Early Preparation Pays Off: New Classifier Pre-tuning for Class-Incremental Semantic Segmentation
Supplementary Materials

Zhengyuan Xie[1], Haiquan Lu[1], Jia-wen Xiao[1], Enguang Wang[1], Le Zhang[3], and Xialei Liu[1,2(✉)]

[1] VCIP, CS, Nankai University
[2] NKIARI, Shenzhen Futian
{xiezhengyuan}@mail.nankai.edu.cn
{xialei}@nankai.edu.cn
[3] SICE, UESTC

A Baseline Details

In this section, we introduce the *baselines* used in experiments.

MiB. In MiB [2], two kinds of loss are used to model the background, *i.e.*, $\mathcal{L}_{unce}$ and $\mathcal{L}_{unkd}$:

$$\begin{aligned}\mathcal{L}_{unce} &= -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\log\tilde{q}_x^t(i,y_i)\,,\\ \mathcal{L}_{unkd} &= -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{c\in\bigcup_{j=1}^{t-1}c_j} q_x^{t-1}(i,c)\log\hat{q}_x^t(i,c)\,,\end{aligned} \tag{1}$$

where $\mathcal{I}$ denotes the pixel set of an image, $y_i \in c_{bg}\cup\mathcal{C}_t$ denotes the ground-truth label of pixel i, q_x^t denotes the output of the model at step t, $\tilde{q}_x^t$ and $\hat{q}_x^t$ denotes the modified output of the current model, considering the old classes for the cross entropy loss and new classes for the knowledge distillation loss.

PLOP. Different from MiB [2], PLOP [4] utilizes pseudo-labeling to address the issue of background shift, as follows:

$$\mathcal{L}_{pseudo} = -\frac{\nu}{WH}\sum_{w,h}^{W,H}\sum_{c\in\mathcal{C}_t}\tilde{S}(w,h,c)\log\hat{S}^t(w,h,c)\,, \tag{2}$$

where $\hat{S}$ denotes the prediction of the model and $\tilde{S}$ denotes the pseudo-labels generated by the old model in the previous step.

It also distills intermediate features by Local POD, as follows:

$$\mathcal{L}_{LocalPod} = \frac{1}{L}\sum_{l=1}^{L}\left|\left|\Phi(f_l^t(I)) - \Phi(f_l^{t-1}(I))\right|\right|\,, \tag{3}$$

where L denotes the number of layers, Φ denotes the operation of Local POD embedding extraction, $f_l^t(I)$ denotes the output feature from the layer l with the input I.

RCIL. RCIL [8] decouple the remembering of old knowledge and the learning of new knowledge by adding a parallel module composed of a convolution layer and a normalization layer for each 3×3 convolution module. At step 0, all parameters are trainable. At the beginning of each incremental step, two branches of the old model are fused into one frozen branch to memorize the old knowledge, while the other branch is learnable. A drop path strategy is also used when fusing the outputs of two branches, which can be denoted as:

$$x_{out} = \eta \cdot x_1 + (1 - \eta) \cdot x_2 \,, \tag{4}$$

where x_{out} denotes the fused output, x_1 and x_2 denotes the outputs from two branches, and η denotes a channel-wise weight vector. For training process, η is sampled from the set $\{0, 0.5, 1\}$ and for evaluation η is set to 0.5. RCIL also proposed a Pooled Cube Knowledge Distillation, using average pooling operation on spatial and channel dimensions.

B Results and Analysis of Disjoint Settings

In *Disjoint* settings, at each step, the *bg* classifier will not see any future class, leading to MiB's initialization struggling in these settings, while our NeST leverages semantic knowledge from old classifiers to generate new classifiers for initialization, the pre-tuning process also benefits the stability of the model. Results of NeST and *baselines* on *15-1 Disjoint* and *10-1 Disjoint* settings are shown in Tab. 1, indicating that NeST can significantly improve the performance of previous methods in *Disjoint* settings.

Table 1: Results of disjoint settings on Pascal VOC 2012 dataset.

Method	*15-1 Disjoint*	*10-1 Disjoint*
MiB	38.6	2.0
MiB+NeST	**41.0**	**20.5**
PLOP	40.7	12.6
PLOP+NeST	**52.7**	**22.4**

C Experiments of COCO-Stuff 10K

To prove the ability to apply our NeST in scenarios with more classes, we introduce another dataset for class incremental semantic segmentation, COCO-Stuff 10K, to evaluate the effectiveness of our method. COCO-Stuff 10K includes 80

thing classes and 91 stuff classes, which is a subset of the original COCO-Stuff dataset. We evaluate our NeST on the *80-91* (2 Steps) overlapped setting, which contains more classes than ADE20K and Pascal VOC 2012. As shown in Tab. 2, our method can handle scenarios with more classes in one step.

Table 2: Results of *80-91* overlapped setting on COCO-Stuff 10K dataset.

Method	*0-80*	*81-171*	*all*
MiB	40.2	18.6	28.9
MiB+NeST	41.9	20.7	**30.7**
PLOP	46.1	17.0	30.8
PLOP+NeST	46.0	18.8	**31.6**

D Comparisons with Transformer-based SOTA Methods

In recent years, many Transformer-based CISS methods have emerged, here we briefly discuss the differences between NeST and these methods. Comformer [1] uses a universal segmentation model Mask2Former [3] to do mask classification for continual panoptic segmentation and continual semantic segmentation. CoinSeg [9] introduces a pretrained Mask2Former model as class-agnostic mask generator. Incrementer [5] sequentially adds tokens of new classes and performs dot production between features and updated class tokens to generate segmentation prediction results. The *baseline* is based on a simple per-pixel classification model SETR with ViT-B as the backbone, while equipped with NeST, it can achieve SOTA performances, as shown in Tab. 3. Moreover, NeST has the potential to be integrated into these transformer-based methods, and we leave it as our future work.

Table 3: Comparisons with Transformer-base SOTA methods.

Method	Backbone	Model	*15-1*	*15-5*	*10-1*
CoinSeg	Swin-B	Deeplab+Mask2Former	75.5	77.6	70.5
Incrementer	ViT-B	Segmenter	75.5	79.9	70.2
MiB	ViT-B	SETR	53.3	80.2	25.5
MiB+NeST (Ours)	ViT-B	SETR	**76.5**	**80.3**	**71.9**

E More Implementation Details

Weight Align. To prevent the new classifiers' weight from being too large during the pre-tuning process, we apply Weight Aligning (WA) [10] as follows:

$$\hat{w}_{new} = w_{new} \cdot \frac{Mean(Norm_{old})}{Mean(Norm_{new})} \tag{5}$$

where $Norm_{old}$ and $Norm_{new}$ denote norms of old and new classifiers' weights, $Mean(\cdot)$ denotes the operation of calculating mean values. Relevant experiment results in Tab. 4 show that WA can correct the biased weight thus boosting the performance of NeST.

Table 4: Ablation study of Weight Align for NeST. All performances are reported on the *15-1* setting.

Method	*0-15*	*16-20*	*all*
MiB+NeST w/o WA	58.4	10.9	47.1
MiB+NeST w/ WA	61.7	20.4	**51.8**
PLOP+NeST w/o WA	72.5	32.4	62.9
PLOP+NeST w/ WA	72.2	33.7	**63.1**

Fix old classifiers. We find that the pseudo-labeling strategy may change the geometric structure of old classifiers severely, which has a detrimental impact on our method. This phenomenon is particularly obvious on the Pascal VOC 2012 dataset. To preserve the old knowledge learned in previous steps, following EWF [7], we fix old classifiers in the formal training steps on settings of Pascal VOC 2012. Relevant experiment results are shown in Tab. 5.

Table 5: Ablation study of fixing previous classifiers for our method based on PLOP [4]. All performances are reported on the *15-1* setting.

Method	*0-15*	*16-20*	*all*
PLOP w/ fix	56.9	11.3	46.0
PLOP+NeST w/o fix	66.8	20.2	55.7
PLOP+NeST w/ fix	72.2	33.7	**63.1**

F Further Analysis

Effectiveness of the importance matrix. For plasticity, we learn to generate a new classifier with relevant old classifiers. In particular, the importance matrix

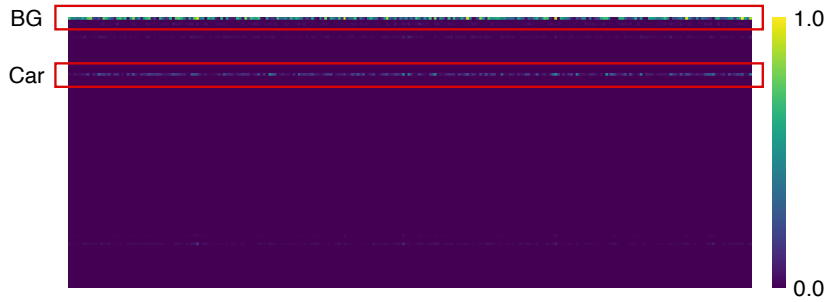

Fig. 1: Visualization of the importance matrix on ADE20K *100-5* step1.

can capture the semantic relationship between old and new classifiers on the channel level. To verify this, we visualize the importance matrix $M \in \mathbb{R}^{100\times256}$ of the new class *van* on ADE20K *100-5* step1. We normalized the absolute values to [0, 1] and a lighter color means a higher value. As shown in Fig. 1, the bg class (row: 0) and car class (row: 21) make the largest contributions. It is intuitive, as the class van may appear in old data, labeled as bg, and van and old class car are closer in semantic relationship.

Different class orders. To evaluate the effectiveness of our method, following PLOP [4], we use five different class orders of Pascal VOC 2012 *15-1* overlapped setting, as follows:

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

More qualitative results. More qualitative results are shown in Fig. 2 and Fig. 3. By applying the pre-tuning process, our method can help the model preserve old knowledge.

Moreover, to validate the effectiveness of the matrix initialization, we also visualize the class activation map for the last class *tv/ monitor*. To visualize segmentation CAMs, we adopt the method proposed in [6]. As shown in Fig. 4, with the designed matrix initialization strategy, the model can pay more attention to areas of new classes.

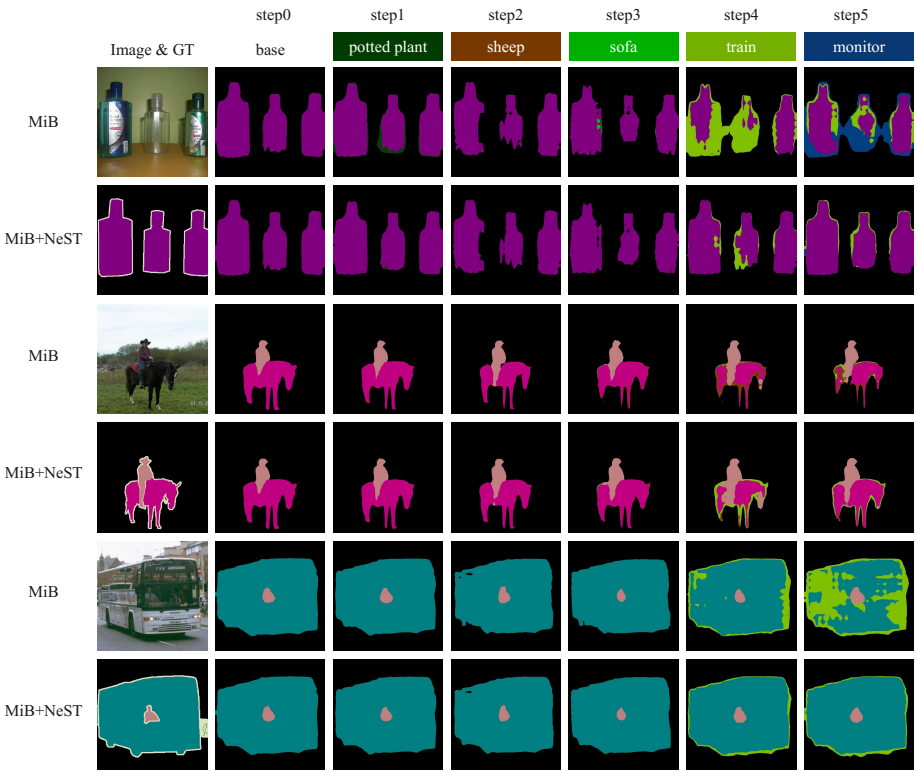

Fig. 2: More qualitative results. All experiments are conducted on the *15-1* setting.

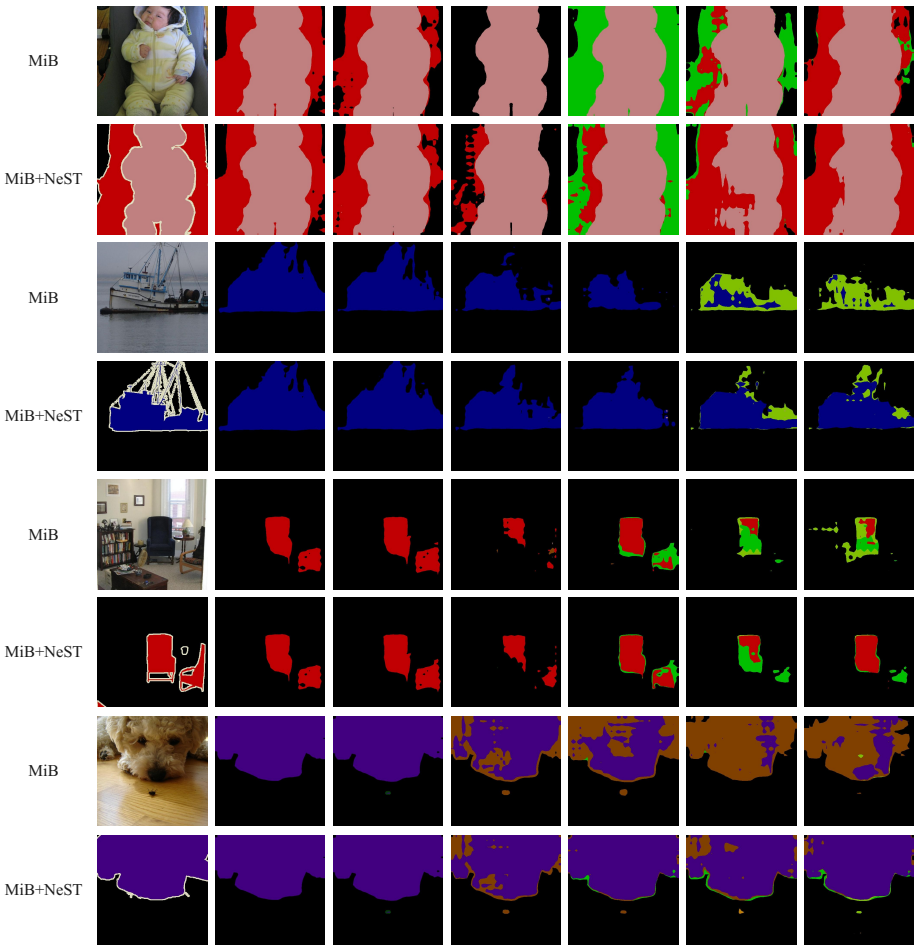

Fig. 3: More qualitative results. All experiments are conducted on the *15-1* setting.

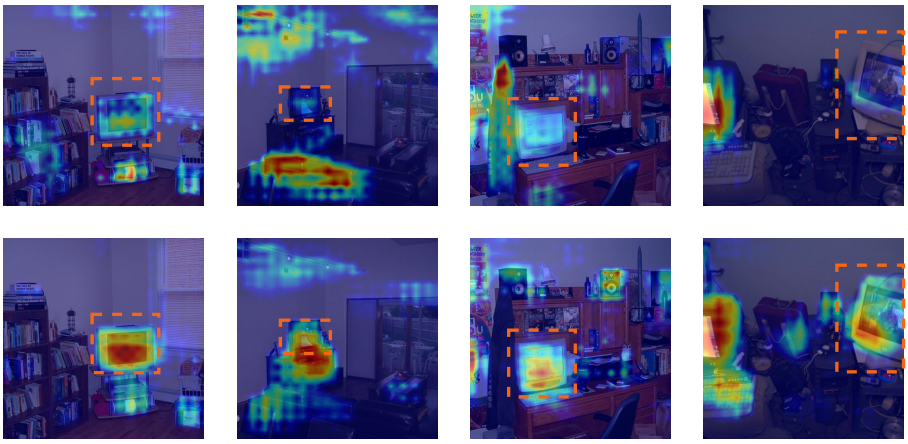

Fig. 4: Class activation maps for the last class *tv/monitor* on the *15-1* setting *w/o* (the top row) and *w/* (the bottom row) our matrix initialization strategy.

References

1. Cermelli, F., Cord, M., Douillard, A.: Comformer: Continual learning in semantic and panoptic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3010–3020 (2023)
2. Cermelli, F., Mancini, M., Bulo, S.R., Ricci, E., Caputo, B.: Modeling the background for incremental learning in semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9233–9242 (2020)
3. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1290–1299 (2022)
4. Douillard, A., Chen, Y., Dapogny, A., Cord, M.: Plop: Learning without forgetting for continual semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
5. Shang, C., Li, H., Meng, F., Wu, Q., Qiu, H., Wang, L.: Incrementer: Transformer for class-incremental semantic segmentation with knowledge distillation focusing on old class. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7214–7224 (2023)
6. Vinogradova, K., Dibrov, A., Myers, G.: Towards interpretable semantic segmentation via gradient-weighted class activation mapping (student abstract). In: AAAI. vol. 34, pp. 13943–13944 (2020)
7. Xiao, J.W., Zhang, C.B., Feng, J., Liu, X., van de Weijer, J., Cheng, M.M.: Endpoints weight fusion for class incremental semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7204–7213 (2023)
8. Zhang, C.B., Xiao, J.W., Liu, X., Chen, Y.C., Cheng, M.M.: Representation compensation networks for continual semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7053–7064 (2022)
9. Zhang, Z., Gao, G., Jiao, J., Liu, C.H., Wei, Y.: Coinseg: Contrast inter-and intra-class representations for incremental segmentation. In: Int. Conf. Comput. Vis. pp. 843–853 (2023)
10. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 13208–13217 (2020)