Supplementary for Standardized Max Logits: A Simple yet Effective Approach for Identifying Unexpected Road Obstacles in Urban-Scene Segmentation

A. Supplementary Material

This supplementary presents the quantitative results on different architectures, hyper-parameter impacts, implementation details, and qualitative results.

A.1. Effects on Different Architecture and Backbone

This section presents the quantitative results of different architecture and backbone (*i.e.,* EfficientPS [11] and ResNeSt [13]) on the FishyScapes Lost & Found validation set. As shown in Table 1, our approach outperforms all other methods in both cases. However, the amount of performance increase is not strictly correlated with the downstream task performance, as also pointed out in the previous work [5, 7].

Architectures	mIoU	Methods	AUROC ↑	AP ↑	FPR_{95} ↓
†EfficientPS [11]	79.3	MSP	84.41	1.46	61.03
		Max Logit	89.39	3.83	48.75
		Ours	**94.17**	**5.93**	**21.93**
DeeplabV3+ w/ ResNeSt [13]	79.1	MSP	87.23	7.89	57.67
		Max Logit	91.91	22.58	51.12
		Ours	**95.32**	**31.38**	**30.37**

Table 1: Results of EfficientPS and DeeplabV3+ with ResNeSt backbone on Fishyscapes Lost & Found validation set. † denotes the results are obtained from the official code with their pre-trained networks.

A.2. Analysis on Hyper-parameters

This section analyzes the impact of hyper-parameters in our proposed method through ablation studies on FishyScapes Lost&Found validation set.

Number of iterations n We report the quantitative results according to the number of iterations n used in iterative boundary suppression, described in Section 3.3.1 of the main paper. Note that we set the initial boundary width r_0 to $2n$ so that $\Delta r = \lfloor \frac{r_0}{n} \rfloor$ equals 2 since we intend to reduce the width by 1 from each side of the boundary. As shown in Table 2, the performances in all metrics consistently increase as n increases up to $n = 4$. While AUROC and FPR_{95} are improved at $n = 5$, AP rather aggravates. Since the number of in-distribution and unexpected pixels are unbalanced, we choose AP for our primary metric, which is invariant to the data imbalance, as done in Fishyscapes. Hence, we use $n = 4$ in our work.

Iterations	AUROC ↑	AP ↑	FPR_{95} ↓
$n = 1$	96.73	36.26	15.48
$n = 2$	96.78	36.44	15.19
$n = 3$	96.84	36.54	14.86
$\boldsymbol{n = 4}$	96.89	**36.55**	14.53
$n = 5$	**96.93**	36.44	**14.22**

Table 2: Quantitative results with respect to n on Fishyscapes Lost & Found. Results are obtained after standardizing the max logit, iterative boundary suppression, and dilated smoothing.

Dilation rate d We present the quantitative results with respect to the dilation rate d used in dilated smoothing, described in Section 3.3.2 of the main paper. As shown in Table 3, taking wider receptive fields improves the performance in AP up to $d = 6$. However, if the size of the receptive field increases further (*e.g.,* after $d = 7$), the performance rather degrades, indicating that a proper size of a receptive field is crucial in properly capturing the consistent local patterns.

Dilation	AUROC ↑	AP ↑	FPR_{95} ↓
$d = 1$	96.86	33.25	14.50
$d = 2$	96.90	34.61	14.36
$d = 3$	96.92	35.57	**14.33**
$d = 4$	**96.93**	36.15	14.39
$d = 5$	96.92	36.46	14.47
$\boldsymbol{d = 6}$	96.89	**36.55**	14.53
$d = 7$	96.86	36.47	14.57
$d = 8$	96.81	36.28	14.66
$d = 9$	96.76	35.99	14.91
$d = 10$	96.70	35.64	15.31

Table 3: Quantitative results according to the dilation rate d on Fishyscapes Lost & Found. Results are obtained after standardizing the max logit, iterative boundary suppression, and dilated smoothing.

A.3. Further Implementation Details

We adopt DeepLabV3+ [2] as our segmentation network architecture and mainly use ResNet101 [3] as the backbone for most of the experiments. Note that, as already shown in the main paper, our proposed method is model-agnostic and achieves the best performance with the MobileNetV2 [12],

ShuffleNetV2 [10], and ResNet50 [3] backbones compared to MSP [6] and max logit [4].

The model is trained with an output stride of 8 and the batch size of 8 for 60,000 iterations with an initial learning rate of 1e-2 and momentum of 0.9. In addition, we apply the polynomial learning rate scheduling [9] with the power of 0.9 and the standard cross-entropy loss with the auxiliary loss proposed in PSPNet [8], where the auxiliary loss weight λ is set to 0.4. Moreover, in order to prevent the model from overfitting, we apply color and positional augmentations such as color jittering, Gaussian blur, random scaling with the range of [0.5, 2.0], random horizontal flipping, and random cropping. We adopt class-uniform sampling [14, 1] with a rate 0.5.

As aforementioned, we set the number of boundary iterations n, the initial boundary width r_0, and the dilation rate d as 4, 8, and 6, respectively. Additionally, we set the sizes of the boundary-aware average pooling kernel and the smoothing kernel size as 3×3 and 7×7, respectively.

A.4. Qualitative Results

This section presents the additional qualitative results. We first demonstrate the qualitative results of our methods and then their comparisons with other baselines. We use the threshold at TPR_{95} and visualize the predicted in-distribution and unexpected pixels as black and white, respectively.

Our results Fig. 1 presents the qualitative results of applying iterative boundary suppression to show the effectiveness of removing the false positives (*i.e.,* in-distribution pixels detected as unexpected). We zoom in particular regions with the red boxes to show the changes in detail. After applying iterative boundary suppression, we significantly remove the false positives in boundary regions.

Additionally, Fig. 2 describes the results of applying all of our methods. The false positives in the boundary regions (*e.g.,* white pixels in the yellow boxes) are removed after applying iterative boundary suppression. Also, as shown in the green boxes, applying dilated smoothing effectively removes the false positives in the non-boundary regions.

Comparison with other approaches We compare our method with MSP [6] and max logit [4] by showing qualitative results. Figs. 3 and 4 show the results obtained from Fishyscapes Lost & Found and Fishsyscapes Static, respectively. Since we visualize the images with the threshold at TPR_{95}, most of the pixels in unexpected objects are identified. However, using MSP and max logit generate a substantial amount of false positives. In contrast, our method produces a negligible amount of false positives, which demonstrates our effectiveness.

References

[1] Samuel Rota Bulo, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5639–5647, 2018. 2

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016. 1, 2

[4] 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*, 2020. 2

[5] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1

[6] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017. 2

[7] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*, 2020. 1

[8] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. In *Proc. of the British Machine Vision Conference (BMVC)*, page 285, 2018. 2

[9] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 2

[10] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 2

[11] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, 2021. 1

[12] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4510–4520, 2018. 1

[13] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 1

[14] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*, pages 8856–8865, 2019. 2

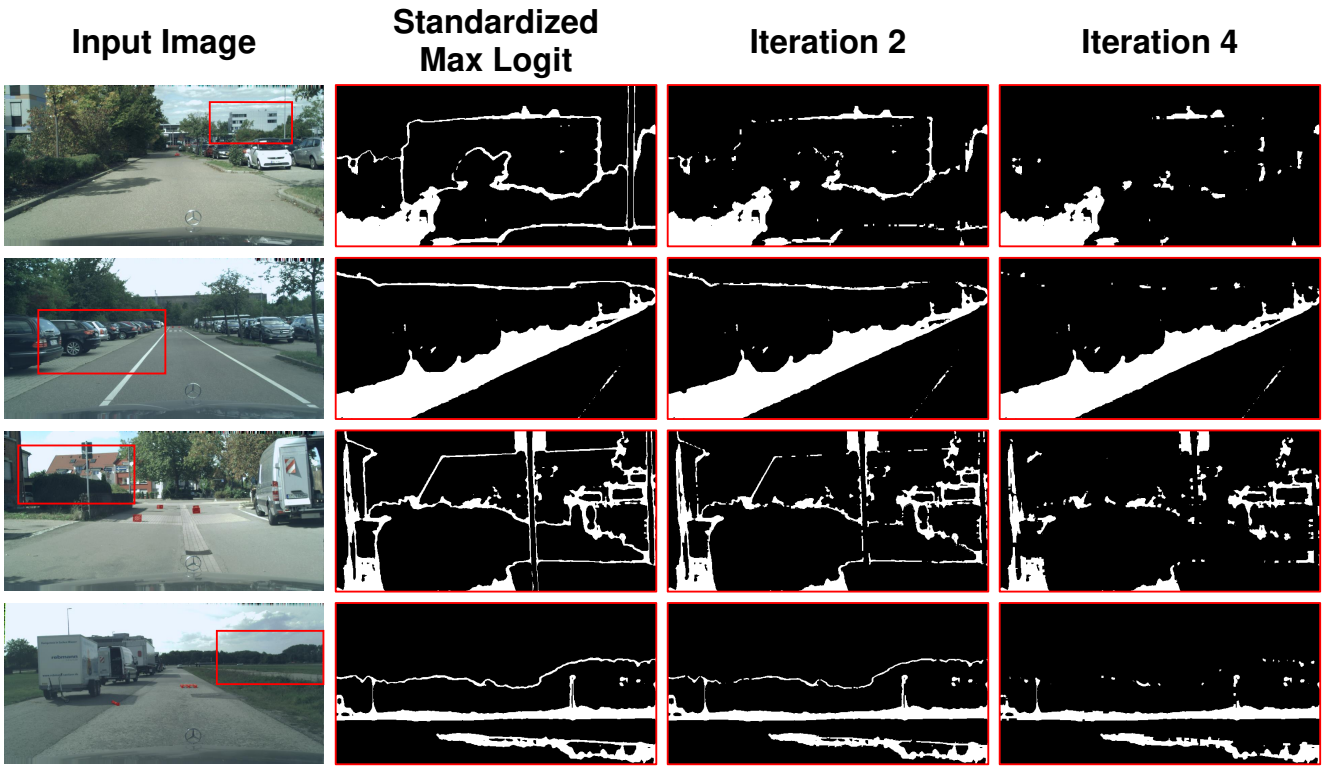

Figure 1: Qualitative results of applying standardized max logit and iterative boundary suppression with iteration 2 and 4, respectively. We report the images of Fishyscapes Lost & Found. The white pixels indicate the pixels predicted as unexpected.

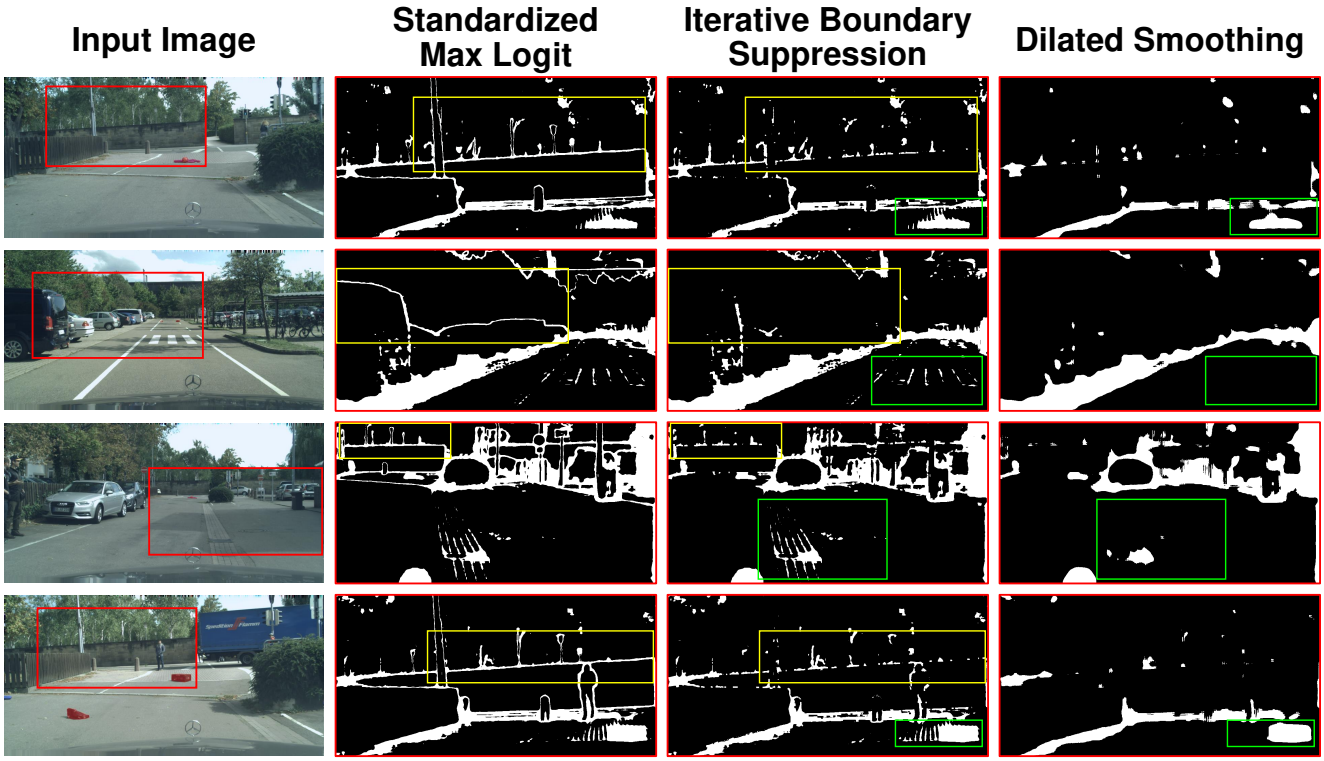

Figure 2: Qualitative results of applying standardized max logit, iterative boundary suppression, and dilated smoothing, respectively. We report the images of Fishyscapes Lost & Found. Yellow boxes and green boxes show that the false positives are effectively removed by applying iterative boundary suppression and dilated smoothing, respectively. The white pixels indicate the pixels predicted as unexpected.

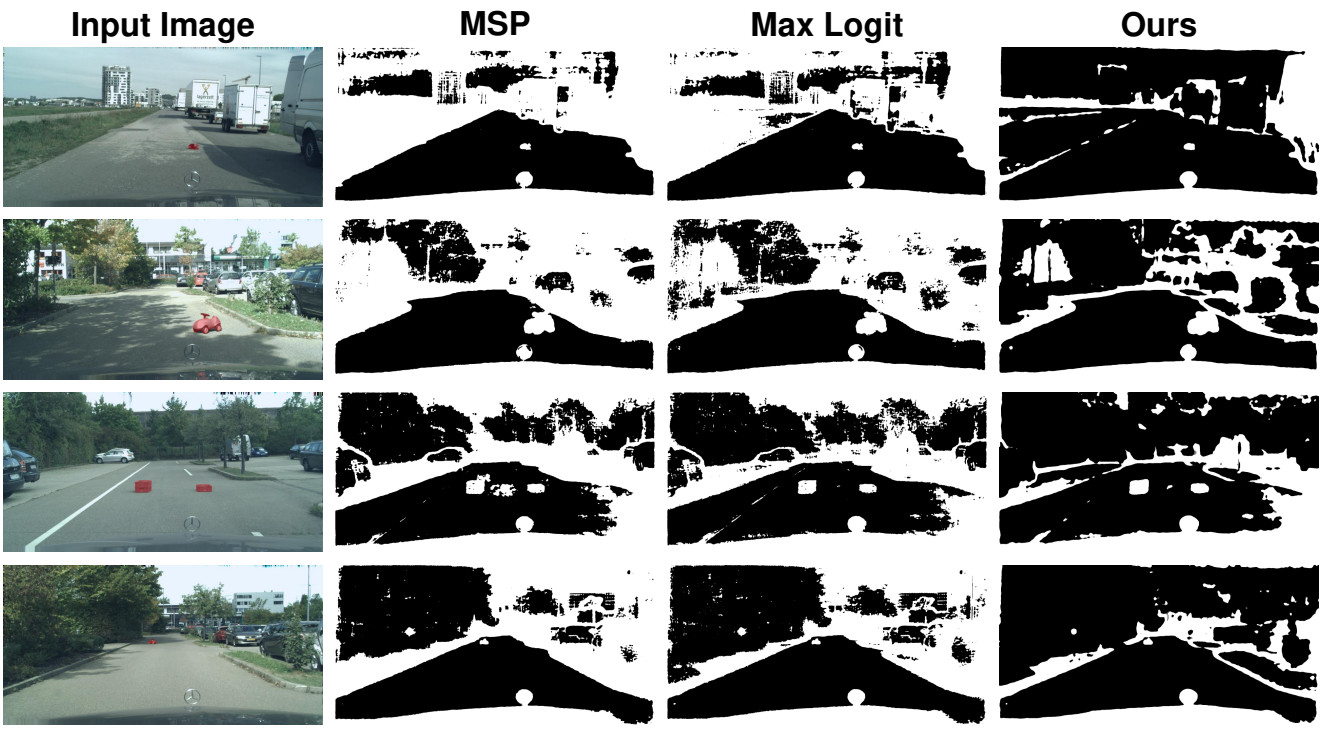

Figure 3: Comparison with MSP, max logit, and ours on Fishyscapes Lost & Found dataset. The white pixels indicate the pixels predicted as unexpected.

Figure 4: Comparison with MSP, max logit, and ours on Fishyscapes Static dataset. The white pixels indicate the pixels predicted as unexpected.