

Online Domain Adaptation for Semantic Segmentation in Ever-Changing Conditions – Supplementary Material

Theodoros Panagiotakopoulos^{1*}

Pier Luigi Dovesi²
Matteo Poggi⁵

Linus Härenstam-Nielsen^{3,4*}

¹King

²Univrse

³Kudan

⁴ Technical University of Munich

⁵University of Bologna

This document introduces further details on the ECCV 2022 paper - "Online Domain Adaptation for Semantic Segmentation in Ever-Changing Conditions". In Section 1, we present results of the OnDA framework on the CityScapes dataset [2] with qualitative samples and a video sequence, in Section 2 we detail the model implementation and hyperparameters, while in Section 3 we further describe the working principle behind Domain Shift Detection. Then we provide additional explanation on some experimental results presented in the paper, in particular we focus on the Standard deviation, Section 4, and the Calibration Error, Section 5. Finally, in Section 6 we expand our Ablation Study and study the impact of the learning rate, the effect of the confidence regularization and Batch Normalization.

1. CityScapes qualitative comparison between Source, Offline and Online Adapted Model.

We provide additional, qualitative results concerning our experiments on the Increasing Storm scenario. The whole adaptation process is also featured in a video sequence (23.46 minutes).

A shorter version of the video (3 minutes) is also available.

- **short version:** : <https://youtu.be/cany-1UNWY8>
- **long version:** <https://youtu.be/igtmgafiurY>

1.1. Video Analysis

To reproduce a realistic online adaptation set-up, we employed the Frankfurt CityScapes video sequence. We divided it in equally sized intervals and then we rendered the rain according the Increasing Storm schema *i.e.* 0mm (*clear*), 25mm, 50mm, 75mm, 100mm, 200mm, 100mm, 75mm, 50mm, 25mm, 0mm. We sub-sample the video to 10fps, with a total length of 23.46 minutes, and the duration of each target domain is 3239 frames, corresponding to around 5.4 minutes (the video is accelerated 2.5 times, hence every domain is presented in 2.09 minutes). It is worth mentioning that the adopted rain rendering methodology [6] does not limit to generating rain random particles, as it uses monocular depth prediction [3] and depth refinement [1] to estimate the rain occlusion in the image. Moreover, other effects are employed to make the sky darker and thus emulate the corresponding cloud coverage.

The OnDA framework employs the Hybrid Switch with same configuration presented in all the other experiments. The only differences consisted in a minor update of the two Hybrid Switch thresholds (T_{c_A} and T_{c_B}) and in the introduction of a debouncing window on the switching indicator function, I_t , as introduced in Section 3.2 of the paper.

In Figure 1 we display video frames and the corresponding segmentation masks collected at regular time intervals so to cover all the deployment domains. The video sequence does not present ground-truth for a quantitative evaluation, anyway the qualitative analysis presents a clear pattern and we feel it significant of a real, online adaptation scenario. The Source model is steadily degraded as the rain intensifies, presenting visible degradation right from the 25mm rain and completely collapsing in the 100mm and 200mm domains. The Offline model instead has been pre-adapted offline on all the rain intensities (for this task we used the CityScapes train split). The Offline model is visibly more robust to the easier targets, while struggling in the 100mm case and collapsing in the 200mm. Finally, the Online model performs similarly or marginally better than the Offline method in all easy targets, while still being robust to the 100mm and 200mm cases.

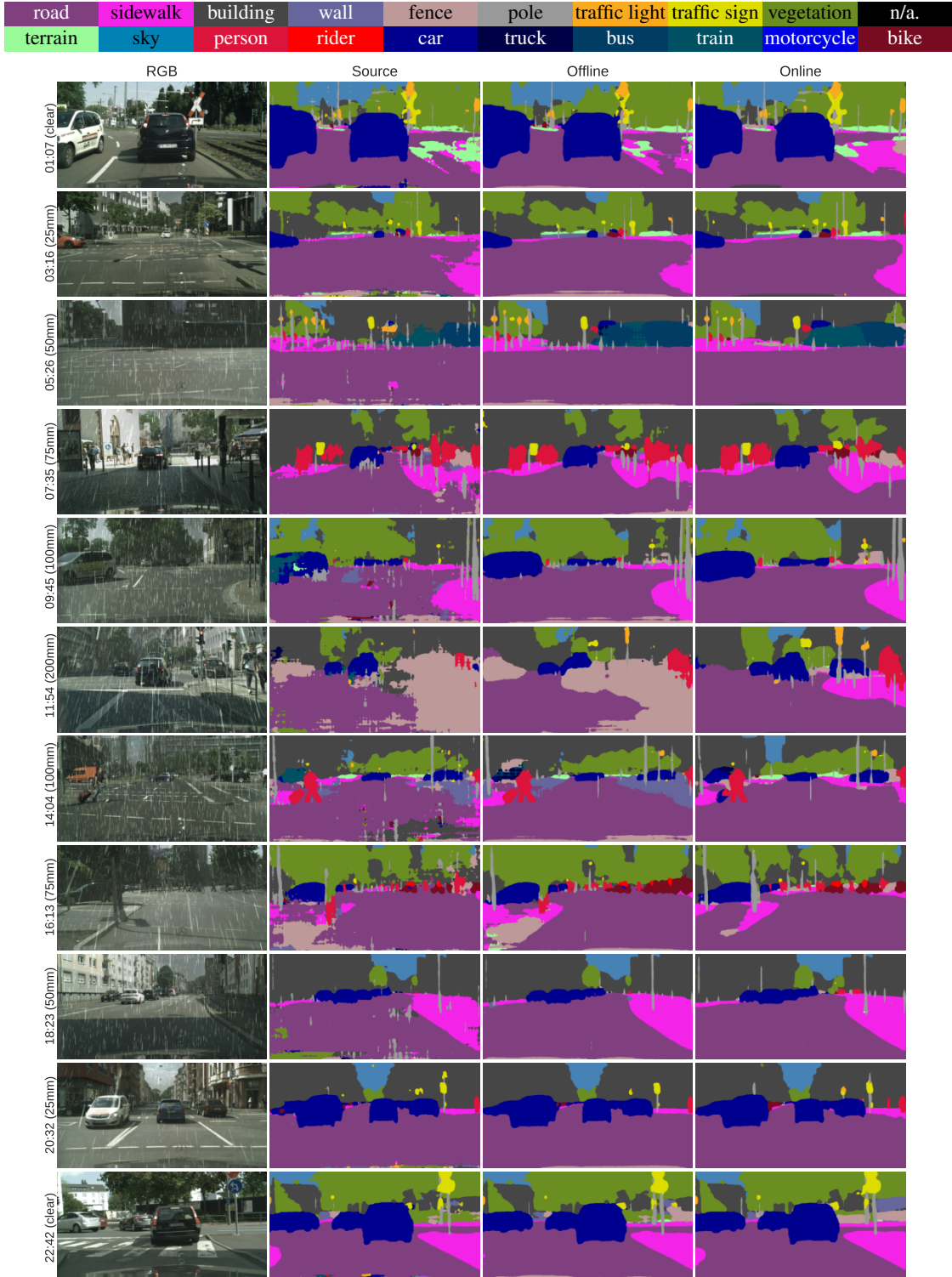


Figure 1. **Samples from the Cityscapes Frankfurt video using artificial rain and comparing the Source, Offline and Online Adapted Model.** The Online model gets adapted on-the-fly exploiting the video footage itself. Images start from frame 1677 (1:07) and are captured every 3239 frames ($\sim 2:09$).

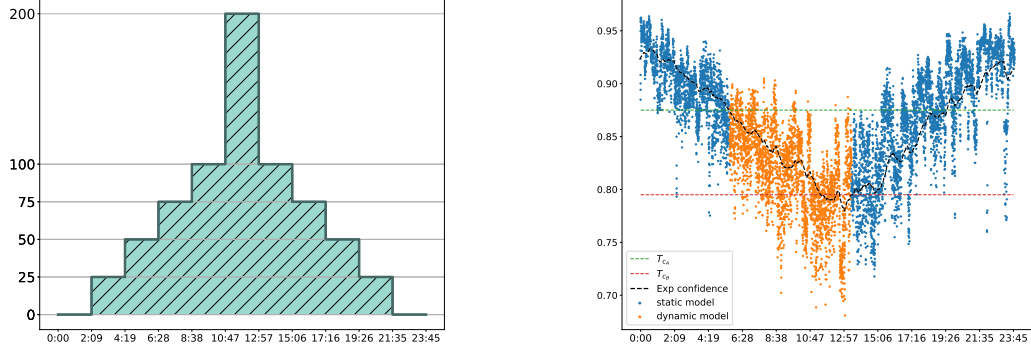


Figure 2. **Domain sequence and switching policy – Increasing Storm.** We show rain intensity over time (left), confidence and switching policy behavior (right) when adapting over the Increasing Storm.

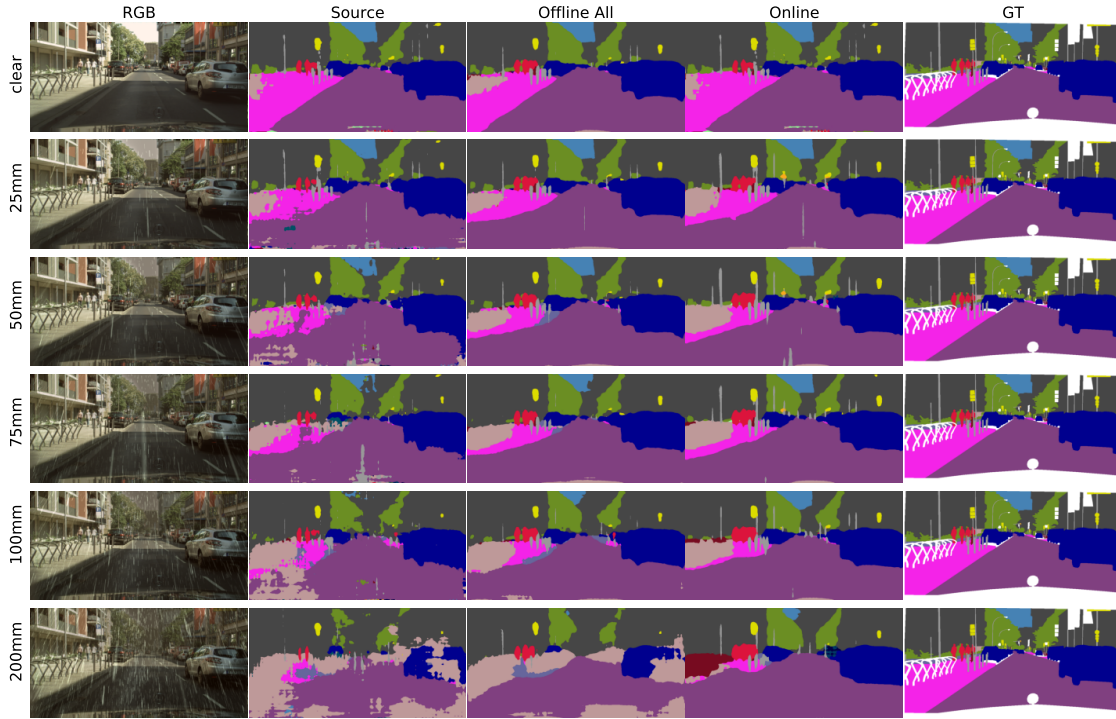


Figure 3. **Qualitative comparison of the Source, Offline and Online Models on rainy images.** From top to bottom, we report a single sample from each domain, from *clear* to 200mm rain intensity.

In Figure 2 we present the rain intensities of the Increasing Storm and the Hybrid Switch mechanism in action over that sequence. In particular we notice how the static prior, h_{static} , is rapidly switched for the dynamic one, h_{dynamic} , during the 50mm domain. This is the consequence of I_t being negative (as z_t is indeed decreasing), since the model is moving further from the source domain. We notice how the model keeps (and updates) h_{dynamic} until the “backward phase” starts, then as the confidence of h_{static} grows – leading to a positive I_t – the model prior is switched to h_{static} again. More details about the switching principle are reported in Sec. 3.

1.2. Validation Set Qualitatives

In Figure 3 and 4 we include more qualitative samples taken from the CityScapes validation set. In Figure 5 we present qualitative samples with fog. In any of these figures, the samples present the comparison between Source, Offline All, Online (Hybrid Switch), and Ground-Truth (GT).



Figure 5. **Qualitative comparison of the Source, Offline and Online Models on foggy images.** From top to bottom, we report a single sample from each domain, from *clear* to 75m visibility.

2. Implementation Details

We report in detail all the hyper-parameters used to train the described methods. The supervised models were trained until convergence (up to 100 epochs) using a learning rate of $2.5\text{e-}4$ and an exponential decay learning rate with a power of 0.9, together with a weight decay of $5\text{e-}4$. For the offline model the same learning policy was used but with an initial learning rate of $1\text{e-}5$ and 10 epochs of training. Moreover, for the models using all of the domains (Supervised All & Offline All), a single epoch was considered as passing through all Domains.

The prototypes update was set to $\lambda = 0.9995$ while the momentum model update to 0.999. The thresholds used for the switches are the following: $T_{cd} = 2\text{e-}4$, $T_c = 0.86$, $T_s = 0.94$, $T_d = 0.82$, $T_{c_A} = 0.9$ and $T_{c_B} = 0.83$. Moreover, the Online models were trained using *SGD* with a momentum of 0.9, learning rates of $8\text{e-}4$ and $1\text{e-}4$ for the feature extractor and segmentation head respectively, and using a shifting window of length $n = 200$. Lastly, the Advent model was trained using the default settings detailed in [7]. All models performed training with a batch size of 4 and images scaled to 512×1024 resolution.

3. Domain Shift Detection

As discussed in the main paper, when performing online adaptation it becomes crucial to properly detect the domain shifts and act accordingly. The confidence h_{static} by the static model can be a good indicator to detect the transition across domains. More precisely, the lower the confidence, the farther the current domain is from the source domain. Then, sudden changes of such a confidence can reflect the transition to a new domain. Moreover, by looking at the sign of such change we can identify if we are moving towards a domain closer to the source one or, vice-versa, if we are going to a new, farther domain. This behavior is encoded by the sign of the derivative of h_{static} .

Figure 6 presents an intuition of this working principle: the blue curve shows the trend of the static model confidence h_{static} , highlighting sudden changes in correspondence of domain changes. The orange curve plots the derivative $\mu_{z_t} - \mu_{z_{t-1}}$ of h_{static} : we can notice how this curve has negative peaks when the confidence drastically drops, i.e. when we approach a new domain farther from the source; on the contrary, we have positive peaks when we switch to a domain closer to the source one.

Based on this principle, we design some of the switching mechanisms introduced and evaluated in our paper.

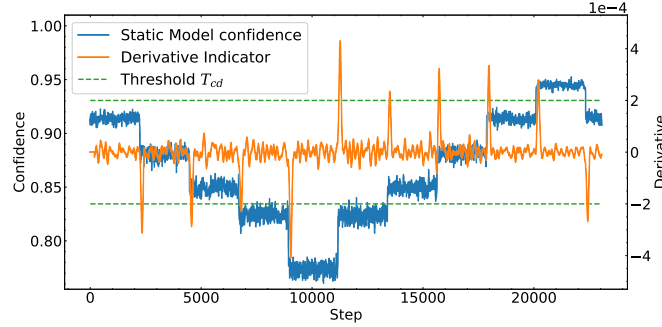


Figure 6. **Working principle of the domain switching detector.** On the back, in blue, the h_{static} confidence is displayed, while with orange color the derivative $\mu_{z_t} - \mu_{z_{t-1}}$ is showed. When switching to more distant domains the value function has negative peaks and a positive on closer domains.

4. Standard Deviation Analysis

Figure 7 displays the progression of the IoU standard deviation across segmentation classes over time. Interestingly, the variance decreases after an adaptation cycle. Before starting adaptation (step 0), we do not observe a specific relationship between the class variance and the domain itself. After a full adaptation cycle, we can notice a correlation between rain intensity and variance – indeed, once the model is back to the *clear* domain, the lower variance (and thus more consistent IoU across all the classes) is achieved on domains closer to the *clear* one, with the highest variance on 200mm (the farthest domain). This hints that the adaptation process itself regularizes the performance across classes, creating an interesting correlation between rain intensity and variance.

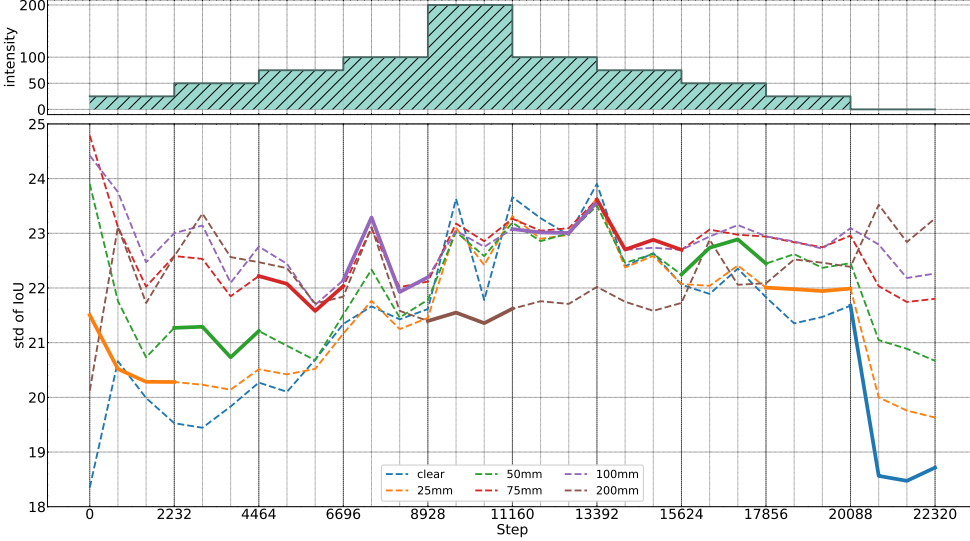


Figure 7. **Illustration of the Standard deviation of the IoU over the classes over time.** Experiment carried out on the Increasing Storm baseline scenario.

5. Calibration Error Analysis

Figure 8 shows confidence, mIoU and Expected Calibration Error (ECE) [4] of h_{static} over the target domains (from *clear* to 200mm). Similarly to several other works in Unsupervised Domain Adaption (UDA) and Self-Training, OnDA heavily bases its working principles on model confidence, in particular, on the confidence of the static model, h_{static} . By studying Figure 8 we can notice how the calibration error remains almost constant for domains closer to the source – despite the steady performance drop – proving that the model confidence is correspondingly decreased to match the output accuracy.

This equilibrium does not hold for domains farther from source: indeed, despite confidence remains in general high (~ 0.80), the mIoU (and accuracy) dramatically drops, consequently leading to significant increase of the ECE. The analysis of this trend, on one hand, suggests that the confidence of h_{static} for domains closer to the source represents a valid heuristic. On the other hand, it proves that the static model confidence validity is not universal as its reliability degrades progressively as we move to further domains. We highlight how our Domain Indicator Function, I_t uniquely exploits the confidence derivative (not the absolute value), hence it is not sensitive to model miscalibration as long as the confidence is decreasing with the target domain distance.

6. Extended Ablation Study

We conclude this document by digging into three further ablation studies, on learning rate, regularization and Batch Norm respectively.

6.1. Learning Rate

According to [5], adapting the weights of the features extractor forces the network to align their representation to those required by the prediction head to achieve good predictions. Inspired by these findings, we investigate the effect of different learning rates applied to the features extractor and the segmentation head separately. Figure 9 shows the behavior of our model while being adapted on the Increasing Storm scenario, focusing respectively on 50mm (left) and 200mm (right) domains. Each curve shows the behavior of one of the two modules (features extractor or prediction head) while varying its learning rate and keeping constant the learning rate for the other module.

Empirically, we found out that setting a higher learning rate for the features extractor (green) allows for better performance, confirming the findings in [5]. Acting on the learning rate of the prediction head (red) shows negligible impact.

6.2. Regularization

Table 1 collects results for adaptation on the Increasing Storm scenario, both when proceeding forward (F) or backward (B) across domains. Empirical results show that confidence regularization does not affect performance substantially all-over

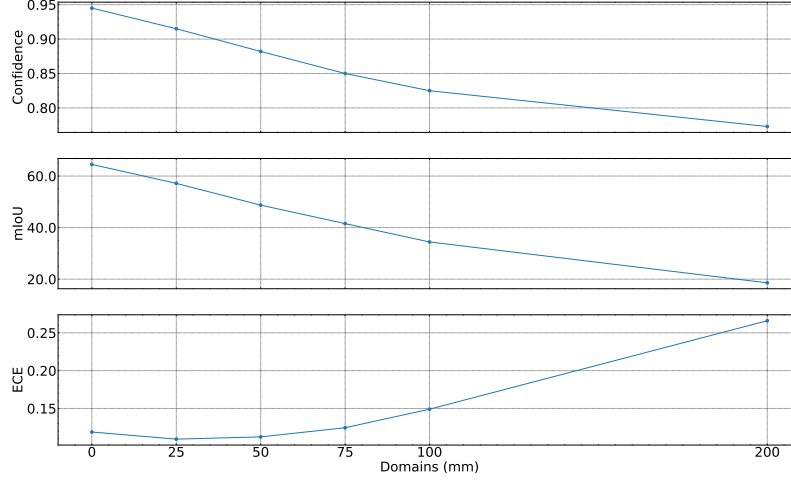


Figure 8. **Comparison between Confidence, MIoU and Confidence Error (ECE) on the static (source) model across domains.** Experiments carried out on the Increasing Storm scenario. ECE is computed over 1000 bins.

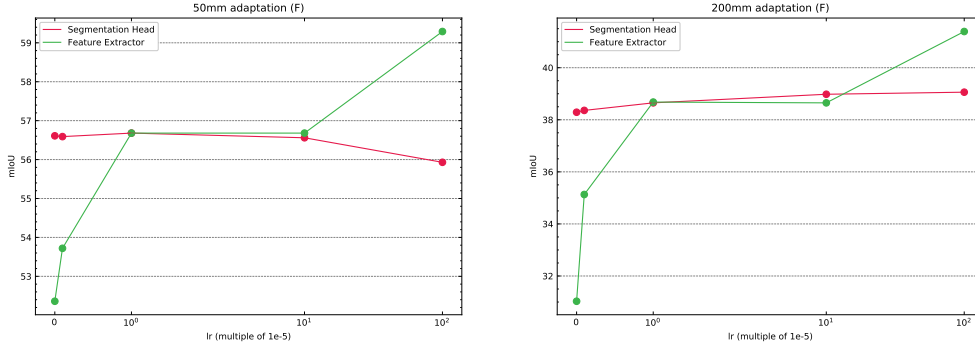


Figure 9. **Learning rate comparison between feature extractor and segmentation head, on the 50mm and 200mm case.** For each testing the learning rate on the opposite sub-network remained fixed at $1e-4$ and $1e-5$ on the feature extractor and segmentation head respectively. Adaptations happens gradually from *clear* to intensities of 25, 50, 75, 100 and 200.

the adaptation process. Nevertheless, by introducing it with a factor 0.1, we are able to increase adaptation performance on the hardest domains, *i.e.* 100mm and 200mm. On the contrary, regularization itself results less effective when adapting backward, leading to slightly lower mIoU when going back to 50mm, 25mm and *clear* domains.

	<i>clear</i>		25mm		50mm		75mm		100mm		200mm		h mean	
	F	B	F	B	F	B	F	B	F	B	F	B	F	B
Regularization factor 0	64.5	65.7	59.7	59.3	54.8	55.1	51.4	50.7	48.7	45.7	38.0	38.0	51.4	50.8
Regularization factor 0.1	64.5	64.8	60.4	57.1	57.3	54.5	54.8	52.2	52.0	49.1	42.2	42.2	54.2	52.4
Regularization factor 0.2	64.5	62.6	59.3	54.2	57.4	52.4	53.9	50.5	50.2	48.4	41.4	41.4	53.4	50.8
Regularization factor 0.3	64.5	60.5	59.9	54.1	56.9	52.2	52.5	50.7	50.2	49.0	41.9	41.9	53.3	50.8

Table 1. **Base adaptation cycle results using the Hybrid Switch approach with different confidence regularization factors.** Experiments in the Increasing Storm scenario

6.3. Batch Normalization

In Figure 10 we present a comparison between the three Batch-Normalization (BN) policies:

- *BN freeze*: freezing the BN when processing samples from the Replay Buffer,

- *BN switching*: swapping BN statistics between target samples and Replay Buffer,
- *Shared BN*: sharing BN statistics.

For simplicity we only present two domains: the source, 0mm (*clear*), and 200mm cases. We underline how these validation metrics are meaningful only when the model is actually exposed to that domain. We can observe that all of the three options perform similarly for the source domain (notice frames 20088 to 22320) on the mIoU (*clear*) plot. However, the *Shared BN* falls behind in terms of adaptation flexibility, as shown in the mIoU (200mm) in the frame interval 8928 to 11160.

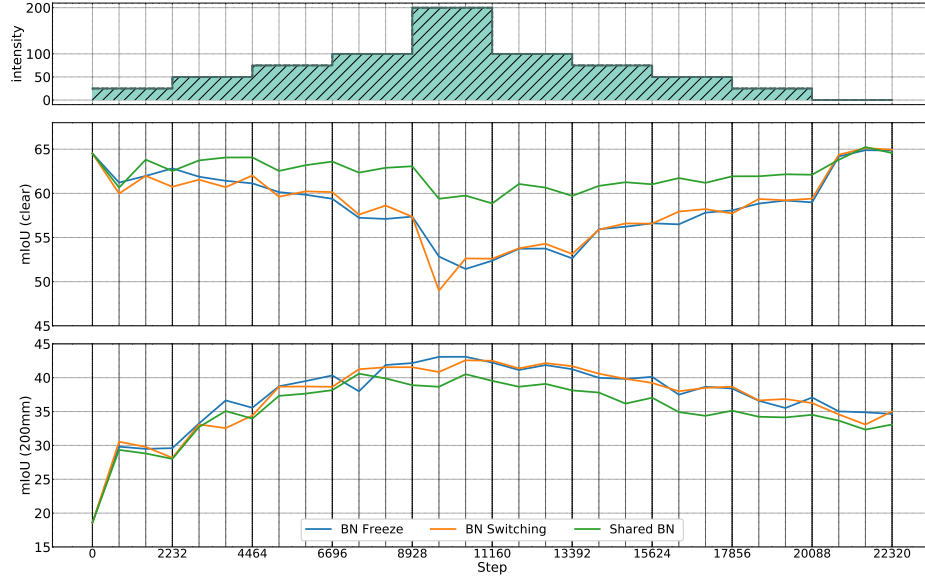


Figure 10. **Comparison between different Batch Normalisation (BN) policies.** The comparison is carried out using the Hybrid Switch on the Increasing Storm baseline scenario.

References

- [1] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *European Conference on Computer Vision*, pages 617–632. Springer, 2016. 1
- [2] 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, 2016. 1
- [3] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1
- [4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, page 1321–1330. JMLR.org, 2017. 7
- [5] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *CoRR*, 2020. 7
- [6] Maxime Tremblay, Shirsendu S. Halder, Raoul de Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision*, 2020. 1
- [7] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. 2019. 6