Semantic Residual Prompts for Continual Learning - Supplementary Materials

Martin Menabue[1], Emanuele Frascaroli[1], Matteo Boschini[1], Enver Sangineto[1], Lorenzo Bonicelli[1], Angelo Porrello[1], and Simone Calderara[1]

[1] AImageLab, University of Modena and Reggio Emilia, Italy
{name.surname}@unimore.it

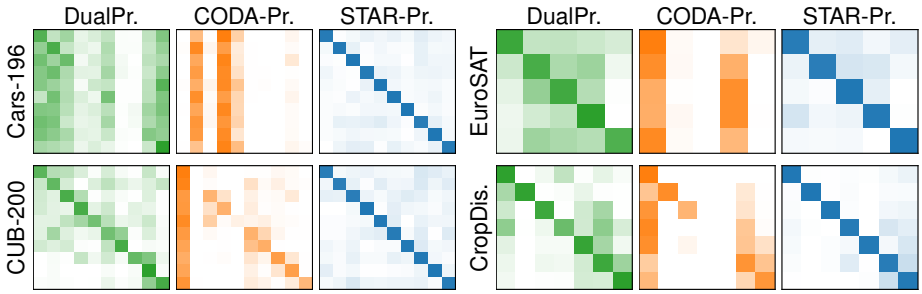

Fig. A: Analysis of prompt selection stability for Split Cars-196, Split EuroSAT, Split CUB-200, and Split CropDiseases. We assess various models at the end of the last task and report results as confusion matrices. The y axis indicates the task of the query sample, while the x axis shows the task of the corresponding selected key.

A Prompt-selection stability

We hereby extend the results outlined in Sec. 3.4 and present outcomes for additional datasets/domains. Figure A provides a visual snapshot at the conclusion of the final task. It is evident that STAR-Prompt exhibits the highest precision in selecting the appropriate prompts. Figure B supplements this analysis with a detailed focus on the initial task. At first glance, CODA-Prompt [7] appears to match the retrieval performance of our approach in Split EuroSAT. However, we ascribe it to a bias of CODA-Prompt towards the prompts learned during the first task, as also observed in Fig. A. These additional results reaffirm our belief that STAR-Prompt strikes the optimal balance between stability and flexibility throughout the training sequence.

B Additional methodological details

In this section, we provide the details concerning $\mathcal{L}_{\text{GR}}^{Q}$, introduced in Sec. 3.5, and which closely follows the computation of $\mathcal{L}_{\text{GR}}^{P}$. Specifically, similarly to Eq. (14),

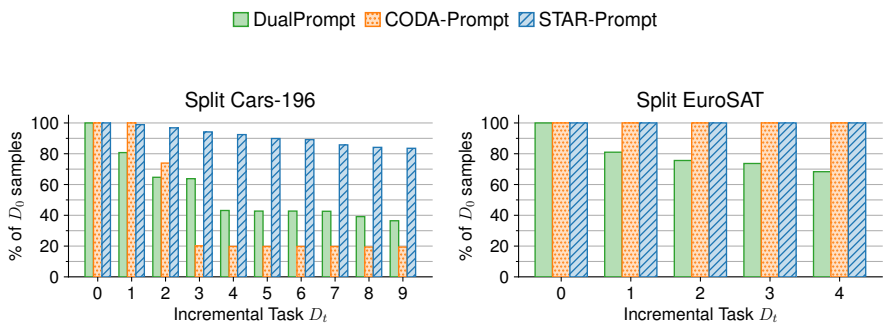

Fig. B: Prompt retrieval on the first task of different datasets.

$\mathcal{L}^Q_{\text{GR}}$ is defined as:

$$\mathcal{L}^Q_{\text{GR}} = -\frac{1}{nNt}\sum_{c=1}^{Nt}\sum_{i=1}^{n}\log p(y_i = c|\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i}), \tag{A}$$

where $p(y_i = c|\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i})$ is the score relating each synthetic representation $\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i} \sim \mathcal{H}_c$ with the corresponding ground-truth class c. We compute it as:

$$p(y_i = c|\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i}) = g_{\theta_{t'}|_{t'\in\{1,\dots,t\}}}(\tilde{\boldsymbol{e}}^{\text{CLS}}_{L,i}). \tag{B}$$

In Eq. (B), $g_{\theta_{t'}|_{t'\in\{1,\dots,t\}}}(\cdot)$ indicates that we use *all* the classification heads of $g(\cdot)$ corresponding to the t tasks so far observed.

C On the computational demands of prompt-based approaches

We use a serial two-stage training approach. First, we learn prompts for the CLIP text encoder, freeze them, and then learn prompts for the ImageNet-pre-trained ViT. Since each phase is independent, the overall cost stems from each respecting phase.

In the first stage, similar to [12], a batch of images requires one forward pass through the frozen CLIP image encoder and another through the text encoder to compute class-level textual embeddings. Due to the very low number of trainable parameters, this stage converges in a few epochs and requires no gradient computation for the image encoder.

In the second stage, we freeze the CLIP text encoder prompts and focus on the ViT prompts. The class prototypes from the text encoder can be cached, eliminating the need for additional forward passes through the text encoder. Again, each batch involves two forward passes: one through the CLIP image

encoder and another through the prompted ViT. This complexity of STAR-Prompt thus matches L2P, Dual-Prompt, and CODA-Prompt, which also use two forward passes but with the same backbone. Our approach uses two distinct backbones, requiring additional GPU memory for the CLIP visual encoder, which we find negligible compared to the performance gain.

D Setting and implementation details

Experimental setting. We here provide any missing information about datasets and experimental settings:

- **Split Imagenet-R**: 10 tasks of 20 classes each; 50 training epochs.
- **Split CIFAR-100**: 10 tasks of 10 classes each; 20 training epochs.
- **Split Cars-196**: 9 tasks of 20 classes each, plus the 10th task comprising the remaining 16 classes; 50 training epochs.
- **Split CUB-200**: 10 tasks of 20 classes each; 50 training epochs.
- **Split EuroSAT**: 5 tasks of 2 classes each; 5 training epochs.
- **Split RESISC45**: 9 tasks of 5 classes each; 30 training epochs.
- **Split CropDiseases**: 7 tasks of 5 classes each; 5 training epochs. From the original dataset [4] of 38 classes, we removed the classes with the lowest number of examples ("*Potato healthy*", "*Apple Cedar*" and "*Peach healthy*").
- **Split ISIC**: 3 tasks of 2 classes each; 30 training epochs. From the original dataset [3] we removed the most frequent class "*Melanocytic nevus*".
- **Split ChestX**: 2 tasks of 3 classes each. 30 training epochs. From the original dataset [9], we took the images without overlapping diseases belonging to the classes "*Cardiomegaly*", "*Consolidation*", "*Edema*", "*Fibrosis*", "*Pleural Thickening*" and "*Pneumothorax*".

Following [11], each experiment is repeated 3 times fixing the seeds 1993, 1996, and 1997.

On instance/batch-wise inference. To maintain consistency with the majority of CL studies, we conduct inference independently for each sample within a batch. This approach, termed *instance-wise* inference, is by far the most predominant in the literature [1, 5–7, 11].

Conversely, L2P and DualPrompt originally presented results using a *batch-wise* setup, where a single prompt is selected for all samples in a batch through majority voting. We reckon that this setup offers an unfair advantage, as it leverages the fact that samples are not shuffled during inference, hence the ground-truth labels of the examples within a mini-batch are typically the same. To ensure a fair comparison with other methods, we thus report L2P and Dual-Prompt results using the instance-wise setup, thereby eliminating any potential advantage these approaches might have had over other techniques.

Table A: The Final Forgetting (*the lower the better*) - **part 1**

Model	Imagenet-R	CIFAR-100	Cars-196	CUB-200	*Avg.*
Fine-tune † (*ViT-B/16*)	83.62	89.95	85.55	90.69	87.45
LwF †	79.68	87.23	76.59	83.81	81.83
DER++ * †	38.88	19.28	24.42	19.16	25.44
CODA-Prompt	4.06	5.56	7.36	**5.65**	5.66
AttriCLIP	6.9	–	18.8	33.35	19.68
STAR-Prompt	**3.92**	**4.71**	**6.14**	6.94	**5.43**

Table B: The Final Forgetting (*the lower the better*) - **part 2**

Model	EuroSAT	RESISC	CropDis.	ISIC	ChestX	*Avg.*
Fine-tune † (*ViT-B/16*)	98.11	95.23	93.07	94.31	66.76	78.23
LwF †	92.88	94.63	90.57	95.06	69.01	77.98
DER++ * †	8.02	53.52	8.57	45.61	61.61	40.84
L2P	43.87	25.41	18.85	37.79	42.60	36.88
DualPrompt	12.88	14.46	10.98	**25.01**	31.35	26.27
CODA-Prompt	16.75	15.05	10.39	22.97	**10.49**	22.41
AttriCLIP	39.13	32.16	62.56	74.72	48.57	51.43
SLCA †	7.74	10.58	4.90	35.25	32.05	27.30
STAR-Prompt	**4.31**	**5.25**	**3.26**	26.36	30.75	**14.61**

E Final Forgetting Metric

Tabs. A and B report the Final Forgetting metric [2] for our experiments. Tab. A lacks L2P, DualPrompt, and SLCA, since [11] does not report forgetting values for their experiments. The same applies to PromptFusion and to the experiments of AttriCLIP on Split CIFAR-100 (taken from [8]).

Moreover, we report in Tab. C the standard deviation of experiments in Tab. 2 of the main paper.

F Hyperparameters

The hyperparameters of STAR-Prompt employed for each experiment are reported in Tabs. D and E.

G Number of Trainable Parameters

The number of trainable parameters varies a lot among the compared methods. As mentioned in Secs. 1 and 2, the two main adaptation strategies are

Table C: The std dev. for experiments in Tab. 2.

Model	EuroSAT	RESISC	CropDis.	ISIC	ChestX
Joint (*STAR-Prompt*)	±0.13	±0.24	±0.10	±0.30	±0.43
Joint † (*ViT-B/16*)	±0.12	±0.10	±0.08	±1.76	±1.40
Fine-tune † (*ViT-B/16*)	±0.13	±2.02	±0.39	±2.22	±0.64
LwF †	±2.78	±0.85	±2.76	±1.98	±0.85
GDumb * †	±1.49	±0.54	±1.35	±3.64	±2.26
DER++ * †	±1.62	±2.89	±1.06	±2.16	±1.22
L2P	±7.86	±3.71	±0.25	±3.84	±1.52
DualPrompt	±4.94	±3.92	±2.68	±1.07	±0.10
CODA-Prompt	±6.30	±5.15	±2.91	±3.50	±3.90
AttriCLIP	±6.15	±4.31	±3.03	±10.38	±1.81
SLCA †	±0.48	±0.35	±0.60	±3.83	±1.80
STAR-Prompt	±0.15	±0.54	±0.60	±0.62	±2.63

	Refs. to Algorithm	Imagenet-R	CIFAR-100	Cars-196	CUB-200
E_1	*First stage* – Line 1	10	10	10	50
λ	*First stage* – Line 4	30	10	30	30
lr	*First stage* – Line 11	0.05	0.05	0.01	0.05
E_2	*Second stage* – Line 14	10	10	10	5
λ	*Second stage* – Line 17	10	5	10	50
lr	*Second stage* – Line 23	0.001	0.001	0.01	0.1
M	*Requirements*	5	5	5	5

Table D: Hyperparameters used for each dataset - **part 1**.

fine-tuning the whole model on the training data of the target dataset(s) or **parameter-efficient learning**, which adapts the model with only a few parameters (e.g., the prompts). In Tab. F we use Split CIFAR-100 as reference scenario and report the number of trainable parameters, *i.e.*, those parameters that are optimized during the incremental learning. For instance, SLCA fine-tunes the whole model, while the trainable parameters of STAR-Prompt are composed of: $\boldsymbol{p}_c, Q_c, A_c$ $(c \in \mathcal{Y}_1, ..., \mathcal{Y}_T)$ and θ_t $(t \in \{1, ..., T\})$.

H Additional ablations

We herein report the ablative studies of Sec. 4.2 over the remaining datasets in Tab. G. These results confirm the conclusion already outlined in the main paper (see Sec. 4.2).

	Refs. to Algorithm	EuroSAT	RESISC	CropDis.	ISIC	ChestX
E_1	*First stage* – Line 1	10	10	10	50	10
λ	*First stage* – Line 4	30	10	30	5	30
lr	*First stage* – Line 11	0.05	0.05	0.01	0.01	0.05
E_2	*Second stage* – Line 14	10	10	10	10	10
λ	*Second stage* – Line 17	5	5	5	10	5
lr	*Second stage* – Line 23	0.1	0.01	0.001	0.01	0.05
M	*Requirements*	5	5	5	5	5

Table E: Hyperparameters used for each dataset - **part 2**.

Model	#params (millions)
Joint, Fine-Tune, LwF	86
GDumb, DER++, SLCA	86
L2P	0.12
DualPrompt	0.41
CODA-Prompt	3.91
AttriCLIP	0.0998
PromptFusion	0.35
STAR-Prompt	3.89

Table F: The number of learnable parameters for each tested method. Note that all these methods use a ViT-B/16 as the main classification architecture (see Sec. 4).

I Additional experimental analysis

Number of Gaussians. Tab. H shows the Final Average Accuracy of STAR-Prompt on Split Imagenet-R using different numbers of Gaussians (M) for each $\mathcal{H}_c$. The results indicate a substantial plateau of the performance when $M \geq 5$, which we hence set as default value in all our experiments.

Prefix tuning *vs.* semantic residuals. In Tab. I we extend the comparison between prefix tuning and semantic residuals reported in Tab. 3 of the main paper. Specifically, differently from *Prefix Tuning* in Tab. 3, in which we used 5 tokens for each key and value, in Tab. I we use *one* token only for each key and value. This way, the total number of parameters per class is comparable with Q_c. Tab. I shows that the Final Average Accuracy scores obtained using prefix tuning are largely inferior to using semantic residuals, confirming the results reported in Tab. 3. This shows that the gap between the two conditioning methods is not due to the number of prompt parameters. The results of prefix tuning in I are drastically inferior to those obtained when using the recipe suggested in [10] (5 tokens for each key and value) and adopted in Tab. 3.

Model	CIFAR-100	CUB-200	RESISC	CropDis.	ChestX
STAR-Prompt	**90.12**$_{\pm 0.32}$	**84.10**$_{\pm 0.28}$	**92.28**$_{\pm 0.54}$	**94.92**$_{\pm 0.60}$	**41.85**$_{\pm 2.63}$
Ablations on two-level prompting					
Classify with first-level keys $\boldsymbol{w}_c$	83.49$_{\pm 0.16}$	81.31$_{\pm 0.20}$	90.80$_{\pm 0.34}$	88.36$_{\pm 0.74}$	31.70$_{\pm 0.77}$
w/o first-level prompts	87.67$_{\pm 0.37}$	81.34$_{\pm 0.17}$	85.85$_{\pm 0.69}$	89.92$_{\pm 1.76}$	37.27$_{\pm 3.96}$
Other secondary ablations					
Prefix Tuning (no residuals)	86.70$_{\pm 0.59}$	82.67$_{\pm 0.14}$	84.26$_{\pm 0.35}$	95.06$_{\pm 0.34}$	39.28$_{\pm 4.14}$
w/o Generative Replay	88.72$_{\pm 0.41}$	82.21$_{\pm 0.39}$	88.2$_{\pm 0.79}$	88.82$_{\pm 1.25}$	37.66$_{\pm 2.12}$
w. Unimodal Generative Replay	90.07$_{\pm 0.34}$	83.16$_{\pm 0.14}$	92.29$_{\pm 0.51}$	94.21$_{\pm 0.44}$	38.77$_{\pm 2.76}$
w/o Confidence Modulation	89.92$_{\pm 0.18}$	83.74$_{\pm 0.30}$	92.05$_{\pm 0.44}$	93.97$_{\pm 0.21}$	39.28$_{\pm 3.17}$

Table G: Ablative studies on STAR-Prompt (Final Avg. Acc. ± std dev).

M	Final Average Accuracy
2	88.94
5	89.37
10	89.16
20	89.58

Table H: Impact of the number of Gaussians.

Conditioning method	Imagenet-R	CIFAR-100	Cars-196	CUB-200
Prefix Tuning (*one token*)	71.45	85.90	52.66	80.12
Semantic Residuals	**89.16**	**90.16**	**86.50**	**85.24**

Table I: Comparing Semantic Residuals with Prefix Tuning. In the latter, we use a single-key and a single-value prompt token.

References

1. Aljundi, R., Lin, M., Goujaud, B., Bengio, Y.: Gradient Based Sample Selection for Online Continual Learning. In: Advances in Neural Information Processing Systems (2019)
2. Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.: Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: Proceedings of the European Conference on Computer Vision (2018)
3. Codella, N.C., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., Dusza, S.W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., et al.: Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). pp. 168–172. IEEE (2018)
4. Hughes, D., Salathé, M., et al.: An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060 (2015)
5. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
6. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCaRL: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2017)
7. 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 conference on Computer Vision and Pattern Recognition (2023)
8. Wang, R., Duan, X., Kang, G., Liu, J., Lin, S., Xu, S., Lü, J., Zhang, B.: Attriclip: A non-incremental learner for incremental knowledge learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2023)
9. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2017)
10. 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: Proceedings of the European Conference on Computer Vision (2022)
11. Zhang, G., Wang, L., Kang, G., Chen, L., Wei, Y.: SLCA: slow learner with classifier alignment for continual learning on a pre-trained model. In: IEEE International Conference on Computer Vision (2023)
12. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision (2022)