Endpoints Weight Fusion for Class Incremental Semantic Segmentation Supplementary Material

Jia-Wen Xiao [1] Chang-Bin Zhang [1] Jiekang Feng [2] Xialei Liu [1]
Joost van de Weijer [3] Ming-Ming Cheng [1]
[1] VCIP, CS, Nankai University [2] Tianjin University
[3] Computer Vision Center, Universitat Autònoma de Barcelona

A. Derivations

Since we use SGD optimizer in the whole training process, we derive the EMA [4] and our EWF in the SGD algorithm. For simplicity, we use constant value 1 to replace the learning rate. Given that EMA algorithm is defined as:

$$v^i = \beta v^{i-1} + (1-\beta)\theta^i \tag{1}$$

The derivation process of the EMA algorithm is as follows:

$$\begin{aligned}
v^n &= \beta v^{n-1} + (1-\beta)\theta^n \\
&= \beta^2 v^{n-2} + (1-\beta)\beta\theta^{n-1} + (1-\beta)\theta^n \\
&= ... \\
&= \beta^n v^1 + (1-\beta)\sum_{k=0}^{n-1}\beta^k\theta^{n-k} \\
&= \beta^n\theta^1 + (1-\beta)(\beta^{n-1}\theta^1 + \beta^{n-2}\theta^2 + ... + \beta^1\theta^{n-1} + \theta^n) \\
&= \beta^n\theta^1 + (1-\beta)(\beta^{n-1}\theta^1 + ... + \theta^1 - \sum_{k=1}^{n-1}\nabla_L(\theta^k)) \\
&= \theta^1 - \sum_{k=1}^{n-1}(1-\beta^{n-k})\nabla_L(\theta^k)
\end{aligned} \tag{2}$$

And the derivation of the EWF algorithm is as follows:

$$\begin{aligned}
\theta_{balance} &= \alpha\theta^n + (1-\alpha)\theta^1 \\
&= \theta^1 - \alpha\sum_{k=1}^{n-1}\nabla_L(\theta^k)
\end{aligned} \tag{3}$$

As shown in Eq.2 and Eq.3, EMA concentrates more on the early steps, and pays less attention to the gradient of subsequent steps. We have shown that the similarity of early steps drops dramatically, leading to noisy information. On the contrary, our method focuses more evenly on the gradient of each part, which results in better performance.

B. Baseline Details

In this section, we describe more about the loss functions and implementation details in our algorithm.

Objectives. In this work, we use distillation to enhance our weight fusion strategy. Feature-based distillation and logits-based distillation are two main schemes in distillation methods. The representative of the former in class incremental semantic segmentation is PLOP [2], while the one for the latter is MiB [1]. In MiB [1], the distillation loss can be formulated as L_{unce} and L_{unkd}.

$$L_{unce} = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\log\hat{p}_t(i, y_i), \tag{4}$$

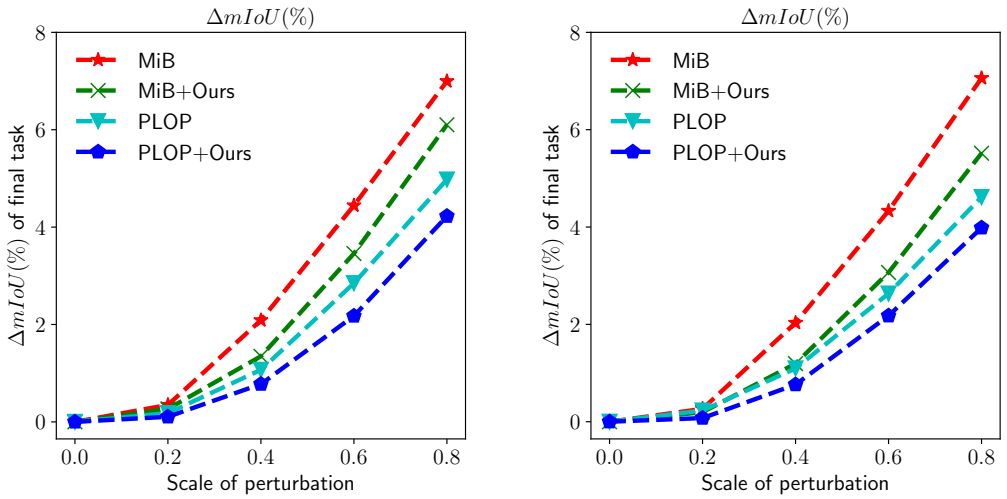
$$L_{unkd} = -\frac{1}{|\mathcal{I}|}\sum_{k\in\mathcal{C}}\sum_{i\in\mathcal{I}} q_{t-1}(i,k)\log\hat{q}_t(i,k), \tag{5}$$

In Eq.4, $y_i \in \{0, C_t\}$ denotes the ground-truth label for the i-th pixel. And $\hat{p}_t(i)$ is modified from the predictions of current model $p_t(i)$, regarding all old classes as *background*. In Eq.5, q_{t-1} is the prediction of the old model, and $\mathcal{C}$ denotes all old classes and the *background* class. The $\hat{q}_t(i)$ is modified by predicted scores $q_t(i)$ of the current model. All new classes are treated as the *background* class. To further stabilize the training process of the classifier, and benefit distillation loss, we fix the previously learned classifiers and only learn the newly added classifier.

In PLOP [2], the distillation can be formulated as L_{pod}.

$$L_{pod} = \frac{1}{L}\sum_{l=1}^{L}||\Phi(f_l^t(x)) - \Phi(f_l^{t-1}(x))||^2 \tag{6}$$

where $f_l^t(x)/f_l^{t-1}(x)$ denotes features of different stages from new and old models, respectively. And Φ denotes the multi-scale strip pooling operation.



(a) $\Delta mIoU(\%)$ on all classes

(b) $\Delta mIoU(\%)$ on old classes

Figure 1. The change of mIoU when perturbing the network parameters for both old classes and all classes (the lower the better).

Method \ Task	A	B	C	D	E	
ILT [3]	9.20	16.74	12.16	11.49	15.60	13.04 ± 2.76
MiB [1]	32.20	20.15	36.05	38.91	53.73	36.21 ± 10.8
PLOP [2]	54.60	47.43	53.43	58.25	47.20	52.18 ± 4.28
RC-IL [5]	59.40	54.05	55.63	55.29	63.19	57.51 ± 3.35
MiB+EWF (ours)	65.56	59.15	63.37	63.52	64.31	63.19 ± 2.16

Table 1. The mIoU (%) of the final step. We conduct experiments on different class orders on 15-1 overlapped setting. The purple denotes the mean mIoU (%) and standard variance over five different class orders.

C. Further Analyses

C.1. Different Class Orders

In order to verify the robustness of different class orders for our method, following PLOP [2], we utilize *15-1 overlapped* setting to run five class orders and show the standard variance and mean value. It contains a sequential order and four random orders provided by the code of PLOP [2]. The result can be found in Table 1. Ours is more robust to different orders and also obtains the best performance in terms of average mIoU. The five orders are defined as:

$$
\begin{aligned}
&A: \{[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15],[16],[17],[18],[19],[20]\},\\
&B: \{[0,12,9,20,7,15,8,14,16,5,19,4,1,13,2,11],[17],[3],[6],[18],[10]\},\\
&C: \{[0,13,19,15,17,9,8,5,20,4,3,10,11,18,16,7],[12],[14],[6],[1],[2]\}\\
&D: \{[0,15,3,2,12,14,18,20,16,11,1,19,8,10,7,17],[6],[5],[13],[9],[4]\},\\
&E: \{[0,7,5,3,9,13,12,14,19,10,2,1,4,16,8,17],[15],[18],[6],[11],[20]\}.
\end{aligned} \tag{7}
$$

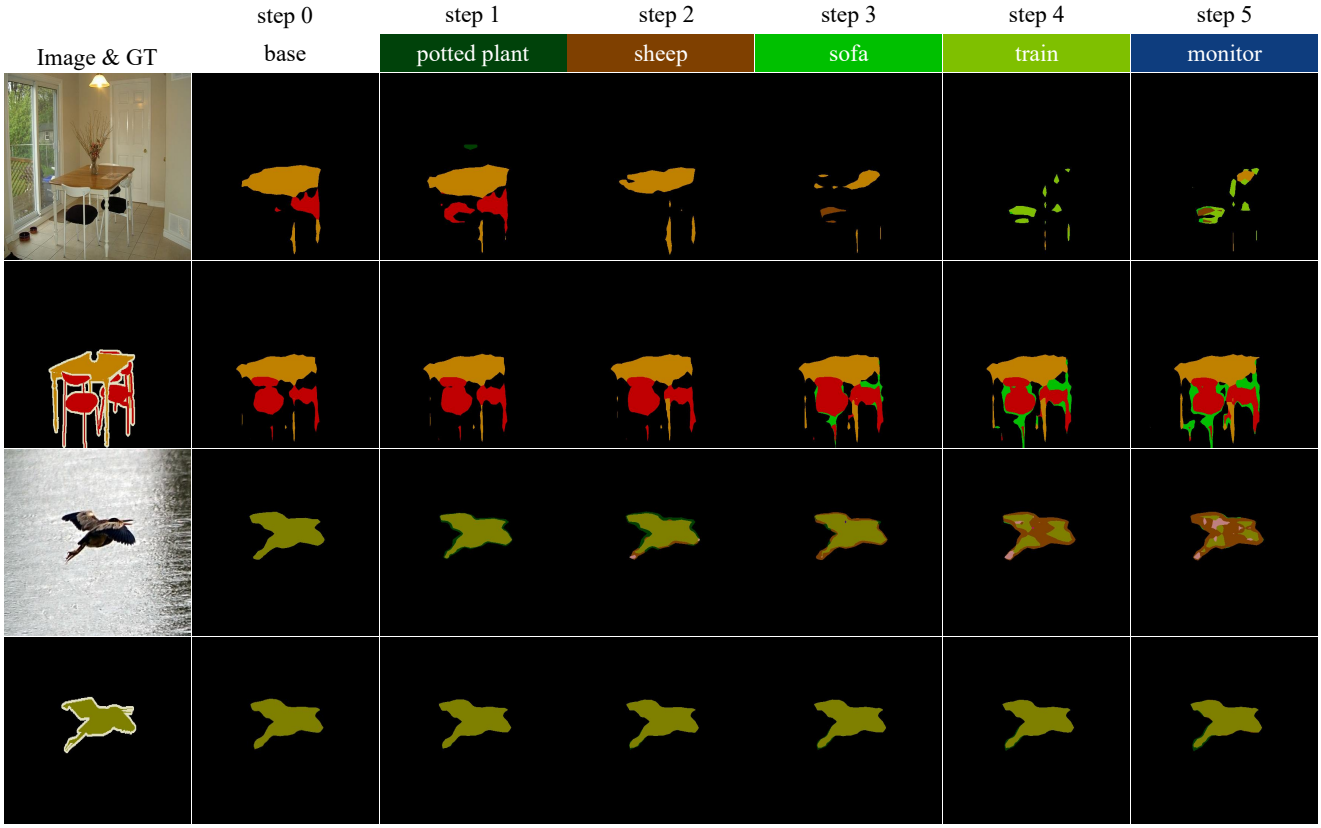
C.2. Model Robustness Analysis

With a new task being learned, f_θ is required to perform well over both old and new tasks. Different from multi-task learning, the second step of optimization does not take into account the optimization goal of the first step. In that case, the model will easily bias to new data and trigger catastrophic forgetting of old data. Nevertheless, model with strong robustness naturally has the capability against catastrophic forgetting. Thus, we test the robustness of the original model (*e.g.,* PLOP [2], MiB [1]) and the improved model against disturbances with our algorithm.

To observe the fluctuation caused by the perturbation ϵ, we choose the parameters of the first three sets of convolutions as the target of perturbation. ϵ is randomly selected from gaussian distribution (*i.e.,* $\mathcal{N}(0,1)$), and it will be scaled with a scale factor $\gamma \in [0,1]$. The fluctuation can be viewed as:

$$
F_\theta^{\gamma\epsilon} = P(\theta + \gamma\epsilon; \mathcal{M}) - P(\theta; \mathcal{M}), \tag{8}
$$

Figure 2. The qualitative comparison between different methods. All prediction results are from the last step of 15-1 overlapped setting. The odd rows are the results of MiB [1], and the even rows are ours.



where $\mathcal{M}$ denotes the whole test set and P denotes the performance gained by parameter θ on $\mathcal{M}$. In the case where the norm of ϵ is bounded, the value of $F_\theta^{\gamma\epsilon}$ can represent the robustness of model θ. As shown in Fig. 1 (a)-(b), our method does enhance the robustness of the original model, both in PLOP [2] and MiB [1].

C.3. More Analysis about EWF Performance Improvement.

The new classes in the main tables include all classes learned after the base step. For instance, class 11 in 10-1 the setting is a new category in the first incremental phase but an old category in subsequent phases. Even if our method has a certain drop of accuracy for the new class during fusion, in the subsequent tasks, the forgetting is greatly reduced by our method, thus resulting in better final accuracy.

C.4. Analysis on performance gap in Pascal VOC 5-3 setting

PLOP adopts pseudo-label strategy and uses a threshold to filter out confident regions. When the number of base classes is small, the model introduces noisy pseudo labels, leading to much worse performance.

C.5. Ablation study for RC-IL's distillation and apple-to-apple comparison with EWF.

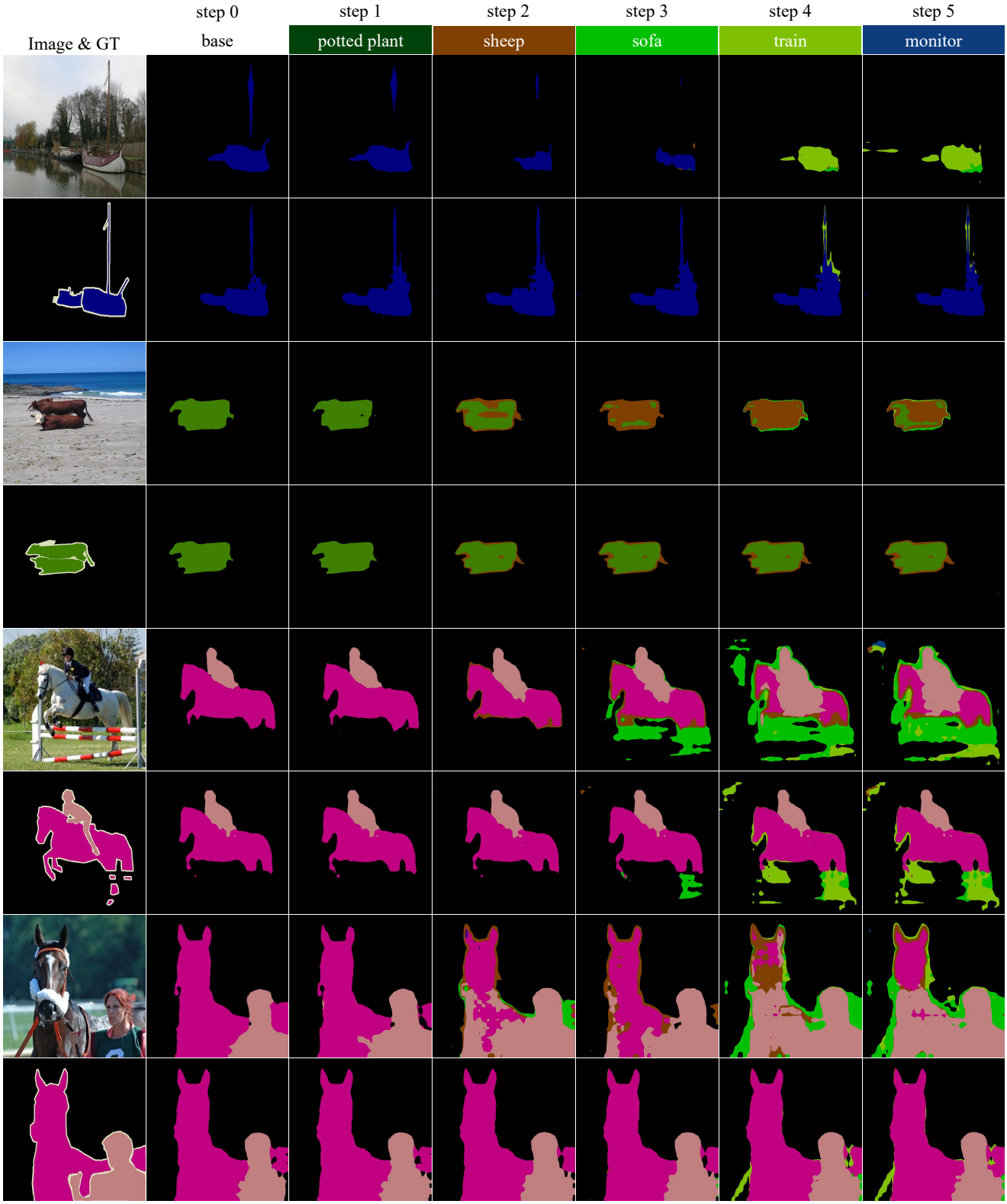
In order to show the adaptability of our method to different distillations, we use the PCKD proposed by the current state-of-the-art RCIL to combine with our method to observe the performance change. We apply PCKD distillation used in RC-IL [5] to compare to RC-IL directly, and it shows that our method is much better than RC-IL [5]. The experiments is conducted in *PASCAL VOC 2012* 15-1 setting and 10-1 setting.

Setting	Method	old	new	all
15-1	RC-IL (PCKD)	70.6	23.7	59.4
	PCKD + EWF	**77.6**	**34.4**	**67.3**
10-1	RC-IL (PCKD)	55.4	15.1	34.3
	PCKD + EWF	**70.0**	**31.6**	**51.8**

C.6. More Qualitative Results.

We display more results in Fig.2 and Fig.3 with MiB [1] and ours for visualization comparison. It is clear that our method has obtained a significant improvement in visual quality.

Figure 3. More visualization comparison between MiB [1] and ours.



References

[1] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *CVPR*, pages 9233–9242, 2020. 1, 2, 3, 4

[2] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *CVPR*, 2021. 1, 2, 3

[3] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *ICCVW*, 2019. 2

[4] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 1

[5] Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *CVPR*, 2022. 2, 3