Generalized Few-shot Semantic Segmentation
Supplementary Material

Zhuotao Tian[1] Xin Lai[1] Li Jiang[2] Shu Liu[3] Michelle Shu[4] Hengshuang Zhao[5,6] Jiaya Jia[1,3]

[1]CUHK [2]MPI Informatics [3]SmartMore [4]Cornell University [5]HKU [6]MIT

Overview

This is the supplementary file for our submission titled *Generalized Few-shot Semantic Segmentation*. This material supplements the main paper with the following content:

- (1) **Implementation Details**.
  - (1.1) **Training Configurations**.
    - (1) General Configuration.
    - (2) Configuration for GFS-Seg.
    - (3) Configuration for FS-Seg.
  - (1.2) **Implementation Details in FS-Seg**.
    - (1) CAPL on PANet.
    - (2) CAPL on PFENet.
  - (1.3) **Transferring FS-Seg Models to GFS-Seg**.
- (2) **Alternative Prototype Fusion Strategies**.
- (3) **Use of Existing Assets**.
- (4) **Visual Illustration**.

1. Implementation Details

1.1. Training Configurations

General Configuration. Our experiments are conducted on Pascal-VOC [4] with augmented data from [5]. Following Pascal-5^i [12], 20 classes in Pascal-VOC are evenly divided into four splits and each split contains 5 classes to cross-validate the performance of models. Specifically, when using one split for providing novel classes, classes of the other three splits and the background class are treated as 16 base classes. Different from FS-Seg where the evaluation is only performed on the novel classes of test samples, GFS-Seg models predict labels from both base and novel classes simultaneously on test images from the validation sets. As the number of either base or novel classes increases, the generalized few-shot task becomes more challenging. Therefore, the experiments on COCO [8] are also included with cross-validation on four splits following [1, 9, 14]. COCO is a rather challenging dataset with an overwhelming number of categories, *i.e.*, 60 base and 20 novel classes for each split.

We build our models on PyTorch. We keep the backbone configuration of PSPNet with the output of the feature extractor being $\frac{1}{8}$ of the input spatial size. The weights of the last 1×1 convolution layer (classifier) of size $[512, N, 1, 1]$ are used as N 512-dimensional prototypes for N classes. The backbone weights are pre-trained on ImageNet [11]; other layers are initialized by the default setting of PyTorch. For later experiments on DeepLab-V3 [3], prototypes have 256 dimensions and the backbone configuration is the same as that of PSPNet. For each benchmark, we take the average of all splits as the final performance, and our experimental results in GFS-Seg are averaged over five different random seeds. All predictions are evaluated on the original labels without resizing.

Data augmentation on training images includes mirroring and re-scaling from 0.5 and 2.0, and random rotation from -10 to 10 degrees. Finally, we randomly crop 473×473 patches as training samples, following the official implementation of PSPNet [18]. We output the prediction without additional post-processing (e.g., fully connected conditional random field (CRF) [6] and multi-scale testing). All experiments are run on NVIDIA Titan X GPU, and predictions are evaluated with the original labels without resizing.

Configuration for GFS-Seg. We use cross-entropy loss in CAPL and SGD for optimization. Weight decay and momentum are set to 0.0001 and 0.9 respectively. The 'poly' learning rate decay [2] is used by multiplying the initial learning rate with $(1 - \frac{current\_iter}{max\_iter})^{power}$, where $power$ is set to 0.9. We train all models for 50 epochs on both Pascal-5^i and COCO-20^i. For models on Pascal-5^i, we set the initial learning rate and training batch size to 0.01 and 6 with a single GPU. For models on COCO-20^i, the initial learning rate and training batch size are 0.01 and 12 with two GPUs. The two-layers MLP for producing γ_{sup} in SCE is constructed by two fully-connected layers with the intermediate ReLU activation function. The output of MLP is then processed by the Sigmoid function.

Configuration for FS-Seg. The backbone configurations are the same as the ones used in [7, 14, 16]. Similar to [15], our models are trained for 30,000 and 60,000 steps on Pascal-5^i and COCO-20^i respectively with the same initial learning rate 0.0005 and batch size 3 on a single GPU. SGD is adopted as the optimizer. Weight decay and momentum for FS-Seg experiments are set to 0.0005 and 0.9, and the learning rate is reduced by 0.1 every 10,000 / 20,000 iterations for Pascal-5^i and by every 20,000 / 40,000 iterations for COCO-20^i. Our 5-shot results are obtained by evaluating the models trained in the 1-shot setting following [9, 14–17].

1.2. Implementation Details in FS-Seg.

In the main paper, we have applied the proposed CAPL to two representative FS-Seg models, *i.e.*, PANet [15] and PFENet [14]. Due to the page limit of the main paper, the detailed descriptions regarding the implementations are presented as follows.

CAPL on PANet. We first apply our method to PANet whose predictions are directly yielded by calculating the cosine similarity between query features and fore-/background prototypes. Since the base classes are not examined during the evaluation phase of FS-Seg, both fore- and background prototypes are the context-providers. Therefore, we simply combine the SCE and DQCE and use two factors $\gamma^i \quad (i \in \{fg, bg\})$ to enrich the contextual information for fore- and background prototypes respectively:

$$\boldsymbol{p}^i_{capl} = \gamma^i * \boldsymbol{p}^i + (1 - \gamma^i) * \boldsymbol{p}^i_{qry}, \quad i \in \{fg, bg\}. \tag{1}$$

Specifically, $\boldsymbol{p}^i$ denotes the prototype yielded by the original method of PANet, *i.e.*, mask pooling the support features, and $\boldsymbol{p}^i_{qry}$ is the query prototypes obtained via Eq. (5) of our main paper. γ^i directly balances the contributions of them to enrich the contextual hints, and the final prediction is obtained by calculating the cosine similarity between $\boldsymbol{p}^i_{capl}$ and the

query features, instead of the original $\boldsymbol{p}^i$. Similar to GFS-Seg where γ^i_{sup} and γ^i_{qry} are independently produced by $\mathcal{G}_{sup}$ (*i.e.*, a two-layers MLP) and $\mathcal{G}_{qry}$ (*i.e.*, cosine similarity), CAPL in FS-Seg also leverages the merits of them by applying both MLP and cosine similarity to $\boldsymbol{p}^i$ and $\boldsymbol{p}^i_{qry}$, written as:

$$\gamma^i = \frac{1}{2} * (Cos(\boldsymbol{p}^i, \boldsymbol{p}^i_{qry}) + MLP(\boldsymbol{p}^i, \boldsymbol{p}^i_{qry})), \quad i \in \{fg, bg\}. \tag{2}$$

The original training loss of PANet consists of the main segmentation loss $\mathcal{L}_{seg}$ and the prototype alignment regularization loss $\mathcal{L}_{PAR}$. PANet+CAPL does not alter these training objectives, but the predictions supervised by $\mathcal{L}_{seg}$ are yielded by the enhanced prototype $\boldsymbol{p}^i_{capl}$, instead of the original prototype $\boldsymbol{p}^i$.

We investigate the effectiveness of applying CAPL to PANet in Table 1 where 'CAPL (Cos)' means only cosine similarity is adopted in Eq. (2) and 'CAPL (Cos+MLP)' represents both Cos and MLP are used for generating γ^i. It is observed that CAPL (Cos) can yield decent performance gain to the baseline PANet and MLP further improves it by introducing an additional data-conditioned adaptation to γ^i.

	1-shot					5-shot				
Methods	Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
PANet	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7
PANet*	42.6	58.0	54.8	43.8	49.8	49.1	65.6	60.9	50.9	56.6
PANet + CAPL(Cos)	58.0	**67.1**	63.0	52.3	60.1	60.1	**71.7**	68.3	60.5	65.2
PANet + CAPL(Cos + MLP)	**58.1**	66.9	**63.8**	**53.7**	**60.6**	**62.9**	**71.7**	**68.7**	**61.2**	**66.1**

Table 1. Ablation study on applying CAPL to models in FS-Seg. PANet denotes the results reported in the original paper. PANet* is reproduced by our training configuration.

CAPL on PFENet. Different from PANet whose predictions are directly produced by the cosine similarity between the fore- and background prototypes, PFENet adopts several convolutional blocks to process the concatenation of middle-level query and support features. Besides, the reasoning between query and support features is guided by an additional prior mask obtained from the pixel-wise correlation between high-level query and support features. In GFS-Seg, CAPL is applied to the last high-level feature map so as to fully exploit the semantic knowledge, therefore we consider adopting CAPL on PFENet to enhance the utilization of high-level features.

Concretely, different from the original prior mask generation method that takes the maximum values from pixel-to-pixel correlation map as the prior mask, we apply mask-pooling on the high-level support features to get the support fore- and background prototypes $\boldsymbol{p}^i$ ($i \in \{fg, bg\}$). Then, the proposed CAPL yields enhanced prototypes $\boldsymbol{p}^i_{capl}$ ($i \in \{fg, bg\}$) by following Eqs. (1)-(2). After that, the high-level prediction $\boldsymbol{y}_{high} \in \mathbb{R}^{h_q \times w_q \times 2}$ is produced by measuring the cosine similarity between the high-level query features and $\boldsymbol{p}^i_{capl}$ ($i \in \{fg, bg\}$). Besides, we apply the Softmax operation to $\tau * \boldsymbol{y}_{high}$ and take the dimension belonging to the foreground as the prior mask $\mathcal{M} \in \mathbb{R}^{h_q \times w_q}$. The pseudo code can be written as:

$$\mathcal{M} = Softmax(\tau * \boldsymbol{y}_{high}, \ \ axis = -1)[:, :, 1], \tag{3}$$

where τ is set to 10 and Eq. (3) is similar to the inference process of Eq. (2) in the main paper. Finally, according to [14], $\mathcal{M}$ is then processed by the min-max normalization and concatenated with the middle-level features for yielding the final prediction $\boldsymbol{y}$.

However, the high-level features are fixed in the original training scheme [14] in order to maintain high generalization ability to unseen classes, hence the updated prototypes $\boldsymbol{p}^i_{capl}$ ($i \in \{fg, bg\}$) are less adaptive. We alternatively make the backbone trainable to be able to be benefited by the proposed CAPL, and the comparison is shown in Table 2 where simply letting the backbone trainable causes considerable 1-shot performance deduction as shown by the results of PFENet† and PFENet*, while the average 5-shot performance of PFENet* is close to that of the default configuration. Contrarily, CAPL significantly improves the baseline model PFENet* and the final model PFENet*+CAPL considerably surpasses the original one in terms of both 1- and 5-shot settings, manifesting its effectiveness.

Further, we investigate whether CAPL is conducive to the middle-level features in Table 2, but results of 'PFENet* + CAPL + CAPL$_{mid}$' show that the middle-level features are less informative than the high-level ones and thus the performance is degraded by further applying CAPL to the middle-level features that are less semantic sensitive. Also, noises contained in the middle-level features might worsen the representation power of the newly formed prototypes, causing performance deduction.

Methods	1-shot					5-shot				
	Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
PFENet	**61.7**	**69.5**	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9
PFENet†	61.0	69.7	57.9	56.5	61.3	62.5	70.4	58.5	57.8	62.3
PFENet† + CAPL	60.5	68.9	58.3	56.2	61.0	61.7	70.2	59.3	57.0	62.1
PFENet*	53.0	66.7	58.8	50.7	57.3	60.2	70.6	62.6	56.5	62.5
PFENet* + CAPL$_{mid}$	52.8	67.3	60.4	50.4	57.7	59.9	70.9	63.5	56.4	62.7
PFENet* + CAPL	61.4	67.6	**64.0**	**55.7**	**62.2**	**66.7**	**72.0**	68.1	**61.6**	**67.1**
PFENet* + CAPL + CAPL$_{mid}$	60.1	68.3	63.9	55.0	61.8	64.7	**72.0**	**68.5**	61.5	66.6

Table 2. Ablation study on applying CAPL to models in FS-Seg. PFENet denotes the results reported in the original paper. PFENet† is reproduced by following the default configuration of PFENet [14] where the backbone is fixed. PFENet* and the models implemented with CAPL are reproduced by our training configuration. The backbone parameters of PFENet* are trainable. It is worth noting that, because the CAPL requires updating the backbone parameters, CAPL can be only applied to PFENet* whose backbone parameters are not fixed. CAPL$_{mid}$ means CAPL is applied to the middle-level features to yield enhanced prototypes.

1.3. Transferring FS-Seg Models to GFS-Seg

In the original codes of CANet [17], SCL [16] and PFENet [14], the final classifier consists of two output channels. The following Softmax layer can well tackle the foreground and background segmentation task, but it causes difficulties for multi-class segmentation tasks in GFS-Seg because the foreground confidence cannot be directly compared between different locations. Therefore, we accordingly set the final output channel as one and replace the following Softmax layer with a Sigmoid layer to adapt these models to GFS-seg. Results in Table 3 show that this modification does not adversely affect the performance of CANet, SCL and PFENet in FS-Seg.

Methods	1-shot					5-shot				
	Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
CANet	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1
CANet*	53.6	67.9	54.3	49.0	56.2	55.4	68.4	54.3	50.6	57.2
PFENet	61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	57.9	61.9
PFENet*	61.0	69.7	57.9	56.5	61.3	62.5	70.4	58.5	57.8	62.3
SCL	63.0	70.0	56.5	57.7	61.8	64.5	70.9	57.3	58.7	62.9
SCL*	63.2	69.5	55.9	58.1	61.7	64.8	70.3	57.7	58.0	62.7

Table 3. Reproduced results of PFENet, CANet, SCL and PANet in the setting of FS-Seg. Results marked with $*$ are reproduced by us.

1.4. Ablation study of GFS-Seg models on COCO.

The effectiveness of CAPL (DECE + SCE) on COCO-20^i is shown in Table 4 where the conclusions obtained on Pascal-5^i still hold – DQCE and SCE are complementary. Similarly, worse results are yielded by DQCE-Sw because the new classifier is controlled by untrustworthy $\boldsymbol{p}_{qry}^{b,i}$. Thank you and the ablation study and analysis are added to the revision.

Methods	MLP	Cos	1-shot			5-shot		
			Base	Novel	Total	Base	Novel	Total
Baseline	N/A	N/A	36.68	5.84	29.06	36.91	7.26	29.59
DQCE	✓	-	43.84	6.84	34.70	44.42	10.74	35.91
DQCE	-	✓	44.14	7.22	35.02	44.86	10.35	36.34
DQCE-Sw	-	✓	16.33	5.45	13.65	17.21	6.49	14.93
SCE	✓	-	41.30	7.24	32.89	42.73	9.29	34.48
SCE	-	✓	40.64	7.36	32.42	42.05	9.81	34.09
SCE-Sw	-	✓	41.14	7.47	32.83	43.30	9.98	35.07
CAPL	N/A	N/A	44.61	7.05	35.46	45.24	11.05	36.80

Table 4. Ablation study on COCO-20^i.

2. Alternative Prototype Fusion Strategies

Impressive improvement has been achieved in AMP [13] by fusing new proxies with the previously learned class signatures. Though both CAPL and AMP use weight imprinting and weighted sum for the base class, CAPL's method differs in two aspects: **1)** AMP involves an additional search after training for a fixed γ that is shared by the weighted summations

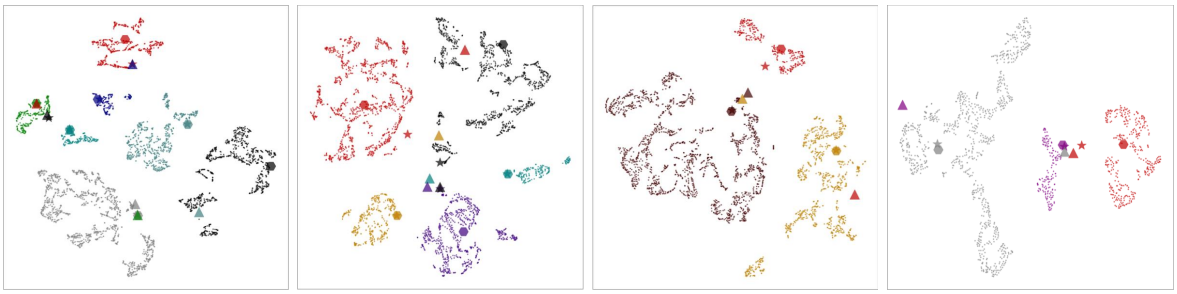

Figure 1. Visual results of t-SNE. Triangles, stars and hexagons are the classifier weights of the baseline, SCE and SCE+DQCE, respectively. Small circles are query features.

of all classes, while our weighting factor is adaptive to different classes because it is conditioned on the input prototypes without additional searching epochs when the training is finished. **2)** The weighted summation is only performed at the *novel class registration phase* in AMP and the model is trained in a normal fashion, but our model meta-learns the behavior during training to better accomplish the context enrichment. Therefore the idea of AMP lies between our baseline and CAPL-Te that incorporates the class representation update during the *novel class registration phase* with a fixed weighting factor, *i.e.*, the converged γ of MLP in CAPL.

To support the discussion above, we show experiments in Table 5 where all results are in the setting of GFS-Seg. Concretely, models IV and VIII represent SCE/DQCE whose γ are determined by the fixed value searched by AMP [13]. Similarly, models V and IX incorporate the SCE/DQCE contextual enrichment strategies but their γ are set to the converged values of models II and VI respectively.

As for SCE, we can observe that even though models IV and V both use the fixed γ values, 'Convg' outperforms 'AMP' because the former is the converged weighting factor that is conditioned on the different input class representations and got through the meta-training phase of CAPL. As the converged γ may not fit the early stage of training, 'Convg' (model V) gets sub-optimal results compared to 'MLP'(model II). However, as MLP is leaning towards minimizing the potential negative impacts brought by the query feature in DQCE, MLP is even less effective than AMP, so the converged values of γ_{qry} also yield inferior performance as shown by model IX. The final CAPL (model X) combines SCE (MLP) and DQCE (Cos).

			1-shot			5-shot		
ID	Methods	γ	Base	Novel	Total	Base	Novel	Total
I	Baseline	N/A	60.47	14.55	49.54	61.88	16.68	51.12
II	SCE	MLP	62.17	17.88	51.63	63.62	20.50	53.35
III	SCE	Cos	59.87	16.80	49.62	61.60	19.92	51.68
IV	SCE	AMP	61.35	14.98	50.31	63.07	17.72	52.27
V	SCE	Convg.	61.40	15.72	50.54	63.59	18.38	52.82
VI	DQCE	MLP	63.25	15.42	51.82	64.12	20.37	53.70
VII	DQCE	Cos	64.16	15.39	52.55	65.26	21.32	54.80
VIII	DQCE	AMP	63.81	15.25	52.25	64.39	20.35	53.90
IX	DQCE	Convg.	63.07	15.03	51.64	63.75	20.15	53.37
X	CAPL	-	**65.48**	**18.85**	**54.38**	**66.14**	**22.41**	**55.72**

Table 5. Comparison of contextual enrichment strategies. 'AMP' represents using the fixed values searched by [13]. 'Convg' means γ is set to the converged value of model-VI whose γ is determined by MLP. Model I is the baseline (PSPNet with ResNet-50) and CAPL (model X) combines SCE (MLP) and DQCE (Cos).

3. Use of Existing Assets

We gratefully thank the creators of the following assets that are very helpful for our project.

- PyTorch 1.6.0 [10]: https://github.com/pytorch/pytorch (The license is available at https://github.com/pytorch/pytorch/blob/master/LICENSE)
- PASCAL-VOC 2012 [4]: http://host.robots.ox.ac.uk/pascal/VOC/voc2012 (The use of image respects the term of use of Flickr at https://www.flickr.com/help/terms)
- SBD [5]: http://home.bharathh.info/pubs/codes/SBD/download.html (The use of image respects the term of use of Flickr at https://www.flickr.com/help/terms)

- COCO 2014 [8] : https://cocodataset.org (Creative Commons Attribution 4.0 License)
- PSPNet [18]: https://github.com/hszhao/semseg (MIT License)
- DeepLab-V3 [3]: https://github.com/pytorch/vision/blob/main/torchvision/models/segmentation/deeplabv3.py (BSD-3-Clause License)
- CANet [17]: https://github.com/icoz69/CaNet (No license published)
- PANet [15]: https://github.com/kaixin96/PANet (No license published)
- SCL [16]: https://github.com/zbf1991/SCL (No license published)
- PFENet [14]: https://github.com/dvlab-research/PFENet (No license published)
- RePRI [1]: https://github.com/mboudiaf/RePRI-for-Few-Shot-Segmentation (No license published)
- HSNet [9]: https://github.com/juhongm999/hsnet (No license published)

4. Visual Illustration

By incorporating DQCE, the classifier dynamically exploits specific contextual hints from each query sample individually, further rectifying the prototypes obtained from support features via SCE. The rectification process is illustrated in Figure 1 where different colors represent different classes. From the t-SNE visualizations, we can observe that the proposed SCE and DQCE are complementary, and are both conducive to the prototype correction.

Visual examples of FS-Seg and GFS-Seg are shown in Figures 2 and 3 respectively. The major issue of the baselines without CAPL is that their results are more likely to be negatively affected by the prototypes of the other classes. Differently, with the enriched context information, CAPL yields fewer false predictions than the baseline. Both our baseline method and CAPL can be easily applied to any normal semantic segmentation model without structural constraints.

References

[1] Malik Boudiaf, Hoel Kervadec, Imtiaz Masud Ziko, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *CVPR*, 2021. 2, 6

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2018. 2

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv*, 2017. 2, 6

[4] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 2010. 2, 5

[5] Bharath Hariharan, Pablo Arbelaez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 2, 5

[6] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 2

[7] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *CVPR*, 2021. 2

[8] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 2, 6

[9] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *ICCV*, 2021. 2, 6

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019. 5

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2

[12] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017. 2

[13] Mennatullah Siam, Boris N. Oreshkin, and Martin Jägersand. AMP: adaptive masked proxies for few-shot segmentation. In *ICCV*, 2019. 4, 5

[14] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *TPAMI*, 2020. 2, 3, 4, 6

[15] Kaixin Wang, JunHao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, 2019. 2, 6

[16] Bingfeng Zhang, Jimin Xiao, and Terry Qin. Self-guided and cross-guided learning for few-shot segmentation. In *CVPR*, 2021. 2, 4, 6

[17] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, 2019. 2, 4, 6

[18] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2, 6

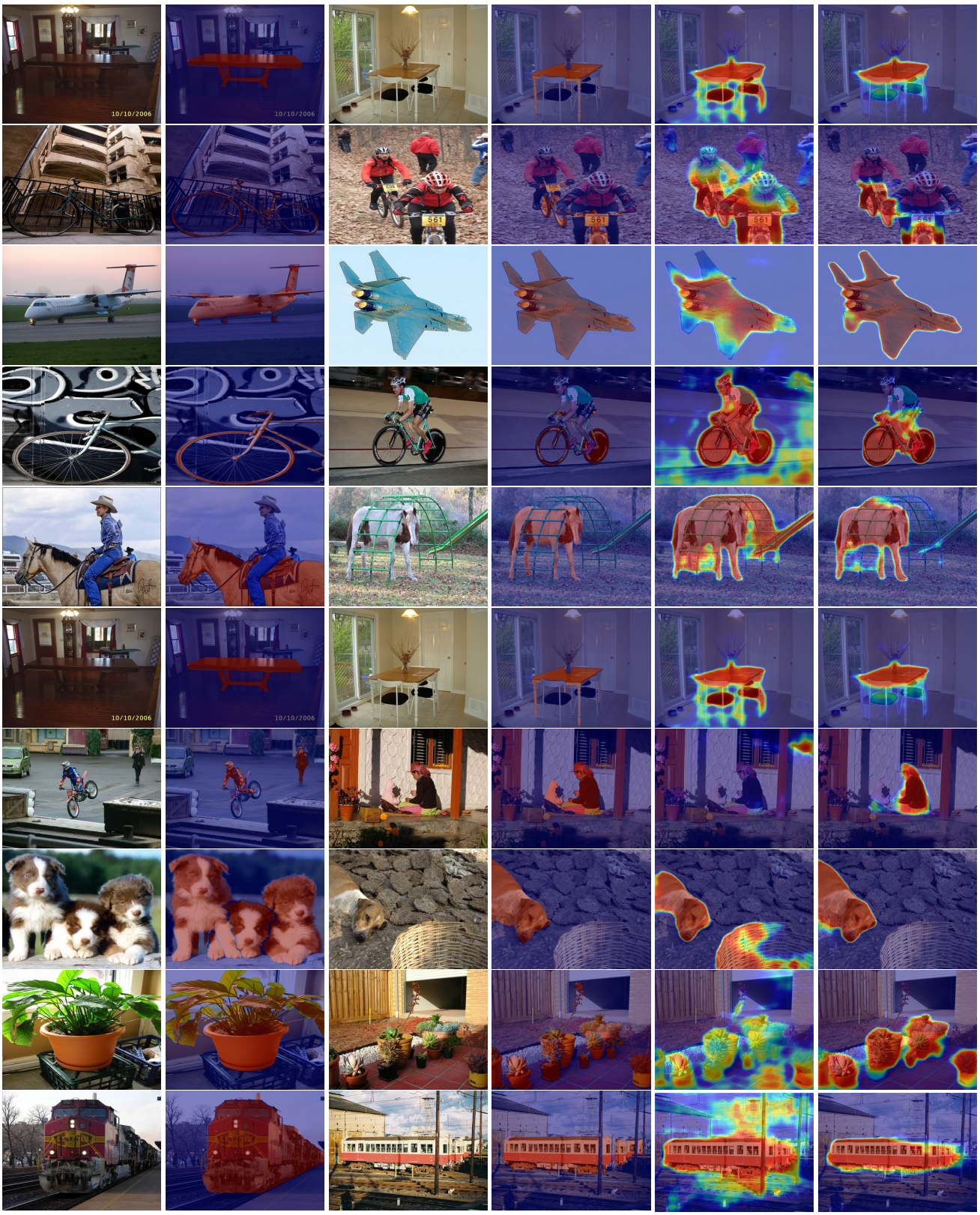

Figure 2. Visual comparison between the baseline and the model incorporated with CAPL.

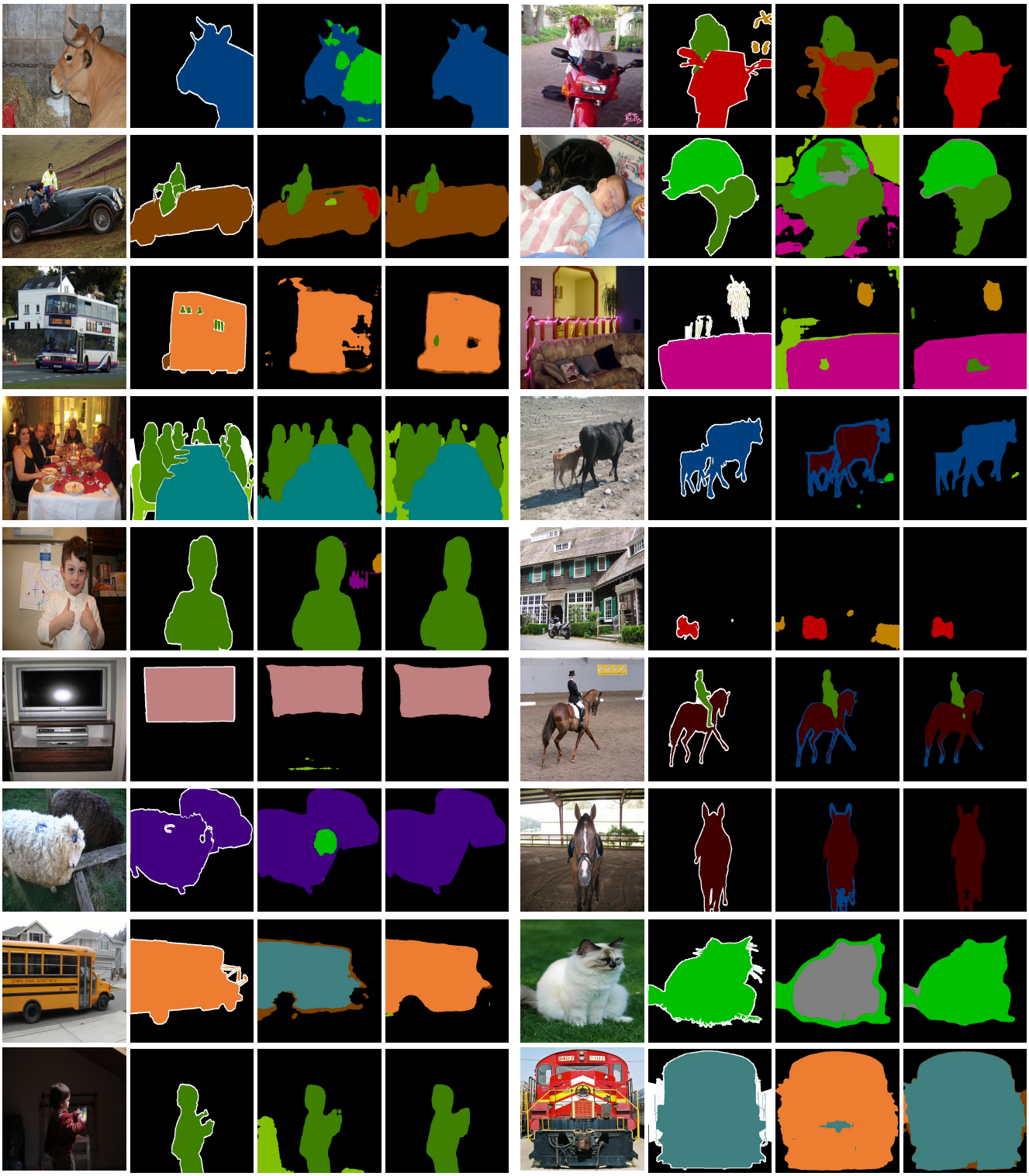

Figure 3. Visual comparison between the baseline and CAPL. Novel classes are bus, car, cat, chair and cow. The white color stands for the label 'Do Not Care'.