PLOP: Learning without Forgetting for Continual Semantic Segmentation: Supplementary Material

Arthur Douillard[1,2], Yifu Chen[1], Arnaud Dapogny[3], Matthieu Cord[1,4]
[1]Sorbonne Université, [2]Heuritech, [3]Datakalab, [4]valeo.ai
arthur.douillard@heuritech.com, {yifu.chen, matthieu.cord}@lip6.fr, ad@datakalab.com

A. Appendix

A.1. Further Work

In our CSS setting, pixels of task T can belong to old $C^{1:t-1}$, current C^t, and future classes $C^{t+1:T}$. In this paper we cover how to better handle old and current classes. Further works should investigate how to exploit the already present future information with Zeroshot [15, 14] as already done in semantic segmentation [12, 1] and explored for continual classification [20, 8].

A.2. Algorithm view of Local POD

In Algo. 1, we summarize the algorithm for the proposed Local POD. The algorithm consists in three functions. First, `Distillation`, loops over all L layers onto which we apply Local POD. Second, `LocalPOD`, computes the L2 distance (L.26) between POD embeddings of the current (L.19) and old (L.20) models. It loops over S different scales (L.14) and Φ computes the POD embedding given two features maps subsets (L.19-20) as defined in Eq. 1. $\| =$ denotes an in-place concatenation.

Algorithm 1 Local POD algorithm

```
 1: function DISTILLATION($f^t$, $f^{t-1}$, $x$, $S$)
 2:     $loss \leftarrow 0$
 3:     for $l \leftarrow 0;\ l < L;\ l{+}{+}$ do
 4:         $\mathbf{h}_l^t \leftarrow f_l^t(\mathbf{x})$
 5:         $\mathbf{h}_l^{t-1} \leftarrow f_l^{t-1}(\mathbf{x})$
 6:         $loss \leftarrow loss + \text{LocalPOD}(\mathbf{h}_l^t, \mathbf{h}_l^{t-1}, S)$
 7:     end for
 8:     return $\frac{loss}{L}$
 9: end function
10:
11: function LOCALPOD($\mathbf{h}^t$, $\mathbf{h}^{t-1}$, $S$)
12:     $\mathbf{P}^t \leftarrow [\,]$
13:     $\mathbf{P}^{t-1} \leftarrow [\,]$
14:     for $s \leftarrow 0;\ s < S;\ s{+}{+}$ do                      ▷ Eq. 3
15:         $w \leftarrow W/2^s$
16:         $h \leftarrow H/2^s$
17:         for $i \leftarrow 0;\ i < W - w;\ i{+}{=}\,w$ do
18:             for $j \leftarrow 0;\ j < H - h;\ j{+}{=}\,h$ do
19:                 $\mathbf{p}^t \leftarrow \Phi(\mathbf{h}^t$[i:i+w, j:j+h]$)$       ▷ Eq. 1
20:                 $\mathbf{p}^{t-1} \leftarrow \Phi(\mathbf{h}^{t-1}$[i:i+w, j:j+h]$)$
21:                 $\mathbf{P}^t \| = \mathbf{p}^t$
22:                 $\mathbf{P}^{t-1} \| = \mathbf{p}^{t-1}$
23:             end for
24:         end for
25:     end for
26:     return $\left\|\mathbf{P}^t - \mathbf{P}^{t-1}\right\|^2$                  ▷ Eq. 5
27: end function
```

A.3. Reproducibility

Datasets: We evaluate our model on three datasets Pascal-VOC [9], ADE20k [23], and Cityscapes [5]. VOC contains 20 classes, 10,582 training images, and 1,449 testing images. ADE20k has 150 classes, 20,210 training images, and 2,000 testing images. Cityscapes contains 2975 and 500 images for train and test, respectively. Those images represent 19 classes and were taken from 21 different cities. All ablations and hyperparameters tuning were done on a validation subset of the training set made of 20% of the images. For all datasets, we resize the images to 512×512, with a center crop. An additional random horizontal flip augmentation is applied at training time.

Implementation details: For all experiments, we use a Deeplab-V3 [4] architecture with a ResNet-101 [10] backbone pretrained on ImageNet [6], as in [2]. For all datasets, we set a maximum threshold for the uncertainty measure of Eq. 7 to $\tau = 1e-3$. We train our model for 30 and 60 epochs per CSS step on Pascal VOC and ADE, respectively, with an initial learning rate of $1e-2$ for the first CSS step, and $1e-3$ for all the following ones. We reduce the learning rate exponentially with a decay rate of $9e-1$. We use SGD optimizer with $9e-1$ Nesterov momentum. The Local POD factor λ is set to $1e-2$ and $5e-4$ for intermediate feature maps and logits, respectively. Moreover, we multiply this factor by the adaptive weighting $\sqrt{|C^{1:t}|/|C^t|}$ introduced by [11] that increases the strength of the distillation the further we are into the continual process. For all feature maps, Local POD is applied before ReLU, with squared pixel values, as in [21, 7]. We use 3 scales for Local POD: 1, $1/2$, and $1/4$, as adding more scales experimentally brought diminishing returns. We use a batch size of 24 distributed on two GPUs. Contrary to many continual models, we don't have access to any task id in inference, therefore our setting/strategy has to predict a class among the set of all seen classes —a realist setting.

Classes ordering details: For all quantitative experiments on Pascal-VOC 2012 and ADE20k, the same class ordering was used across all evaluated models. For Pascal-VOC 2012 it corresponds to [1, 2, ..., 20] and ADE20k to [1, 2, ..., 150] as defined in [2]. For continual-domain cityscapes, the order of the domains/cities is the following: `aachen`, `bremen`, `darmstadt`, `erfurt`, `hanover`, `krefeld`, `strasbourg`, `tubingen`, `weimar`, `bochum`, `cologne`, `dusseldorf`, `hamburg`, `jena`, `monchengladbach`, `stuttgart`, `ulm`, `zurich`, `frankfurt`, `lindau`, and `munster`.

In the main paper we showcased a boxplot featuring 20 different class orders for Pascal-VOC 2012 15-1. For the sake of reproducibility, we provide details on these orders:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
[12, 9, 20, 7, 15, 8, 14, 16, 5, 19, 4, 1, 13, 2, 11, 17, 3, 6, 18, 5]
[9, 12, 13, 18, 2, 11, 15, 17, 10, 8, 4, 5, 20, 16, 6, 14, 19, 1, 7, 3]
[13, 19, 15, 17, 9, 8, 5, 20, 4, 3, 10, 11, 18, 16, 7, 12, 14, 6, 1, 2]
[15, 3, 2, 12, 14, 18, 20, 16, 11, 1, 19, 8, 10, 7, 17, 6, 5, 13, 9, 4]
[7, 13, 5, 11, 9, 2, 15, 12, 14, 3, 20, 1, 16, 4, 18, 8, 6, 10, 19, 17]
[12, 9, 19, 6, 4, 10, 5, 18, 14, 15, 16, 3, 8, 7, 11, 13, 2, 20, 17, 1]
[13, 10, 15, 8, 7, 19, 4, 3, 16, 12, 14, 11, 5, 20, 6, 2, 18, 9, 17, 1]
[3, 14, 13, 1, 2, 11, 15, 17, 7, 8, 4, 5, 9, 16, 19, 12, 6, 18, 10, 20]
[1, 14, 9, 5, 2, 15, 8, 20, 6, 16, 18, 7, 11, 10, 19, 3, 4, 17, 12, 13]
[16, 13, 1, 11, 12, 18, 6, 14, 5, 3, 7, 9, 20, 19, 15, 4, 2, 10, 8, 17]
[10, 7, 6, 19, 16, 8, 17, 1, 14, 4, 9, 3, 15, 11, 12, 2, 18, 20, 13, 5]
[7, 5, 3, 9, 13, 12, 14, 19, 10, 2, 1, 4, 16, 8, 17, 15, 18, 6, 11, 20]
[18, 4, 14, 17, 12, 10, 7, 3, 9, 1, 8, 15, 6, 13, 2, 5, 11, 20, 16, 19]
[5, 4, 13, 18, 14, 10, 19, 15, 7, 9, 3, 2, 8, 16, 20, 1, 12, 11, 6, 17]
[9, 12, 13, 18, 7, 1, 15, 17, 10, 8, 4, 5, 20, 16, 6, 14, 19, 11, 2, 3]
[3, 14, 13, 18, 2, 11, 15, 17, 10, 8, 4, 5, 20, 16, 6, 12, 19, 1, 7, 9]
[7, 5, 9, 1, 15, 18, 14, 3, 20, 10, 4, 19, 11, 17, 16, 12, 8, 6, 2, 13]
[3, 14, 6, 1, 2, 11, 12, 17, 7, 20, 4, 5, 9, 16, 19, 15, 13, 18, 10, 8]
[1, 2, 12, 14, 6, 19, 18, 17, 5, 20, 8, 4, 9, 16, 10, 3, 15, 13, 11, 7]
```

In the 15-1 setting, we first learn the first fifteen classes, then increment the five remaining classes one by one. Note that the special class `background` (0) is always learned during the first task.

Hardware and Code: For each experiment, we used two Titan Xp GPUs with 12 Go of VRAM each. The initial step $t = 1$ for each setting is common to all models, therefore we re-use the weights trained on this step. All models took less than 2 hours to train on Pascal-VOC 2012 15-1, and less than 16 hours on ADE20k 100-10. We distributed the batch size equally on both GPUs. All models are implemented in PyTorch [18] and runned with half-precision for efficiency reasons with Nvdia's APEX library (https://github.com/NVIDIA/apex) using O1 optimization level. Our code base is based on [2]'s code (https://github.com/fcdl94/MiB) that we modified to implement our strategy. It is available at https://github.com/arthurdouillard/CVPR2021_PLOP.

Table 1: Ablations of PLOP on the Pascal-VOC 2012 dataset in 15-5 and 15-1. Scores are measured on a validation subset made of 20% of the training set.

	15-5 (2 tasks)		**15-1** (6 tasks)	
Model	*all*	*avg*	*all*	*avg*
CE	13.85	46.91	3.99	19.37
Pseudo	66.19	73.07	19.74	44.48
Pseudo + Local POD	70.29	75.13	50.41	64.95
νPseudo + Local POD	**71.43**	**75.70**	**52.31**	**65.71**

A.4. Additional Experiments

Model ablation: Table 1 shows the construction of our model component by component on Pascal-VOC 2012 in 15-5 and 15-1. For this experiment, we train our model on 80% of the training set and evaluate on the validation set made of the remaining 20%. We report the mIoU at the final task ("*all*") and the average of the mIoU after each task ("*avg*"). We start with a crude baseline made of solely cross-entropy (CE). Pseudo-labeling by itself increases by a large margin performance (eg. 3.99 to 19.74 for 15-1). Applying Local POD reduces drastically the forgetting leading to a massive gain of performance (eg. 19.74 to 50.41 for 15-1). Finally our adaptive factor ν based on the ratio of accepted pseudo-labels over the number of background pixels further increases our overall results (eg. 50.41 to 52.31 for 15-1). The interest of ν arises when PLOP faces hard images where few pseudo-labels will be created due to an overall high uncertainty. In such a case, current classes will be over-represented, which can in turn lead to strong bias towards new classes (*i.e.* the model will have a tendency to predict one of the new classes for every pixel). The ν factor therefore decreases the overall classification loss on such images, and empirical results confirm its effectiveness.

Pascal-VOC 2012 Disjoint: In the main paper, we reported results on Pascal-VOC 2012 Overlap. For reasons mentioned previously, Overlap is a more realist setting than Disjoint. Nevertheless, for the sake of comparison, we also provide results in Table 2 in the Disjoint setting. While PLOP has similar performance to MiB in 15-5 (the differences are not significant), it significantly outperforms previous state-of-the-art methods in both 19-1 and 15-1.

Pascal-VOC 2012 Overlap with more baselines: In Table 3, we report results on Pascal-VOC 2012 Overlap with more baselines. In addition to the models presented in the main paper, we add a naive Fine Tuning, two continual models based on weights constraints (PI [22] and RW [3]), and one continual model based on knowledge distillation (LwF [16]). PLOP surpasses these methods in all CSS scenarios.

Table 2: Mean IoU on the Pascal-VOC 2012 dataset for different incremental class learning scenarios, all in Disjoint. † denotes results from Cermelli et al.[2].

	19-1 (2 tasks)				**15-5** (2 tasks)				**15-1** (6 tasks)			
Method	0-19	20	*all*	*avg*	0-15	16-20	*all*	*avg*	0-15	16-20	*all*	*avg*
Fine Tuning†	5.80	12.30	6.20		1.10	33.60	9.20		0.20	1.80	0.60	
PI† [22]	5.40	14.10	5.90		1.30	34.10	9.50		0.00	1.80	0.40	
EWC† [13]	23.20	16.00	22.90		26.70	37.70	29.40		0.30	4.30	1.30	
RW† [3]	19.40	15.70	19.20		17.90	36.90	22.70		0.20	5.40	1.50	
LwF† [16]	53.00	9.10	50.80		58.40	37.40	53.10		0.80	3.60	1.50	
LwF-MC† [19]	63.00	13.20	60.50		67.20	41.20	60.70		4.50	7.00	5.20	
ILT† [17]	69.10	16.40	66.40		63.20	39.50	57.30		3.70	5.70	4.20	
MiB† [2]	69.60	25.60	67.40		**71.80**	**43.30**	**64.70**		46.20	12.90	37.90	
PLOP	**75.37**	**38.89**	**73.64**	75.71	**71.00**	**42.82**	**64.29**	72.05	**57.86**	**13.67**	**46.48**	62.67

Table 3: Mean IoU on the Pascal-VOC 2012 dataset for different incremental class learning scenarios, all in Overlap. † denotes results from Cermelli et al. [2], all other results are from us.

	19-1 (2 tasks)				**15-5** (2 tasks)				**15-1** (6 tasks)			
Method	0-19	20	*all*	*avg*	0-15	16-20	*all*	*avg*	0-15	16-20	*all*	*avg*
Fine Tuning†	6.80	12.90	7.10		2.10	33.10	9.80		0.20	1.80	0.60	
PI† [22]	7.50	14.00	7.80		1.60	33.30	9.50		0.00	1.80	0.50	
EWC† [13]	26.90	14.00	26.30		24.30	35.50	27.10		0.30	4.30	1.30	
RW† [3]	23.30	14.20	22.90		16.60	34.90	21.20		0.00	5.20	1.30	
LwF† [16]	51.20	8.50	49.10		58.90	36.60	53.30		1.00	3.90	1.80	
LwF-MC† [19]	64.40	13.30	61.90		58.10	35.00	52.30		6.40	8.40	6.90	
ILT† [17]	67.10	12.30	64.40		66.30	40.60	59.90		4.90	7.80	5.70	
ILT [17]	67.75	10.88	65.05	71.23	67.08	39.23	60.45	70.37	8.75	7.99	8.56	40.16
MiB† [2]	70.20	22.10	67.80		75.50	49.40	69.00		35.10	13.50	29.70	
MiB [2]	71.43	23.59	69.15	73.28	**76.37**	49.97	**70.08**	**75.12**	34.22	13.50	29.29	54.19
PLOP	**75.35**	**37.35**	**73.54**	**75.47**	75.73	**51.71**	**70.09**	**75.19**	**65.12**	**21.11**	**54.64**	**67.21**

References

[1] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[2] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] Arslan Chaudhry, Puneet Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2018.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. In *arXiv preprint library*, 2017.

[5] M. Cordts, M. Omran, S. Ramos, T. Reheld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2020.

[8] Arthur Douillard, Eduardo Valle, Charles Ollion, and Matthieu Cord. Insights from the future for continual learning. In *arXiv preprint library*, 2020.

[9] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. In *International Journal of Computer Vision (IJCV)*, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] Naoki Kato, Toshihiko Yamasaki, and Kiyoharu Aizawa. Zero-shot semantic segmentation via variational mapping. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshop*, 2019.

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017.

[14] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] C. H. Lampert, H. Nickisch, and S. Hermeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[16] Z. Li and D. Hoiem. Learning without forgetting. *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2016.

[17] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshop*, 2019.

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2017.

[19] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] Kai Wang, Luis Herranz, Anjan Dutta, and Joost van de Weijer. Bookworm continual learning: beyond zero-shot learning and continual learning. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV) Workshop*, 2020.

[21] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[22] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, 2017.

[23] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.