# Introducing Language Guidance in Prompt-based Continual Learning

Muhammad Gul Zain Ali Khan[1,2] Muhammad Ferjad Naeem[3] Luc Van Gool[3] Didier Stricker[1,2]
Federico Tombari[4,5] Muhammad Zeshan Afzal[1,2]
[1]RPTU [2]DFKI [3]ETH Zürich [4]TUM [5]Google

## Abstract

*Continual Learning aims to learn a single model on a sequence of tasks without having access to data from previous tasks. The biggest challenge in the domain still remains catastrophic forgetting: a loss in performance on seen classes of earlier tasks. Some existing methods rely on an expensive replay buffer to store a chunk of data from previous tasks. This, while promising, becomes expensive when the number of tasks becomes large or data can not be stored for privacy reasons. As an alternative, prompt-based methods have been proposed that store the task information in a learnable prompt pool. This prompt pool instructs a frozen image encoder on how to solve each task. While the model faces a disjoint set of classes in each task in this setting, we argue that these classes can be encoded to the same embedding space of a pre-trained language encoder. In this work, we propose Language Guidance for Prompt-based Continual Learning (LGCL) as a plug-in for prompt-based methods. LGCL is model agnostic and introduces language guidance at the task level in the prompt pool and at the class level on the output feature of the vision encoder. We show with extensive experimentation that LGCL consistently improves the performance of prompt-based continual learning methods to set a new state-of-the art. LGCL achieves these performance improvements without needing any additional learnable parameters.*

## 1. Introduction

In Class Incremental Continual Learning, we task a model to learn a sequence of non-overlapping tasks consisting of new classes being introduced at each task. This presents a challenge different from the common supervised learning setting as the data distribution is continuously changing, and the independent and identically distributed (i.i.d.) data assumption does not hold. As a result, a model trained with our usual training recipe of optimising a loss function on incoming data leads to catastrophic forgetting [33] *i.e.*, the model forgets the previously seen classes since the loss only incentivises performance on the current task. There have been several attempts to address this challenge. One popular line of works aims to identify model parameters most important for performance on each task and prevent them from changing too much through subsequent tasks [19, 65, 25, 1]. These regularization-based methods, however, achieve sub-optimal performance as we move to very complex tasks where the model needs to share parameters between different tasks to learn a robust representation.

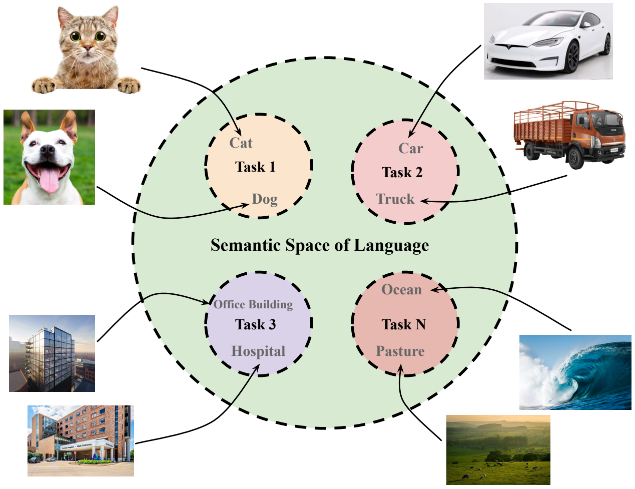

Figure 1: In Computer Vision, Continual Learning in the class incremental setting aims to learn a single model on a sequence of tasks where each task consists of disjoint classes. While each task represents a disjoint set of classes, we argue that they can be mapped to the same semantic space of a pretrained language encoder. Based on this principle, we propose to introduce language guidance in a continual learner to mitigate catastrophic forgetting.

Another line of work takes a very simple but effective approach of storing a chunk of training data. Rehearsal-based methods [7, 8, 12] maintain a rehearsal buffer which is a finite set of training data stored across each task. The key intuition is that to prevent forgetting on previously seen classes, the model simply uses the examples in the rehearsal buffer when optimising for new tasks. However, these methods require large buffer sizes as the number of tasks increases and hence become expensive. Moreover, they have been criticised for being impractical in real-world settings where privacy concerns prevent storing data. Architecture-based methods [49, 64, 23, 28, 46, 66] take an orthogonal approach where these works reserve specialised parts of the network for each task and take an approach similar to multi-task learning. However, this can bring a significant increase in the number of learnable parameters. Moreover, this requires knowing the task identity at test time to select the relevant network module for each task which is not a realistic setting.

Recently, Learning to Prompt (L2P)[59] has proposed an exciting new direction for continual learning. Instead of learning the parameters of the model for each new task, the authors propose to use a pre-trained vision encoder and learn the prompts that can instruct this pre-trained model to solve new tasks. This technique is called Prompt Tuning and is popularized by its success in Natural Language Processing (NLP). A learned prompt instructs the model on how to solve a new task using the wide set of knowledge it has stored during pre-training. These methods have shown incredible performance boosts at a fraction of learnable parameters. L2P initialises a learnable prompt pool where each prompt is attached with a learnable key. The authors propose to use the CLS feature from the pre-trained vision encoder to perform a lookup with this learnable pool. The selected prompts are then appended with the patchwise embeddings of the image into the pre-trained model, and the output representations of the selected prompt tokens are used to learn a linear classification layer. This surprisingly simple formulation has brought impressive performance gains without the need to store any data in a rehearsal buffer.

While the sequence of data from each task in continual learning has a changing distribution, we argue that the classes of each task can be mapped to the same semantic space. In this work, we argue that language represents such a robust representation space where all tasks can be sufficiently mapped to. Hence if we encode the features of the continual learner to map to a semantic space of language, this can present an avenue to mitigate catastrophic forgetting and result in a more robust continual learner. We use this insight to develop a novel method **Language Guidance for Prompt-based Continual Learning (LGCL)** that can introduce language guidance into any prompt-based continual learning method. We achieve this by introducing language guidance at two levels. First, we introduce the task-level language guidance by incentivising the model to map the learnable keys of the prompt pool into a shared language representation of all classes in the task. Secondly, we incentivise the model to map the output features of the visual encoder after prompting it to align with the language representation of its respective class. The model learns a robust representation of all tasks by aligning these representations with a pre-trained semantic space of language.

Our contributions are as follows: 1) We present a novel perspective that entails introducing language guidance in continual learning to mitigate catastrophic forgetting. 2) We propose Language Guidance for Prompt-based Continual Learning (LGCL), a novel method that introduces language guidance in prompt-based continual learning methods. 3) Without any additional learnable parameters or extra memory requirements at inference, LGCL improves the performance of prompt-based continual learning methods and achieves state-of-the-art performance on two challenging continual learning benchmarks.

## 2. Related Work

**Continual Learning** tasks a model to learn a sequence of tasks while mitigating catastrophic forgetting [33]. Methods in continual learning have traditionally been divided into three categories, namely regularization-based methods, rehearsal-based methods, and architecture-based methods. Regularization-based methods [19, 65, 25, 1] aim to find important parameters for each task and limit their plasticity in future tasks by adjusting the learning rate. These methods work without storing any labelled examples; however, they are unable to achieve satisfactory performance in challenging and complex datasets [30, 47, 61].

Rehearsal-based methods [7, 8, 12] maintain a buffer to save data from older tasks and use it for training while future tasks become available. Several works improve upon it with training tricks like knowledge distillation [47, 61, 6, 3] and self-supervised learning [5, 40]. Rehearsal-based methods address catastrophic forgetting by simply retraining on stored data from all tasks the model has seen at any given stage. Although conceptually very simple, these methods have been very competitive and consistently rank among state-of-the-art [37, 30]. However, these methods suffer from performance degradation as the replay buffer gets smaller or the number of classes increases significantly. Moreover, these methods can not be used when data privacy is a concern [51].

Architecture-based methods aim to specialize parts of the model for each task. These modules are added as additional blocks [49, 64, 23, 28, 46, 66], or specialising task specific sub-networks [31, 50, 56, 16]. Since these models specialise parts of the model for each task, they often require the task identity as an input to the model at test time which limits their use in realistic class-incremental and task-agnostic settings. Some methods infer task identity from the data [60], while others infer it using a rehearsal buffer [63, 40]. However, these methods require significantly more learnable parameters, often as many as the core model. Prompt-based methods [59, 57] have recently emerged as a new exciting fourth direction in continual learning. These methods use a pre-trained feature extractor and learn each task as a set of prompts that specialise the pre-trained model for the task. These methods are highly parameter efficient as prompts are small sequences of learnable tokens. These methods achieve this by encoding the task information in the learnable prompts rather than storing input data. Moreover, these methods do not require the task identity as input, thanks to a clever lookup formulation conditioned on the input to select the prompt.

**Prompt Learning** has emerged as a popular transfer learning technique in Natural Language Processing (NLP). Instead of retraining the model, prompt learning learns a set of prompts that instructs the pre-trained model to process the new task. To this end, several works introduce prompts as learnable tokens achieving impressive performance on transfer learning [22, 24]. These methods are incredibly efficient with respect to learnable parameters compared to competitors [53, 39, 14].

**Language Guidance** has been extensively explored in various vision tasks, including open set learning [43, 11, 55], zero-shot learning [36, 34, 35, 17], and metric learning [48]. Methods in open set learning [43, 11] learn a vision encoder that can map to the same embedding space as language. The model can then generalise to new classes by generating the embeddings of the class names without requiring labelled visual data. Methods in zero-shot learning use word embeddings from pre-trained language models [62, 38, 52] and knowledge graphs [54, 15, 2, 35, 32] to encode semantic similarities between seen and unseen classes. Unseen classes can then be inferred by measuring a distance metric between a vision encoding and a language feature from a pre-trained model. Integrating language supervision in vision models allows the model to adapt to new classes efficiently, as these classes lie in the same semantic space as previously seen classes.

Our method lies at the intersection of prompt-based continual learning and language guidance. To the best of our knowledge, we provide the first method for integrating language guidance in prompt-based continual learning methods for challenging class-incremental continual learning.

## 3. Background

**Notations.**

Continual Learning aims to train a Machine Learning model on a stream of data from a sequence of tasks. We denote the sequence of tasks as $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_\mathrm{T}\}$ where each task consists of tuples of input data $(x_i^t, y_i^t)$ where $x_i^t \in \mathcal{X}$ represents the input image in an RGB color space and $y_i^t \in \mathcal{Y}$ represents the corresponding label for the task $t$. In line with previous works, the tasks are non-overlapping, *i.e*. images and labels are not repeated in subsequent tasks, and the model does not have access to the training data from the previous tasks. We focus on class incremental setting in which task identity is unknown at test time. Moreover, we assume that a pre-trained feature extractor $\mathcal{F}$ is available for images and is kept frozen throughout the training [57, 59]. Similarly, we assume that a pre-trained feature extractor is available for language and is kept frozen throughout the training [43]. To be consistent with previous works, vision and language feature encoders are each independently trained. We consider the class incremental setting in which task boundaries are defined clearly, and task identity is unknown during test time [41].

### 3.1. Prompt-based Continual Learning

Since our method aims to improve prompt-based continual learning methods, we provide an overview in this section. The vast majority of continual learning works maintain a Replay Buffer consisting of labelled samples of previous tasks. This buffer is used to avoid catastrophic forgetting by continuously training on previous tasks. However, rehearsal buffers are expensive to store and do not scale well to large dataset or a large number of tasks. Recently, prompt tuning has emerged as an alternative to rehearsal buffers. Methods in this direction [57, 59] use a pre-trained vision encoder and rely on prompt learning to learn the tasks continually, instead of replaying samples from previous tasks. This is achieved by storing the knowledge of each task in a learnable pool of prompts without explicitly defining a pool for each task.

Given a pre-trained image feature extractor $\mathcal{F}$, an image transformer, these methods aim to learn prompts that can be used to instruct the pre-trained model to solve the encoded task. Given an image $x$, they do a forward pass to extract the CLS token corresponding to the global feature of the image. This feature is used to look up the relevant prompt from the prompt pool, which we introduce in the next section.

**Introducing a learnable Prompt Pool.** Prompt learning has emerged as a powerful technique in NLP to use a general pre-trained language model and re-purpose it for a downstream task by introducing a set of learnable tokens without changing the parameters of the pre-trained model. Prompt Tuning [22] introduces a set of learnable tokens for a pre-trained language model like T5 [44] to condition the pre-trained model to solve a new task. These tokens encode the task instructions and instruct the pre-trained model to solve the NLP task at hand [26]. On the other hand, another form of utilizing learnable prompts is prefix tuning. In prefix tuning, the learnable prompt is appended to the keys and values of attention blocks [57].

Introducing Prompting for Continual Learning involves some clever design choices. We want to utilize the prompt to fine-tune the internal representations of the vision transformer for our task-specific distribution without tuning the model parameters. The simplest approach is to learn one set of prompt tokens for each task capturing the task-specific information in its tokens. However, this has a significant limitation in that the model needs the task id as an input to

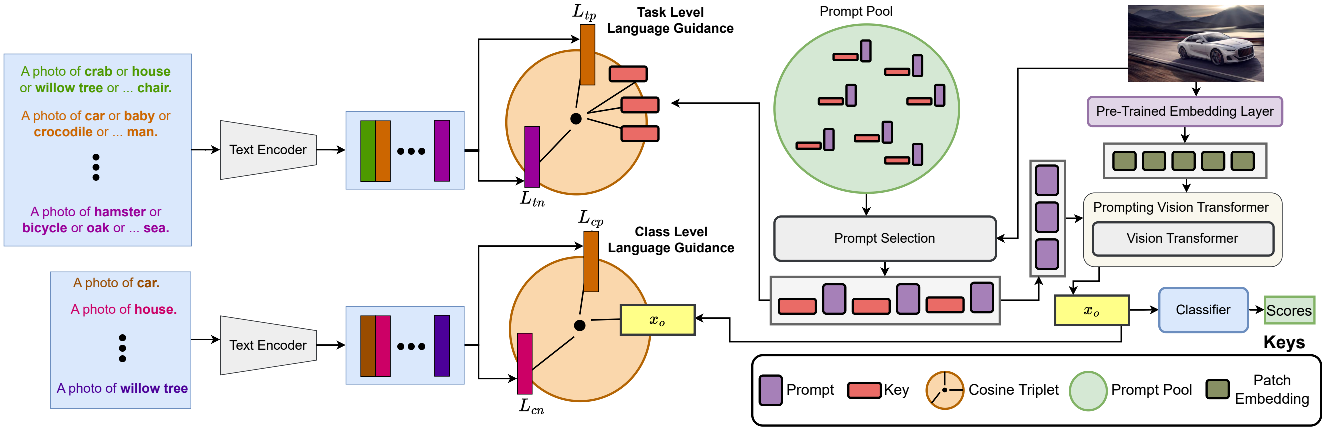
A photo of crab or house or willow tree or ... chair.
A photo of car or baby or crocodile or ... man.
A photo of hamster or bicycle or oak or ... sea.
Text Encoder
$L_{tp}$
Task Level Language Guidance
$L_{tn}$
A photo of car.
A photo of house.
A photo of willow tree
Text Encoder
$L_{cp}$
Class Level Language Guidance
$L_{cn}$
$x_o$
Prompt Pool
Prompt Selection
Pre-Trained Embedding Layer
Prompting Vision Transformer
Vision Transformer
$x_o$
Classifier
Scores
Keys
Prompt
Key
Cosine Triplet
Prompt Pool
Patch Embedding

Figure 2: Our novel **Language Guidance for Prompt-based Continual Learning (LGCL)** introduces Language Guidance in prompt-based continual learning methods. We introduce language guidance at two levels. At task level, we encode the language feature corresponding to the classes the model will encounter in the selected keys from the prompt pool. At class level, we encode the language feature corresponding the the ground truth class in the output feature of the vision transformer. Together the two modules improve the baseline prompt-based continual learning method and bring performance improvements without introducing additional learnable parameters.

select the correct prompt. Moreover, this does not allow the model to create a joint representation that can share similar information between tasks. Learning to Prompt (L2P) [59] instead cleverly introduces a pool of prompts where each prompt can encode knowledge without explicitly attaching it to a task. The prompt pool is defined as:

$$\mathbf{P} = \{P_1, P_2, \cdots, P_M\}, \quad M = \text{total } \# \text{ of prompts}, \quad (1)$$

where $P_j \in \mathbb{R}^{L_p \times E}$ is a single prompt with token length $L_p$ and the same embedding size $E$ as $x_p$. Each prompt is attached to a learnable key $k_j$. Given the CLS feature corresponding to an input image $x$ as a query, the model can look up the relevant prompt encoding the knowledge for its task by a key query look up. Learning to Prompt [59] uses top N prompts corresponding to the lookup as the selected prompts for tuning. The selected prompts are used as additional input to the pre-trained Vision Transformer, along with patch embeddings. Dual Prompt[57] instead uses prefix-tuning and directly injects these prompts in the Multi-headed attention layers of the Vision Transformer by prepending the learnable prompt with keys and values of the Multi-Head Attention layer. We name the pre-trained frozen vision transformer prompted with learnable prompts the Prompting Vision Transformer.

**Encoding task information in the selected prompts.** We define $x_o$ as the output feature of the Vision Transformer to be used for classification. In Dual Prompt[57] this corresponds to the CLS feature of the Vision transformer after injecting the selected prompts. In L2P[59] this refers to the average pooled output of the tokens corresponding to the selected prompts. This feature is trained for classification with a supervised loss like Cross-Entropy for classification tasks. The training loss incentivises the model to store task-specific features in the prompt pool. Moreover, since the prompt selection is dependent on the input image and not the task, the task representations are shared in the pool and evolve over the training period to encode the different tasks. Dual Prompt [57] improves upon L2P [59] by additionally introducing a global learnable prompt which learns a shared representation across all tasks. This global prompt is used with the prompts as input to the Vision Transformer.

## 4. Language Guidance for Prompt-based Continual Learning (LGCL).

Continual Learning addresses the task of learning a changing distribution of data coming from different tasks. While the visual data of these tasks changes, their task definition or classification targets can lie in the same space of language. Language consists of a compact representation of the world and storing language cues like class names is available to a model for free as it has access to them from the current and previous tasks. We propose to integrate this language guidance into the prompt-based continual learning methods to further mitigate catastrophic forgetting. More specifically, we propose to use a text encoder $\mathcal{T}$ from a pre-trained model to encode the task knowledge and class knowledge into the prompt pool and learned features of the continual learner. LGCL is a generic framework that can be incorporated into any prompt-based method for continual learning without requiring any additional learnable parameters. We give an overview of our method in Figure 2.

### 4.1. Introducing Task Level Language Guidance

Given the $t-th$ task, we denote the class names of classes represented in this task with the set $\mathcal{Y}^t$. The task $t$ involves correctly classifying the classes contained in $\mathcal{Y}^t$. Therefore, we propose to represent the language representation of the task as a prompt of class names as follows.

"A photo of {**class 1**} or {**class 2**} $\cdots$ or {**class n**}"

where {**class 1**},$\cdots$, {**class n**} are replaced with the class names of the task. The prompt is input to the pre-trained text encoder to extract the feature corresponding to the output token to represent $L_t \in \mathbb{R}^E$, the language representation of the task $t$ with embedding dimension $E$. Since prompt-based continual learning methods learn the lookup operation for selecting the prompts against learnable keys, we aim to encode the task definition in these keys. Given $P_s = \{P_{s_1}; \cdots; P_{s_N}\}$ are the $N$ prompts selected for the task $t$, with learnable keys $K_s = \{k_{s_1}; \cdots; k_{s_N}\}$, we aim to encode the $L_t$ in these learnable keys. For each key $k \in \mathbb{R}^E$ in $K_s$, we compute the cosine similarity between the key and the language encoded task feature $L_t$ as follows:

$$S(k, L_t) = \frac{k \cdot L_t}{|k||L_t|} \tag{2}$$

We optimise the cosine similarity with a triplet loss to incentivise the model to align the selected keys close to the language representation of their respective task and away from the language representation of other tasks. Given the task $t$, the model only has access to the task definitions of the current task and the tasks before $t$. Therefore, when optimising the loss for the current task $t$, $L_{tp}$ denotes the language feature of the task $t$ as the positive and the language features of the previous tasks are randomly sampled as the negative $L_{tn}$ in each optimisation step. For a selected key $k$, we optimise the following loss:

$$\mathcal{L}_{task}(k, L_{tp}, L_{tn}) = 1 - S(k, L_{tp}) + S(k, L_{tn}) \tag{3}$$

By aligning the lookup keys with the language feature of the task, the model learns a feature representation of keys that comes from the same distribution of language and is less likely to diverge between tasks while training. Since the performance of prompt-based continual learning methods depends on the correctness of the selected prompts, learning better keys can allow for better performance.

### 4.2. Introducing Class Level Language Supervision.

The prompt pool represents the task-level knowledge for the model. We further want to guide the class-level feature of the image with language. For a given training sample $(x, y)$ consisting of image $x$ with label $y$, we take the class name for $y$ and represent it in language as the following prompt:

"A photo of {**class name**}"

Similar to the last module, the prompt is input to the pre-trained text encoder to extract the feature corresponding to the output token to represent $L_c \in \mathbb{R}^E$, the language representation of class $y$ with embedding dimension $E$. We want to encode this language representation in the output feature $x_o \in \mathbb{R}^E$ of the vision transformer used for classification. This feature aims to represent the class-level information of the task. We introduce language guidance in this feature representation through a cosine triplet loss similar to the last module. Our positive example consists of the language-encoded feature of class $y$ as $L_{cp}$. For the negative example, we randomly sample a class from the classes of the previous tasks as $L_{cn}$ for each optimization step. We optimise the following loss for introducing language guidance in our continual learner:

$$\mathcal{L}_{class}(x_o, L_{cp}, L_{cn}) = 1 - S(x_o, L_{cp}) + S(x_o, L_{cn}) \tag{4}$$

By aligning output image features to the classwise language representation, we incentivise the model to map to the same semantic space of the pre-trained language encoder across each task. We keep all other aspects of the baseline methods the same from their respective authors.

**Inference.** The model does not require language guidance at inference, and the baseline prompt-based methods can be used with their original formulation. The $x_o$ features are extracted and classified with a linear layer.

## 5. Experiments

**Experiment Protocol.** Consistent with previous works[59, 40], we used ViT B/16[10] pre-trained on ImageNet 1k as our Image feature extractor. This is kept frozen during training. On the language side, we use the text transformer of CLIP L/14[43] for our main experiments. We use the Adam optimizer [18] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set the batch size to 24 for Dual Prompt [57] and 16 for L2P [59]. We train on one A100-40GB GPU with the code released by the authors of each method. Input images are resized to $224 \times 224$ and normalized to the range of [0,1]. We follow [4, 59, 57] and train for multiple epochs. For L2P [59] Split CIFAR-100 [20], we train 5 epochs, for L2P [59] Split ImageNet-R [57] we train 50 epochs. We train Dual Prompt [57] for 20 epochs on Split CIFAR-100 [20] and for 50 epochs on Split ImageNet-R [57]. For comparison with state-of-the-art, we use the widely adopted Average accuracy (higher is better) and Forgetting (lower is better) to compare model performance [29, 7, 30]. Since prompt-based continual learning is a very recent development, we use the two most recent baselines Learning to Prompt[59] and Dual Prompt[57] and incorporate LGCL in training. To make the comparison fair, we use the same hyperparameters

| Method | Buffer size | Split CIFAR-100 |  | Buffer size | Split ImageNet-R |  |
|---|---|---|---|---|---|---|
|  |  | Avg. Acc (↑) | Forgetting (↓) |  | Avg. Acc (↑) | Forgetting (↓) |
| ER [8] | 1000 | 67.87±0.57 | 33.33±1.28 | 1000 | 55.13±1.29 | 35.38±0.52 |
| BiC [61] |  | 66.11±1.76 | 35.24±1.64 |  | 52.14±1.08 | 36.70±1.05 |
| GDumb [42] |  | 67.14±0.37 | - |  | 38.32±0.55 | - |
| DER++ [3] |  | 61.06±0.87 | 39.87±0.99 |  | 55.47±1.31 | 34.64±1.50 |
| Co$^2$L [5] |  | 72.15±1.32 | 28.55±1.56 |  | 53.45±1.55 | 37.30±1.81 |
| ER [8] | 5000 | 82.53±0.17 | 16.46±0.25 | 5000 | 65.18±0.40 | 23.31±0.89 |
| BiC [61] |  | 81.42±0.85 | 17.31±1.02 |  | 64.63±1.27 | 22.25±1.73 |
| GDumb [42] |  | 81.67±0.02 | - |  | 65.90±0.28 | - |
| DER++ [3] |  | 83.94±0.34 | 14.55±0.73 |  | 66.73±0.87 | 20.67±1.24 |
| Co$^2$L [5] |  | 82.49±0.89 | 17.48±1.80 |  | 65.90±0.14 | 23.36±0.71 |
| FT-seq | 0 | 33.61±0.85 | 86.87±0.20 | 0 | 28.87±1.36 | 63.80±1.50 |
| EWC [19] |  | 47.01±0.29 | 33.27±1.17 |  | 35.00±0.43 | 56.16±0.88 |
| LwF [25] |  | 60.69±0.63 | 27.77±2.17 |  | 38.54±1.23 | 52.37±0.64 |
| L2P [58] | 0 | 83.86±0.28 | 7.35±0.38 | 0 | 61.57±0.66 | 9.73±0.47 |
| **L2P + LGCL (Ours)** |  | 84.33±0.06 | 5.83±0.23 |  | 62.51±0.05 | 8.9±0.17 |
| DualPrompt | 0 | 86.51±0.33 | 5.16±0.09 | 0 | 68.13±0.49 | 4.68±0.20 |
| **DualPrompt + LGCL (Ours)** |  | **87.23±0.21** | **5.10±0.15** |  | **69.46±0.04** | **4.2±0.06** |
| Upper-bound | - | 90.85±0.12 | - | - | 79.13±0.18 | - |

Table 1: **Results on class incremental learning.** We compare LGCL with baseline and previous methods. Following [57], we group methods by buffer size. Our method is proposed for prompt-based methods like [57, 59] and therefore, require no rehearsal buffer. We observe LGCL outperforms previous baseline methods in Split-ImageNet-R [57] and Split CIFAR-100 [20] consistently.

for L2P [59] and Dual Prompt [57] as provided in their code repositories and paper. We do not perform any hyperparameter optimisation for LGCL. Since our $\mathcal{L}_{task}$ and $\mathcal{L}_{class}$ require negatives from previous tasks, they are used once the first task is learned. We compare with regularization and rehearsal-based methods in Table 1 as these can be trained with the same transformer-based visual encoder. We further compare with architecture-based methods in Table 2. Since these models are trained with different visual encoders, we compare performance against them as a difference from supervised performance.

### 5.1. Datasets

**Split Imagenet-R.** The split ImageNet-R [57] is built on ImageNet-R [13]. It contains 200 classes that are split into 10 disjoint tasks, with each task containing 20 classes. The dataset is divided into 24,000 training images and 6000 test images. Split ImageNet-R [57] has more diversity in the images and is closer to the complicated real-world images.

**Split CIFAR-100.** Split CIFAR-100 is a widely used dataset for continual learning. Split CIFAR-100 is made of 10 disjoint tasks with 10 classes per task taken from the original CIFAR-100 [20]. Compared to Split ImageNet-R, it is a simpler dataset for classification, however, it is sufficient to expose the large forgetting rate of CL methods in class-incremental learning [57].

### 5.2. Comparison with State-of-the Art

We compare with various regularization-based, rehearsal based and prompt-based methods for continual learning in Table 1. We observe that Dual Prompt[57] paired with our model LGCL achieves the best results and sets a new state-of-the-art. We further observe that prompt-based methods significantly outperform regularization-based and rehearsal-based methods on both datasets.

As we compare Dual Prompt with LGCL + Dual Prompt[57], we see that the introduction of language guidance brings a decent improvement. On Split CIFAR-100, LGCL improves Dual Prompt by 0.72% on average accuracy, the measure of average performance across all tasks. Similarly, on Split ImageNet-R, LGCL improves Dual Prompt by an impressive 1.33%.

As we compare the second best method L2P[59], we see that the introduction of LGCL brings similar performance improvements. On Split CIFAR-100, the method sees an improvement of 0.47% in average accuracy and an impressive 1.52% on the forgetting metric. Similarly, on Split ImageNet-R, the method sees an improvement of 0.94% in average accuracy and 0.83% in forgetting. This validates our hypothesis that the introduction of language guidance with LGCL improves model performance without requiring any additional learnable parameters.

We compare the performance of LGCL and prompt-based

| **Method** | **Backbone** | **Avg. Acc (↑)** | **Diff (↓)** | **Buffer size** | **Additional Parameters** | |
|---|---|---|---|---|---|---|
| | | | | | MB | % |
| Upper-bound | ResNet18 | 80.41$^\dagger$ | - | - | - | - |
| SupSup [60] | | 28.34±2.45$^\ddagger$ | 52.07 | 0 | 3.0 | 6.5% |
| DualNet [40] | | 40.14±1.64$^\ddagger$ | 40.27 | 1000 | 5.04 | 10.9% |
| RPSNet [45] | | 68.60$^\dagger$ | 11.81 | 2000 | 181 | 404% |
| DynaER [63] | | 74.64$^\dagger$ | 5.77 | 2000 | 19.8 | 43.8% |
| Upper-bound | ResNet152 | 88.54$^\dagger$ | - | - | - | - |
| DynaER [63] | | 71.01±0.58$^\ddagger$ | 17.53 | 2000 | 159 | 68.5% |
| Upper-bound | ViT-B/16 | 90.85±0.12$^\ddagger$ | - | - | - | - |
| L2P [58] | | 83.86±0.28$^\ddagger$ | 6.99 | 0 | 1.94 | 0.56% |
| L2P + LGCL (Ours) | | 84.33±0.06 | 6.52 | 0 | 1.94 | 0.56% |
| DualPrompt | | 86.51±0.33$^\ddagger$ | 4.34 | 0 | **1.90** | **0.55%** |
| **DualPrompt + LGCL (Ours)** | | **87.23**±**0.21** | **3.45** | 0 | **1.90** | **0.55%** |

$^\dagger$Reported by the original papers. $^\ddagger$ Reproduced using their original codebases.

Table 2: **Comparison with Architecture Based methods on Split-CIFAR-100.** The Upper-Bound denotes the model performance when trained in a fully supervised, non-continual setting, i.e., with access to all tasks at the same time. Following [57], we use `Diff = Upper-Bound Acc - Method Acc` (lower is better). This measures how close the model is to the supervised performance across different model backbones. We observe LGCL outperforms baseline methods and consistently improves the performance of prompt-based continual learning methods.

methods with architecture-based methods in Table 2. These methods are trained on different backbones. To be consistent with previous works[57, 59], we report the difference between supervised performance and the model performance as the metric. We observe from the Table 2 that LGCL again sets a new state-of-the-art in this setting too. LGCL consistently outperforms methods with big buffer sizes. As we compare the performance of Dual Prompt with and without LGCL, we again notice an improvement. LGCL further pushes Dual Prompt towards the upper bound supervised performance with a difference of only 3.45% from the supervised performance.

## 5.3. Ablation on the components of LGCL.

We test each component of our model LGCL on the challenging Split Imagenet-R dataset and report the results in Table 3 for both L2P [59] and Dual Prompt [57]. Comparing rows a) and b), the introduction of $\mathcal{L}_{task}$ leads to a slight improvement in both L2P and Dual Prompt. Comparing rows a) and c), we see a similar conclusion where the introduction of class-level language loss leads to a decent improvement in both datasets. Comparing rows b) and c), we observe that class-level language guidance leads to a bigger improvement than only task-level language guidance. Finally, as we observe from row d), our full model LGCL uses both task-level and class-level language supervision in training and achieves more than a full point improvement on both baseline methods indicating the effectiveness of both modules of our model. We, therefore, conclude that the introduction of both task-level and class-level language guidance is complementary and consistently improves the prompt-based continual learning methods. We once again want to emphasise that this improvement is achieved without introducing any additional learnable parameters.

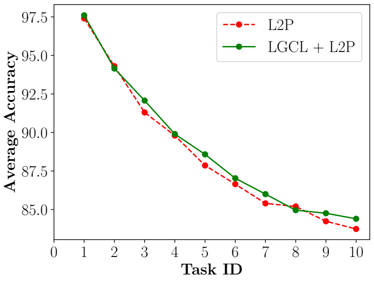

Figure 3: Comparison of average accuracy at each task of L2P [59] + LGCL. We observe that LGCL on average prevents a drop in performance across tasks.

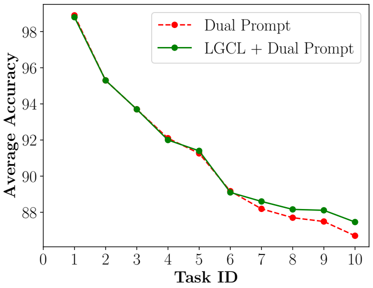

Figure 4: Comparison of average accuracy at each task of Dual Prompt [57] + LGCL. We observe that LGCL on average prevents a drop in performance across tasks.

|  | Components |  | L2P |  | Dual Prompt |  |
|---|---|---|---|---|---|---|
|  | $\mathcal{L}_{task}$ | $\mathcal{L}_{class}$ | Acc | Forg | Acc | Forg |
| a) |  |  | 61.57 | 9.73 | 68.13 | 4.68 |
| b) | ✓ |  | 61.77 | 10.03 | 69.43 | 4.26 |
| c) |  | ✓ | 62.36 | 8.53 | 69.02 | 4.7 |
| d) | ✓ | ✓ | **62.51** | **4.2** | **69.46** | **4.2** |

Table 3: **Ablating over LGCL** on the challenging Split Imagenet-R dataset, we confirm the importance of each component of our model. We conclude that our method benefits from the introduction of language guidance at both the task level and at the class level. This performance improvement is achieved without introducing any additional learning parameters.

| Text Encoder | Avg. Accuracy | Forgetting |
|---|---|---|
| RoBERTa [27] | 87.04 | 5.4 |
| BERT [21] | 87.11 | **4.90** |
| CLIP [43] | **87.23** | 5.10 |

Table 4: **Ablation over different text encoders.** We test our proposed method with CLIP [43], BERT [9] and RoBERTa [27] text encoders. All experiments in this table are conducted on Dual Prompt [57] + LGCL. We observe that CLIP [43] demonstrates the highest performance.

|  | Keys | Avg. Accuracy | Forgetting |
|---|---|---|---|
| a) | Frozen CLIP Keys | 86.15 | **3.93** |
| b) | Learnable Keys | **87.23** | 5.10 |

Table 5: **Ablation over different keys $K_s$.** We replace the keys with CLIP [43] CLS tokens and use our loss function.

### 5.4. Per task performance improvement.

We plot the Average Accuracy of the model through the ten tasks in Figure 3 for L2P and Figure 4 for Dual Prompt on Split CIFAR-100 dataset. As we compare the model performance with and without LGCL on L2P in Figure 3, we see that the model with language guidance is slow in dropping accuracy as each additional task is introduced. We see that performance improvement of introducing language can be observed at each training stage. We similarly compare the performance of Dual Prompt with and without LGCL in Figure 4 and see a similar trend where the introduction of language guidance results in smaller drops in performance as the model is trained for more tasks. This again validates our hypothesis that the introduction of language guidance can mitigate catastrophic forgetting without including any additional trainable parameters.

### 5.5. Ablation on Text Transformer.

In previous experiments, we use the text transformer from a pre-trained CLIP model. CLIP was pre-trained on images and their captions from the internet and therefore learns image-informed text embeddings. In Table 4, we additionally ablate over text transformers from pre-trained language-only models, namely BERT [21] and RoBERTa [27]. We perform this ablation with Dual Prompt + LGCL on Split CIFAR-100. We observe from the table that the CLIP Text Transformer achieves the best result in Average Accuracy since it is pre-trained with both image and text data. However, we see a reasonable performance gain with Language Only pre-trained Text Encoders RoBERTa and BERT. This validates that LGCL is fairly robust to the choice of text encoder.

### 5.6. Ablation on Keys of the Prompt Pool.

We ablate over the design choice for the keys of the prompt pool in Table 5. We ablate using the Split CIFAR-100 dataset with Dual Prompt + LGCL. The keys of the prompt pool are learnable and responsible for selecting the most relevant prompt(s) for the task with a query key lookup from the `CLS` feature of the Image Transformer. Therefore improving the keys can result in performance improvement. We test two different strategies here. In row a), we replace the keys with the CLS tokens from the CLIP Text Transformer and keep them frozen. These keys represent the targets we optimise with our $\mathcal{L}_{task}$. We observe that this, while competitive, does not reflect the performance gains of LGCL over Dual Prompt. In row b) we notice that the learnable keys with our $\mathcal{L}_{task}$ achieve the best performance indicating the effectiveness of our formulation.

## 6. Conclusion.

We introduce a novel perspective of introducing language guidance in prompt-based continual learning in this work. The key intuition behind our approach is that even though the task distributions change between tasks, their label space can be mapped to the same language space. A model that can learn to map to this space can mitigate catastrophic forgetting, leading to performance improvement. We introduce language guidance at two levels; namely task-level and class-level. At task-level, we introduce language guidance for prompt pool, where the model needs to select relevant prompts for class conditioning of a pre-trained vision transformer. By improving the key lookup of the prompt pool, we allow the model to be more robust across different tasks. To this end, we encourage the model to map the keys to its respective task-level language representation. Secondly, we introduce language guidance at the class-level in the output feature of the vision transformer. At this stage, we incentivise the model to map the output feature to the class level language representation. Without any additional learning parameters, our method improves the performance of baseline prompt-based continual learning methods to set a new state-of-the-art.

## References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 1, 2

[2] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Generating visual representations for zero-shot classification. In *ICCV Workshops*, 2017. 3

[3] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020. 2, 6

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *ArXiv*, abs/2004.07211, 2020. 5

[5] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *ICCV*, 2021. 2, 6

[6] Arslan Chaudhry, Albert Gordo, Puneet Kumar Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. *arXiv preprint arXiv:2002.08165*, 2(7), 2020. 2

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 1, 2, 5

[8] 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. 1, 2, 6

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 8

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021. 5

[11] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 3

[12] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *ICRA*, 2019. 1, 2

[13] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329, 2020. 6

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3

[15] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P Xing. Rethinking knowledge graph propagation for zero-shot learning. In *CVPR*, 2019. 3

[16] Zixuan Ke, Bing Liu, and Xingchang Huang. Continual learning of a mixed sequence of similar and dissimilar tasks. *NeurIPS*, 33, 2020. 2

[17] Muhammad Gul Zain Ali Khan, Muhammad Ferjad Naeem, Luc Van Gool, Alain Pagani, Didier Stricker, and Muhammad Zeshan Afzal. Learning attention propagation for compositional zero-shot learning. In *WACV*, 2023. 3

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] 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. 1, 2, 6

[20] Alex Krizhevsky. Learning multiple layers of features from tiny images. In *UoT*, 2009. 5, 6

[21] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942, 2019. 8

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

[23] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *ICML*, pages 3925–3934. PMLR, 2019. 2

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

[25] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2017. 1, 2, 6

[26] 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. *ACM Computing Surveys*, 55(9):1–35, 2023. 3

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 8

[28] Noel Loo, Siddharth Swaroop, and Richard E Turner. Generalized variational continual learning. *arXiv preprint arXiv:2011.12328*, 2020. 2

[29] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *NeurIPS*, 2017. 5

[30] 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. 2, 5

[31] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018. 2

[32] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Learning graph embeddings for open world compositional zero-shot learning. In *T-PAMI*. IEEE, 2022. 3

[33] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1, 2

[34] Muhammad Ferjad Naeem, Muhammad Gul Zain Ali Khan, Yongqin Xian, Muhammad Zeshan Afzal, Didier Stricker, Luc Van Gool, and Federico Tombari. I2mvformer: Large language model generated multi-view document supervision for zero-shot image classification. In *CVPR*, 2023. 3

[35] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *CVPR*, 2021. 3

[36] Muhammad Ferjad Naeem, Yongqin Xian, Luc Van Gool, and Federico Tombari. I2dformer: Learning image to document attention for zero-shot image classification. In *NeurIPS*, 2022. 3

[37] 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. 2

[38] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 3

[39] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020. 3

[40] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *NeurIPS*, 34, 2021. 2, 5, 7

[41] Quang Hong Pham, Chenghao Liu, and Steven C. H. Hoi. Dualnet: Continual learning, fast and slow. In *NeurIPS*, 2021. 3

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

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 3, 5, 8

[44] 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. 3

[45] Jathushan Rajasegaran, Munawar Hayat, Salman H Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. *NeurIPS*, 32, 2019. 7

[46] Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[47] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 2

[48] Karsten Roth, Oriol Vinyals, and Zeynep Akata. Integrating language guidance into vision-based deep metric learning. In *CVPR*, 2022. 3

[49] 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. 2

[50] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, pages 4548–4557, 2018. 2

[51] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proc SIGSAC conference on computer and communications security*, 2015. 2

[52] Richard Socher, Milind Ganjoo, Hamsa Sridhar, Osbert Bastani, Christopher D Manning, and Andrew Y Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013. 3

[53] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020. 3

[54] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018. 3

[55] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *NeurIPS*, 2022. 3

[56] Zifeng Wang, Tong Jian, Kaushik Chowdhury, Yanzhi Wang, Jennifer Dy, and Stratis Ioannidis. Learn-prune-share for lifelong learning. In *ICDM*, 2020. 2

[57] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *ECCV*, 2022. 2, 3, 4, 5, 6, 7, 8

[58] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. *CVPR*, 2022. 6, 7

[59] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Learning to prompt for continual learning. *CVPR*, 2021. 2, 3, 4, 5, 6, 7

[60] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *arXiv preprint arXiv:2006.14769*, 2020. 2, 7

[61] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 2, 6

[62] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. In *ACL*, 2020. 3

[63] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *CVPR*, pages 3014–3023, 2021. 2, 7

[64] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 2

[65] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017. 1, 2

[66] Tingting Zhao, Zifeng Wang, Aria Masoomi, and Jennifer Dy. Deep bayesian unsupervised lifelong learning. *Neural Networks*, 149:95–106, 2022. 2