Enhancing Visual Continual Learning with Language-Guided Supervision
——— Supplementary Materials ———

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

In this supplementary material, we provide additional details regarding the main manuscript. More specifically:

- In Sec. 1, we provide further explanation of the datasets, protocols and metrics.
- In Sec. 2, we provide the detailed hyperparameters of different continual learning settings.
- In Sec. 3, we provide additional experiments and results.

1. Datasets, Protocols and Metrics

In Sec. 1.1, we present the statistical information for the datasets used in our experiments. In Sec. 1.2, we describe the continual learning protocols that are commonly used in the literature. Finally, in Sec. 1.3, we introduce the evaluation metrics used to measure the performance comprehensively.

1.1. Datasets statistics

- *Split-CIFAR-100 (Task-IL)*. The CIFAR100 dataset [8] comprises 60,000 32×32 images belonging to 100 classes. In task-incremental learning setting, Split-CIFAR-100 splits the original CIFAR-100 [8] into 10 tasks, 10 disjoint classes per task.

- *CIFAR-100 (Class-IL)*. In the class-incremental learning setting, we divide the classes into mutually exclusive sets. The first task consists of B classes, and each subsequent task consists of C classes.

- *ImageNet-100 (Class-IL)*. ImageNet-100 [13] is the subset of ImageNet1000 [3] containing 100 classes [13]. These classes are selected from the first 100 classes after a random shuffle with seed 1,993 [19]. Each image is represented by 224×224 pixels.

- *OfficeHome (Domain-IL)*. OfficeHome [16] consists of images from four different domains: Artistic images, Clip Art, Product images and Real-World images. For each domain, the dataset contains images of 65 object categories found typically in Office and Home settings. Each image is represented by 224×224 pixels.

We use the official categories provided by the respective dataset creators for all datasets, which can be accessed through the dataset resources [3, 8, 16]. These categories are also presented in Fig. 1, Fig. 3, and Fig. 2 for CIFAR100, ImageNet100, and OfficeHome, respectively.

1.2. Continual Learning Protocols

In continual learning (CL), the model is trained in a task-by-task manner. We define a sequence of tasks denoted by $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$. The t-th task, denoted by $\mathcal{D}_t = \{(\boldsymbol{x}_i^t, y_i^t)\}_{i=1}^{n_t}$, comprises tuples consisting of an input sample $\boldsymbol{x}^t \in \mathcal{X}_t$ and its corresponding label $y^t \in \mathcal{Y}_t$. Depending on the target set and the number of training samples, CL protocols can be divided into four common categories:

- *Task-incremental learning* where the target set of test sample $\boldsymbol{x}^t$ is $\mathcal{Y}_t$.
- *Class-incremental learning* where the target set of test sample $\boldsymbol{x}^t$ is $\cup_{i=1}^{t}\mathcal{Y}_i$.
- *Few-shot Class-incremental learning* where the target set of test sample $\boldsymbol{x}^t$ is $\cup_{i=1}^{t}\mathcal{Y}_i$ and the $n_t(t>1)$ of training set is limited.
- *Domain-incremental learning* where each task shares the same target set, *i.e.*, $\mathcal{Y}_1 = \mathcal{Y}_2 = \cdots = \mathcal{Y}_T$.

1.3. Evaluation Metrics

Formally, suppose the model is conducted for N tasks and let $A_{i.j}$ denote the classification accuracy evaluated on the test set of the task i after the incremental learning of the j-th task is $A_{i.j}$. Our method is extensively evaluated by three commonly used metrics:

⋆Equal contribution. †Corresponding author.

Method	*B=50, C=10*			*B=50, C=5*			*B=50, C=2*		
	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
Oracle	77.6	77.6	-	77.6	77.6	-	77.6	77.6	-
w/ Ours	78.0	78.0	-	78.0	78.0	-	78.0	78.0	-
Distillation-based methods									
LUCIR [5]	65.9±1.7	55.8±1.7	24.9±1.7	60.9±1.1	51.4±0.4	26.7±1.5	52.9±0.5	42.5±1.5	34.0±0.6
w/ Ours	**67.5**±0.8	**57.1**±1.5	**22.5**±0.6	**62.1**±1.9	**53.2**±2.0	**22.4**±1.6	**53.5**±1.2	**43.4**±1.6	**31.2**±0.5
	(+1.6)±1.2	**(+1.3)**	**(-2.4)**	**(+1.2)**	**(+1.8)**	**(-4.3)**	**(+0.6)**	**(+0.9)**	**(-2.8)**
BiC [17]	63.5±0.9	51.2±0.5	16.2±1.0	57.0±0.1	45.0±0.9	15.9±1.0	44.6±0.0	33.2±0.8	11.1±0.7
w/ Ours	**64.7**±0.9	**52.4**±0.8	**11.8**±1.0	**58.6**±0.5	**46.3**±0.9	**14.5**±0.8	**46.7**±0.6	**35.1**±0.9	**6.3**±0.8
	(+1.2)	**(+1.2)**	**(-4.4)**	**(+1.6)**	**(+1.3)**	**(-1.4)**	**(+2.1)**	**(+1.9)**	**(-4.8)**
Rectification-based methods									
CwD [15]	66.9±0.3	57.4±0.8	23.4±0.9	62.3±0.8	52.5±0.7	26.9±0.6	56.3±0.4	44.7±0.8	36.2±0.5
w/ Ours	**68.0**±0.4	**58.4**±0.2	**22.8**±0.9	**63.3**±1.0	**53.6**±0.5	**26.0**±0.5	**58.3**±0.2	**46.1**±0.5	**34.1**±0.3
	(+1.1)	**(+1.0)**	**(-0.6)**	**(+1.0)**	**(+1.1)**	**(-0.9)**	**(+2.0)**	**(+1.4)**	**(-2.1)**
IL2M [2]	65.7±0.1	55.9±0.3	25.2±0.7	59.9±0.6	49.9±0.1	29.7±0.2	52.5±0.8	42.0±0.3	35.3±0.6
w/ Ours	**68.5**±0.2	**59.2**±0.6	**21.2**±0.2	**60.8**±0.9	**49.8**±0.7	**29.4**±0.5	**54.0**±0.4	**45.7**±0.5	**29.2**±0.3
	(+2.8)	**(+3.3)**	**(-4.0)**	**(+0.9)**	**(-0.1)**	**(-0.3)**	**(+1.5)**	**(+3.7)**	**(-6.1)**

Table 1. Results on class-incremental experiments on CIFAR100 of Average accuracy (%), last phase accuracy (%) and forgetting rate $\mathcal{F}$ (%) with and without language-guided representation at various CL settings. B denotes the number of classes at initial task, C denotes the number of classes in each task after the initial one.

Method	*K=4*			*K=8*			*K=16*			*K=32*		
	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
Buffer size = 0												
LUCIR [5]	13.9±0.4	7.3±0.6	40.9±0.4	23.1±0.8	10.7±1.0	52.3±0.6	30.9±0.3	12.8±0.1	53.1±0.1	32.3±0.3	12.9±0.8	57.4±0.8
w/ Ours	**13.9**±0.2	**6.0**±0.9	**30.2**±0.5	**36.9**±0.4	**12.7**±0.5	**38.1**±0.3	**45.4**±0.8	**20.0**±0.9	**35.9**±0.2	**46.3**±0.6	**20.4**±0.5	**41.6**±0.1
	(+0.0)	**(-1.3)**	**(-10.7)**	**(+13.8)**	**(+2.0)**	**(-14.2)**	**(+14.5)**	**(+7.2)**	**(-17.2)**	**(+14.0)**	**(+7.5)**	**(-15.8)**
Buffer size = 1												
LUCIR [5]	39.1±0.2	15.2±0.3	29.1±0.9	40.2±0.1	18.7±0.5	35.0±0.4	31.1±0.8	19.2±0.4	31.7±0.4	38.7±0.9	23.9±0.0	37.9±0.3
w/ Ours	**41.6**±0.2	**19.3**±0.9	**10.6**±0.2	**44.9**±0.8	**21.8**±0.8	**13.3**±0.2	**38.9**±0.3	**24.7**±0.2	**13.8**±0.5	**44.9**±0.6	**29.2**±0.3	**18.9**±0.9
	(+2.5)	**(+4.1)**	**(-18.5)**	**(+4.7)**	**(+3.1)**	**(-21.7)**	**(+7.8)**	**(+5.5)**	**(-17.9)**	**(+6.2)**	**(+5.3)**	**(-19.0)**

Table 2. Results on few-shot class-incremental experiments on ImageNet100 under $B = 50, C = 10$.

- *Last-step accuracy (Last)* which measures the overall performance at last:

$$Last = \frac{1}{N}\sum_{i=1}^{N} A_{i,N} \tag{1}$$

- *Average incremental accuracy (Avg)* which measure the performance evolution along the learning trajectory:

$$Avg = \frac{1}{N}\sum_{j=1}^{N}(\frac{1}{j}\sum_{i=1}^{j} A_{i,j}) \tag{2}$$

- *Forgetting rate (Forget)* which measures the degree of forgetting on learned tasks:

$$Forget = \frac{1}{N-1}\sum_{i=1}^{N-1} \max\{A_{i,1}, \cdots, A_{i,N-1}\} - A_{i,N} \tag{3}$$

Besides, following [12], we perform a subspace similarity analysis to measure the representation drifting. Given the input from the same task, 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. We compute the PCA decomposition of $\mathbf{F}_t$, *i.e.*, the eigenvectors $(v_1, v_2, \cdots)$ of $\mathbf{F}_t^\top \mathbf{F}_t$. Let $\mathbf{V}_{k,t}$ are the top-k principal directions of $\mathbf{F}_t$, and $\mathbf{V}_{k,t'}$ the corresponding matrix for $\mathbf{F}_{t'}$. The representation drifting from the t-th task to the t'-th can be defined as:

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

$\frac{1}{k}\|\mathbf{V}_{k,t}^T \mathbf{V}_{k,t'}\|_F^2$ measures the similarity of the subspaces spanned by $\mathbf{F}_t$ and $\mathbf{F}_{t'}$. The smaller the similarity between the subspaces at task t and t', the greater the representation drifting.

2. Hyperparameter details

We provide the detailed hyperparameters of class-incremental learning, task-incremental learning, and domain-incremental experiments in Sec. 2.1, Sec. 2.2 and Sec. 2.3, respectively.

#	Template	Avg (↑)	Last (↑)	$\mathcal{F}$ (↓)
1	Baseline	65.9%	55.8%	24.9%
2	`{object}`	67.5%	57.1%	22.5%
3	`a photo of a {object}`	67.5%	57.5%	22.5%
4	Templates ensemble [11]	67.2%	57.9%	21.7%

Table 3. Comparison with different prompting techniques on CIFAR-100 under class-incremental setting $B = 50, C = 10$.

2.1. Class-incremental learning

For CNN-based methods [2, 5, 9, 15, 17], we employ the SGD optimizer [14] with an initial learning rate of 0.1, a momentum of 0.9, and a batch size of 128. In the experiments performed on CIFAR100, all models are trained for 160 epochs within each task, with the learning rate decreased by a factor of 10 at the 80-th and 120-th epochs. For ImageNet100, all models are trained for 90 epochs within each task, with the learning rate reduced by a factor of 10 at the 30-th and 60-th epochs.

For ViT-based methods such as DyTox [4], we follow the original hyperparameters. We train the model for 500 epochs per task with Adam [6] with a learning rate of 5e-4, including 5 epochs of warmup. At the end of each task (except the first), we finetune the model for 20 epochs with a learning rate of 5e-5 on a balanced dataset.

2.2. Task-incremental learning

The learning rate starts from 1e-4 and decays at epochs 30 and 60 with a multiplier of 0.1. The total epochs are 80. The batch size is set to 32. The regularization coefficient of EWC [7], MAS [1] and SI [18] are set to 100, 0.1 and 10, respectively.

2.3. Domain-incremental learning

We use the Adam [6] optimizer with an initial learning rate 0.001, and a batch size of 128. The epochs are 80 and the learning rate is decay by 10 at the 40-th and 60-th epochs. The regularization coefficients of EWC [7], MAS [1], SI [18] and GEM [10] are set to 100, 0.1, 0.3 and 5, respectively.

3. Additional Experiments Analysis

In Sec. 3.1 and Sec. 3.2, we present additional results on class-incremental learning and few-shot class-incremental learning experiments, respectively. Moreover, in Sec. 3.3, we offer an analysis of the prompting technique.

3.1. Class-incremental Learning

In the main manuscript, Table 1 presents the results of class-incremental learning (CIL) experiments on CIFAR100 under the setting where the number of base classes (B) equals the number of incremental classes (C). To provide further insights, we supplement additional results under the setting where $B = 50$ in Tab. 1. The results show that our proposed method consistently and significantly improves the performance across all metrics under the $B = 50$ setting. These results provide further evidence of the effectiveness of our approach in various CIL settings.

3.2. Few-shot Class-incremental Learning

The results of few-shot class-incremental learning are displayed in Figure 7 of the main manuscript. To provide more quantitative results, we also present them in Tab. 2. It is evident that our proposed approach consistently achieves significant performance gains, with or without buffers. These findings provide further evidence that our method facilitates effective knowledge transfer from the initial well-learned task.

3.3. Prompting Technique

Prompting [11] is a widely used technique to transfer knowledge from pretrained language models. In Tab. 3, we compare three different settings for prompting. In setting #1, we used the category name as input without any additional templates. In setting #2, we used the template `a photo of a {object}`. Finally, in setting #3 [11], we averaged the results of 80 different templates. Our results show that the use of templates can slightly ease the forgetting, which we attribute to the fact that the template ensemble enhances the stability of the generated features and reduces the effect of noise. These findings highlight the robustness and generalizability of our approach.

apple	clock	lion	plate	pepper
aquarium	cloud	lizard	poppy	table
fish	cockroach	lobster	porcupine	tank
baby	couch	man	possum	telephone
bear	crab	maple	rabbit	television
beaver	crocodile	tree	raccoon	tiger
bed	cup	motorcycle	ray	tractor
bee	dinosaur	mountain	road	train
beetle	dolphin	mouse	rocket	trout
bicycle	elephant	mushroom	rose	tulip
bottle	flatfish	oak	sea	turtle
bowl	forest	tree	seal	wardrobe
boy	fox	orange	shark	whale
bridge	girl	orchid	shrew	willow
bus	hamster	otter	skunk	tree
butterfly	house	palm	skyscraper	wolf
camel	kangaroo	tree	snail	woman
can	computer	pear	snake	worm
castle	keyboard	pickup	spider	
caterpillar	lamp	truck	squirrel	
cattle	lawn	pine	streetcar	
chair	mower	tree	sunflower	
chimpanzee	leopard	plain	sweet	

Figure 1. The categories of CIFAR100.

Alarm Clock	Eraser	Monitor	Screwdriver
Backpack	Exit Sign	Mop	Shelf
Batteries	Fan	Mouse	Sink
Bed	File Cabinet	Mug	Sneakers
Bike	Flipflops	Notebook	Soda
Bottle	Flowers	Oven	Speaker
Bucket	Folder	Pan	Spoon
Calculator	Fork	Paper Clip	TV
Calendar	Glasses	Pen	Table
Candles	Hammer	Pencil	Telephone
Chair	Helmet	Postit Notes	ToothBrush
Clipboards	Kettle	Printer	Toys
Computer	Keyboard	Push Pin	Trash Can
Couch	Knives	Radio	Webcam
Curtains	Lamp Shade	Refrigerator	
Desk Lamp	Laptop	Ruler	
Drill	Marker	Scissors	

Figure 2. The categories of OfficeHome.

eastern hog-nosed snake	silver salmon
rooster	remote control
wardrobe	chain mail
corkscrew	swim trunks / shorts
isopod	white stork
beaver	teddy bear
acorn	moped
goldfinch	horse chestnut seed
Siamese cat	holster
chiffonier	ping-pong ball
bittern bird	purse
screw	indigo bunting
Cairn Terrier	wolf spider
valley	lighthouse
lens cap	sturgeon
Brittany dog	toaster
Appenzeller Sennenhund	Arctic fox
entertainment center	doormat
Greater Swiss Mountain Dog	southern black widow
Band-Aid	high-speed train
dhole	vending machine
sea anemone	cricket insect
ice cream	longhorn beetle
threshing machine	African rock python
bell or wind chime	red wine
sunglasses	assault rifle
can opener	carbonara
microphone	CRT monitor
quail	candy store
brussels griffon	academic gown
computer keyboard	cannon
hand-held computer	music speaker
eel	African wild dog
Norwegian Elkhound	farm plow
mailbox	koala
leopard	crutch
mitten	Groenendael dog
Cocker Spaniel	Norwich Terrier
split-rail fence	cardboard box / carton
dowitcher	combination lock
tennis ball	candle
Afghan Hound	Windsor tie
parking meter	pan flute
snow leopard	rose hip
spiny lobster	small white butterfly
monarch butterfly	space shuttle
hook	Chow Chow
drumstick	wool
toilet paper	ring binder
sawmill	alligator lizard

Figure 3. The categories of ImageNet100.

References

[1] 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. 3

[2] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *CVPR*, pages 583–592, 2019. 2, 3

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[4] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *CVPR*, 2022. 3

[5] 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. 2, 3

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[7] 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. 3

[8] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 1

[9] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *CVPR*, pages 2544–2553, 2021. 3

[10] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 3

[11] 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. 3

[12] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *ICLR*, 2021. 2

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

[14] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. 3

[15] 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. 2, 3

[16] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for

unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017. 1

[17] 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, 3

[18] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995, 2017. 3

[19] Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Practical block-wise neural network architecture generation. In *CVPR*, pages 2423–2432, 2018. 1