Enhancing Visual Continual Learning with Language-Guided Supervision

Bolin Ni[1,2*], Hongbo Zhao[1,2*], Chenghao Zhang[1,2], Ke Hu[2]
Gaofeng Meng[1,2,3†], Zhaoxiang Zhang[1,2,3], Shiming Xiang[1,2]

[1]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences.

[2]School of Artificial Intelligence, University of Chinese Academy of Sciences.

[3]Centre for Artificial Intelligence and Robotics, HK Institute of Science & Innovation, Chinese Academy of Sciences.

nibolin2019@ia.ac.cn   gfmeng@nlpr.ia.ac.cn

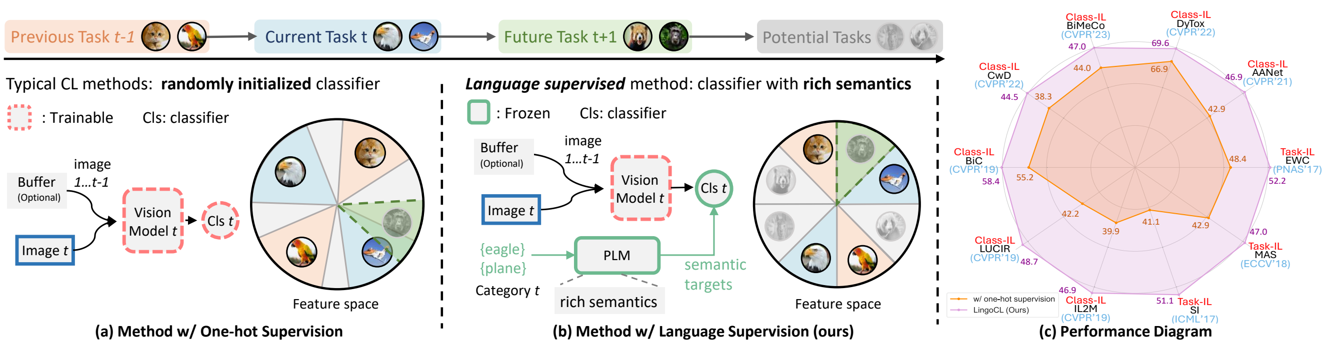


Figure 1. We introduce LingoCL, a simple yet effective continual learning paradigm leveraging language-guided supervision, which can be integrated into most existing approaches seamlessly. (a) Overview of the typical methods which are supervised only by one-hot labels. (b) Overview of the proposed LingoCL which is supervised by semantic targets generated from the pretrained language model. (c) LingoCL is versatile, which significantly enhances the performance of mainstream methods in class-, task- and domain-incremental scenarios.

Abstract

Continual learning (CL) aims to empower models to learn new tasks without forgetting previously acquired knowledge. Most prior works concentrate on the techniques of architectures, replay data, regularization, etc. *However, the category name of each class is largely neglected. Existing methods commonly utilize the one-hot labels and randomly initialize the classifier head. We argue that the scarce semantic information conveyed by the one-hot labels hampers the effective knowledge transfer across tasks. In this paper, we revisit the role of the classifier head within the CL paradigm and replace the classifier with semantic knowledge from pretrained language models (PLMs). Specifically, we use PLMs to generate semantic targets for each class, which are frozen and serve as supervision signals during training. Such targets fully consider the semantic correlation between all classes across tasks. Empirical studies show that our approach mitigates forgetting by alleviating representation drifting and facilitating knowledge transfer across tasks. The proposed method is simple to implement and can seamlessly be plugged into existing methods with negligible adjustments. Extensive experiments based on eleven mainstream baselines demonstrate the effectiveness and generalizability of our approach to various protocols. For example, under the class-incremental learning setting on ImageNet-100, our method significantly improves the Top-1 accuracy by 3.2% to 6.1% while reducing the forgetting rate by 2.6% to 13.1%.*

1. Introduction

The main challenge in continual learning (CL) is *catastrophic forgetting*, where models experience significant performance degradation on earlier tasks when new tasks are introduced. To address this, researchers have developed various strategies, including architecture-based [26, 27, 39, 49], replay-based [3, 35, 40], distillation-based [12, 16, 38], and regularization-based methods [7, 22, 52], alongside other notable contributions [21, 47].

However, most existing approaches overlook the significance of the semantic knowledge contained in category names. The prevailing trend in prior work leans towards

*Equal contribution. †Corresponding author.

using one-hot labels, coupled with the randomly initialized classifier head, and optimizing the encoder and classifier head jointly. Such a methodology is de facto paradigm for stationary environments. Nevertheless, in the CL scenarios, this practice presents two issues. Firstly, the problem of *representation drifting* emerges. When the model encounters new tasks, the feature space could drift or even be overwritten, compromising the *stability* of models. This drift arises because the optimization of the semantic target of each class is narrowly focused on its current task. Due to the limited access to old data and the unpredictability of new data, the model struggles to be compatible with the previous and future classes. For example, as shown in Fig. 1(a), the potential future class "chimpanzee" may erase the feature space of the learned class "plane", exacerbating the forgetting of old tasks. Secondly, this particularity of data in CL also results in *inefficient knowledge transfer*. Since the semantic targets in the classifier are randomly initialized without any prior knowledge, and are then optimized within individual tasks, it struggles to capture the semantic correlation across all tasks. This incompleteness in semantic correlations impedes the model's knowledge transfer, thereby affecting its *plasticity*.

In this work, we study how to enhance CL performance by leveraging the semantic knowledge in category names from a classifier perspective. Inspired by the impressive generalization capabilities of pretrained language models (PLMs) [5, 36], we propose a simple yet effective approach, language-guided supervision for CL (LingoCL), which employs PLM to generate the semantic targets. Specifically, for the incoming task, we first use the category name of each class as input to the language model and take the outputs as the weights in the classifier. Then, the classifier is kept frozen during CL training, guiding the learning of the encoder. Our approach is motivated by the rich knowledge and strong generalization abilities of PLMs. Even with the limitations in previous and future data, PLMs ensure that each generated semantic target implicitly considers the semantic correlations between all classes. Therefore, these targets can be used to direct the learning of the encoder. For instance, as illustrated in Fig. 1(b), PLMs can provide the prior knowledge that the "eagle" in current tasks shares a similar semantic target with the learned "parrot" class, facilitating the knowledge transfer between the learned classes and new classes. We explore two types of language models in this work: self-supervised models on unimodal data and vision-supervised models on multimodal data. Our results demonstrate that both types of models can serve as excellent classifier heads, constantly improving performance. Moreover, the analysis in Sec. 3.3 demonstrates that the improvements come from alleviating the representation drift and facilitating knowledge transfer, instead of the individual gains at each task.

Each row of the classifier's weights represents the semantic target for its corresponding class.

Without loss of generality, we choose eleven methods as baselines and incorporate the text-supervised classifiers for them. Comprehensive experiments demonstrate the proposed methods are generally effective. In particular, under the class-incremental learning setting, LingoCL can improve the accuracy on ImageNet-100 by 3.2% to 6.1%, and reduce the forgetting rate by 2.6% to 13.1%. In task- and domain-incremental learning, LingoCL improves the accuracy by 3.9% to 9.7% and 1.2% to 4.0%, respectively.

The contributions can be summarized as follows:

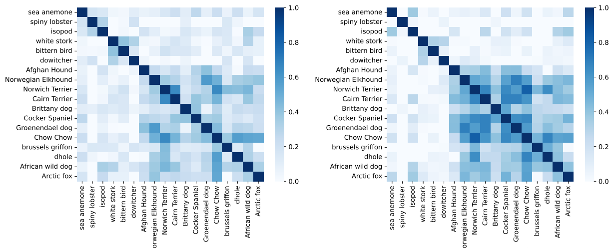
- We point out that the semantic knowledge in category names is largely neglected by existing methods when initializing classifiers, which could have two issues, *i.e.*, representation drifting and insufficient knowledge transfer.
- We propose LingoCL, a new CL paradigm with language-guided supervision. With the rich semantic knowledge in PLMs, we alleviate the abovementioned issues and thus enhance the performance of mainstream CL methods.
- The proposed LingoCL has several key advantages: 1) computation efficiency; 2) orthogonality to existing methods; 3) flexibility with various PLMs; and 4) versatility in diverse CL scenarios. Extensive experiments are conducted to systematically examine our method.

2. Related Work

Continual learning. To alleviate catastrophic forgetting, researchers have explored various routes. *Regulation-based methods* [2, 7, 22, 52] aim to prevent catastrophic forgetting by penalizing the changes of network parameters when learning current tasks. *Replay-based methods* entail selecting a subset of data from previous tasks [3, 35, 54] or using generative models to produce synthetic data [18, 20, 34] as "replayed" data to preserve the knowledge of previous tasks. *Distillation-based methods* take the model trained on the previous task as the teacher to supervise the learning of the current model. These methods can be divided into logits distillation [38, 48], feature distillation [12, 16], and relational distillation [41, 42]. *Architecture-based methods* involve dynamic allocation of different parameters for each task through architecture expansion [13, 26, 39, 49] or mask operation [29, 30]. *Rectification-based methods* analyzes the abnormal behaviors in CL models compared to oracle models and tries to rectify them. These methods usually focus on the imbalance in the feature embedding [4, 27, 40] or network weights [6, 48, 53].

Most existing methods commonly use one-hot labels coupled with randomly initialized classifiers, ignoring the category names seriously. In contrast to them, our work studies whether and how to improve CL by leveraging the semantic information contained in the category names.

Cross-modality adaptation. In recent years, transferring language knowledge to visual modeling has emerged as a new paradigm. For example, contrastive language-image



(a) w/ one-hot supervision (b) w/ language supervision (ours)

Figure 2. Comparison of the inter-class correlation maps. LingoCL facilitates more efficient knowledge transfer among similar classes.

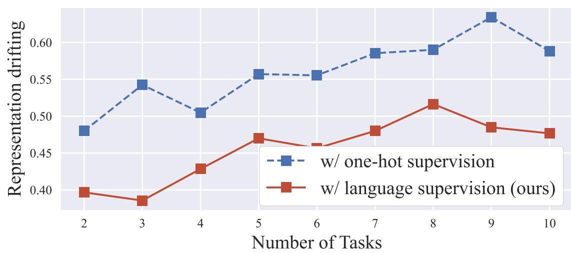


Figure 3. Quantitative analysis of representation drifting on ImageNet-100 with 10 tasks. LingoCL effectively alleviates the representation drifting in the CL process.

pretraining demonstrates impressive "zero-shot" transfer and generalization capacities [19, 36, 51]. Moreover, some works explore how to model the vision input using pretrained language models in order to transfer the ability of language models [1, 24, 43]. Another line of work focuses on how to improve the vision encoder with the guidance of the language information. For instance, Tex [45] proposes to use language models to reduce the bias in the classifier of fine-tuned visual models. DUET [8] integrates the latent semantic knowledge from PLMs to vision models for better zero-shot recognition ability. Additionally, Lei *et al*. [25] designed a suite of evaluation tasks across various perception aspects and showed that language models can learn visual features from vast amounts of data, including shape, texture, and color, and that vision supervision can enhance the comprehension of visual concepts. In this work, we are pioneering the exploration of how to transfer knowledge in language models to address the catastrophic forgetting issue in continual learning.

3. Methodology

3.1. Revisiting Classifier in Existing CL Paradigms

We first review the typical CL paradigms from a classifier perspective. In CL scenarios, the vision encoder, denoted by g_V, is sequentially optimized over tasks. Each task typically requires an individual classifier head. For t-th task, we symbolize the training dataset as $\mathcal{D}_t = \{(\boldsymbol{x}_t, y_t)\}$ that contains C_t disjoint classes and the classifier head as $\mathbf{W}_t \in \mathbb{R}^{C_t \times d}$. The vision encoder g_V and semantic targets in $\mathbf{W}_t$ are optimized jointly, following the learning objective in the stationary environment:

$$g_V^*, \mathbf{W}_t^* = \underset{\Theta_V, \mathbf{W}_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}_t, y_t \sim \mathcal{D}_t} \big[\mathcal{L}(\operatorname{sim}(\mathbf{W}_t, g_V(\boldsymbol{x}_t)), y_t)\big], \tag{1}$$

where $\mathbf{W}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. The function $\operatorname{sim}(\mathbf{W}, g_V(\boldsymbol{x}_t))$ calculates the similarity between the image embedding $g_\theta(\boldsymbol{x})$, and every semantic target in $\mathbf{W}_t$, using the inner product.

However, this approach encounters challenges specific to CL. Firstly, CL benefits from knowledge transfer across tasks, such as forward and backward transfer, but randomly initialized classifiers struggle to capture the semantic similarity among classes across tasks, resulting in *inefficient knowledge transfer*. Secondly, the optimization of each semantic target is confined to its current task. This narrow focus overlooks the broader compatibility between semantic targets spanning all tasks. Such a shortsighted learning approach can cause conflicts between the semantic targets of different tasks, inducing *representation drifting or erasure* in the feature space.

3.2. Our Proposed Language-Guided Supervision

To address these issues, we utilize the rich semantic knowledge contained in pretrained language models to guide the learning process for each task. Specifically, for an incoming task t, the procedure is as follows:

(i) Gathering the category names $[\boldsymbol{l}_1, \cdots, \boldsymbol{l}_{C_t}]$ of task t.

(ii) Feeding these category names into PLM to generate the semantic targets for the classifier $\tilde{\mathbf{W}}_t$:

$$\tilde{\mathbf{W}}_t = g_T([\boldsymbol{l}_1, \cdots, \boldsymbol{l}_{C_t}]), \tag{2}$$

(iii) Optimizing the vision encoder g_V, while keeping the classifier $\tilde{\mathbf{W}}_t$ *frozen*:

$$g_V^* = \underset{\Theta_V}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}_t, y_t \sim \mathcal{D}_t} \big[\mathcal{L}(\operatorname{sim}(\tilde{\mathbf{W}}_t, g_V(\boldsymbol{x}_t)), y_t)\big]. \tag{3}$$

The classifier $\tilde{\mathbf{W}}_t$ is kept frozen to preserve the semantic knowledge from being disturbed or forgotten in the CL process. As these generated semantic targets are optimized with sufficient data and concepts, they effectively serve as supervision signals, directing the vision encoder's optimization.

In light of the outlined methodology, LingoCL has several key advantages: 1) it is *computation-efficient*; leveraging category names requires only a single forward propagation, with a negligible cost comparison to overall training; 2) it provides flexibility to utilize knowledge from *various* language models, promoting easy integration of the latest PLM advancements; 3) it is *orthogonal* to most of existing CL methods, allowing for seamless integration; 4) it is *versatile*, and compatible with diverse CL scenarios such as class-, task- and domain-IL.

Method	Backbone	*B=10, C=10*			*B=5, C=5*			*B=2, C=2*		
		Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
Oracle	ResNet-18	77.6	77.6	-	77.6	77.6	-	77.6	77.6	-
w/ LingoCL		78.0	78.0	-	78.0	78.0	-	78.0	78.0	-
Architecture-based methods										
AANet [27]	ResNet-18	64.6±0.2	49.1±0.2	-	62.5±0.3	42.5±0.3	-	57.7±0.5	37.6±0.5	-
w/ LingoCL		**65.4**±0.1	**50.4**±0.1	-	**63.2**±0.2	**44.5**±0.4	-	**58.7**±0.4	**38.6**±0.5	-
		(+0.8)	**(+1.3)**		**(+0.7)**	**(+2.0)**		**(+1.0)**	**(+1.0)**	
DyTox [13]	ConViT	69.5±0.0	52.8±0.2	33.0±0.0	67.4±0.1	48.1±0.3	37.8±0.0	64.5±0.2	44.8±0.3	41.3±0.1
w/ LingoCL		**71.9**±0.0	**58.9**±0.1	**24.9**±0.0	**70.0**±0.1	**52.3**±0.3	**30.5**±0.1	**65.9**±0.1	**46.3**±0.3	**36.1**±0.2
		(+2.4)	**(+6.1)**	**(-8.1)**	**(+2.6)**	**(+4.2)**	**(-7.3)**	**(+1.4)**	**(+1.5)**	**(-5.2)**
BiMeCo [33]	MobileNet-V2†	59.7±0.5	45.9±0.5	35.4±0.6	52.5±0.4	38.7±0.3	43.5±0.7	37.2±1.0	25.9±1.3	50.7±1.1
w/ LingoCL		**60.2**±0.2	**46.4**±0.1	**32.3**±0.1	**54.2**±0.7	**39.1**±0.6	**41.8**±0.5	**43.4**±0.9	**32.5**±0.8	**45.0**±0.9
		(+0.5)	**(+0.5)**	**(-3.1)**	**(+1.7)**	**(+0.4)**	**(-1.7)**	**(+6.2)**	**(+6.6)**	**(-5.7)**
Distillation-based methods										
LUCIR [16]	ResNet-18	60.2±0.4	46.5±0.7	37.3±0.5	54.8±0.7	41.7±1.0	42.0±0.5	45.6±1.1	36.2±1.6	44.5±0.9
w/ LingoCL		**61.9**±0.3	**47.5**±0.6	**36.5**±0.1	**56.3**±0.5	**44.3**±0.4	**39.8**±0.8	**46.8**±1.2	**37.0**±0.6	**42.3**±1.0
		(+1.7)	**(+1.0)**	**(-0.8)**	**(+1.5)**	**(+2.6)**	**(-2.2)**	**(+1.2)**	**(+0.8)**	**(-2.2)**
BiC [48]	ResNet-18	57.8±0.9	41.2±1.0	26.7±1.0	50.1±0.6	34.7±0.1	28.7±0.7	38.1±1.0	23.6±0.4	38.6±0.9
w/ LingoCL		**60.1**±0.5	**43.4**±1.0	**20.5**±1.0	**51.6**±0.6	**36.6**±1.3	**19.6**±0.6	**44.9**±1.0	**31.1**±0.8	**29.7**±0.6
		(+2.3)	**(+2.2)**	**(-6.2)**	**(+1.5)**	**(+1.9)**	**(-9.1)**	**(+6.8)**	**(+7.5)**	**(-8.9)**
Rectification-based methods										
CwD [40]	ResNet-18	60.0±0.5	46.7±0.5	36.9±0.9	54.4±0.6	42.2±0.5	41.5±0.4	40.2±1.2	34.0±1.2	44.6±0.2
w/ LingoCL		**60.6**±0.7	**47.6**±1.1	**35.3**±1.0	**55.7**±0.9	**44.3**±0.5	**39.2**±0.7	**46.0**±0.7	**38.4**±0.8	**36.9**±0.8
		(+0.6)	**(+0.9)**	**(-1.3)**	**(+1.3)**	**(+2.1)**	**(-2.3)**	**(+5.8)**	**(+4.4)**	**(-7.7)**
IL2M [4]	ResNet-18	57.8±0.3	44.3±0.8	41.0±0.3	52.6±0.8	40.5±0.4	45.3±1.0	44.0±1.1	34.2±0.8	48.5±0.5
w/ LingoCL		**62.1**±0.0	**48.1**±1.0	**37.8**±0.3	**56.6**±0.4	**44.0**±1.0	**43.0**±1.2	**48.0**±0.7	**39.6**±0.0	**42.8**±0.2
		(+4.3)	**(+3.8)**	**(-3.2)**	**(+4.0)**	**(+3.5)**	**(-2.3)**	**(+4.0)**	**(+5.4)**	**(-5.7)**

Table 1. Results on class-incremental experiments on CIFAR100 of Average accuracy (%), last phase accuracy (%) and forgetting rate $\mathcal{F}$ (%) with and without text-supervised classifier at various CL settings. B denotes the number of classes at the initial task, and C denotes the number of classes in each task after the initial one. † denotes a modified version of the backbone as adapted by the original authors. Notably, for each metric, ↑ (↓) indicates that the larger (the smaller) values, the better results are.

3.3. Quantitative Analysis

Next, we examine our method to answer the two questions mentioned above: *1) Does our method alleviate representation drifting,* and *2) Does it facilitate knowledge transfer?*

To answer the first question, we perform a subspace analysis [37] on challenging class-incremental learning protocol. Given the same input, let $\mathbf{F}_t, \mathbf{F}_{t'} \in \mathbb{R}^{n\times d}$ denote the output of the encoder after the t-th task and after the t'-th task ($t' > t$), respectively. $\mathbf{V}_{k,t}$ and $\mathbf{V}_{k,t'}$ are the top-k principal directions of $\mathbf{F}_t$ and $\mathbf{F}_{t'}$, respectively. The representation drifting from the t-th task to the t'-th can be defined as:

$$\mathrm{RepreDrift}_k(\mathbf{F}_t, \mathbf{F}_{t'}) = 1 - \frac{1}{k}\|\mathbf{V}_{k,t}^T\mathbf{V}_{k,t'}\|_F^2, \quad (4)$$

where a smaller value indicates less representation drifting. We adopt LUCIR [16] as the baseline. Fig. 3 shows the evolution of the first task's representations as the training progresses. LingoCL significantly reduces the representation drifting, demonstrating the capacity to enhance the stability of the CL model.

For the second question, we sample the first 18 classes in ImageNet-100 and calculate the inter-class correlation of the embeddings produced by the encoder with vanilla classifier and LingoCL. Note that these classes are scattered among different tasks. Fig. 2 shows that LingoCL exhibits certain inter-class correlations, indicating that a well-considered semantic target can facilitate knowledge transfer and improve the performance of CL. The above analyses offer an initial demonstration of the effectiveness of LingoCL. A more in-depth exploration of these issues is presented in Tab. 3.

4. Experiments

4.1. Experimental Setup

Continual learning protocols. We evaluate LingoCL on four common CL protocols, including: *class-incremental learning (CIL)*, *general few-shot class-incremental learning*, *task-incremental learning*, and *domain incremental learning*.
Datasets. We use CIFAR100 [23] for task-IL, both CIFAR100 and ImageNet-100 [38] for class-IL, and Office-Home [44] for domain-IL. More details about datasets are shown in the *supplementary material.*
Architecture. We employ MobileNetV2 for BiMeCo [33] and ResNet18 [15] for other CNN-based methods. For ViT-

Method	*B=50, C=10*			*B=50, C=5*			*B=10, C=10*			*B=5, C=5*		
	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
Oracle	80.6	80.6	-	80.6	80.6	-	80.6	80.6	-	80.6	80.6	-
w/ LingoCL	80.6	80.6	-	80.6	80.6	-	80.6	80.6	-	80.6	80.6	-
Architecture-based methods												
AANet [27]	75.3±1.0	66.2±0.7	-	72.7±0.7	61.2±1.3	-	57.8±0.0	42.9±1.2	-	48.5±1.1	37.9±1.3	-
w/ LingoCL	**75.7**±0.6	**68.7**±0.5	-	**72.9**±1.1	**61.7**±0.9	-	**62.3**±0.4	**46.9**±0.3	-	**50.1**±1.4	**41.3**±0.7	-
	(+0.4)	**(+2.5)**	-	**(+0.2)**	**(+0.5)**	-	**(+4.5)**	**(+4.0)**	-	**(+1.6)**	**(+3.4)**	-
DyTox [13]	79.8±0.4	72.5±1.1	10.8±0.3	75.5±1.1	65.4±1.3	15.3±1.4	78.1±0.0	66.9±0.8	15.6±0.8	75.4±0.9	61.3±1.3	22.9±0.3
w/ LingoCL	**80.6**±0.2	**72.8**±0.2	**6.9**±1.2	**76.7**±0.4	**66.8**±0.6	**12.3**±0.2	**79.5**±1.0	**69.6**±0.9	**12.3**±1.0	**76.2**±0.3	**63.0**±1.4	**18.3**±0.5
	(+0.8)	**(+0.3)**	**(-3.9)**	**(+1.2)**	**(+1.4)**	**(-3.0)**	**(+1.4)**	**(+2.7)**	**(-3.3)**	**(+0.8)**	**(+1.7)**	**(-4.6)**
BiMeCo [33]	71.2±0.2	60.7±0.5	15.8±0.3	68.9±0.1	59.7±0.1	22.5±0.4	59.0±0.1	44.0±0.2	41.1±0.2	47.6±0.3	35.5±0.5	49.0±0.6
w/ LingoCL	**73.0**±0.1	**63.4**±0.3	**11.8**±0.3	**71.1**±0.2	**63.1**±0.2	**18.4**±0.3	**60.0**±0.7	**47.0**±0.5	**38.6**±0.2	**49.8**±0.1	**39.4**±0.2	**45.1**±0.2
	(+1.8)	**(+2.7)**	**(-4.0)**	**(+2.2)**	**(+3.4)**	**(-4.1)**	**(+1.0)**	**(+3.0)**	**(-2.5)**	**(+2.2)**	**(+3.9)**	**(-3.9)**
Distillation-based methods												
LUCIR [16]	70.2±0.0	59.7±1.0	21.4±0.7	67.7±0.8	56.7±0.2	23.3±1.4	57.1±0.8	42.2±0.4	44.5±0.7	47.5±0.5	35.5±1.0	48.5±1.5
w/ LingoCL	**73.4**±1.1	**66.0**±1.3	**8.3**±0.9	**71.5**±0.6	**62.3**±1.3	**10.3**±1.0	**62.2**±0.9	**48.7**±1.4	**38.6**±0.7	**53.6**±1.0	**41.9**±1.4	**45.9**±0.8
	(+3.2)	**(+6.3)**	**(-13.1)**	**(+3.8)**	**(+5.6)**	**(-13.0)**	**(+5.1)**	**(+6.5)**	**(-5.9)**	**(+6.1)**	**(+6.4)**	**(-2.6)**
BiC [48]	72.0±0.7	63.6±0.3	7.2±1.4	68.1±1.4	57.5±0.9	7.1±0.8	66.4±1.0	55.2±1.0	15.1±0.8	59.0±0.5	44.8±0.7	20.9±0.7
w/ LingoCL	**73.8**±0.9	**65.8**±0.1	**6.3**±1.0	**72.8**±0.4	**63.9**±0.4	**4.1**±0.4	**68.9**±0.5	**58.4**±1.5	**14.3**±1.0	**61.9**±0.1	**49.6**±0.4	**18.8**±1.2
	(+1.8)	**(+2.2)**	**(-0.9)**	**(+4.7)**	**(+6.4)**	**(-3.0)**	**(+2.5)**	**(+3.2)**	**(-0.8)**	**(+2.9)**	**(+4.8)**	**(-2.1)**
Rectification-based methods												
CwD [40]	72.0±1.1	62.2±1.3	21.4±0.3	69.2±1.3	59.5±0.5	24.6±0.8	53.6±0.0	38.3±0.6	46.4±0.7	38.9±0.4	26.8±1.1	55.0±1.1
w/ LingoCL	**73.4**±1.1	**65.6**±0.8	**10.9**±1.2	**71.8**±0.3	**62.8**±1.3	**15.6**±0.2	**55.6**±1.3	**41.5**±1.0	**43.0**±0.6	**41.0**±1.3	**29.9**±0.4	**53.0**±0.5
	(+1.4)	**(+3.4)**	**(-10.5)**	**(+2.6)**	**(+3.3)**	**(-9.0)**	**(+2.0)**	**(+3.2)**	**(-3.4)**	**(+2.1)**	**(+3.1)**	**(-2.0)**
IL2M [4]	67.5±1.4	54.6±0.7	30.0±0.4	63.6±0.8	50.8±0.9	33.7±0.0	55.0±1.4	39.9±0.1	50.2±0.1	46.4±0.3	34.6±0.0	50.1±0.0
w/ LingoCL	**71.7**±0.6	**60.9**±0.9	**23.8**±1.2	**69.1**±0.5	**57.4**±1.3	**26.6**±1.3	**59.9**±1.3	**46.9**±0.3	**43.5**±0.3	**51.7**±0.4	**41.4**±1.2	**48.0**±0.5
	(+4.2)	**(+6.3)**	**(-6.2)**	**(+5.5)**	**(+6.6)**	**(-7.1)**	**(+4.9)**	**(+7.0)**	**(-6.7)**	**(+5.3)**	**(+6.8)**	**(-2.1)**

Table 2. Results on class-incremental experiments on ImageNet-100.

based methods, such as DyTox [13], we follow the original implementation and use ConViT [14]. As for the pretrained language model, we utilize the text transformer in CLIP-B/32 [36] pretrained on WIT-400M [36]. Results about more language models are explored in Tab. 6.

Metrics. Following [10, 32], LingoCL is extensively evaluated by three metrics: last-step accuracy (Last), average incremental accuracy (Avg), and forgetting rate ($\mathcal{F}$).

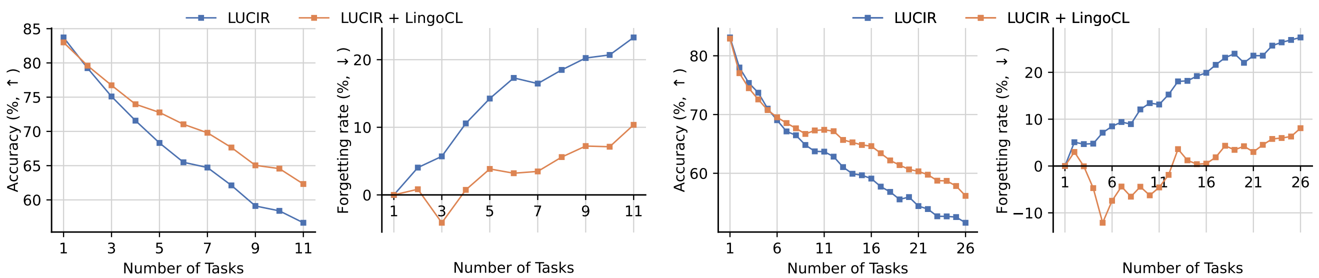
Baselines. We comprehensively evaluate the effectiveness of the proposed method on eleven baselines, spanning various continual learning approaches. These include distillation-based methods such as LUCIR [16] and BiC [48], architecture-based methods like DyTox [13] and AANet [27], rehearsal-based methods such as GEM [28], regularization-based methods including EWC [22], MAS [2], and SI [52], and rectification-based methods like IL2M [4] and CwD [40]. To ensure a fair comparison, we implement all the baseline methods using their officially released code or the widely recognized CL library [17, 31] in the research community and keep their original hyperparameters unchanged.

4.2. Class-incremental Learning Experiments

Benchmark protocol. We denote the number of classes in the initial task by B and the number of new classes learned per task after the initial one by C. We adopt two popular protocols [38, 48]: 1) $B = 50$: where the initial task covers half of the total number of classes and the remaining classes are equally divided among the subsequent tasks, and (2) $C = B$, where each task within the data stream involves an equal number of classes. The memory size for each class is set to 20 in all datasets. All approaches are evaluated under the same class order [16, 27, 38] for fair comparison.

Implementation details. We follow the original hyperparameters of all methods. We select exemplars as memory based on the herding strategy following previous works [38]. Detailed hyperparameters are in the *supplementary material*.

Results. We conduct extensive experiments by incorporating our method into various baselines of different routes. Tab. 2 and Tab. 1 present the results on ImageNet-100 and CIFAR100, respectively, which illustrate that our method consistently and significantly improves all metrics. Taking LUCIR on ImageNet-100 as an example, our method improves the average accuracy by $3.2\% \sim 6.1\%$ across different settings. Importantly, our method barely impacts the performance of the oracle model, implying that the performance gains stem from reducing forgetting instead of the individual gains at each task. This is evidenced by the significant reductions in forgetting rate by $2.6\% \sim 13.1\%$.



(a) ImageNet-100 (B=50, C=5) (b) ImageNet-100 (B=50, C=2)

Figure 4. The evolution curve of accuracy and forgetting rate for each task on class-incremental experiments on ImageNet-100. Significantly, LingoCL exhibits negative forgetting, i.e., the learning of subsequent tasks leads to improved performance on prior tasks. This phenomenon evidences LingoCL's effective facilitation of knowledge transfer.

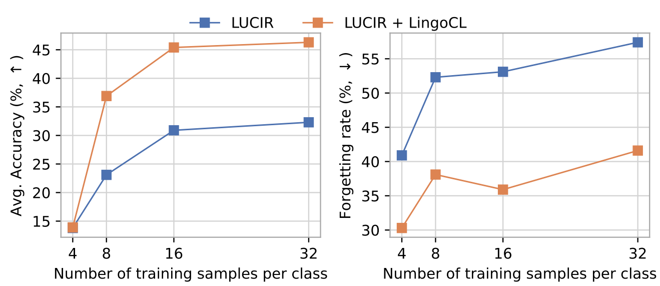


Figure 5. Results on general few-shot class-incremental learning.

Moreover, we observe that the gains in last accuracy are usually larger than that in average accuracy, indicating that our method benefits more on long task sequences which are commonly more challenging. This observation is also supported by the experimental results. For instance, when the number of task increases from 6 to 11 (B=50,C=10 to B=50,C=5), the gains of CwD and LUCIR increase from 1.4% and 3.2% to 2.6% and 3.8%, respectively.

Finally, in addition to the quantitative results, Fig. 4 displays the accuracy and forgetting rate curves for LUCIR in long sequence settings. Our method achieves a smoother forgetting rate curve, with a gradual and consistent improvement in accuracy at each task. More importantly, our method even achieves negative forgetting rate, indicating that learning the later tasks helps to improve the accuracy of the previous ones. We attribute these gains to the fact that our method utilizes the semantic similarity among classes to guide the CL process, which promotes backward knowledge transfer.

4.3. General Few-shot Class-IL Experiments

Benchmark protocol. General few-shot CIL is a more realistic setting where the initial task has sufficient training data to initialize the model, while the subsequent tasks only have K training samples per class. No rehearsal buffer is available. In this study, we use ImageNet-100 as the benchmark dataset with $B = 50$ and $C = 10$ settings.
Implementation details. K is set to 4/8/16/32 in our experiments. We choose LUCIR [16] as the baseline.
Results. Due to the scarcity of data, few-shot CIL requires the model to not only overcome forgetting but also transfer as much learned knowledge as possible. As shown in Fig. 5, our proposed method shows greater improvements in this challenging setting. Specifically, when K is 32, our method achieves an improvement in accuracy of 14.0% and a reduction in the forgetting rate of 15.8%. This demonstrates that our method achieves more effective knowledge transfer from the initial well-learned task by pre-allocating the semantic target for each class. We also observe that when K is 4, although the gain of accuracy is marginal, the forgetting rate is reduced by 10.6%. It demonstrates that our method can alleviate the representation drifting in the feature space. See *supplementary material* for more results.

4.4. Task-incremental Learning Experiments

Benchmark protocol. The benchmark in this study is 10-split-CIFAR100, which involves dividing CIFAR100 into 10 tasks with non-overlapping classes. Following [46, 47], the last accuracy and forgetting rate are reported.
Implementation details. The learning rate is 1e-4 and epochs is 80. More detailed hyperparameters are shown in the *supplementary material*.
Results. As shown in Tab. 4, our method improves the accuracy by $3.9\% \sim 9.7\%$ and reduces the forgetting by $2.2\% \sim 9.6\%$. Additionally, we observe that rehearsal-free methods can be competitive with rehearsal-based methods. SI [52] with our method achieves 51.1% accuracy, surpassing the Rehearsal baseline by 3.0%. This finding suggests that our approach facilitates the effective use of learned knowledge, thus mitigating the reliance on old data.

4.5. Domain-incremental Learning Experiments

Benchmark protocol. OfficeHome [44] comprises four different domains, each treated as a distinct task. As the label set is consistent across all tasks, a shared classifier is utilized. We report the last accuracy and forgetting rate.

Init.	Trainable	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
Random	✓	57.1%	42.2%	44.5%
PLM	✓	60.3%	47.8%	42.0%
PLM	✗	**62.2%**	**48.7%**	**38.6%**

(a) Study on the effects of frozen weights.

Corr. in weights	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
Learned	57.1%	42.2%	44.5%
Orthogonal	60.2%	47.5%	42.6%
PLM	**62.2%**	**48.7%**	**38.6%**

(b) Study on the effects of semantic correlation.

Sup. signal	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
One-hot	57.1%	42.2%	44.5%
Oracle	61.6%	**49.7%**	39.2%
PLM	**62.2%**	48.7%	**38.6%**

(c) Study on different supervision signal.

Table 3. Component analysis. The first cell (gray) represents the baseline with a vanilla classifier which is trainable, randomly initialized, and supervised by one-hot label. Default settings are marked in gray. “Init.”: Initialization. “Sup.”: Supervision. “Corr.”: Correlation.

Method	Task-IL		Domain-IL	
	Last (↑)	$\mathcal{F}$ (↓)	Last (↑)	$\mathcal{F}$ (↓)
Oracle	77.6	-	99.1	-
w/ LingoCL	78.0	-	99.1	-
Regularization-based methods				
EWC [22]	41.5±1.9	45.2±1.7	48.4±1.7	45.9±0.4
w/ LingoCL	**45.4**±0.5	**42.5**±1.2	**52.2**±0.9	**39.9**±0.4
	(+3.9)	**(-2.7)**	**(+3.8)**	**(-6.0)**
MAS [2]	42.9±0.5	44.5±1.3	54.1±1.3	41.9±0.5
w/ LingoCL	**47.0**±1.5	**42.3**±0.4	**58.1**±2.0	**36.0**±1.1
	(+4.1)	**(-2.2)**	**(+4.0)**	**(-5.9)**
GEM* [28]	-	-	58.5±0.8	38.4±1.5
w/ LingoCL	-	-	**59.7**±0.5	**34.6**±0.8
			(+1.2)	**(-3.8)**
SI [52]	41.4±1.7	45.0±1.2	53.2±1.8	43.0±1.1
w/ LingoCL	**51.1**±1.5	**35.4**±0.8	**56.7**±0.7	**35.8**±1.9
	(+9.7)	**(-9.6)**	**(+3.5)**	**(-7.2)**

Table 4. Task-incremental and domain-incremental results. The method using an extra data buffer is marked with *.

Implementation details. The regularization coefficients of EWC, MAS, SI and GEM are set to 100, 0.1, 0.3 and 5, respectively. More details are in the *supplementary material*.
Results. Tab. 4 reports that our method improves accuracy by 1.2% $\sim$ 4.0%, while simultaneously reducing the forgetting rate by 3.8% $\sim$ 7.6%. Due to the variability of the image domains, the semantic targets often shift or are limited to the current domains only. In contrast, the semantic targets generated by PLMs can utilize the rich source of domain knowledge in PLMs, ensuring a more representative distribution of these targets.

4.6. Analysis and Ablation

In this subsection, we conduct comprehensive ablation studies and analyses to systematically examine LingoCL. Unless stated otherwise, the experiments are based on LUCIR [16] and ImageNet-100 (B=10, C=10).
Analysis of freezing the language-guided classifier. We delve into two pivotal components in the design of the language-guided classifier: 1) freezing the weights, and 2) the semantic correlation in the weights. We first analyze the effect of freezing the weights in Tab. 3a. The comparison between updating and freezing weights reveals that updates lead to a decrease in accuracy (from 61.7% to 60.3%). This performance drop is attributed to catastrophic forgetting in semantic targets, triggered by updating weights for each task. It highlights the necessity of preserving the semantic knowledge sourced from pretrained language models. However, it’s also notable that even with updated weights, performance exceeds that of random initialization, suggesting that strong initialization with rich semantics plays a crucial role in CL.

Analysis of the semantic correlation in the classifier. Furthermore, we ablate the semantic correlation in the classifier. By orthogonalizing the semantic targets output of the pretrained language model, we construct a classifier that removes semantic correlations among classes. The orthogonal classifier is kept frozen during training. As indicated in Tab. 3b, the removal of semantic correlation leads to a 2.0% decrease in accuracy and a 4.0% increase in the forgetting rate. Nevertheless, the orthogonal classifier still surpasses traditional vanilla classifiers by 3.1% in accuracy. This suggests that the frozen, orthogonal targets help to reduce interference between different tasks, thereby diminishing feature drift in the feature space. On the other hand, the absence of semantic correlation appears to impede knowledge transfer across tasks. These findings underscore the dual significance of maintaining a frozen state and preserving semantic correlation in the classifier.

Comparison with oracle classifier. To thoroughly assess the impact of our language-guided supervision, we introduce an oracle classifier as a benchmark for oracle supervision. Initially, an idealized oracle model is trained with data from all tasks, typically considered the performance upper bound in CL. Subsequently, we replace the baseline model’s vanilla classifier with this oracle classifier, which remains frozen during training. As shown in Tab. 3c, our language-guided classifier not only matches but surpasses the oracle classifier in both average accuracy and forgetting rate. This superiority is likely attributable to the fact that the dataset for pretraining language models is conceptually more diverse and sufficient than that used for the oracle model, providing semantically richer targets for each class. These results highlight the exceptional efficacy of our approach.

Comparison with logits rectification-based methods. Tab. 5 presents a comparison of LingoCL with other meth-

Method	Forward compatible	Semantic correlation	Avg (↑)	$\mathcal{F}$ (↓)
Baseline			63.0%	29.9%
w/ BiC [48]	✗	✗	65.4%	25.1%
w/ E2E [48]	✗	✗	65.0%	26.1%
w/ Div. head [48]	✓	✗	63.7%	27.2%
w/ LingoCL	✓	✓	**67.5%**	**22.5%**

Table 5. Comparison with logits rectification-based methods.

Method	Pretraing data	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
Baseline		65.9%	55.8%	24.9%
multimodal pretraining				
CLIP [36]	WIT-400M	67.5%	57.1%	22.5%
OpenCLIP [9]	LAION-2B	68.0%	57.9%	22.0%
unimodal pretraining				
BERT [11]	3.3B	66.6%	57.1%	23.8%
XLNet [50]	32.8B	67.0%	57.1%	22.8%

Table 6. Ablation study on the pretrained language models.

ods that modify the classifier to address anomalies. We use a simple CIL baseline with rehearsal and distillation on CIFAR100 under the setting of B=50, C=10. BiC [48] addresses classifier bias by adding an extra linear layer, while EEIL [6] finetunes the classifier using balanced data. Divergence head [13] utilizes an additional classifier to separate the features of old and new tasks to preserve the feature space for future classes. Existing methods mainly focus on addressing the compatibility with old tasks using statistical corrections, whereas our method stands out by considering the semantic correlation among all classes, including the past and the future. Notably, LingoCL does not entirely conflict with these methods; in fact, LingoCL can complement it to further enhance performance, It is evidenced in Tab. 1 and Tab. 2, where LingoCL notably enhances the efficacy of BiC.

Ablation on different language models. In Tab. 6, we explore two types of language models: multimodal pretraining models and unimodal pretraining models. The overall results indicate that the multimodal pretraining language models perform better, which we attribute to the pretraining aligned with images allowing the language models to learn more semantic information from visual cues. Although the semantic targets generated by the unimodal pretraining models are not aligned with images, they still can be easily fitted with trainable vision encoders. Furthermore, we found that increasing the amount of pretraining data can effectively improve performance (67.5% → 68.0%, 66.6% → 67.0%), as the language model learns more concepts.

Effect of the number of exemplars for replay. We investigate the effect of the number of old exemplars on model performance. The results in Fig. 6 show that LingoCL can consistently improve the accuracy and reduce the forgetting rate under all settings, especially when the number of re-

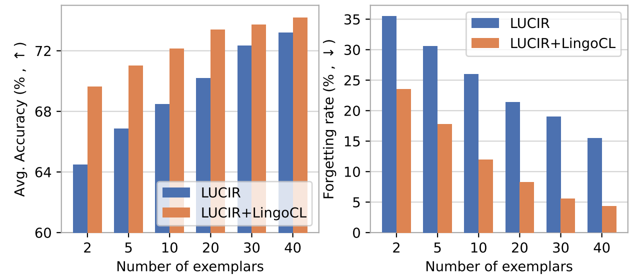


Figure 6. Effect of the number of exemplars for replay.

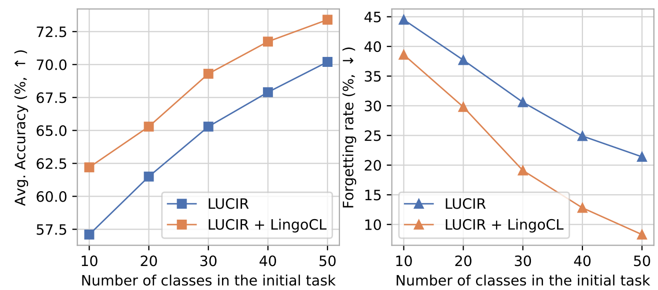


Figure 7. Impact of the number of classes in the initial task.

served exemplars is quite small. Notably, integrated with LingoCL, the baseline with reserving only 2 exemplars per class is comparable to the vanilla version by utilizing 20 exemplars per class (69.6% *v.s.* 70.2%), further verifying the power of the proposed LingoCL.

Impact of the number of classes in the initial task. In this ablation, we discuss the effect of the number of classes learned in the initial task. As shown in Fig. 7, the X-axis represents the number of classes in the initial task, and the remaining classes are incremented with 10 classes per task. We can observe that LingoCL can bring a $+3.2\% \sim 5.1\%$ acreage accuracy improvement and reduce the forgetting rate by $+5.9\% \sim 13.1\%$, which illustrates the effectiveness and robustness of our method.

5. Conclusion

In this work, we present a new perspective on CL, *i.e.*, how to utilize the semantic knowledge in category names. Specifically, we use pretrained language models to generate the semantic target for each class. Empirical study shows that our method alleviates the representation drifting and facilitates knowledge transfer. Extensive experiments across various scenarios demonstrate the effectiveness of our method.

Acknowledgements. This work was supported partially by the National Natural Science Foundations of China (Grants No.62376267, 62076242), the Pre-Research Project on Civil Aerospace Technologies (No.D030312), the National Defense Basic Scientific Research Program of China(No.JCKY2021203B063) and the innoHK project.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 3

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018. 2, 5, 7

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019. 1, 2

[4] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *CVPR*, pages 583–592, 2019. 2, 4, 5

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 2

[6] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018. 2, 8

[7] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, pages 532–547, 2018. 1, 2

[8] Zhuo Chen, Yufeng Huang, Jiaoyan Chen, Yuxia Geng, Wen Zhang, Yin Fang, Jeff Z Pan, and Huajun Chen. Duet: Cross-modal semantic grounding for contrastive zero-shot learning. In *AAAI*, pages 405–413, 2023. 3

[9] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 8

[10] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE TPAMI*, 44(7):3366–3385, 2021. 5

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 8

[12] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020. 1, 2

[13] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *CVPR*, 2022. 2, 4, 5, 8

[14] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *ICML*, pages 2286–2296. PMLR, 2021. 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[16] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pages 831–839, 2019. 1, 2, 4, 5, 6, 7

[17] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. In *NeurIPS Workshop*, 2018. 5

[18] Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao Tao, Dongyan Zhao, Jinwen Ma, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *ICLR*, 2019. 2

[19] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916. PMLR, 2021. 3

[20] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018. 2

[21] Muhammad Gul Zain Ali Khan, Muhammad Ferjad Naeem, Luc Van Gool, Didier Stricker, Federico Tombari, and Muhammad Zeshan Afzal. Introducing language guidance in prompt-based continual learning. In *ICCV*, pages 11463–11473, 2023. 1

[22] 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, 5, 7

[23] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 4

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 3

[25] Lei Li, Jingjing Xu, Qingxiu Dong Dong, Ce Zheng, Qi Liu, Lingpeng Kong, and Xu Sun. What does vision supervision bring to language models? a case study of clip. *OpenReview*, 2022. 3

[26] 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. 1, 2

[27] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *CVPR*, pages 2544–2553, 2021. 1, 2, 4, 5

[28] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 5, 7

[29] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773, 2018. 2

[30] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, pages 67–82, 2018. 2

[31] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE TPAMI*, pages 1–20, 2022. 5

[32] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE TPAMI*, 45(5):5513–5533, 2022. 5

[33] Xing Nie, Shixiong Xu, Xiyan Liu, Gaofeng Meng, Chunlei Huo, and Shiming Xiang. Bilateral memory consolidation for continual learning. In *CVPR*, pages 16026–16035, 2023. 4, 5

[34] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *CVPR*, pages 11321–11329, 2019. 2

[35] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, pages 524–540, 2020. 1, 2

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 3, 5, 8

[37] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *ICLR*, 2021. 4

[38] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542, 2017. 1, 2, 4, 5

[39] 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. 1, 2

[40] Yujun Shi, Kuangqi Zhou, Jian Liang, Zihang Jiang, Jiashi Feng, Philip HS Torr, Song Bai, and Vincent YF Tan. Mimicking the oracle: an initial phase decorrelation approach for class incremental learning. In *CVPR*, pages 16722–16731, 2022. 1, 2, 4, 5

[41] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *ECCV*, pages 254–270, 2020. 2

[42] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, pages 12183–12192, 2020. 2

[43] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *NeurIPS*, pages 200–212, 2021. 3

[44] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017. 4, 6

[45] Junyang Wang, Yuanhong Xu, Juhua Hu, Ming Yan, Jitao Sang, and Qi Qian. Improved visual fine-tuning with natural language supervision. In *ICCV*, 2023. 3

[46] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *CVPR*, pages 184–193, 2021. 6

[47] 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. In *CVPR*, pages 139–149, 2022. 1, 6

[48] 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, 4, 5, 8

[49] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *CVPR*, pages 3014–3023, 2021. 1, 2

[50] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*, 32, 2019. 8

[51] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 3

[52] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995, 2017. 1, 2, 5, 6, 7

[53] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shutao Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, pages 13205–13214, 2020. 2

[54] Hongbo Zhao, Bolin Ni, Haochen Wang, Junsong Fan, Fei Zhu, Yuxi Wang, Yuntao Chen, Gaofeng Meng, and Zhaoxiang Zhang. Continual forgetting for pre-trained vision models. *arXiv preprint arXiv:2403.11530*, 2024. 2