Semantic Self-adaptation: Enhancing Generalization with a Single Sample

Sherwin Bahmani[1,*] Oliver Hahn[1,*] Eduard Zamfir[2,*,†]
Nikita Araslanov[3,†] Daniel Cremers[3] Stefan Roth[1,4]

[1]TU Darmstadt [2]University of Würzburg [3]TU Munich [4]hessian.AI

Abstract

Previous studies on out-of-distribution generalization relied on the assumption of a static model: once the training process is complete, model parameters remain fixed at test time. We challenge this assumption with a self-adaptive *approach for semantic segmentation that adjusts the inference process to each test sample. Self-adaptation operates on two levels: (i) It employs a self-supervised loss that customizes the parameters of convolutional layers to the input image; (ii) in Batch Normalization layers, self-adaptation approximates the mean and the variance of the entire test distribution, which is assumed unavailable. It achieves this by interpolating between the training and the reference distribution derived from a* single *test sample. Following a rigorous evaluation protocol, our analysis leads to a surprising conclusion: Using a standard training procedure – unlike previous works – self-adaptation significantly outperforms strong baselines and sets new state-of-the-art segmentation accuracy on out-of-distribution test domains.*

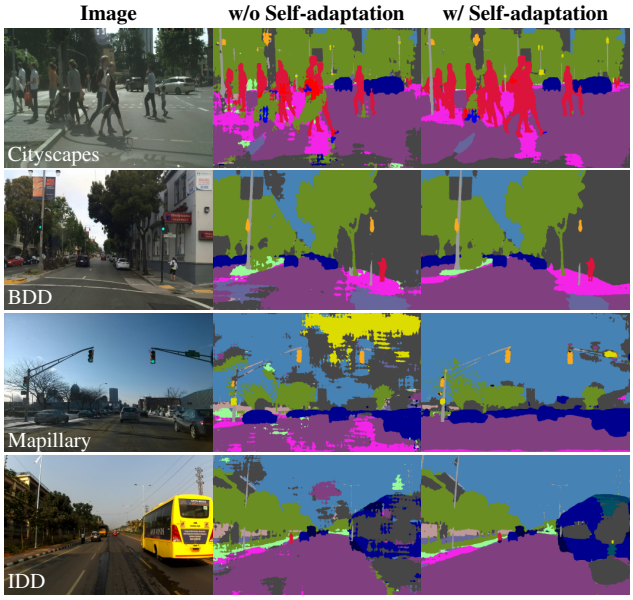

Figure 1. Self-adaptation perceptibly improves real-world generalization of semantic segmentation models (here, trained on GTA).

1. Introduction

In this work, we study the out-of-distribution (OOD) generalization problem of semantic segmentation from synthetic data [23,24] through the lens of adaptation. In contrast to previous work that focused on the training process [4,5,32], we leave the training stage unchanged, but replace the standard inference procedure with a technique inspired by domain adaptation methods [1,16]. The technique, that we term *self-adaptation*, leverages a self-supervised loss, which allows for adapting to a single test sample with a few parameter updates. Complementary to these loss-based updates, self-adaptation integrates feature statistics of the training data with those of the test sample in Batch Normalization (BN) layers [11]. Expanding upon related studies [25], we find that this normalization strategy improves the segmentation accuracy as well as the prediction uncertainty.

Related work. Domain randomization has been the prevalent approach in previous work on generalization of semantic segmentation models [4,9,14,15,22,32]. A few other techniques have also been successful in learning domain-invariant features, such as instance-selective whitening loss [6], swapping channel-wise statistics in normalization layers [27], meta-learning [13], distillation [5] and instance normalization (IN) layers [21]. At their core, all these training strategies are instantiations of the Empirical Risk Minimization (ERM), since they minimize the training loss w.r.t. samples from the training distribution. However, in the OOD setting studied here, this distribution is assumed to be distinct from the test distribution, hence the premise of ERM, the i.i.d. assumption of the training and test distribution, does not apply. In contrast to these works, we do not alter the training process or the model architecture, but instead focus on the inference process by adjusting the model to each input sample at test time. See Fig. 1 for result preview.

Updating model parameters at test time is not new [8,12].

*Equal contribution; † work primarily done while at TU Darmstadt.
Code and pre-trained models: https://github.com/visinf/self-adaptive.

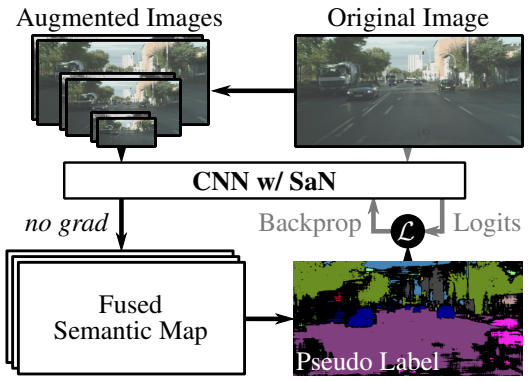

Figure 2. *Overview of self-adaptation operating on a single sample.* Sec. 2 elaborates on this process in more detail.

However, its surprising effectiveness in dealing with OOD samples has only recently begun to emerge [26, 29]. BN [11] and other normalization techniques have also been increasingly linked to OOD generalization [10, 25]. Schneider *et al*. [25] combine the source and target statistics during inference, where the statistics are weighted depending on the number of samples that these statistics aggregate. Nado *et al*. [18] propose using batch statistics during inference from the target domain instead of the training statistics acquired from the source domain. Note that these works [25, 26, 29] focus on *domain adaptation* in the context of image classification and typically assume access to more than a single sample from the target distribution. We here address *domain generalization* (DG) for semantic segmentation, which is fundamentally different, as it only allows access to *a single datum* from the test set [26].

2. Self-adaptation: Adapting to a single sample

Our approach, visualized in Fig. 2, uses data augmentation to create a mini-batch of images for each test sample. Based on the original test image, we first create a set of N augmented images by multi-scaling, horizontal flipping, and grayscaling. This augmented mini-batch passes through the CNN. We transform the produced semantic maps from the model back to the image plane of the original image using inverse similarity transformations, and obtain $m_{i,:,:,:}$ for every sample i in the mini-batch. This allows the model to have multiple predictions for one pixel. We then compute the mean $\bar{m}$ of these softmax probabilities along the mini-batch dimension i for class c and pixel (j, k) on the spatial grid, as

$$\bar{m}_{c,j,k} = \frac{1}{N} \sum_i m_{i,c,j,k}. \tag{1}$$

Using hyperparameter $\psi \in (0, 1)$, we compute a threshold value t_c from the maximum probability of every class to yield a class-dependent threshold t_c:

$$t_c = \psi \cdot \max(\bar{m}_{c,:,:}). \tag{2}$$

We finally extract the dominant class $c^*_{j,k}$ for every pixel by

$$c^*_{j,k} = \arg\max(\bar{m}_{:,j,k}). \tag{3}$$

We ignore low-confidence predictions using our class-dependent threshold t_c. Specifically, all pixels with a softmax probability below the threshold are set to an ignore label, while the remaining pixels use the dominant class $c^*_{j,k}$ as the pseudo label $u_{j,k}$,

$$u_{j,k} = \begin{cases} c^*_{j,k}, & \text{if } \max(\bar{m}_{:,j,k}) \geq t_{c^*_{j,k}} \\ \text{ignore}, & \text{else.} \end{cases} \tag{4}$$

The pseudo ground truth u for the test image is used to fine-tune the model for N_t iterations with gradient descent using cross-entropy. We determine all hyperparameters, *i.e*. resolution of the scales, threshold ψ, number of iterations N_t, and learning rate η, based on a validation dataset, which is distinct from the test domain (see Sec. 3.1). After the self-adaptation process, we produce the final prediction for the test sample using the updated model weights. Since no knowledge about the complete test distribution must leak into the model in the DG setting, we reset these weights to their initial values to process the next sample.

The `argmax` operation in Eq. (3) and applying the threshold in Eq. (4) aim to select only the most confident pixel predictions. If the confidence values are miscalibrated (*e.g*., due to the domain shift), a significant fraction of incorrect pixel labels will end up in the pseudo mask. To mitigate this issue, we introduce *self-adaptive normalization* (SaN). It improves the test-time behavior of BN layers by combining the inductive bias coming in the form of the running statistics from the source domain with statistics extracted from a single test instance. Let the *source* mean $\hat{\mu}_s$ and the variance $\hat{\sigma}^2_s$ denote the running average of the sample statistics in a BN layer (for some feature channel) at training time. If we had the *target* domain knowledge expressed by the sufficient statistics $\hat{\mu}_t$ and $\hat{\sigma}^2_t$, we could use those in place of $\hat{\mu}_s$ and $\hat{\sigma}^2_s$ in BN to compensate for the covariate shift. However, at test time we only have access to the sample estimates, μ_t and σ^2_t, provided by a single datum from the target distribution:

$$\mu_t = \tfrac{1}{HW} \sum_{j,k} z_{0,j,k}\,, \quad \sigma^2_t = \tfrac{1}{HW} \sum_{j,k} \left(z_{0,j,k} - \mu_t\right)^2, \tag{5}$$

where $z_0 \in \mathbb{R}^{H,W}$ is a spatial feature channel in a CNN of the target sample. The leading index of 0 emphasizes that only one image sample is available at test time. At inference time, we propose to compute the new mean and variance, $\hat{\mu}_t$ and $\hat{\sigma}_t$, as follows:

$$\hat{\mu}_t := (1-\alpha)\hat{\mu}_s + \alpha\mu_t\,, \quad \hat{\sigma}^2_t := (1-\alpha)\hat{\sigma}^2_s + \alpha\sigma^2_t, \tag{6}$$

where the hyperparameter $\alpha \in [0, 1]$ is chosen on the validation set and is fixed for all test images. Notably, this does not affect the behavior of the BN layers at training time; $\hat{\mu}_t$ and $\hat{\sigma}^2_t$ apply in the BN layers only at test time.

Table 1. *Segmentation accuracy and model calibration using SaN. (a)* The mean IoU (%) on three target domains (Cityscapes, BDD, IDD) across both backbones, trained on GTA and SYNTHIA. *t*-BN denotes train BN [11], while *p*-BN refers to prediction-time BN [18]. *(b)* The *ECE* (%) on three target domains (Cityscapes, BDD, IDD) across both backbones trained on GTA and SYNTHIA and compare to MC-Dropout [7]. The in-domain bounds on ECE when directly trained on Cityscapes/BDD/IDD are 7.46%/14.99%/9.48% for ResNet-50.

(a) Method	*IoU (%, ↑), Source: GTA / SYNTHIA*					
	CS		BDD		IDD	
ResNet-50						
w/ *t*-BN	30.95	31.83	28.52	24.30	32.78	24.73
w/ *p*-BN	**37.71**	33.83	31.67	23.36	30.85	23.39
w/ SaN (*Ours*)	37.54	**36.14**	**32.79**	**26.66**	**34.21**	**26.37**

(b) Method	*ECE (%, ↓), Source: GTA / SYNTHIA*					
	CS		BDD		IDD	
ResNet-50	37.28	37.50	35.61	43.19	27.73	40.11
w/ SaN (*Ours*)	30.57	30.96	30.94	33.27	26.90	36.31
w/ MC-Dropout	30.29	34.82	29.80	37.30	24.17	36.63
w/ both (*Ours*)	**25.50**	**30.66**	**27.36**	**33.06**	**22.62**	**35.60**

3. Experiments

3.1. Designing principled evaluation

Previous studies [4, 5, 21, 32] on DG for semantic segmentation used divergent evaluation methodologies, which exacerbates the comparison and reproducibility in follow-up research. We revise the evaluation protocol which allows us to follow the best practice in machine learning, yet will not disadvantage previous work in the empirical comparison. In particular, we follow four principles: *(i)* the test set must comprise multiple domains; *(ii)* a single model must be used for all domains; *(iii)* the validation set must be clearly specified; *(iv)* no test images may be used for model selection. These principles follow naturally from the requirements of DG, with *(iii)* and *(iv)* also being widely accepted in the research community. Nevertheless, we found that *no previous work on DG for semantic segmentation has yet fulfilled all of these principles.* To implement these principles, we assume access to two data distributions for model training and validation, the *source data* and the *validation set*. We assess the generalization ability of the model yielded by the validation process on qualitatively distinct *target sets*. We now concretize the datasets used in this study, which focus on traffic scenes for compatibility with previous work [4, 21, 32].

Source data. We train our model on the training split of two synthetic datasets (mutually exclusive) with low-cost ground truth: GTA [23] and SYNTHIA [24]. Importantly, these datasets exhibit domain shift w.r.t. real world.

Validation set. For model selection and hyperparameter tuning, we use the validation set of WildDash [33]. It is understood to be of limited quantity (compared to the source data), owing to its more costly annotation compared to the source data. In contrast to the training set, however, it may bear closer visual resemblance to the target domains.

Target data. Our test domain comprises multiple real-world segmentation benchmarks: Cityscapes [23], BDD [31], IDD [28] and Mapillary [20]. For consistency with previous work, we use the validation sets of these datasets, which are not accessible to the model until the test time.

3.2. SaN improves OOD accuracy and calibration

For both source domains (GTA, SYNTHIA) in combination with all main target domains (Cityscapes, BDD, IDD), we investigate the effect of SaN on out-of-domain segmentation accuracy and calibration. We first select α on the validation set and report the results in Tab. 1. In Tab. 1a, we compare the SaN accuracy on the target domains with *t*-BN and *p*-BN [18]. While *t*-BN uses the running average $\hat{\mu}_s$ and $\hat{\sigma}_s^2$ in BN layers for the test data, as was originally suggested [11], *p*-BN employs μ_t and σ_t^2 of the input test sample instead, which can improve the prediction accuracy in the OOD scenario [18]. Remarkably, SaN improves the mean IoU not only of the *t*-BN baseline (*e.g.*, by 4.1% IoU with ResNet-50 trained on GTA, on average), which represents an established evaluation mode, but also over *p*-BN. When trained on SYNTHIA, SaN yields stable improvement over the *t*-BN baseline even despite *p*-BN being significantly worse than *t*-BN in this scenario. Furthermore, we found that the calibration of our models, in terms of the expected calibration error (ECE) [19], also improves. As shown in Tab. 1b, not only does SaN substantially enhance the baseline, but it even tends to outperform the widely used approach for uncertainty estimation, the MC-Dropout [7]. Rather surprisingly, SaN exhibits a complementary effect with MC-Dropout: the calibration of the predictions improves even further when both methods are used jointly. This observation holds even for the segmentation accuracy. For example, our model trained on GTA and tested with SaN *and* MC-Dropout (*i.e.*, by averaging the predictions) improves the IoU of the SaN-only inference on Cityscapes, BDD, IDD by 1.3%, 2.34%, 1.29%, whereas MC-Dropout alone does not benefit model accuracy. Overall, the combined results from Tab. 1 demonstrate that SaN improves both the model's prediction accuracy and calibration on OOD samples.

3.3. Self-adaptation: New state of the art

We compare self-adaptation with state-of-the-art DG methods in Tab. 2. Most of the other methods report their results on weakly tuned baselines, and we empirically found

Table 2. *Mean IoU (%) comparison to state-of-the-art DG methods* for both source domains (GTA, SYNTHIA) as well as three target domains (Cityscapes, Mapillary, BDD). In-domain training to obtain the upper bounds uses our baseline DeepLabv1 following the same schedule as with the synthetic datasets. (‡), (†) and (††) denote the use of FCN [17], DeepLabv2 [2] and DeepLabv3+ [3], respectively.

	Method	*Backbone: ResNet-50*			*Backbone: ResNet-101*		
		CS	Mapillary	BDD	CS	Mapillary	BDD
	In-domain Bound	71.23	58.39	58.53	73.84	62.81	61.19
GTA	No Adapt	32.45	25.66	26.73	33.56	28.33	27.76
	DRPC‡ [32]	37.42 ↑*4.97*	34.12 ↑*8.46*	32.14 ↑*5.41*	42.53 ↑*8.97*	38.05 ↑*9.72*	38.72 ↑*10.96*
	No Adapt	35.16	31.29	29.71	35.73	33.42	34.06
	WildNet†† [15]	44.62 ↑*9.46*	46.09 ↑*14.77*	38.42 ↑*8.71*	45.79 ↑*10.06*	47.08 ↑*13.66*	**41.73** ↑*7.67*
	No Adapt	29.32	28.33	25.71	30.64	28.65	27.82
	SAN-SAW† [22]	39.75 ↑*10.43*	41.86 ↑*13.53*	37.34 ↑*11.63*	45.33 ↑*14.69*	40.77 ↑*12.12*	41.18 ↑*13.36*
	No Adapt	30.95	34.56	28.52	32.90	36.00	32.54
	Self-adaptation *(Ours)*	**45.13** ↑*14.18*	**47.49** ↑*12.93*	**39.61** ↑*11.09*	**46.99** ↑*14.09*	**47.49** ↑*11.49*	40.21 ↑*7.67*
SYNTHIA	No Adapt	28.36	27.24	25.16	29.67	28.73	25.64
	DRPC‡ [32]	35.65 ↑*7.29*	32.74 ↑*5.50*	31.53 ↑*6.37*	37.58 ↑*7.91*	34.12 ↑*5.39*	34.34 ↑*8.70*
	No Adapt	23.18	21.79	24.50	23.85	21.84	25.01
	SAN-SAW† [22]	38.92 ↑*15.74*	34.52 ↑*12.73*	**35.24** ↑*10.74*	40.87 ↑*17.02*	37.26 ↑*15.42*	**35.98** ↑*10.97*
	No Adapt	31.83	33.41	24.30	37.25	36.84	29.32
	Self-adaptation *(Ours)*	**41.60** ↑*9.77*	**41.21** ↑*7.80*	33.35 ↑*9.05*	**42.32** ↑*5.07*	**41.20** ↑*4.36*	33.27 ↑*3.95*

such suboptimal baselines to be the easiest to improve upon. Nevertheless, we show consistent improvements even over a carefully tuned, hence substantially stronger baseline regardless of the backbone architecture or source data. Our single model with self-adaptation even outperforms DRPC [32] and FSDR [9] on most benchmarks (*e.g.*, by $4.2 - 9.4\%$ on Mapillary with ResNet-101). These methods train individual models for each target domain; FSDR [9] further uses the target domains for hyperparameter tuning, hence violates our OOD evaluation protocol. Note that ASG [5] and CSG [4] (as well as DRPC [32]) require access to a distribution of real images for training, while IBN-Net [21] modifies the model architecture. Our approach requires neither. WildNet [15] appears to be more accurate than self-adaptation on BDD with ResNet-101. However, it uses a more advanced architecture (DeepLabv3+ *vs.* DeepLabv1). In fact, we tested self-adaptation with DeepLabv3+ as well, and found it to outperform WildNet in this setup by 2.79% IoU, as expected. Similarly, SAN-SAW [22] reaches higher accuracy on BDD, if trained on SYNTHIA, presumably due to the ASPP module [2] that we do not use. Self-adaptation considerably outperforms SAN-SAW in all other scenarios. Overall, despite adhering to a stricter evaluation practice and a simpler model architecture, self-adaptation overwhelmingly exceeds the segmentation accuracy of previous work.

Comparison to Tent [29]. Like self-adaptation, Tent [29] also updates model parameters at test time. However, different from constructing the pseudo labels based on well-calibrated predictions in our self-adaptation, Tent simply minimizes the entropy of a single-scale prediction. Tent also limits the adaptation to updating only the BN parameters, whereas our self-adaptation generalizes this process to convolutional layers. To demonstrate these advantages, we train HRNet-W18 [30] on GTA and compare the IoU on Cityscapes to the equivalent configuration of Tent. *Under a comparable computational budget of 10 model update iterations*, self-adaptation substantially outpaces Tent, by a remarkable 7.7% IoU (from 36.4% to 44.1%). Notably, SaN alone reaches 40.0%, hence already outperforms Tent significantly with a single forward pass, by 3.6%.

Qualitative analysis (Fig. 1). Self-adaptation yields a clearly perceivable improvement over the baseline, especially in terms of image boundary consistency. It exhibits more homogeneous semantic masks and reduces spatial irregularities of the baseline (*e.g.*, "sidewalk" errors).

4. Conclusion

We presented and studied a self-adaptive *inference* process. Our analysis clearly demonstrates that a single sample from the test domain can already suffice to substantially improve model predictions. The accuracy improvement shown by our experiments is remarkably substantial, despite no changes to the training process or the model architecture, unlike in previous works. We hope that these encouraging results will incentivize our research community to study self-adaptive techniques in other application domains, such as panoptic segmentation, or monocular depth prediction.

References

[1] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. In *CVPR*, 2021. 1

[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. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. 4

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 4

[4] Wuyang Chen, Zhiding Yu, Shalini De Mello, Sifei Liu, Jose M. Alvarez, Zhangyang Wang, and Anima Anandkumar. Contrastive syn-to-real generalization. In *ICLR*, 2021. 1, 3, 4

[5] Wuyang Chen, Zhiding Yu, Zhangyang Wang, and Animashree Anandkumar. Automated synthetic-to-real generalization. In *ICML*, 2020. 1, 3, 4

[6] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. RobustNet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *CVPR*, 2021. 1

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 3

[8] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *ICCV*, 2009. 1

[9] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. FSDR: Frequency space domain randomization for domain generalization. In *CVPR*, 2021. 1, 4

[10] Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. In *CVPR*, 2019. 2

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 1, 2, 3

[12] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1409–1422, 2012. 1

[13] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Pin the memory: Learning to generalize semantic segmentation. In *CVPR*, 2022. 1

[14] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R. Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *ICCV*, 2021. 1

[15] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. WildNet: Learning domain generalized semantic segmentation from the wild. In *CVPR*, 2022. 1, 4

[16] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *ICLR*, 2017. 1

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 4

[18] Zachary Nado, Shreyas Padhy, D. Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv:2006.10963 [cs.LG]*, 2020. 2, 3

[19] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI*, 2015. 3

[20] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The Mapillary Vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 3

[21] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via IBN-Net. In *ECCV*, 2018. 1, 3, 4

[22] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *CVPR*, 2022. 1, 4

[23] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 1, 3

[24] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 1, 3

[25] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020. 1, 2

[26] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020. 2

[27] Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris N. Metaxas. CrossNorm and SelfNorm for generalization under distribution shifts. In *ICCV*, 2021. 1

[28] Girish Varma, Anbumani Subramanian, Anoop M. Namboodiri, Manmohan Chandraker, and C. V. Jawahar. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*, 2019. 3

[29] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 2, 4

[30] Jingdong Wang, Ke Sun, and Tianheng Cheng et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3349–3364, 2021. 4

[31] 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 *CVPR*, 2020. 3

[32] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto L. Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *ICCV*, 2019. 1, 3, 4

[33] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernández Domínguez. WildDash – Creating hazard-aware benchmarks. In *ECCV*, 2018. 3