Learning from Itself: Mining Internal Knowledge from Vision Language Models for Continual Learning

Supplementary Material

This appendix provides comprehensive supplementary materials to support the main paper. We present detailed implementation specifications, extended experimental results, and thorough ablation studies that validate our approach. Sec. A describes the external caption generation methodology used for baseline comparisons. Sec. B provides complete experimental configurations including dataset specifications, baseline implementations, and computational infrastructure. Sec. C presents additional benchmark results across diverse continual learning protocols and model architectures. Sec. D contains extensive ablation studies examining hyperparameter sensitivity, compatibility with parameter-efficient methods, and scaling properties. These supplementary materials ensure full reproducibility of our work and provide deeper insights into the behavior of Learning from Itself across various experimental conditions.

A. Details on External Caption Generation

This appendix provides comprehensive implementation details for the external caption generation experiments discussed in the main text. These experiments serve as a comparative baseline to evaluate the effectiveness of our self-generated tokens against descriptive captions produced by external multimodal language models.

Caption generation methodology. We employ Qwen3-VL-8B-Instruct [21], an open-weight multimodal language model, to generate descriptive captions for each image given its corresponding class label. To ensure reproducibility and standardization of results, we utilize the following structured prompt template:

```
System: Write {num_captions} captions for
        the given image and its class label.
Follow COCO Captions style:
{examples_str}

Guidelines:
- Be descriptive but concise (10-20 words
    per caption)
- Focus on visual attributes, context, and
    scene
- Incorporate the class label naturally
- Describe what you see: colors, positions,
    background, actions
- Vary the captions when generating multiple
    ones
- Use natural, simple language

User: Class label: {class_name}
{image}
```

We set `num_captions` to 5 by default to generate multiple diverse descriptions per image, though only the first caption is utilized during training to maintain consistency. The `examples_str` field contains 10 randomly sampled captions from the COCO Captions dataset [2] (including train2014, val2014, train2017, and val2017 splits) to provide stylistic reference. The default reference captions are:

```
'The interior of a bathroom on an airplane'
'there is a orange train coming up the
 tracks'
'A breakfast sandwich with a fruit cup .'
'A man and a woman camping on a beach.'
'Someone surfing waves in the ocean on their
 surf board'
'A man in a Mexican hat is riding a horse in
 the street.'
'An empty room with a hard floor, with a
 toilet and other objects in the background.'
'A horse standing on the ground behind a
 fence.'
'A bathroom with a black door and a white
 tub'
'A young boy on a wake board in a lake.'
```

Generation process and quality control. We leverage the structured output capabilities of the model to ensure consistent parsing of generated captions. The generation process follows a deterministic seeding strategy: we initialize with seed 42 for each trial and increment sequentially (43, 44, 45, ...) if the returned number of captions does not match our requirement. This iterative process continues until exactly `num_captions` valid captions are obtained.

Inference infrastructure. To ensure reproducible results and eliminate variability from different inference providers, we deploy the model using a self-hosted infrastructure. Specifically, we utilize:

- *Model variant*: Qwen3-VL-8B-Instruct-FP8[1] (officially quantized version)
- *Inference engine*: vLLM 0.11.0 [12]
- *Software stack*: PyTorch [13] 2.8.0 with CUDA 12.6
- *Hardware*: Single NVIDIA RTX 4090 (24GB VRAM)
- *Context length*: 36K tokens (maximum capacity within memory constraints)
- *Quantization*: FP8 quantization for improved efficiency with negligible performance degradation

[1] Hugging Face ID: `Qwen/Qwen3-VL-8B-Instruct-FP8`

- *Sampling parameters*: Default chat template with standard sampling configuration

Representative caption examples. The following examples illustrate the quality and descriptive nature of generated captions:

```
Dataset: CIFAR-100
Image ID: 2
Class label: apple
Generated Captions:
1. "A red apple with a green stem sits on a
    wooden surface, slightly out of focus."
2. "Blurry image of a shiny red apple
    resting on a light-colored wooden table."
3. "A single apple, mostly red with green
    top, appears soft-focused on a wooden
    background."
4. "Red apple with a hint of green, captured
    in a shallow depth-of-field shot on wood."
5. "A round, red apple with a stem, resting
    on a wooden surface with a soft, blurred
    look."

Dataset: CUB-200
Image ID: Black_Footed_Albatross_0009_34.jpg
Class label: black-footed albatross
Generated Captions:
1. "A black-footed albatross soars
    gracefully over the open ocean with
    wings spread wide."
2. "This black-footed albatross glides
    effortlessly above the calm sea,
    showcasing its dark plumage."
3. "In mid-flight, the black-footed
    albatross displays its long wings
    against the vast ocean backdrop."
4. "The black-footed albatross flies low
    over the water, its silhouette sharp
    against the gray sky."
5. "A majestic black-footed albatross in
    flight, wings extended, above the
    rippling ocean surface."
```

Training integration. During continual learning, these external captions are incorporated through an auxiliary CLIP contrastive loss, identical to the formulation used for our self-generated tokens. This ensures a fair comparison between external caption-based and self-generated token-based approaches, with the only difference being the source of the descriptive text—external multimodal models versus internal CLIP knowledge mining.

B. Additional Experimental Details

This section provides comprehensive specifications for all experimental configurations referenced in the main paper. We detail the exact experimental setups used for introductory analyses, complete dataset descriptions with class distributions and splitting protocols, baseline method implementations including reproduction details and source attributions, and the computational infrastructure employed throughout our experiments. These details ensure complete reproducibility and clarify the experimental conditions under which our results were obtained.

B.1. Configuration for Introductory Analysis

All empirical analyses presented in Fig. 1 and Fig. 2 utilize OpenAI CLIP ViT-B/16 as the base architecture. For the distribution gap analysis in Fig. 1, we compute CLIP loss and similarity metrics using the original pretrained model without any adaptation. The contrastive loss is calculated with a batch size of 64. For the COCO Captions [2] reference dataset, which contains multiple textual descriptions per image, we aggregate the text embeddings by computing their mean representation to establish one-to-one image-text correspondences. The continual learning performance with external captions is evaluated by incorporating an auxiliary CLIP contrastive loss between the generated captions and their corresponding images during training.

For the performance comparison analysis in Fig. 2, we conduct evaluations in two phases. First, we assess zero-shot performance using SimpleCIL for vision-only classification (constructing prototypical classifiers from pretrained features) and standard CLIP for vision-language inference. Subsequently, we evaluate post-training performance: SimpleCIL with training involves standard finetuning after prototypical classifier initialization, with classifiers reconstructed from the original pretrained vision encoder at each new task arrival. For CLIP training, we perform full model finetuning with all parameters unfrozen except the temperature scalar.

B.2. Dataset Specifications

Our experimental evaluation encompasses six diverse visual recognition benchmarks:

CIFAR-100 [11]: A fundamental benchmark comprising 100 object categories with 600 32×32 pixel images per class (500 training, 100 testing). Despite its low resolution, CIFAR-100 remains challenging for continual learning due to significant inter-class visual similarity and intra-class variation.

ImageNet-R [5]: A robustness benchmark containing 30,000 images across 200 ImageNet classes, featuring various artistic renditions including paintings, cartoons, sculptures, and sketches. This dataset tests model generalization under significant domain shift from natural photographic images.

ImageNet-100 [3]: A carefully curated 100-class subset of ImageNet-1K, following the standard split from the `continuum` library [4]. Each class contains approximately 1,300 training images and 50 validation images at 224×224

Table A1. Continual learning results on OpenCLIP ViT-B/16 with 6-task protocol across ImageNet-R, Stanford Cars, and Food-101.

Method	ImageNet-R		Stanford Cars		Food-101	
	last	*avg*	*last*	*avg*	*last*	*avg*
L2P [20]	66.77	72.82	65.64	76.37	73.13	80.42
DualPrompt [19]	67.58	73.22	67.55	76.88	72.75	80.00
CODA-P [16]	68.05	73.71	64.19	75.06	74.13	80.98
SimpleCIL [22]	74.05	76.37	86.95	89.17	85.73	88.13
Cont-CLIP [17]	74.53	76.49	79.69	82.49	85.42	87.59
Finetune	82.49	84.64	86.49	89.33	89.02	91.48
CoOp [25]	77.67	80.19	81.48	84.87	85.39	88.87
LoRA [6]	81.78	84.70	86.67	89.44	88.18	91.02
RAPF [7]	70.23	76.10	63.19	75.87	81.17	85.53
PROOF [24]	80.30	82.32	89.54	90.53	84.74	87.52
ENGINE [23]	80.98	83.63	90.03	91.61	83.94	86.89
LfI (ours)	**84.12**	**85.23**	**91.92**	**92.66**	**89.93**	**91.89**
Joint	86.07	—	92.69	—	91.87	—

resolution, providing a balanced evaluation of large-scale visual recognition.

CUB-200-2011 [18]: A fine-grained classification dataset containing 11,788 images of 200 North American bird species. The dataset poses unique challenges with high intra-class variation and subtle inter-class differences, requiring models to learn discriminative features at fine granularity.

Stanford Cars [10]: A fine-grained vehicle recognition dataset comprising 16,185 images across 196 car models, spanning various makes, models, and years (1991-2012). For continual learning protocols with 10 tasks, the final task contains 16 classes while others contain 20 classes each, accommodating the non-divisible class count.

Food-101 [1]: A large-scale food recognition dataset with 101,000 images across 101 food categories (1,000 images per class). For experimental consistency, we randomly sample 100 classes to maintain uniform task splitting across protocols.

B.3. Baseline Implementation and Reproduction

To ensure fair comparison and reproducibility, we categorize baseline results by their source. Training-free methods (SimpleCIL [22] and Continual-CLIP [17]) and finetuning approaches (standard finetuning, CoOp [25], and LoRA [6]) are reproduced in our experimental framework using identical hardware and software configurations. Results for other methods on OpenCLIP are sourced from PROOF [24] and ENGINE [23], while OpenAI CLIP results are obtained from MG-CLIP [8]. In CoOp and LoRA implementations, we utilize 16 learnable prompt tokens and a rank of 8 for LoRA adapters on attention layers, following standard practices.

For experimental consistency, we adopt the random seed conventions of the respective source papers: seed 1993 for OpenCLIP experiments (following PROOF [24] and ENGINE [23] protocols) and seed 42 for OpenAI CLIP experiments (following MG-CLIP [8]). This ensures our comparisons align with previously reported results while maintaining reproducibility.

Table A2. Continual learning performance on OpenAI CLIP ViT-B/16 with 10-task protocol for fine-grained datasets.

Method	CUB-200		Stanford Cars		Food-101	
	last	*avg*	*last*	*avg*	*last*	*avg*
SimpleCIL	76.03	83.86	75.04	81.94	86.72	91.08
Cont-CLIP	55.85	68.82	63.41	71.31	88.36	92.54
Finetune	65.84	78.64	72.91	81.39	90.34	93.88
CoOp [25]	57.61	72.09	65.18	75.57	89.46	93.37
LoRA [6]	63.03	76.89	71.98	81.37	90.17	93.81
LfI (ours)	**80.64**	**85.82**	**82.51**	**87.29**	**91.89**	**94.34**
Joint	83.19	—	89.01	—	93.00	—

B.4. Implementation Infrastructure

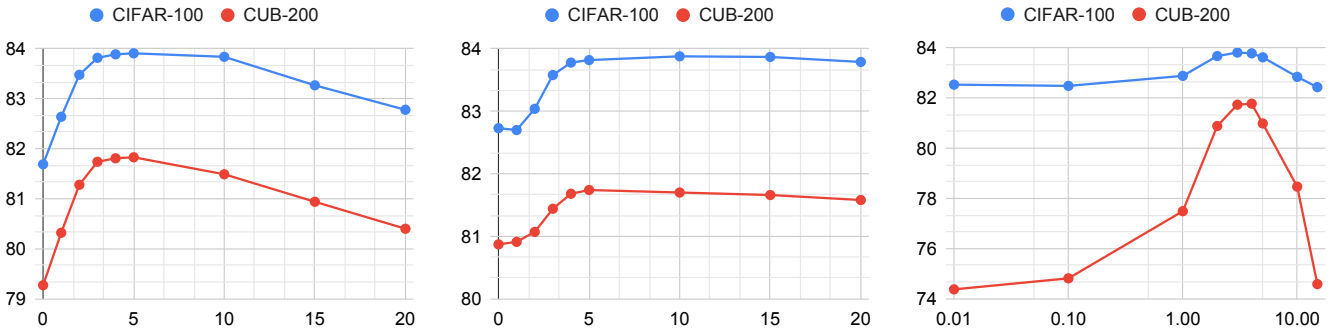
All experiments, excluding external caption generation, are conducted on a single NVIDIA RTX 3090 GPU with 24GB memory. We employ `bfloat16` automatic mixed precision (AMP) to accelerate training and reduce memory consumption without sacrificing numerical precision or model performance. Our implementation utilizes PyTorch 2.8.0 with CUDA 12.6, providing optimal compatibility with modern GPU architectures.

Pretrained CLIP models are accessed through the `open_clip` library (version 3.1.0) [9], which provides standardized interfaces for both model variants. Specifically, we utilize: *OpenCLIP* with model identifier `ViT-B-16` with variant `laion400m_e32`, representing the model trained on LAION-400M for 32 epochs, and *OpenAI CLIP* with model identifier `ViT-B-16-quickgelu` with variant `openai`, implementing the original CLIP architecture with QuickGELU activation.

C. Additional Benchmark Results

This section extends the benchmark evaluation presented in the main paper with additional experimental protocols and model variants. We provide more results on alternative task splitting configurations (6-task protocols), evaluation on fine-grained datasets not covered in prior work, and comprehensive comparisons across different CLIP architectures. These extended results demonstrate the consistency and robustness of our approach across diverse experimental settings.

Tab. A1, as a supplemental table to Tab. 1, reveals how different methods handle domain shift and fine-grained classification. On ImageNet-R's artistic renditions, prompt-based approaches struggle severely (L2P: 66.77%, CODA-Prompt: 68.05%), while LfI achieves 84.12%, approaching joint training (86.07%). Stanford Cars highlights the vision-text per-



(a) Number of learnable tokens K per sample. (b) Vocabulary alignment neighbors M. (c) Adaptive mixing sharpness γ (log scale).

Figure A1. Analysis of key hyperparameters on CIFAR-100 and CUB-200.

formance gap—SimpleCIL's prototypical approach reaches 86.95% while RAPF drops to 63.19%. LfI achieves 91.92%, nearly matching joint training (92.69%) and surpassing ENGINE by 1.89%. Food-101 shows competitive performance across methods, with LfI maintaining superiority at 89.93%, confirming robustness across diverse visual domains.

Tab. A2 explores fine-grained classification (not previously evaluated in the MG-CLIP benchmark) on OpenAI CLIP, revealing dramatic performance disparities. On CUB-200, SimpleCIL achieves 76.03% through visual features alone, while Continual-CLIP drops to 55.85%, suggesting OpenAI CLIP's text encoder struggles with fine-grained bird distinctions. LfI successfully bridges this gap at 80.64%. Stanford Cars shows similar patterns—traditional finetuning (72.91%) outperforms CoOp (65.18%), indicating that CLIP's frozen structure can hinder adaptation. Food-101 presents an exception where Continual-CLIP achieves competitive 88.36%, suggesting better alignment with food semantics. Across all datasets, LfI consistently leads (80.64%, 82.51%, 91.89%), with particularly striking gains on Stanford Cars where we exceed the next best by nearly 7.5%.

D. Additional Ablation Studies

This section presents detailed ablation studies that examine the sensitivity and robustness of our method. We analyze the impact of key hyperparameters on model performance, investigate the compatibility of our approach with parameter-efficient finetuning methods, and explore how our method scales with increased model capacity and pretraining data. These studies provide deeper insights into the mechanisms underlying Learning from Itself and guide practical deployment decisions.

D.1. Hyperparameter Analysis

Fig. A1 examines the sensitivity of our method to three critical hyperparameters. For the number of learnable tokens K (Fig. A1a), performance peaks at 3-5 tokens (83.82% on CIFAR-100 with $K = 3$), while excessive tokens ($K > 10$) lead to overfitting and adversarial solutions, causing performance degradation. Notably, using only class names ($K = 0$) produces negative effects (81.69%), confirming that bare class labels create harmful distribution mismatch. The vocabulary alignment parameter M (Fig. A1b) shows stable performance above 4 neighbors, with our choice of $M = 5$ achieving optimal results. Without vocabulary alignment ($M = 0$), performance drops by 1.09% on CIFAR-100, validating its importance in preventing semantic drift. The mixing sharpness γ (Fig. A1c) exhibits a clear optimal range around 3-4, where the adaptive weighting effectively balances knowledge from both heads. Small values ($\gamma < 1$) approximate uniform mixing, reducing to ineffective symmetric mutual learning, while large values ($\gamma > 10$) create overly rigid targets that hinder distillation.

Table A3. Integration of LfI with parameter-efficient methods. Performance gains are consistent across different adaptation strategies.

Method	**CIFAR-100**		**CUB-200**	
	last	*avg*	*last*	*avg*
Finetune	80.91	87.33	71.06	82.73
+ LfI (reported)	**83.82**	**88.65**	**81.74**	**87.23**
CoOp [25]	69.83	78.77	62.32	75.33
+ LfI	**72.38**	**81.23**	**67.69**	**80.08**
LoRA [6]	79.69	86.45	69.52	81.41
+ LfI	**82.34**	**87.59**	**80.18**	**86.58**

D.2. Compatibility with PEFT Methods

Tab. A3 demonstrates that our approach successfully enhances existing parameter-efficient finetuning (PEFT) methods. When combined with CoOp [25], LfI improves performance by 2.55% on CIFAR-100 and 5.37% on CUB-200, suggesting that generated tokens provide valuable training signals even with prompt tuning. Integration with LoRA [6] yields substantial gains (2.65% and 10.66% respectively), with particularly dramatic improvement on fine-grained tasks

Table A4. Performance scaling with model size (ViT-L/14) and pretraining data (LAION-2B). LfI maintains consistent improvements across different model scales.

Model size	Pretraining data	Method	CIFAR-100		CUB-200	
			last	*avg*	*last*	*avg*
ViT-B/16	LAION-400M [14]	SimpleCIL [22]	76.68	81.41	79.57	86.17
		Continual-CLIP [17]	71.11	77.86	64.15	76.70
		Finetune	80.91	87.33	71.06	82.73
		CoOp [25]	69.83	78.77	62.32	75.33
		LoRA [6]	79.69	86.45	69.52	81.41
		LfI (ours)	**83.82**	**88.65**	**81.74**	**87.23**
		Joint	89.44	—	83.64	—
ViT-L/14	LAION-400M [14]	SimpleCIL [22]	81.44	85.53	84.79	90.02
		Continual-CLIP [17]	76.63	83.07	72.97	81.38
		Finetune	85.47	90.03	77.49	86.05
		CoOp [25]	78.40	85.20	69.81	80.92
		LoRA [6]	84.62	89.55	76.37	85.20
		LfI (ours)	**88.12**	**91.48**	**86.21**	**91.17**
		Joint	91.59	—	87.18	—
ViT-B/16	LAION-2B [15]	SimpleCIL [22]	80.40	84.63	80.86	87.11
		Continual-CLIP [17]	77.19	82.91	70.04	79.99
		Finetune	82.61	88.10	73.71	84.30
		CoOp [25]	77.66	83.81	67.38	79.38
		LoRA [6]	82.08	87.77	73.32	83.40
		LfI (ours)	**85.76**	**90.19**	**82.89**	**88.93**
		Joint	89.79	—	84.26	—

where mutual distillation transfers crucial visual knowledge. These results confirm that LfI's knowledge mining is complementary to parameter-efficient adaptations, offering a practical enhancement for resource-constrained scenarios.

D.3. Scaling Analysis

Tab. A4 explores how our method scales with increased model capacity and pretraining data. With ViT-L/14, LfI achieves 88.12% on CIFAR-100, maintaining a 2.65% improvement over finetuning despite the stronger baseline. The larger model particularly benefits fine-grained classification—CUB-200 reaches 86.21%, approaching joint training performance (87.18%). When scaling pretraining data from LAION-400M [14] to LAION-2B [15] while maintaining ViT-B/16 architecture, the enhanced pretrained representations lift all methods, yet LfI preserves its advantage (85.76% vs. 82.61% finetuning). Interestingly, the performance gap between LfI and baselines remains consistent across scales (approximately 3-4% on CIFAR-100, 8-10% on CUB-200), indicating that our knowledge mining approach effectively leverages improved representations regardless of their source.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014. 3

[2] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 1, 2

[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. 2

[4] Arthur Douillard and Timothée Lesort. Continuum: Simple management of complex continual learning scenarios. *arXiv preprint arXiv:2102.06253*, 2021. 2

[5] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical

analysis of out-of-distribution generalization. In *ICCV*, 2021. 2

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 3, 4, 5

[7] Linlan Huang, Xusheng Cao, Haori Lu, and Xialei Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *ECCV*, 2024. 3

[8] Linlan Huang, Xusheng Cao, Haori Lu, Yifan Meng, Fei Yang, and Xialei Liu. Mind the gap: Preserving and compensating for the modality gap in clip-based continual learning. In *ICCV*, 2025. 3

[9] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 3

[10] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 3

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[12] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. 1

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 1

[14] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 5

[15] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 5

[16] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 2023. 3

[17] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022. 3, 5

[18] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds 200. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 3

[19] 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. In *ECCV*, 2022. 3

[20] 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*, 2022. 3

[21] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 1

[22] Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *IJCV*, 133(3):1012–1032, 2025. 3, 5

[23] Da-Wei Zhou, Kai-Wen Li, Jingyi Ning, Han-Jia Ye, Lijun Zhang, and De-Chuan Zhan. External knowledge injection for clip-based class-incremental learning. In *ICCV*, 2025. 3

[24] Da-Wei Zhou, Yuanhan Zhang, Yan Wang, Jingyi Ning, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Learning without forgetting for vision-language models. *IEEE TPAMI*, 47(6): 4489–4504, 2025. 3

[25] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 3, 4, 5