Supplementary material of Deep Metric Learning for Open World Semantic Segmentation

Jun Cen Peng Yun Junhao Cai Michael Yu Wang Ming Liu
The Hong Kong University of Science and Technology
{jcenaa, pyun, jcaiaq}@connect.ust.hk, {mywang, eelium}@ust.hk

A. Open-set semantic segmentation

In this section, we provide these additional details for Section 5.1:

- Details of each open-set semantic segmentation datasets.
- Details of open-set semantic segmentation implementation.
- Open-set semantic segmentation results under various β and γ, which are hyperparameters of Equation 10.
- Open-set semantic segmentation results under various T, which is the non-zero element of the prototypes.

A.1. Datasets

StreetHazards dataset [1] contains 5125 images for training, 1031 images without anomalous objects for validation, and 1500 images for testing with anomalies. Twelve classes are involved during training, including sky, road, street lines, traffic signs, sidewalk, pedestrian, vehicle, building, wall, pole, fence, and vegetation. We include 250 unique anomaly models of diverse types in the test dataset, while most of them are large rare transportation machines.

Lost and Found dataset [2] comprises 112 stereo video sequences with 2104 annotated frames. The whole dataset is only used for evaluating the anomaly segmentation performance and is not involved in training. 37 different obstacle types are contained and most of them are small items left on the street.

Road Anomaly dataset [3] is composed of 60 images for evaluating the anomalous objects. This dataset is no longer constrained in urban scenarios but contains images of villages and mountains. Animals, rocks, lost tires, trash cans are some anomalous examples in this dataset.

A.2. Implementation

For StreetHazards, we train a PSPNet decoder [4] with a ResNet-101 encoder [5] for 20 epochs with batch size 8. We train both the encoder and decoder using SGD with the momentum of 0.9, the learning rate of 2×10^{-2}, and the learning rate decay of 10^{-4}.

For Lost and Found and Road Anomaly, we use the training set from BDD100k [6] as these two datasets do not contain the training set themselves. The training procedure is as same as for StreetHazards because the number of training images in BDD100k is 4116, which is close to the number of training images in Streethazards.

A.3. Varying β and γ

Equation 10 describes the way of using MMSP to suppress the middle response of EDS. Pixels whose EDS score is smaller than γ are suppressed by MMSP, and β controls the suppressing effect. The ablation experiment results are in Table a. Some qualitative results are shown in Fig. a.

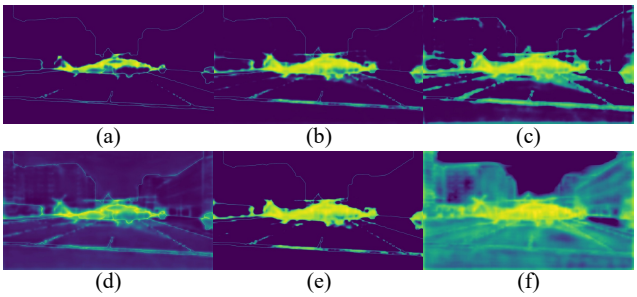
From Fig. a we can see that OOD objects are more obvious using Equation 10, but the fact is the mixture of EDS and MMSP provides similar metrics to EDS alone according to Table a. This is because: (1) MMSP will not only suppress in-distribution pixels, but also OOD pixels. For example, in (a) of Fig. a, some pixels of the helicopter are also suppressed. (2) All three anomaly segmentation related metrics are threshold-independent and used to measure whether anomalous scores of OOD pixels and in-distribution pixels are distinguishable, not the absolute difference value. EDS is already able to differentiate OOD pixels and in-distribution as shown in the Fig. 4 of our manuscript. However, the mixture map of EDS and MMSP can give labelers a better view and tells them the location of OOD objects, so they can make annotations more easily and pass them to our next incremental few-shot learning module.

A.4. Varying the non-zero element T of prototypes

T is the non-zero element of all prototypes as discussed in Section 4.2. It controls the positions of all prototypes in the metric space. Here we vary T, while the loss function is hybrid loss and the unknown identification criterion is EDS. The result is Table b. We find that different T has similar close-set mIoU indicating that T has little influence on the close-set segmentation. The most related metric among all anomaly segmentation metrics is FPR95, meaning that the

γ	β	AUPR↑	AUROC↑	FPR95↓	mIoU↑
×	×	**14.7**	**93.7**	17.3	
0.9		12.4	93.0	15.9	
0.8		14.1	93.5	18.0	
0.7	20	14.6	93.6	17.5	
0.6		**14.7**	**93.7**	17.3	53.9
0.5		**14.7**	**93.7**	**17.2**	
	5	12.0	93.2	17.9	
0.8	20	14.1	93.5	18.0	
	50	14.4	93.4	18.6	

Table a. **Ablation experiment results of β and γ.** The unknown identification criterion of the first row is EDS without MMSP. In our experiments, mIoU values are same because the close-set segmentation submodule is not influenced by Equation 10. It is shown that β and γ do not have huge impact on the performance.



(a) (b) (c)

(d) (e) (f)

Figure a. **Visualization results of different β and γ.** (a) $\beta = 50, \gamma = 0.9$. (b) $\beta = 50, \gamma = 0.7$. (c) $\beta = 50, \gamma = 0.5$. (d) $\beta = 5, \gamma = 0.8$. (e) $\beta = 20, \gamma = 0.8$. (f) EDS only. From these visualization results we can see that MMSP can suppress the middle response of EDS.

appropriate T can reduce the false-positive detection.

T	AUPR↑	AUROC↑	FPR95↓	mIoU↑
1	14.2	88.1	35.1	53.6
2	14.9	92.2	22.0	**54.1**
3	14.7	93.7	**17.3**	53.9
4	**15.0**	**93.9**	17.4	53.9
5	14.1	93.4	19.0	53.8
6	13.7	93.6	21.4	53.8

Table b. **Ablation experiment results of T.** We find the DMLNet has nice anomaly segmentation performance when $T = 3$ and $T = 4$.

B. Incremental few-shot learning

In this section, we provide the following details for Section 5.2:

- Details of the network architecture and training implementation.
- Incremental few-shot learning results of the novel prototype method (NPM) under various λ_{novel} of the Equation 12.
- Incremental learning under the few-shot and non-few-shot condition.
- Incremental learning using the pseudo labels and ground truth labels.

B.1. Implementation

The DMLNet we adopt for incremental few-shot learning is based on DeeplabV3+ [7], as shown in Fig. b.

We train the base model on the Cityscapes dataset [8] containing high quality pixel-level annotations of 5000 images (2975 and 500 for the training and validation respectively). The labels of 3 classes including car, truck and bus are set to be 255, so they are ignored during training. We train the encoder and decoder using SGD with the momentum of 0.9, the learning rate decay of 10^{-4}, and the initial learning rate of 0.01 and 0.1 respectively for 3×10^4 iterations. The batch size is 8 and the crop size is 762 due to the GPU memory limitation.

For the novel prototype method (NPM), we do not have to retrain the model for incremental few-shot learning as discussed in Section 4.3. For the pseudo label method (PLM), the architecture of the backbone and final branch head are demonstrated in Fig. b. When we apply the PLM for each novel class, we fix the trained backbone and heads and decrease the initial learning rate to 0.01 and 0.001 for 5 shot and 1 shot respectively. Total iteration numbers are both 500 for 5 shot and 1 shot but the batch size is 5 and 1 respectively.

B.2. Varying the λ_{novel} of NPM

The λ_{novel} in Equation 12 controls the distance threshold for the novel class classification of NPM. We conduct the ablation experiments for various λ_{novel} and the results are in Table c. Large λ_{novel} will cause more false-positive detection while small λ_{novel} will cause more false-negative detection for the novel class. $\lambda_{novel} = 1.5$ achieves the best performance according to Table c.

B.3. Few-shot and non-few-shot

In the paper, we increase the knowledge base of the DMLNet through the incremental few-shot learning module. This is because: (1) Few-shot learning requires much fewer labels compared to non-few-shot learning, while making segmentation labels is extremely time-consuming. (2) The training process of the few-shot learning also consumes less time. (3) Incremental few-shot learning is not well studied so far, and we provide two methods including PLM and NPM as the baseline in this area.

However, PLM and NPM perform worse than the upper bound which regards the novel class as one of the original in-distribution classes and retrain the model using the whole

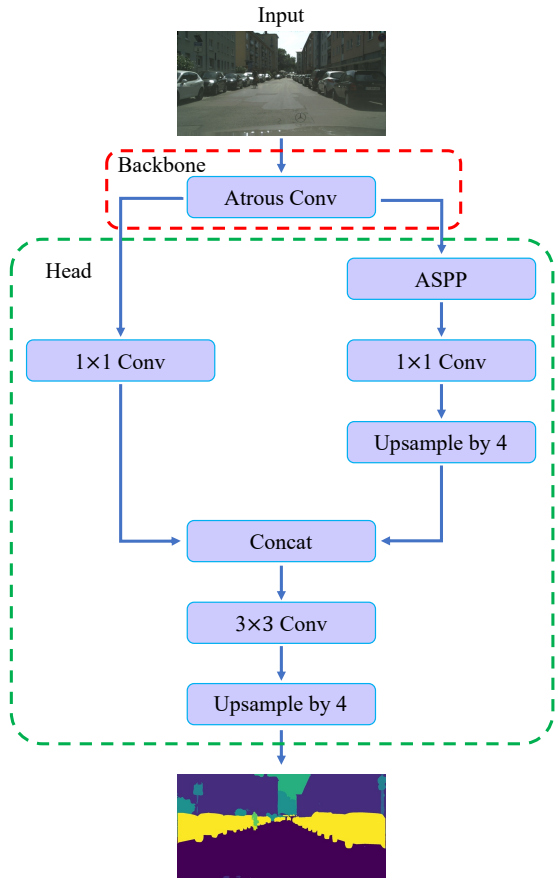


Figure b. **DMLNet architecture based on DeeplabV3+**. This is the network architecture we adopt for the incremental few-shot learning module. The backbone and head in Fig. 5 of our manuscript are demonstrated specifically in this Fig. b.

16+1 setting	λ_{novel}	mIoU	$\text{mIoU}_{\text{novel}}$	mIoU_{old}	$\text{mIoU}_{\text{harm}}$
	2	**67.8**	61.9	**68.2**	64.9
5 shot	1.5	67.4	**64.6**	67.6	**66.1**
	1	63.4	41.8	64.7	50.8
	2	**67.1**	55.4	**67.9**	61.0
1 shot	1.5	66.5	**60.1**	66.9	**63.3**
	1	62.5	38.0	64.1	47.7
16+3 setting					
	2	55.4	24.1	61.3	34.6
5 shot	1.5	**58.2**	**26.1**	**64.2**	**37.1**
	1	56.6	20.2	63.4	30.7
	2	54.6	24.2	60.3	34.6
1 shot	1.5	**56.6**	**25.9**	62.3	**36.5**
	1	55.5	18.9	**62.4**	29.0

Table c. **Ablation experiment results of** λ_{novel}. We find that $\lambda_{novel} = 1.5$ has the best performance among all settings.

dataset. As the upper bound is under the non-few-shot condition, there are two possible reasons that make our incremental few-shot learning methods perform worse than the upper bound. The first one is that the training samples are insufficient, so the DMLNet cannot extract representative features. The second one is that our methods themselves constrain the DMLNet to obtain good performance. Therefore, we conduct experiments using more training samples to find out the reason.

From Table 4 of our manuscript, we notice that PLM have a better performance on the novel class than NPM under 5 shot condition. This is because the network architecture and the metric space of PLM will grow to fit the new classes. Therefore, PLM is more suitable for the non-few-shot condition. We conduct ablation experiments of PLM using a different number of training samples Q. The results are shown in Table d. We find the performance of PLM improves with more training samples, but the performance still not reaches the upper bound when using the whole training set of the Cityscapes dataset. Therefore, both the limited number of training samples and the PLM itself constrain the performance under few-shot condition.

16+1 setting	mIoU	$\text{mIoU}_{\text{novel}}$	mIoU_{old}	$\text{mIoU}_{\text{harm}}$
$Q = 1$	60.4	64.5	60.1	62.2
$Q = 5$	64.4	75.7	63.7	69.2
$Q = 100$	70.7	85.5	69.8	76.9
$Q = 1000$	71.9	90.1	70.7	79.2
$Q = 2975$	**72.2**	**91.8**	**71.0**	**80.1**
All 17	74.9	94.8	-	-

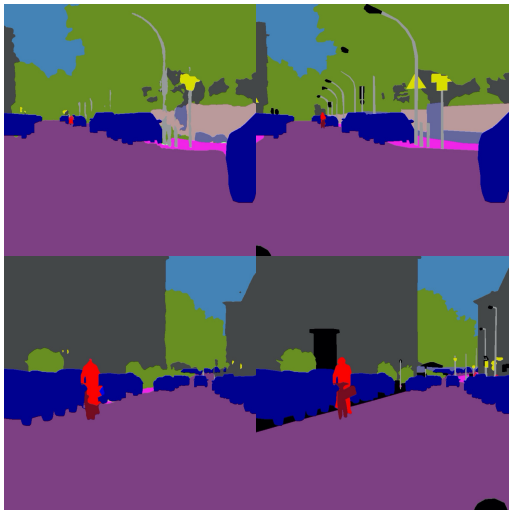
Table d. **Ablation experiment results of** Q. **All 17** is the upper bound. The performance of PLM improves with more training samples.

B.4. Pseudo label and ground truth label

In the pseudo label method (PLM), the old trained final branch heads provide the prediction of old classes, and these predictions combine with the annotation of the new class to generate the pseudo label of the training sample. In this way, labelers only need to give annotations for the new class and do not need to annotate for every pixel of the training samples. To verify that the PLM is reasonable, we conduct experiments using the ground truth labels for incremental learning rather than the pseudo labels. The results are in Table e. Compared to Table d, we find that the performance of using pseudo labels is similar to the performance of using ground truth labels, demonstrating the effectiveness of our PLM method. Some visualization of pseudo labels is shown in Fig. c.

16+1 setting	mIoU	$\text{mIoU}_{\text{novel}}$	mIoU_{old}	$\text{mIoU}_{\text{harm}}$
$Q = 1$	58.9	60.2	58.8	59.5
$Q = 5$	61.2	72.3	60.5	65.9
$Q = 100$	70.2	85.8	69.2	76.6
$Q = 1000$	72.0	91.8	70.8	79.9
$Q = 2975$	**72.0**	**91.9**	**70.8**	**80.0**

Table e. **Incremental learning results using the ground truth under various** Q. Compared to Table d, this table shows the incremental learning results using the ground truth labels are similar to the results using pseudo labels.



Pseudo labels Ground truth labels

Figure c. **Pseudo labels and ground truth labels**. The labels of the novel class *car* are the same, while in other places the ground truth labels provide more precise details.

References

[1] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019.

[2] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1099–1106, 2016.

[3] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2152–2161, 2019.

[4] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2633–2642, 2020.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[8] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.