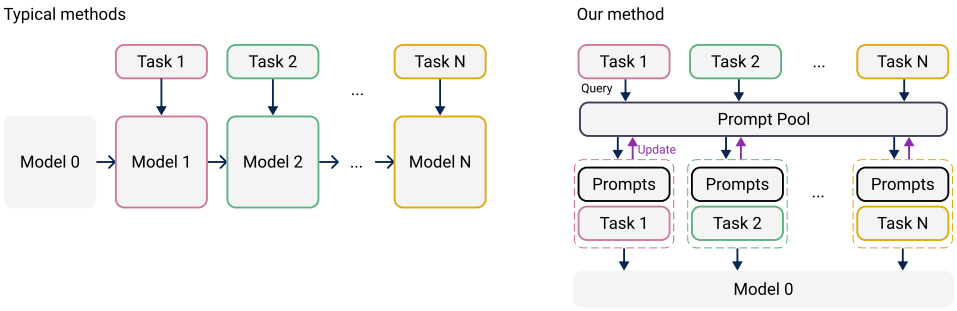

Figure 1: Overview of the L2P framework. Compared with typical continual learning methods (left) that adapt model weights to tasks sequentially, L2P (right) uses a single backbone model and learns a prompt pool to adapt tasks.

from a different perspective: Can we encode task-specific information of continual tasks into a shared parameterized prompt space in order to allow a pre-trained model to perform conditional prediction during the continual learning process?

To this end, we propose a new continual learning method called Learning to Prompt for Continual Learning (L2P). Figure 1 gives an overview of our method and demonstrates how it differs from typical continual learning methods. L2P leverages the representative features from pretrained models; however, instead of tuning the parameters during the continual learning process, L2P keeps the pretrained model untouched, and instead learns a set of prompts that dynamically help models solve corresponding tasks, thus mitigating catastrophic forgetting. The prompts are structured in a key-value shared memory space called the prompt pool, and we design a query mechanism to dynamically lookup a subset of task-relevant prompts based on the instance-wise input features. The prompt pool, which is optimized jointly with the supervised loss, ensures that shared prompts encode shared knowledge for knowledge transfer, and unshared prompts encode task-specific knowledge that help maintain model plasticity. The instance-wise query mechanism removes the necessity of knowing the task identity or boundaries, enabling *task-agnostic* continual learning. The selected prompts are then prepended to the input embeddings (Figure 2), which implicitly add task-relevant guidance to pretrained models, so that the model can use the most useful pretrained features to conduct corresponding tasks. In summary, this work makes the following contributions:

1. We propose a novel method, called L2P, that addresses multiple challenges in continual learning: (1) we leverage pretrained models and prompting techniques to mitigate catastrophic forgetting; (2) we design a novel key-value paired prompt pool to achieve knowledge sharing and maintain model plasticity; and (3) we devise an instance-wise query mechanism to enable task-agnostic learning.
2. We conduct comprehensive experiments to demonstrate the effectiveness of L2P on multiple continual learning benchmarks, including class-incremental, task-agnostic, and domain-incremental settings. The proposed L2P outperforms previous works in terms of forgetting on all datasets, beating rehearsal based methods on certain benchmarks and providing practical advantages over them by avoiding privacy issues of task data sharing present in some applications (Delange et al., 2021). Moreover, when equipped with a rehearsal buffer in applications with less strict privacy constraints, L2P matches the performance of training all tasks together, which is often regarded as an upper bound in continual learning.
3. To the best of our knowledge, we are the first to introduce the idea of prompting in the field of continual learning to address some of the key challenges in continual learning.

2 Related Work

Continual learning. There are three main categories of recent continual learning algorithms: *Regularization-based* methods (Kirkpatrick et al., 2017; Zenke et al., 2017; Li & Hoiem, 2017; Aljundi et al., 2018) limit the plasticity of the model by limiting the learning rate on important parameters for previous tasks. Although these methods address catastrophic forgetting to some extent, they cannot get satisfactory performance under more challenging settings, *e.g.*, class-incremental

setting (Mai et al., 2021). *Rehearsal-based* methods (Chaudhry et al., 2018; 2019; Hayes et al., 2019) construct a buffer to save samples from older tasks to train with data from the current task. These methods are state-of-the-art on various benchmarks (Parisi et al., 2019; Mai et al., 2021). However, rehearsal-based methods are not applicable to scenarios where data privacy should be taken into account (Shokri & Shmatikov, 2015). *Architecture-based* methods either expand the network (Rusu et al., 2016; Yoon et al., 2017) or prune the network (Mallya & Lazebnik, 2018; Wang et al., 2020). The former suffers from scalability issue as parameters scale up linearly with the number of tasks, and the latter are sensitive to hyperparameters.

Prompting. Prompting, or prompt-based learning, has been widely explored in the field of natural language processing (Kumar et al., 2016; McCann et al., 2018; Radford et al., 2019; Schick & Schütze, 2020). The high-level idea of prompting is to apply a function to modify the input text, so that the language model gets additional information about the task. However, the design of a prompting function is challenging and requires heuristics. Recent work, including prompt tuning (Lester et al., 2021) and prefix tuning (Li & Liang, 2021), seek to address this problem by applying learnable prompts in a continuous space, achieving satisfactory performance on transfer learning for pretrained language models. Nevertheless, to the best of our knowledge, the idea of prompting has never been studied systematically in continual learning.

3 PREREQUISITES

3.1 CONTINUAL LEARNING PROTOCOLS

Continual learning is usually defined as training machine learning models on non-stationary data from sequential tasks. We define a sequence of tasks $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$, where the t-th task $\mathcal{D}_t = \{(\boldsymbol{x}_i^t, y_i^t)\}_{i=1}^{n_t}$ contains tuples of the input sample $\boldsymbol{x}_i^t \in \mathcal{X}$ and its corresponding label $y_i^t \in \mathcal{Y}$. The goal is to train a single model $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by θ, such that it predicts the label $y = f_\theta(\boldsymbol{x}) \in \mathcal{Y}$ given an unseen test sample $\boldsymbol{x}$ from arbitrary tasks. Data from the previous tasks may not be seen anymore when training future tasks.

Depending on the task transition environment, continual learning can be categorized into multiple settings with slightly different challenges. The common task, class, and domain incremental setting assumes task data $\mathcal{D}_t$ arrives in sequence $t = \{1, ..., T\}$ in a discrete manner. Task-incremental assumes task identity is known at test time while class-incremental does not. Different from the task and class incremental settings where each task has different classes, domain-incremental learning maintains the same set of classes for every task and only changes the distribution of $\boldsymbol{x}$ by task. In the more challenging task-agnostic setting, task data in $\mathcal{D}$ changes smoothly, and the task identity t is unknown. Our paper tackles the more challenging class-incremental, task-agnostic, and domain-incremental settings.

3.2 PROMPT-BASED LEARNING AND BASELINES

Prompt-based learning is an emerging technique in NLP. In contrast to traditional supervised fine-tuning, this type of methods design task-specific prompt functions to enable pre-trained models perform corresponding tasks (Liu et al., 2021). One of recent techniques, Prompt Tuning (PT) (Lester et al., 2021), proposes to simply condition frozen T5-like language models (Raffel et al., 2020) to perform down-streaming NLP tasks by learning prompt parameters that are prepended to the input tokens. While prompt-based learning has demonstrated success in NLP, to the best of our knowledge, the related research in computer vision and its application to continual learning remains under-investigated. Without loss of generality, here we introduce the definition of PT using the image modality given vision transformer-based models (Dosovitskiy et al., 2021; Vaswani et al., 2017). The definition is easy to generalize to other modalities and sequence-based models.

Given an input of 2D image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ and a pretrained ViT (excluding the classification head) $f = f_r \circ f_e$, where f_e is the input embedding layer, and f_r represents a stack of self-attention layers (Dosovitskiy et al., 2021). Images are reshaped to a sequence of flattened 2D patches $\boldsymbol{x}_p \in \mathbb{R}^{L \times (S^2 \cdot C)}$, where L is the token length, *i.e.*, the number of patches, S is the patch size and C is the original number of channels. To simplify notation, we assume the first token in $\boldsymbol{x}_p$ is the `[class]` token as part of pre-trained model (Dosovitskiy et al., 2021). The pretrained embedding layer $f_e : \mathbb{R}^{L \times (S^2 \cdot C)} \to \mathbb{R}^{L \times D}$ projects the patched image to the embedding feature

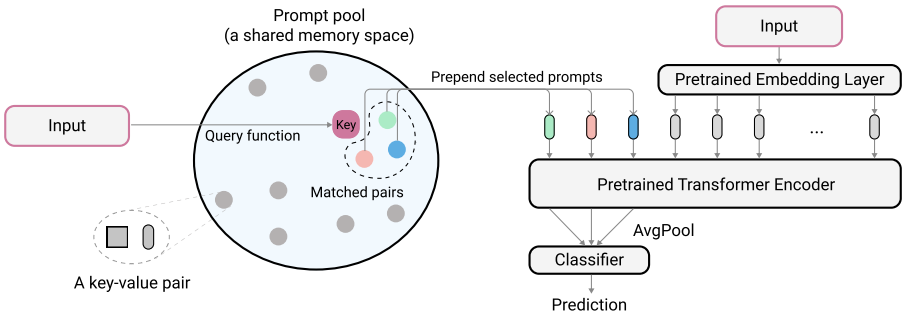

Figure 2: The illustration of L2P at test time. During training time, we follow the same procedure and optimize the model as described in Section 4.3.

$\boldsymbol{x}_e = f_e(x) \in \mathbb{R}^{L \times D}$, where D is the embedding dimension. When solving multiple downstreaming tasks, we keep the large-scale pre-trained backbone frozen to maintain its generality following PT. The direct application of PT is to prepend learnable parameters $P_e \in \mathbb{R}^{L_p \times D}$, called a prompt, to the embedding feature $\boldsymbol{x}_p = [P_e; \boldsymbol{x}_e]$, and feed the extended sequences to the model function $f_r(\boldsymbol{x}_p)$ for performing classification tasks. Different tasks have independent prompts and share one copy of the large model.

Compared with ordinary fine-tuning classification heads with a fixed backbone, literature shows that prompt-based learning results in a sequence-based model with higher capacity to learn features (Liu et al., 2021; Lester et al., 2021). PT can be applied to task-incremental continual learning by learning independent prompts for each task. However, in more challenging settings when no task identity is available, choosing a prompt is more difficult.

4 Learning to Prompt

Our proposed method, Learning to Prompt for Continual Learning (L2P) is depicted in Figure 2. First, we select a subset of prompts from a key-value pair *prompt pool* based on our proposed instance-wise query mechanism. We then prepend the selected prompts to the input embedding. Finally, we feed the extended input embedding to the model, and optimize the classification loss and the prompt pool jointly. In the remainder of this section, we will introduce the critical designs of our method in detail, and discuss how L2P mitigates catastrophic forgetting and addresses some of the other challenges in continual learning (Hadsell et al., 2020), and describe the training procedure.

4.1 From Prompt to Prompt Pool

The motivations of introducing prompt pool are threefold. First, the task index at test time is unknown so training task-independent prompts is not feasible. Second, even if the task-independent prompt can be known at test time, it prevents possible knowledge sharing between similar tasks (Hadsell et al., 2020). Third, while the simple way of learning a single shared prompt for all tasks enables knowledge sharing, it is challenging when tasks are diverse (see Section 5.3). Ideally one would learn a model that is able to share knowledge when tasks are similar, while maintaining knowledge independence otherwise. Thus, we propose using a *prompt pool* to store encoded knowledge, which can be flexibly grouped as an input to the model. The prompt pool is defined as

$$\mathbf{P} = \{P_1, P_2, \cdots, P_M\}, \quad M = \text{total number of prompts}, \tag{1}$$

where $P_j \in \mathbb{R}^{L_p \times D}$ is a single prompt with token length L_p and the same embedding size D as $\boldsymbol{x}_e$. Following the notations in Section 3.2, we let $\boldsymbol{x}$ and $\boldsymbol{x}_e = f_e(\boldsymbol{x})$ be the input and its corresponding embedding feature, respectively. Note that we omit the task index t of $\boldsymbol{x}$ in our notation as our method is general enough to the task-agnostic setting. Denoting $\{s_i\}_{i=1}^N$ as a subset of N indices from $[1, M]$, we can then adapt the input embedding as follows:

$$\boldsymbol{x}_p = [P_{s_1}; \cdots; P_{s_N}; \boldsymbol{x}_e], \quad 1 \leq N \leq M, \tag{2}$$

where ; represents concatenation along the token length dimension. P are free to compose, so they can jointly encode knowledge (e.g. visual features or tasks) for the model to process. Ideally, we

want to achieve a more fine-grained knowledge sharing scheme via prompt combinations at the instance-wise level: similar inputs tend to share more common prompts, and vice versa. We next elaborate our prompt selection strategy and training in the following sections.

4.2 INSTANCE-WISE PROMPT QUERY

We design a key-value pair based query strategy to dynamically select suitable prompts for different inputs. This key-valued memory query mechanism shares some design principles with methods in other fields, such as Differentiable Neural Computer (Graves et al., 2016) and VQ-VAE (Oord et al., 2017), which have external memory to maintain, and employs them for a different purpose. With a slight abuse of notation, we associate each prompt as value to a learnable key: $\mathbf{P} = \{(\boldsymbol{k}_1, P_1), (\boldsymbol{k}_2, P_2), \cdots, (\boldsymbol{k}_M, P_M)\}$, where $\boldsymbol{k} \in \mathbb{R}^{D_k}$. Ideally, we would like to let the input instance itself decide which prompts to choose through query-key matching. To this end, we introduce a query function $q : \mathbb{R}^{H \times W \times C} \to \mathbb{R}^{D_k}$ that encodes input $\boldsymbol{x}$ to the same dimension as the key. Moreover, q should be a deterministic function with respect to different tasks and has no learnable parameters. We directly use the whole pretrained model as a frozen feature extractor to get the query features: $q(\boldsymbol{x}) = f(\boldsymbol{x})[0, :]$ (we use the feature vector corresponding to `[class]`). Other feature extractors like ConvNet are feasible.

Denote $\gamma : \mathbb{R}^{D_k} \times \mathbb{R}^{D_k} \to \mathbb{R}$ as a function to score the match between the query and prompt key (we find cosine distance works well). Given an input $\boldsymbol{x}$, we use $q(\boldsymbol{x})$ to lookup the top-N keys by simply solving the objective:

$$\mathbf{P}_{\boldsymbol{x}} = \underset{\{s_i\}_{i=1}^{N} \subseteq [1,M]}{\operatorname{argmin}} \sum_{i=1}^{N} \gamma\left(q(\boldsymbol{x}), \boldsymbol{k}_{s_i}\right). \tag{3}$$

Note that the design of this key-value strategy decouples the query mechanism learning and prompt learning processes, which has been experimentally shown to be critical (see Section 5.3). Furthermore, querying prompts is done in an instance-wise fashion, which makes the whole framework *task-agnostic*, meaning that the method works without needing clear task boundaries during training, nor task identifications at test time.

Optionally diversifying prompt-selection. Although our method does not need task boundary information, in real-world scenarios and experimental datasets, it is quite common that the task transition is discrete and so task boundaries are known at train time. We find that adding such a prior into our framework can help the model learn better task-specific prompts, especially when tasks have high diversity. To this end, we propose an additional technique for adding task boundaries which is optional for the L2P framework.

During training of task t, we maintain a prompt frequency table $H_t = [h_1, h_2, \cdots, h_M]$, where each entry represents the normalized frequency of prompt P_i being selected up until task $t-1$. To encourage the query mechanism select diverse prompts, we modify equation 3 to

$$\mathbf{P}_{\boldsymbol{x}} = \underset{\{s_i\}_{i=1}^{N} \subseteq [1,M]}{\operatorname{argmin}} \sum_{i=1}^{N} \gamma\left(q(\boldsymbol{x}), \boldsymbol{k}_{s_i}\right) \cdot h_{s_i}, \tag{4}$$

where h_{s_i} penalizes the frequently-used prompts being selected to encourage diversified selection. Equation 4 is only applicable during training; at test time, only equation 3 is needed.

4.3 OPTIMIZATION OBJECTIVE FOR L2P

At every training step, after selecting N prompts following the aforementioned query strategy, the adapted embedding feature $\boldsymbol{x}_p$ is fed into the rest of the pretrained model f_r and the final classifier g_ϕ parametrized by ϕ. Overall, we seek to minimize the end-to-end training loss function:

$$\min_{\mathbf{P}, \phi} \quad \mathcal{L}(g_\phi(f_r^{\text{avg}}(\boldsymbol{x}_p)), y) + \lambda \sum_{\mathbf{P}_{\boldsymbol{x}}} \gamma\left(q(\boldsymbol{x}), \boldsymbol{k}_{s_i}\right), \quad \textit{s.t.}, \quad \mathbf{P}_{\boldsymbol{x}} \text{ is obtained with equation 3}, \tag{5}$$

where $f_r^{\text{avg}} = \text{AvgPool}(f_r(\boldsymbol{x}_p)[N \cdot L_p, :])$, i.e., the output hidden vectors corresponding to the $N \cdot L_p$ prompt locations are averaged before the classification head. The first term is the softmax cross-entropy loss, the second term is a surrogate loss to pull selected keys closer to corresponding query features. λ is a scalar to weight the loss.

5 EXPERIMENTS

To evaluate the proposed L2P, we closely follow the settings proposed in prior works (Lopez-Paz & Ranzato, 2017; Zeno et al., 2018; Van de Ven & Tolias, 2019), and conduct comprehensive experiments. In particular, we consider (1) the class-incremental setting, where the task identity is unknown during inference; (2) the domain-incremental setting, where the input domain shifts over time; (3) the task-agnostic setting, where there is no clear task boundary. Moreover, we conduct extensive ablation studies to provide a deeper understanding of our method.

Evaluation metrics. For settings with task boundaries and where each task has an associated test set, we use two metrics, *Average accuracy* (A) and *Forgetting* (F), which are widely used in previous works (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018; Mai et al., 2021). Denoting by $a_{t,i}$ the accuracy of the i-th task after finishing training on task t, we can compute the corresponding average accuracy A_t and forgetting F_t up until the current task t as follows:

$$A_t = \frac{1}{t}\sum_{i=1}^{t} a_{t,i}, \quad F_t = \frac{1}{t-1}\sum_{i=1}^{t-1} \max_{i' \in \{1,\cdots,t-1\}} \left(a_{i',i} - a_{t,i}\right). \tag{6}$$

We report the final performance A_T and F_T after training on all T tasks. For settings without task boundary or where there is only a single test set available, we only report the final test accuracy following the protocol in previous work (Lomonaco & Maltoni, 2017; Shanahan et al., 2021).

Comparing methods. We compare L2P against several baselines and state-of-the-art continual learning methods. Note that we used the same pretrained ViT-B/16 model (Dosovitskiy et al., 2021) as a starting point for every method to ensure fair comparison. (1) *FT-iid* is the usual supervised finetuning under the i.i.d. setting, which is the possible upper bound performance a continual learning method could achieve. (2) *FT-seq-frozen* is the naive sequential fine-tuning approach with the pretrained model frozen. (3) *FT-seq* is the naive sequential fine-tuning approach (model weights are updated). (4) *EWC* (Kirkpatrick et al., 2017) is a regularization-based approach aiming at limiting the learning rate of parameters that are important for previous tasks. (5) *LwF* (Li & Hoiem, 2017) applies the idea of knowledge distillation (Hinton et al., 2015) to preserve knowledge from past tasks. To further demonstrate the effectiveness of our method, we introduce two state-of-the-art rehearsal-based methods, which require additional memory buffer to save samples from past tasks: (6) *ER* (Chaudhry et al., 2019; Hayes et al., 2019) mixes samples from buffer with samples the from current task in the training process. (7) *GDumb* (Prabhu et al., 2020) simply constructs the buffer from the sequence of tasks and trains on the buffered samples jointly, so forgetting metric is not applicable to this method. GDumb can outperform many state-of-the-art methods under various settings (Prabhu et al., 2020; Mai et al., 2021). Following the experiment setting in Prabhu et al. (2020), we store an average of 50 samples per class, *e.g.*, a buffer size of 5,000 for CIFAR100, as this is a relatively large choice of buffer size that guarantees SOTA performance.

Experiment details. For L2P, we train all models using Adam (Kingma & Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a batch size of 128, and a constant learning rate of 0.03 for all settings. Input images are resized to 224×224 and normalized to the range of $[0, 1]$ to match the pretraining setting. As pointed out by Buzzega et al. (2020), training multiple epochs for each task disentangles the effects of possible underfitting from forgetting. Thus, we train every task for 5 epochs in the class- and domain-incremental settings. However, in the task-agnostic setting where we don't have the concept of a task, we follow Shanahan et al. (2021) to train every batch only once. We set $M = 10, N = 5, L_p = 5$ for all CIFAR-100 based datasets and CORe50. For 5-datasets, we use $M = 20, N = 4, L_p = 5$. Prompts only add $46,080$ and $92,160$ parameters to the original pretrained model for these two settings, leading to a small 0.05% and 0.11% total parameter increase, respectively. We find λ in equation 5 is not sensitive and works well in a large range, so we set $\lambda = 0.5$ consistently for all datasets.

5.1 RESULTS ON CLASS-INCREMENTAL LEARNING

Split CIFAR-100. This dataset randomly splits the original CIFAR-100 dataset (Krizhevsky et al., 2009) into 10 tasks, where each task consist of 10 disjoint classes. Since the tasks are from a single original dataset, they share some similarities and some classes are even from the same superclass.

5-datasets. This dataset (Ebrahimi et al., 2020) consists of five image classification datasets: CIFAR-10, MNIST (LeCun, 1998), Fashion-MNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011),

Table 1: Results on class-incremental learning. Accuracy and forgetting are reported. All methods start from the same pre-trained ViTB/16 model and train on each task for 5 epochs. Methods are separated based on whether rehearsal is applied. All results are shown in percentage (%) and are averaged over 3 runs.

Method	**Split CIFAR-100**		**5-datasets**	
	Average Acc (↑)	Forgetting (↓)	Average Acc (↑)	Forgetting (↓)
Upper bound:				
FT-iid	90.85±0.12	-	93.93±0.18	-
Non-rehearsal based methods:				
FT-seq-frozen	17.72±0.34	59.09±0.25	39.49±0.12	42.62±0.20
FT-seq	33.61±0.85	86.87±0.20	20.12±0.42	94.63±0.68
EWC	47.01±0.29	33.27±1.17	50.93±0.09	34.94±0.07
LwF	60.69±0.63	27.77±2.17	47.91±0.33	38.01±0.28
L2P (ours)	**83.83±0.04**	**7.63±0.30**	**81.14 ±0.93**	**4.64 ±0.52**
Rehearsal based methods:				
ER	82.53±0.17	16.46±0.25	89.30±0.94	8.08±0.53
GDumb	81.67±0.02	-	70.76±0.12	-
L2P-R (ours)	**86.31±0.59**	**5.83±0.61**	**91.92±0.78**	**3.34±0.71**

Table 2: Results on task-agnostic continual learning, in terms of test accuracy. We use Gaussian scheduled CIFAR-100 as the evaluation benchmark. All results are shown in percentage (%) and are averaged across 3 runs.

Category	**Method**	Test Acc (↑)
Upper bound	FT-iid	90.85±0.12
Rehearsal	ER	82.53±0.17
	GDumb	81.67±0.02
Non-rehearsal	EWC	63.04±0.42
	LwF	69.46±0.35
	L2P (ours)	**88.34±0.14**

Table 3: Results on domain-incremental learning, in terms of test accuracy. We use CORe50 as the evaluation benchmark. All results are shown in percentage (%) and are averaged across 3 runs.

Category	**Method**	Test Acc (↑)
Upper bound	FT-iid	82.15 ±0.37
Rehearsal	ER	80.10±0.56
	GDumb	74.92±0.25
Non-rehearsal	EWC	74.82±0.60
	LwF	75.45±0.40
	L2P (ours)	**78.33±0.06**

and notMNIST (Bulatov, 2011). Although each dataset alone is not hard, the sequential training of them is fairly challenging to even ImageNet pre-trained models, since models are more susceptible to forgetting when the tasks are diverse (Mehta et al., 2021). We apply the optional strategy introduced in 4.2 to enhance prompt selection diversity.

Table 1 summarizes the results on these two class-incremental benchmarks. Similar to what Mehta et al. (2021) have shown: in the simpler task-incremental setting, pre-trained models can overall improve these benchmarks when integrated with existing methods. However, the forgetting rate remains prominent in the class-incremental setting as we shown, suggesting the importance of innovating technologies in pre-trained models beyond applying existing methods.

Our method, L2P, achieves superior performance in terms of both average accuracy and forgetting. In particular, our method: (1) outperforms all non-rehearsal based methods by a large margin, including beating rehearsal-based methods on split CIFAR-100 without rehearsal; and (2) our method improves upon state-of-the-art rehearsal-based methods when incorporating the rehearsal strategy, closing a significant part of the gap to the upper bound performance when doing finetuning under the i.i.d. setting; and (3) compared to the performance of FT-seq-frozen with our method, we can see that naive sequential training is not able to fully take advantage of the pretrained features, further demonstrating the advantages of introducing the prompting strategy.

Table 4: Ablation study on 5-datasets. All results are shown in percentage (%).

Method	5-datasets	
	Average Acc (↑)	Forgetting (↓)
L2P without prompt pool	51.96	26.60
L2P without key-value pair	58.33	20.45
L2P without diversified prompt selection	62.26	17.84
L2P	**81.14**	**4.64**

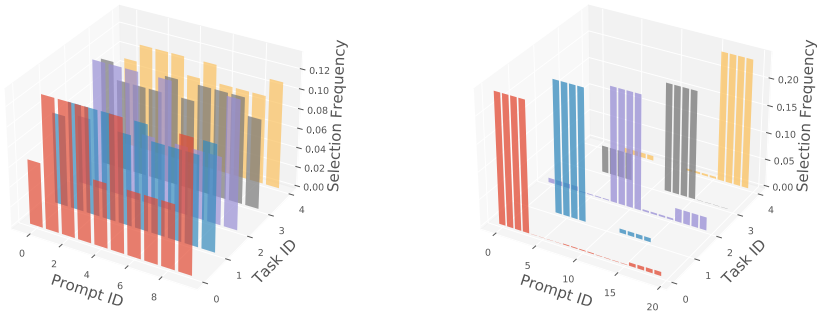

Figure 3: Prompt selection histograms for (left) Split CIFAR-100 and (right) 5-datasets. Note that we only show the first 5 tasks for Split CIFAR-100 for better readability.

5.2 Results on task-agnostic and domain incremental settings

Gaussian scheduled CIFAR-100. In this task-agnostic setting, the distribution of data shifts gradually throughout the learning process (Shanahan et al., 2021), the probability that a class is present in a batch follows a Gaussian distribution centered at some time step. There is no explicit task boundaries between batches, thus requiring methods to be able to implicitly adapt to non-stationary data distribution without utilizing any task-specific information during training and inference.

Table 2 summarizes the results. L2P achieves the best performance among all methods, including rehearsal based ones. The task-agnostic setting is usually considered more challenging than the class-incremental setting. Since these two benchmarks have the same test test, we can compare them deeper. Interestingly, EWC and LwF both achieve higher accuracy than that on split CIFAR-100, indicating that a well-pretrained model itself may serve as a better starting point for task-agnostic continual learning. Similar observations has been reported on a simpler task-incremental setting in Mehta et al. (2021). Moreover, L2P achieves a test accuracy 88.34%, which is very close to the upper bound performance 90.85% shown in Table 1, suggesting strongly reduced forgetting rate.

CORe50. This is a dataset specifically designed for continual object recognition (Lomonaco & Maltoni, 2017). It is a collection of 50 objects collected in 11 distinct domains, where 8 of them (120,000 samples) are used for training, and the rest are considered as a single test set (45,000 examples). Methods are trained on each domain sequentially.

Table 3 summarizes the results on the domain-incremental setting. Although L2P still achieves better performance than most methods, surprisingly, all methods are quite close to the upper bound performance FT-iid. This indicates that a well pretrained model has the potential to accumulate knowledge from different domains without much interference. However, more comprehensive experiments are required to further confirm this observation, which we leave to future work.

5.3 Effectiveness of core designs

We further conduct ablation studies to demonstrate the effectiveness of the core designs of L2P.

Prompt pool. To further confirm the importance of the prompt pool, we design a counterpart of our method with only a single prompt instead of the prompt pool. This variation of our method keeps the same prompt capacity as L2P in equation 2. From Table 4 (row 1 and 4), we can see that L2P significantly outperforms its counterpart with a single prompt, suggesting that the prompt pool encodes task-relevant and task-specific knowledge well.

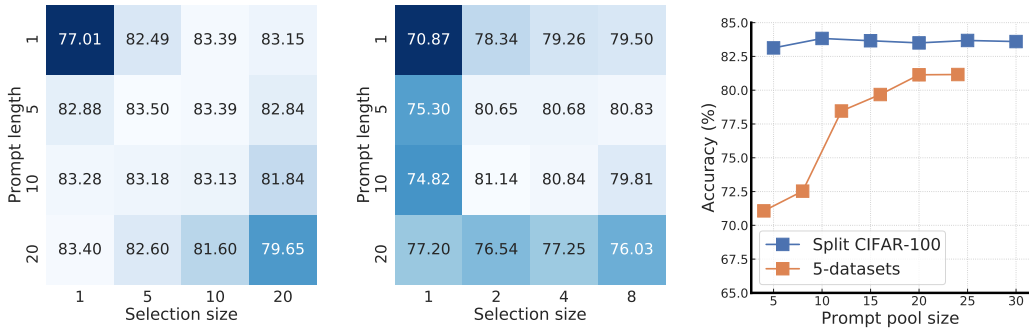

Figure 4: Left-Middle: Average accuracy w.r.t prompt length L_p and prompt selection size N for Split CIFAR-100 and 5-datasets, respectively, given $M = 20$. Right: Average accuracy (%) w.r.t. prompt pool size M, given $L_p = 5$, $N = 5$ for Split CIFAR-100 and $L_p = 5$, $N = 4$ for 5-datasets.

Key-value pair design. We remove the learnable key associated with prompts and directly use mean of prompts as keys and the mean of input embedding as query features, as they reside in the same space. From Table 4 (row 2), we can see this results in a significant drop, demonstrating the importance of introducing learnable keys to decouple the query and prompt learning process.

Diversified prompt selection. This technique is used by default on 5-dataset only. When we remove it, (Table 4 row 3), we basically allow instances from different tasks to choose prompts freely. The decrease in performance demonstrates that when tasks are diverse, adding the diversified prompt selection strategy can indeed reduce unnecessary knowledge sharing and thus mitigating interference between unrelated tasks.

To better understand the prompt selection mechanism, we plot the prompt selection histograms for each task in both split CIFAR-100 and 5-datasets in Figure 3 under the best-performing parameters settings, respectively. From the plot of Split CIFAR-100 (left), the tasks largely share all prompts, meaning that our prompt selection mechanism encourages more knowledge sharing between similar tasks. In contrast, in the plot of 5-datasets (right), diverse tasks tends to choose more task-specific prompts and share less.

Effect of hyperparameters for L2P. Recall that there are three key hyperparameters, including the size of the prompt pool M, length of a single prompt L_p, and the selection size N used as model input. Intuitively, M decides the total capacity of learnable prompt parameters. L_p decides capacity of a singe prompt (which jointly encodes certain knowledge), and $L_p \times N$ decides the total size used to prepend the input. From the results on both datasets (Figure 4 (left-middle)), a smaller L_p always negatively affects results. We hypothesize that a reasonable capacity of a single prompt is critical to encode a certain aspect of shared knowledge. Increasing the prompt pool size shows positive effect for performance as shown in Figure 4 (right), especially on 5-datasets, suggesting a large enough pool size is needed to encode task-specific knowledge when tasks are diverse.

6 Conclusion

This paper presents a novel method to address some of the key challenges in continual learning with a method that can achieve strong performance without a need for rehearsal and task identity. L2P introduces prompt-based learning to continual learning and proposes a novel technique to enable a single pre-trained model to adapt to sequential tasks via a shared prompt pool, successfully mitigating the catastrophic forgetting problem. The resulting method achieves good results on challenging continual learning problems, including class-incremental, domain-incremental, and task-agnostic settings, demonstrating the effectiveness of the method, as well as its advantages to satisfy the practical data privacy requirement when storing data as rehearsal buffer is prohibited.

Although our method is demonstrated on vision models, it does not make any assumption of modalities. We leave exploration on other modalities as future work. Additionally, L2P assumes there are pre-trained sequence-based models. While they have become common assets in advanced communities, how to generalize our framework to ConvNets could another appealing research direction.

7 Ethics Statement

L2P is a strong continual learning method and has great potential to be applied in various fields. However, there are some ways it could be misused. Our method takes a well-pretrained model as a backbone, thus any bias and fairness issues (Mehrabi et al., 2021) in the original model may be carried over during the continual learning process. We encourage any users to thoroughly check the pretrained model to mitigate any bias and fairness issues. Moreover, the method could be deployed in safety-critical applications, such as autonomous driving systems (Grigorescu et al., 2020), which may present potential security issues in terms of adversarial attacks (Madry et al., 2017). We would recommend testing the robustness of our method in future work and design corresponding defense techniques to deal with potential security concerns.

8 Reproducibility

To make the results presented in our work reproducible, we include all experiment setups and details, evaluation metrics, and comparing methods in Section 5. We test our method on multiple publicly available datasets and under different settings. We report the average and corresponding standard deviations over multiple runs using different randoms seeds for our main results (Table 1, 2 and 3). Our results are also verified on different hardwares, including TPU and GPU. We plan to make the code publicly available upon acceptance.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 2

Yaroslav Bulatov. notmnist dataset, 2011. URL http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html. 7

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020. 6

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 3, 6

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 3, 6

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3, 6

Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *ECCV*, 2020. 6

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538 (7626):471–476, 2016. 5

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. 10

Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 2020. 1, 4

Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *ICRA*, 2019. 3, 6

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 6

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, 2017. 2, 6

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*, 2016. 3

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 6

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 1, 3, 4

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 1, 3

Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2017. 2, 6

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 1, 3, 4

Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, 2017. 6, 8

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *NeurIPS*, 2017. 6

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 10

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *arXiv preprint arXiv:2101.10423*, 2021. 1, 3, 6

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018. 3

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018. 3

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989. 1

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021. 10

Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. *ICML Workshop on Theory and Foundation of Continual Learning*, 2021. 7, 8

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS*, 2011. 6

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017. 5

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 1, 3

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 6

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21:1–67, 2020. 1, 3

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 3

Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020. 3

Murray Shanahan, Christos Kaplanis, and Jovana Mitrović. Encoders and ensembles for task-free continual learning. *arXiv preprint arXiv:2105.13327*, 2021. 6, 8

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020. 1

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proc SIGSAC conference on computer and communications security*, 2015. 3

Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 6

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 3

Zifeng Wang, Tong Jian, Kaushik Chowdhury, Yanzhi Wang, Jennifer Dy, and Stratis Ioannidis. Learn-prune-share for lifelong learning. In *ICDM*, 2020. 3

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 3

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017. 2

Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. Task agnostic continual learning using online variational bayes. *arXiv preprint arXiv:1803.10123*, 2018. 6