Supplementary Material for Open-World Dynamic Prompt and Continual Visual Representation Learning

Youngeun Kim[1,*†], Jun Fang[2,*‡], Qin Zhang[2], Zhaowei Cai[3], Yantao Shen[2], Rahul Duggal[2], Dripta S. Raychaudhuri[2], Zhuowen Tu[2], Yifan Xing[2], Onkar Dabeer[2]

[1] Yale University
[2] AWS AI Labs
[3] Amazon AGI
youngeun.kim@yale.edu {junfa, qzaamz, zhaoweic, yantaos, dugrahul, driptarc, ztu, yifax, onkardab}@amazon.com

In this supplementary material, we expand our discussions on Dynamic Prompt and Representation Learner (DPaRL) with additional analysis and experiments. In particular, we discuss the following:

- We provide the detailed data information in Section A on our established open-world continual representation learning tasks with four benchmarks;
- We compare the number of prompt tokens and learnable parameters for Prompt-based Continual Learning (PCL) methods in Section B;
- We present the dynamic changes in the accuracy curve across all continual learning stages with various PCL methods in Section C;
- We study the few-shot performance in Section D with various PCL methods and demonstrate the superior performance of our DPaRL approach;
- We illustrate the effectiveness of our DPaRL in closed-world evaluation benchmarks in Section E;
- We investigate the influence of various DPG encoders in the PCL methods in Section F;
- We report the empirical training and testing time in Section G for different continual learning methods.

A Data Information

In Table A, we provide the detailed number of classes and data samples we used for four open-world image retrieval benchmarks: Cars [2], In-Shop [4], SOP [5] and iNat2018 [11].

The Cars dataset [2] offers 16,185 images of 196 car classes. Models were trained on the first 98 classes and tested on the subsequent 98. The In-Shop dataset [4] provides 72,712 clothing images across 7,986 classes, with the first half

* Equal contribution.
† Work conducted during an internship at Amazon.
‡ Corresponding author.

utilized for training and the latter half, which is divided into a query and gallery set, for testing. The SOP [5] includes 120,053 product images spanning 22,634 classes and 24 superclasses, bifurcated approximately in the middle for training and testing. Lastly, iNat2018 [11] is a fine-grained image retrieval dataset with 461,939 images featuring a diverse range of animal and plant species, comprising 5,690 training classes and 2,452 testing classes.

We split the number of classes almost evenly across all 10 or 100 Continual Learning (CL) stages in the training data. Except for the first CL stage, all other stages have an equal number of classes. Detailed number of classes in the first and other stages are listed in the Table A.

Table A: Data information details for four datasets: Cars, In-Shop, SOP, and iNat2018. We report the total number of images, total number of classes, and the average number of images per class in both training and testing. We split the number of classes almost evenly across all 10 or 100 Continual Learning (CL) stages. Except for the 1st CL stage, all other stages have an equal number of classes.

Dataset Information		Cars	In-Shop	SOP	iNat2018
Training Data	Number of images	8,054	25,882	59,551	325,846
	Number of classes	98	3,997	11,318	5,690
	# Images per class	82.2	6.5	5.3	57.3
10 CL Stages	# Cls in 1st stage	8	397	1139	569
	# Cls in other stages	10	400	1131	569
100 CL Stages	# Cls in 1st stage	N/A	37	131	47
	# Cls in other stages	N/A	40	113	57
Testing Data	Number of images	8,131	26,830	60,502	136,093
	Number of classes	98	3,985	11,316	2,452
	# Images per class	83.0	6.7	5.3	55.5

For Cars, In-Shop, and SOP, the class labels in each CL stage are following the order of label indices. In another word, the class labels in the i-th CL stage are $\{L_i, L_i + 1, L_i + 2, ..., L_{i+1} - 1\}$, where $L_0 = 0$ and $L_{i+1} - L_i$ is the number of classes in the i-th CL stage. For the iNat2018 dataset, we split the class labels in each CL stage by randomly shuffling the class indices. This step is important for such large-scale and fine-grained dataset as varying class orders can change the task complexity [9].

B Number of Prompt Tokens and Learnable Parameters

In this section, we provide a detailed comparison of the number of prompt tokens fed into the discriminative backbone model, as well as the total number of learnable parameters for all PCL methods, inlcuding L2P [14], DualPrompt [13], CodaPrompt [9], and our DPaRL.

Prior PCL methods utilize a static prompt pool to select and combine several prompt tokens (prompt length), which are then fed into the ViT backbone

Table B: Number of prompt tokens and learnable parameters in PCL methods.

Method	Prompt Token Information			Learnable Parameters
	Pool Size	Length	Depth	
L2P [14]	30	20	5	2.42M
Dual [13]	10	26	5	0.49M
Coda [9]	100	8	5	3.84M
DPaRL (Ours)	**0**	8	5	8.13M

model with multiple layers (prompt depth). Therefore, the learnable parameters come from the construction of the prompt pool and the prompt token formation mechanism. The number of these parameters, listed in Table B, ranges from 0.5M to 3.8M, making these methods parameter-efficient.

In contrast, our DPaRL is a dynamic prompt generation method that does not rely on a static prompt pool. To investigate the influence on prompt formation, we follow CodaPrompt [9] by setting the same prompt length (8) and prompt depth (5). We generate prompt tokens dynamically, resulting in the same prompt token size of $8 \times 768 \times 5$. However, our DPG requires 8.1 million learnable parameters, more than double the size of CodaPrompt. This increase is primarily from the specialized mapping function (7.9M) and the LoRA layer weights (0.07M), indicating a stronger representation power compared to methods relying on static prompt pools.

C Performance across Continual Learning Stages

In this section, we illustrate the dynamic change of Recall@1 performance throughout the continual learning stages in the Figure A for PCL methods, including Learning to Prompt (L2P) [14], DualPrompt (Dual) [13], CodaPrompt (Coda) [9] and our Dynamic Prompt and Representation Learner (DPaRL).

Remarkably, from the performance trend, achieving superior performance in initial stages offers a significant advantage, given the unpredictability of when the model might be evaluated in real-world scenarios. From the figure, we see the common trends: With more continual learning stages training with more data, Recall@1 performance improves; The methods achieving high Recall@1 in the last stage are likely to show higher Recall@1 in the early stages.

Importantly, our DPaRL approach exhibits the capacity to produce generalizable features from the early stages of continual learning, marking a clear advantage over traditional static prompt pool-based techniques.

D Performance in the Few-Shot Setting

In the literature, prompt learning has recently emerged as a prominent technique in the domain of few-shot learning with small number of training samples per

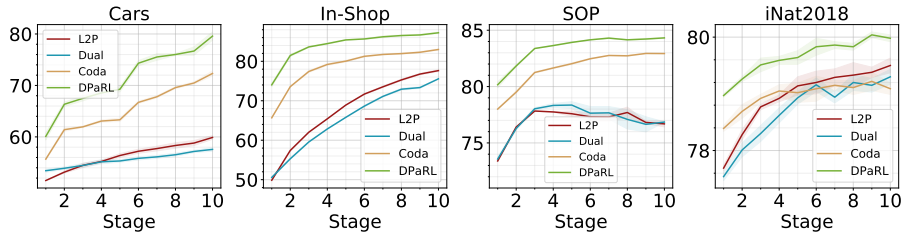

Fig. A: The change of accuracy in Recall@1 across 10 continual learning stages. Learning to Prompt (L2P) [14], DualPrompt (Dual) [13], CodaPrompt (Coda) [9] are static prompt pool-based methods. Our method DPaRL is a dynamic prompt generation method. The plots are best viewed in color.

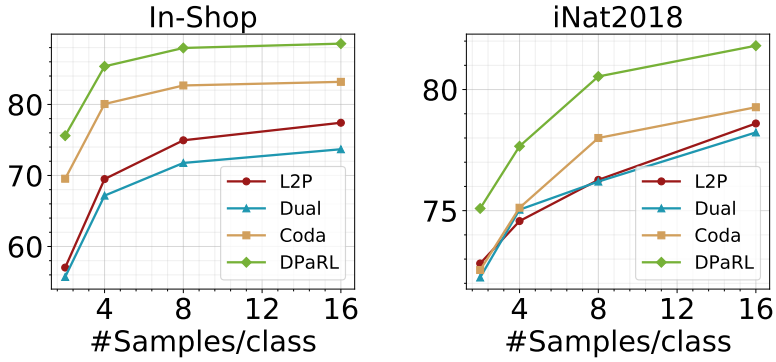

Fig. B: Accuracy in Recall@1 with respect to the different few-shot continual learning settings with 10 CL stages. The plots are best viewed in color.

class. Consequently, we investigate the efficacy of PCL-based methods within a few-shot continual learning context.

In Figure B, we present the outcomes for the M-shot scenarios, where each class is represented by M samples, with $M = 2, 4, 8, 16$, on In-Shop and iNat2018 datasets. It's worth noting that the SOP dataset is inherently designed for few-shot learning, containing approximately 5 samples per class.

The experimental results consistently demonstrate the superior performance of our DPaRL methodology across varied few-shot learning scenarios. Specifically, our method achieves 6.06% and 2.31% higher performance than CodaPrompt on In-Shop and iNat2018 datasets, respectively, with difficult 2-shot learning setting. This highlights the robust generalization capabilities of our method with dynamic prompts generation and effective representation backbone learning in the few-shot setting.

E Performance on Closed-World Benchmarks

In this section, we assess the effectiveness of our method on the closed-world evaluation benchmarks. By following [9, 13, 15], we evaluate our DPaRL on 3 challenging closed-world benchmarks: dual-shift *ImageNet-R* dataset, *DomainNet* and *VTAB* datasets with diverse classes from multiple complex realms. Table C presents a comparison to prior methods, with baseline numbers sourced from the original papers. Despite being primarily designed for open-world scenarios, our DPaRL approach demonstrates effectiveness in closed-world benchmarks by outperforming other methods. Particularly, our DPaRL achieves up to 1.47% of accuracy improvement over the previous best baselines, indicating the superiority of our method.

Table C: Accuracy (%) on closed-world benchmarks. We mark "–" if prior work did not report evaluation results on the target dataset.

Methods	Venue	ImageNet-R	DomainNet	VTAB
L2P [14]	CVPR 2022	71.66	70.54	77.11
DualPrompt [13]	ECCV 2022	71.32	70.73	83.36
CodaPrompt [9]	CVPR 2023	75.45	73.24	85.09
ADAM /w Adapter [15]	ArXiv 2023	72.35	–	85.95
ADP [10]	ICCV 2023	73.27	–	–
HiDePrompt [12]	NeurIPS 2023	75.06	–	–
DPaRL (Ours)	**Ours 2024**	**76.05** (↑0.60)	**74.71** (↑1.47)	**86.63** (↑0.68)

F Robustness against Various DPG Encoders

In the DPG network, we use pre-trained encoders to dynamically generate prompt tokens on the fly. We assess the robustness of our DPG design against various individual encoders, including CLIP [7] and DINO-V2 [6], as well as their combinations. All methods utilize the same discriminative representation backbone model pre-trained on ImageNet-21k, allowing us to solely study the effectiveness of the encoder in generating prompt tokens.

We note that CLIP and DINO-V2 process extensive datasets (>100M) containing rich semantic information, which can provide strong prompt instructors for open-world task. Moreover, none of them rely on labeled data or possess awareness of class information relevant to our open-world tasks. They align with the principles of our open-world setting.

From the results in Table D, we observed an average performance enhancement with a single encoder trained on a larger dataset with CLIP or DINO-V2, compared to ImageNet-21k with 14M training samples. Furthermore, no single model attains peak performance across all datasets, indicating the varied suitability of pre-trained encoders for distinct data domains.

However, we can combine multiple encoders to dynamically generate prompt tokens via concatenation to feed into our DPG network (denoted as DPG++) for harnessing their individual strengths to further enhance the final accuracy. Here, we limit our design to two encoders, CLIP and DINO-V2, for two reasons: one of these two models can achieve the best performance across these four tasks, and adding more encoders would substantially increase compute and memory requirements. The results of DPG++ in Table D demonstrate substantial performance improvements of 2.64% from 79.77% to 82.41% on average across all datasets. Moreover, by leveraging DPG++ with our Dynamic Prompt and Representation Learner, denoted as DPaRL++, it achieves another 1.96% boost to 84.37%, which is very close to the upper bound of 85.78%. This emphasizes the robustness of our DPG and DPaRL design in dynamically harnessing diverse pre-trained encoders to enhance open-world visual representation learning capability. It accentuates the practical impact of our method in addressing challenges posed by open-world continual learning problems.

Table D: Accuracy in Recall@1 with various pre-trained encoders for prompt generation with the same discriminative backbone. DPG++ and DPaRL++ use two encoders to generate and combine dynamic prompts, achieving more advanced performance.

Pre-trained Encoder	Method	Car	In-Shop	SOP	iNat2018	Average
ImageNet-21K [8]	Coda [9]	65.23	78.61	81.62	78.97	76.11
ImageNet-21K [8]	DPG	70.62	84.09	82.69	80.02	79.36 (↑3.25)
CLIP [7]	DPG	71.88	82.83	82.75	80.57	79.51 (↑3.40)
DINO-V2 [6]	DPG	70.67	84.47	81.77	82.17	79.77 (↑3.66)
CLIP + DINO-V2	DPG++	76.63	87.10	83.46	82.45	82.41 (↑6.30)
CLIP + DINO-V2	DPaRL++	**80.25**	**89.23**	**85.44**	**82.59**	**84.37** (↑8.26)

G Training and Testing Time Comparisons

Table E outlines the training and testing wall times, measured in minutes (mins), for non-PCL methods, various PCL methods, and our DPaRL method. The results highlight that traditional non-PCL methods, such as ER [1] and LwF [3], exhibit faster training and testing runtimes compared to PCL methods. This discrepancy arises because PCL methods necessitate an additional network for prompt token generation, either from a static prompt pool (L2P, Dual, Coda) or in a dynamic manner (our DPaRL).

The primary distinction between our DPaRL and other PCL methods lies in the prompt generation mechanism and the parameter-efficient fine-tuning on the backbone model. As discussed in Section B, our DPaRL introduces additional parameters in the dedicated mapping function, stage tokens, and low-rank adaption layers, resulting in more number of learnable parameters compared to other PCL methods. Consequently, our DPaRL incurs about 38% additional training time

overhead. However, during inference, our DPaRL exhibits comparable runtime to other PCL methods, indicating its deployment advantage with higher accuracy performance. Moreover, we can achieve even higher accuracy by utilizing two pre-trained encoders in our DPG network for dynamic prompt generation, denoted as DPaRL++, albeit at a higher cost of inference runtime.

Table E: Training and testing wall time, in minutes (mins), for different methods on iNat2018 dataset. The training wall time denotes the duration required to complete training across all CL stages. On the other hand, the testing wall time represents the duration for conducting the image retrieval evaluation benchmark on the testing data.

Method	Method Type	Training	Testing
ER [1]	Rehearsal-based	476.6 mins	7.6 mins
LwF [3]	Regularization-based	500.9 mins	7.6 mins
L2P [14]	Rehearsal-free PCL	524.8 mins	8.4 mins
Dual [13]	Rehearsal-free PCL	510.9 mins	8.4 mins
Coda [9]	Rehearsal-free PCL	526.2 mins	8.5 mins
DPaRL (Ours)	Rehearsal-free PCL	727.6 mins	8.5 mins
DPaRL++ (Ours)	Rehearsal-free PCL	865.5 mins	10.2 mins

References

1. Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P.K., Torr, P.H., Ranzato, M.: On tiny episodic memories in continual learning. arXiv preprint arXiv:1902.10486 (2019)
2. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013)
3. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
4. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1096–1104 (2016)
5. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4004–4012 (2016)
6. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
7. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
8. Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses. arXiv preprint arXiv:2104.10972 (2021)

9. Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11909–11919 (2023)
10. Tang, Y.M., Peng, Y.X., Zheng, W.S.: When prompt-based incremental learning does not meet strong pretraining. In: ICCV. pp. 1706–1716 (2023)
11. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8769–8778 (2018)
12. Wang, L., Xie, J., Zhang, X., Huang, M., Su, H., Zhu, J.: Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. NeurIPS (2023)
13. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: European Conference on Computer Vision. pp. 631–648. Springer (2022)
14. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 139–149 (2022)
15. Zhou, D.W., Ye, H.J., Zhan, D.C., Liu, Z.: Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. arXiv preprint arXiv:2303.07338 (2023)