Supplementary Material for
Attribution-aware Weight Transfer: A Warm-Start Initialization for Class-Incremental Semantic Segmentation

Dipam Goswami[†§] René Schuster[†] Joost van de Weijer[‡] Didier Stricker[†]
dipamgoswami01@gmail.com rene.schuster@dfki.de joost@cvc.uab.es didier.stricker@dfki.de
[†] DFKI - German Research Center for Artificial Intelligence, Kaiserslautern
[§] Birla Institute of Technology and Science, Pilani [‡] Computer Vision Center, Barcelona

Introduction

In this supplementary material to our main paper *Attribution-aware Weight Transfer: A Warm-Start Initialization for Class-Incremental Semantic Segmentation*, we discuss the details of the gradient-based attribution method, Integrated Gradients [11] used in our Attribution-aware Weight Transfer (AWT) initialization. We further share more details of our implementation for better reproducibility, and perform additional ablative experiments to analyze the impact of the proposed warm-start initialization. Finally, we present the qualitative results of AWT with MiB [2] and SSUL [3] on Pascal-VOC 2012.

1. Attribution Method

Integrated Gradients: Consider a deep neural network as a function $F : \mathbb{R}^n \rightarrow [0, 1]$. Let $x \in \mathbb{R}^n$ be the input image and $x' \in \mathbb{R}^n$ be a baseline black image. Integrated Gradients (IG) [11] computes and accumulates the gradients at all points along the straight line path (in $\mathbb{R}^n$) from the baseline to the input.

Let $\frac{\partial F(x)}{\partial x_i}$ be the gradient of $F(x)$ along the i^{th} dimension. Then the integrated gradient along the i^{th} dimension for an input x and baseline x' is defined as follows:

$$IG_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} \, d\alpha \quad (1)$$

Note that the attributions add up to the difference between $F(x)$ and $F(x')$.

Layer Integrated Gradients: Layer Integrated Gradients [10] is designed for computing attributions corresponding to inputs or outputs of a specific layer of the network. For a given layer, the size of the attribution maps is the same as the layer's input or output dimensions, based on whether we attribute to the inputs or outputs of that layer. In our method, we compute the attributions for the inputs to the final classifier layer. We obtain the attributions corresponding to a given target class (*background* class in our method).

2. Reproducibility

Datasets: We evaluate our models on Pascal-VOC 2012 [8], ADE20K [13] and Cityscapes [5]. VOC contains 10,582 images for training and 1,449 images for testing. ADE20K contains 20,210 and 2,000 images for training and testing respectively. Cityscapes contains 2,975 training images and 500 testing images.

Implementation details: We use Deeplab-v3 [4] with ResNet-101 [9] backbone pretrained on ImageNet [6] having output stride of 16. In-place activated batch normalization [1] is used to reduce memory requirements. Similar to [2, 7, 12], the data augmentation from [4] has been applied along with random cropping of 512×512 pixels for training and a center crop of the same size for testing. A random horizontal flip is performed on the training set only.

We re-implement SSUL by training for 60 epochs on ADE20K dataset. We follow the same training settings for SSUL as proposed in [3] for VOC and ADE20K. For Cityscapes, we trained SSUL with a learning rate of 0.01 and a batch size of 24. We train the other models of FT, PLOP, RCIL for Cityscapes with SGD and a learning rate of 2×10^{-2} for the first step only and 10^{-3} for the incremental steps.

Class order: For all the quantitative experiments, we order the classes by increasing class id, *i.e.* the default order of the respective datasets.

For the ablation experiment using random orders on VOC 15-1, we sampled the following 10 class sequences:
[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, 10]
[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]
[7, 5, 9, 1, 15, 18, 14, 3, 20, 10, 4, 19, 11, 17, 16, 12, 8, 6, 2, 13]
[12, 9, 19, 6, 4, 10, 5, 18, 14, 15, 16, 3, 8, 7, 11, 13, 2, 20, 17, 1]

Table 7: Ablation study for significance of weight transfer on Pascal-VOC 2012.

		VOC (15-1)		
New Classifier Init	Iterations	0-15	16-20	all
Random	$\times$ 1	45.7	5.3	36.1
Random	$\times$ 2	39.7	6.6	31.8
Random	$\times$ 4	29.9	7.5	24.6
Weight transfer - MiB [2]	$\times$ 1	48.1	15.8	40.4
Weight transfer - AWT (Ours)	$\times$ 1	**59.1**	**17.2**	**49.1**

Table 8: Ablation study for selection of threshold using MiB+AWT on Pascal-VOC 2012.

	VOC (15-1)		
Threshold for channel selection	0-15	16-20	all
Top 10%	51.0	11.0	41.5
Top 25%	**59.1**	**17.2**	**49.1**
Top 50%	58.3	17.6	48.6
Top 75%	56.8	14.9	46.8

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

3. Additional Ablation Experiments

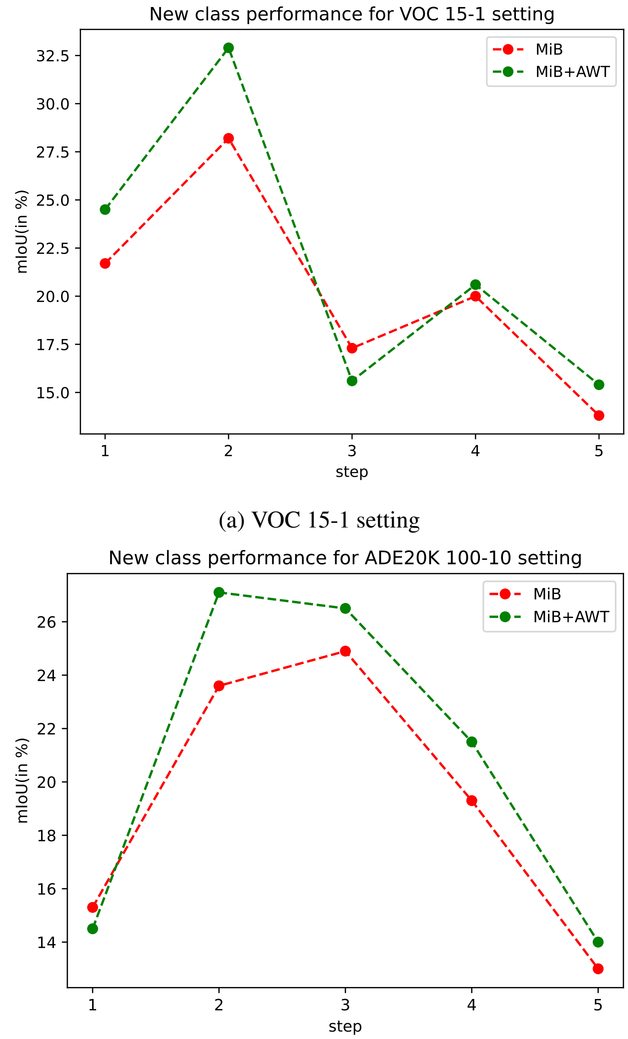
Additional experiments are performed to analyze the effect of the initialization and the number of training iterations per step. We show in Table 7 that training the model with random initialization for a higher number of iterations ($\times 2$, $\times 4$) cannot reach the performance of AWT initialization or even the one proposed by [2]. Instead, training for more iterations causes higher forgetting of old classes.

Furthermore, we vary the threshold k to select the most significant 10%, 25%, 50% and 75% of the channels for weight transfer. Based on the results of this experiment shown in Table 8, our final AWT uses a ratio of 25% for all our experiments in the main paper.

To discuss the role of AWT on reducing the effect of background shift, we analyze the performance of the newly added classes after every step of training for VOC 15-1 and ADE20K 100-10 settings in Figure 7. We observe that MiB+AWT better learns the new set of classes which transitions from the previous background to current foreground. This indicates reduced effect of the background shift with AWT across multiple steps.

4. Additional Qualitative Evaluation

Figure 8 shows the comparison of predictions using MiB, MiB+AWT, SSUL, and SSUL+AWT on some test samples of Pascal-VOC 2012 using models trained in the 10-1 setting. Over both the methods, AWT improves the predictions for multiple classes like *TV*, *car*, *aeroplane*, *bird*, *chair*, *table*, *horse*, *person*, *dog*, and many more.



(a) VOC 15-1 setting

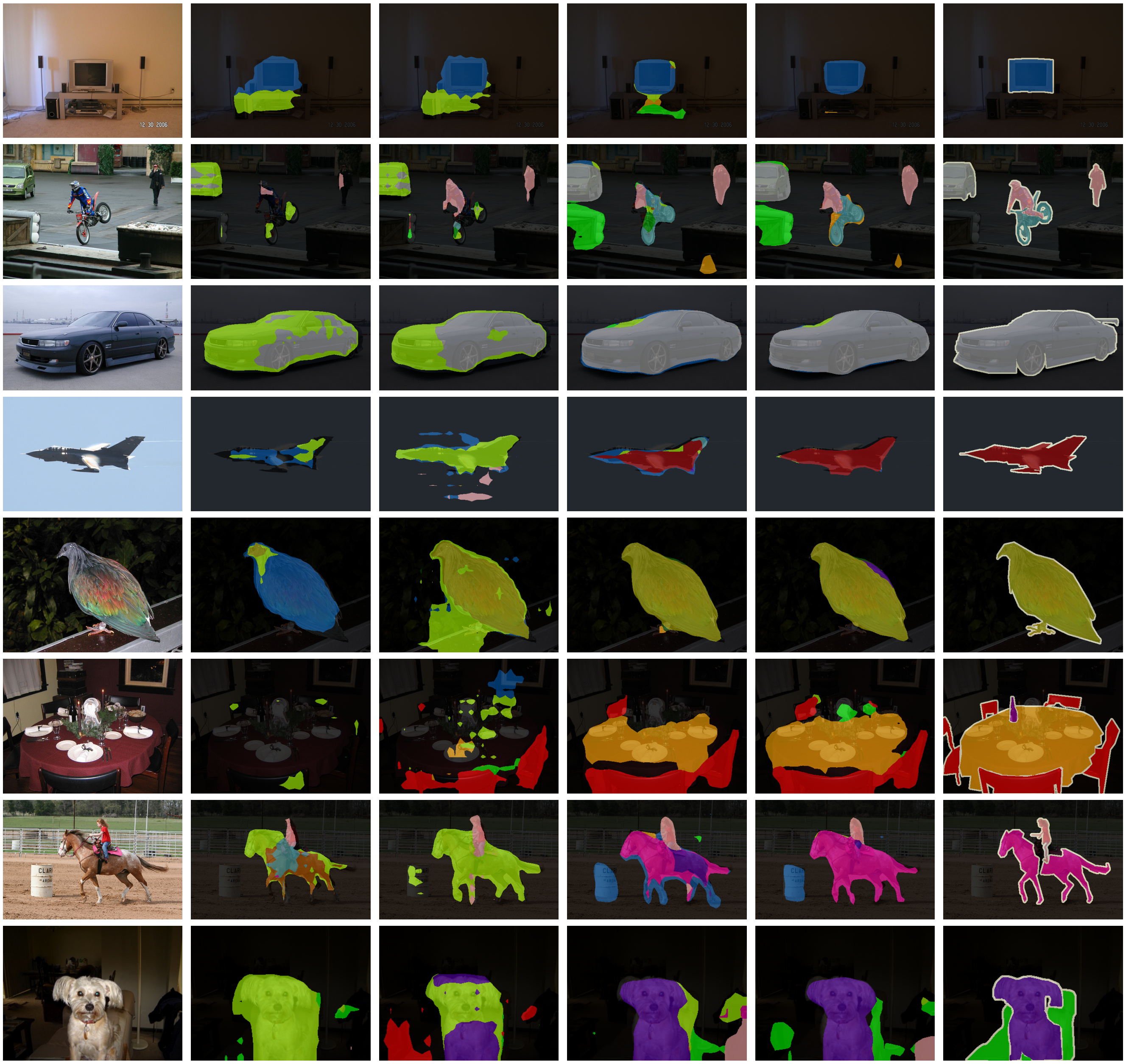
(b) ADE20K 100-10 setting

Figure 7: Analysis of the learning of new classes at every step

References

[1] Samuel Rota Bulo, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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



(a) Image (b) MiB [2] (c) MiB+AWT (d) SSUL [3] (e) SSUL+AWT (f) GT

Figure 8: Visualization of predictions using MiB, MiB+AWT, SSUL and SSUL+AWT in 10-1 setting on test images of Pascal-VOC 2012.

[3] Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[8] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 2015.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

[11] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[12] Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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