Texture Learning Domain Randomization for Domain Generalized Segmentation (Supplementary Material)

Appendix

In Appendix, we provide more details and additional experimental results of our proposed Texture Learning Domain Randomization (TLDR). The sections are organized as follows:

- A: Details of Dimension Estimation
- B: Loss Graph
- C: Theoretical analysis on $\mathcal{L}_{\text{TR}}$ and $\mathcal{L}_{\text{TG}}$
- D: Details of t-SNE Visualizations
- E: Experiment on Class Uniform Sampling
- F: Hyperparameter Analysis
- G: Experiment on Multi-source Setting
- H: Pseudocode and Source Code Implementation
- I: More Qualitative Results

A. Details of Dimension Estimation

Figure S1. Visualization of image pairs with a certain semantic concept: (a) texture pair and (b) shape pair. The image pairs are generated by using the Style Transfer Module (STM).

In Section 5.3, we conduct an experiment to verify whether our TLDR enhances the ability to encode texture information in latent representations. Esser *et al*. proposed a method for estimating the dimensions of semantic concepts in latent representations [4], and Islam *et al*. utilized the method on texture and shape [8].

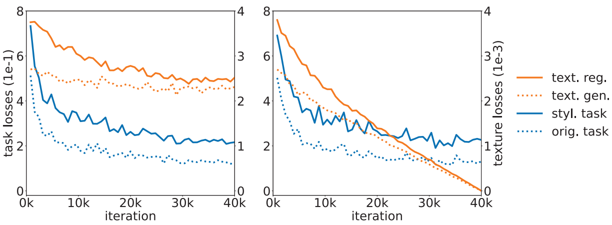
Let I_a and I_b be a pair of images that are similar in terms of a certain semantic concept, as shown in Figure S1. The latent representations z_i^a and z_i^b correspond to the images I_a and I_b, respectively, and are estimated by the task model at the i^{th} layer. The method hypothesizes that high mutual information between two latent representations indicates that the model effectively encodes the corresponding semantic concept. It is known that the mutual information between latent representations obeys the lower bound of Equation S1 [5].

$$\texttt{MI}(z_i^a, z_i^b) \geq -\frac{1}{2}\log(1 - \texttt{corr}(z_i^a, z_i^b)), \tag{S1}$$

where $\texttt{corr}(\cdot)$ represents correlation, and $\texttt{MI}(\cdot)$ represents mutual information. We calculate the mutual information by assuming it satisfies the inequality with a tight condition. We use mutual information to measure the scores of encoding texture and shape. Lastly, the percentage of semantic concepts in latent representations is calculated by taking a softmax function over each of the three scores, the texture score, the shape score, and a fixed baseline score.

In the experiment, we create image pairs using the Style Transfer Module (STM). A texture pair consists of two stylized images that share the same style but generated from two different content images using the STM. Conversely, a shape pair comprises two stylized images derived from a single content image but with distinct styles, again using the STM. The dimensionality is then calculated for the texture and shape pairs that share one common image (*e.g.*, I_a in Figure S1). We conduct the experiment on 200 sets of shape-texture image pairs and estimate the dimensions by taking the average.

B. Loss Graph



(a) w/o LDF (b) w/ LDF

Figure S2. Graphs of the original task loss, the stylized task loss, the texture regularization loss, and the texture generalization loss (a) without LDF and (b) with LDF.

In Section 4.4, we introduce the Linear Decay Factor (LDF), which is based on the observation that the texture regularization and generalization losses exhibit relatively constant behavior compared to the task losses. In Figure S2a, we can observe that the original task loss and the stylized task loss continue to decrease with each iteration, whereas the texture regularization loss and the texture generalization loss remain relatively constant. Figure S2b shows that the application of LDF results in aligned scales for the losses. Higher-order functions such as cosine annealing [11] can also be applied to improve performance.

C. Theoretical analysis on $\mathcal{L}_{\text{TR}}$ and $\mathcal{L}_{\text{TG}}$

It is known that texture can be represented as low-level statistics of image features. Meanwhile, [10] theoretically showed that matching the Gram-matrices in the l_2 norm is equivalent to minimizing the Maximum Mean Discrepancy (MMD) with a second-order polynomial kernel. This minimization implies aligning the low-level statistics between features. Thus, $\mathcal{L}_{\text{TR}}$ and $\mathcal{L}_{\text{TG}}$ are designed to compare only the low-level statistics (*i.e.*, texture) using Gram-matrices, while excluding the high-level statistics present in entire features.

D. Details of t-SNE Visualizations

In Figure 3, we demonstrate the significance of utilizing texture by presenting t-SNE [16] plots of the shape and texture features for the road, sidewalk, and terrain classes. In the Cityscapes [3] dataset, we select 500 random instances from each class that contained more than $5k$ pixels. For feature extraction, we utilize feature maps from the final layer of Segformer-B5 [17] model pre-trained on ImageNet. Subsequently, the shape features were derived using Canny edge [1], while texture features were extracted via Gram-matrix.

E. Experiment on Class Uniform Sampling

	Case	C	B	M	S
1	w/ CUS	**48.63**	**45.49**	**50.06**	38.45
2	w/o CUS	47.58	44.88	48.80	**39.35**

Table S1. Experiment on Class Uniform Sampling (CUS) on TLDR. The model is trained on GTA [13] using ResNet-101 as the encoder and evaluated on Cityscapes [3], BDD [18], Mapillary [12], and SYNTHIA [14]. The default setting is marked in gray.

Some existing methods [2, 9, 20] used Class Uniform Sampling (CUS) technique [7] to alleviate class imbalance problem in DGSS. We conduct experiments without CUS in the default setting for a fair comparison with DGSS methods without CUS. Table S1 shows the performance ablation results of TLDR when trained using CUS. The model is trained on GTA [13] using ResNet-101 as the encoder. One can see that our TLDR achieves better results with CUS (cf. row 1) compared to without CUS (cf. row 2).

F. Hyperparameter Analysis

In hyperparameter analysis, we use ResNet-101 as the encoder, train on GTA [13], and evaluate on Cityscapes [3], BDD [18], Mapillary [12], and SYNTHIA [14]. The best results are **highlighted**.

	α_{orig}	α_{styl}	C	B	M	S
1	0.1	0.9	46.58	43.98	**49.09**	38.04
2	0.3	0.7	47.41	43.71	47.09	38.57
3	0.5	0.5	**47.58**	**44.88**	48.80	39.35
4	0.7	0.3	47.05	44.11	47.01	38.57
5	0.9	0.1	44.25	43.11	40.33	**39.97**

Table S2. Hyperparameter analysis on the original and stylized task loss weights. The model is trained on GTA using ResNet-101 as the encoder and evaluated on Cityscapes, BDD, Mapillary, and SYNTHIA. The default setting is marked in gray.

Task loss weights. We analyze the performance changes resulting from variations in the task loss weights α_{orig} and α_{styl}. Table S2 shows the results for the analysis. The best performance is achieved when the values of α_{orig} and α_{styl} are both 0.5 (cf. row 3). We assume that the best performance is achieved in this case because it balances the objectives for texture and shape. Additionally, we observe that setting α_{orig} to 0.1 leads to relatively good performance (cf. row 1), while setting α_{styl} to 0.1 significantly decreases performance (cf. row 5). We assume that the reason is that shape provides the primary prediction cues, while texture serves as a complementary cue to shape.

	Case	C	B	M	S
1	u_l=$5 \times 10^{-l-3}$	47.32	44.60	44.99	**39.52**
2	u_l=$5 \times 10^{-l-2}$	**47.58**	**44.88**	**48.80**	39.35
3	u_l=$5 \times 10^{-l-1}$	46.41	44.49	44.60	39.33
4	v_l=$5 \times 10^{-l-3}$	46.56	44.66	45.92	**40.15**
5	v_l=$5 \times 10^{-l-2}$	**47.58**	**44.88**	**48.80**	39.35
6	v_l=$5 \times 10^{-l-1}$	45.35	43.17	45.33	40.01

Table S3. Hyperparameter analysis on the texture regularization and generalization parameters. The model is trained on GTA using ResNet-101 as the encoder and evaluated on Cityscapes, BDD, Mapillary, and SYNTHIA. The default setting is marked in gray.

Weighting factors. To examine the effects of the weighting factors u_l and v_l on the texture regularization and generalization losses, we conduct ablation experiments by manipulating the scale of the weighting factors. As shown in Table S3, we vary the weight factors by decreasing them by 0.1 times (cf. rows 1 and 4) and increasing them by 10 times (cf. rows 3 and 6) relative to the default setting (cf. rows 2 and 5). The experimental results indicate that the default setting achieves the best performance.

G. Experiment on Multi-source Setting

	Train on G+S		
Methods	C	B	M
RobustNet [2]	37.69	34.09	38.49
SHADE [20]	47.43	40.30	47.60
TLDR (ours)	**48.83**	**42.58**	**47.80**

Table S4. Experimental results on a multi-source setting. The model is trained on GTA and SYNTHIA using ResNet-50 as the encoder and evaluated on Cityscapes, BDD, and Mapillary. Our method is marked in gray.

We compare our method against other DGSS methods [2, 20] in a multi-source setting. In the experiment, we train on GTA [13] and SYNTHIA [14], and evaluate on Cityscapes [3], BDD [18], and Mapillary [12]. As shown in Table S4, our method consistently showed superior performance across all benchmarks.

H. Pseudocode and Source Code Implementation

Algorithm 1 is the PyTorch-style pseudocode for the proposed TLDR. The pseudocode contains the process of computing the original task loss, the stylized task loss, the texture regularization loss, and the texture generalization loss for one training iteration. For more details on the implementation of TLDR, please refer to the source code. The source code to reproduce TLDR is provided at https://github.com/ssssshwan/TLDR. A detailed description of the source code is explained in the contained `README.md` file.

Algorithm 1: PyTorch-style pseudocode for one training iteration of TLDR.

```
# STM : Style Transfer Module
# n, CE : L2 matrix norm, Cross Entropy Loss
# f_T, f_I : get task, ImageNet L (total number of layers) feature maps list
# h_T : Semantic segmentation decoder
# g :  Gram-matrix from a feature map
def g(F):
    B, C, H, W = F.size()
    F = F.view(B, C, H * W)
    G = torch.bmm(F, F.transpose(1,2))
    return G.div(C * H * W)

x_s, y = source_loader()
x_r = style_loader()
x_sr = STM(x_s, x_r)
p_s, p_sr = h_T(f_T(x_s)), h_T(f_T(x_sr)) # Inference x_s and x_sr
L_orig, L_styl = CE(p_s, y), CE(p_sr, y) # Calculate task losses

for l in range (L): # Iterate L layers (L_tr = 0, L_tg = 0)
    G_I_s, G_T_s = g(f_I(x_s)[l]).detach(), g(f_T(x_s)[l])
    L_tr += n(G_I_s - G_T_s) # Texture reg loss in layer l

    G_T_r, G_T_sr = g(f_T(x_r)[l]), g(f_T(x_sr)[l])
    diff = G_T_sr - G_T_s
    mask = diff > threshold # Random Style Masking
    L_tg += n((G_T_r - G_T_sr) * mask)# Texture gen loss in layer l
```

I. More Qualitative Results

This section compares the qualitative results between DGSS methods [2, 19] and our proposed TLDR using ResNet-50 [6] encoder. We obtain qualitative results in two settings. *First*, Cityscapes [3] to RainCityscapes [7] (Figure S3) and Foggy Cityscapes [15] (Figure S4). *Second*, GTA [13] to Cityscapes [3] (Figure S5), BDD [18] (Figure S6), Mapillary [12] (Figure S7), and SYNTHIA [14] (Figure S8). Our TLDR demonstrates superior results compared to the existing DGSS methods across the various domains.

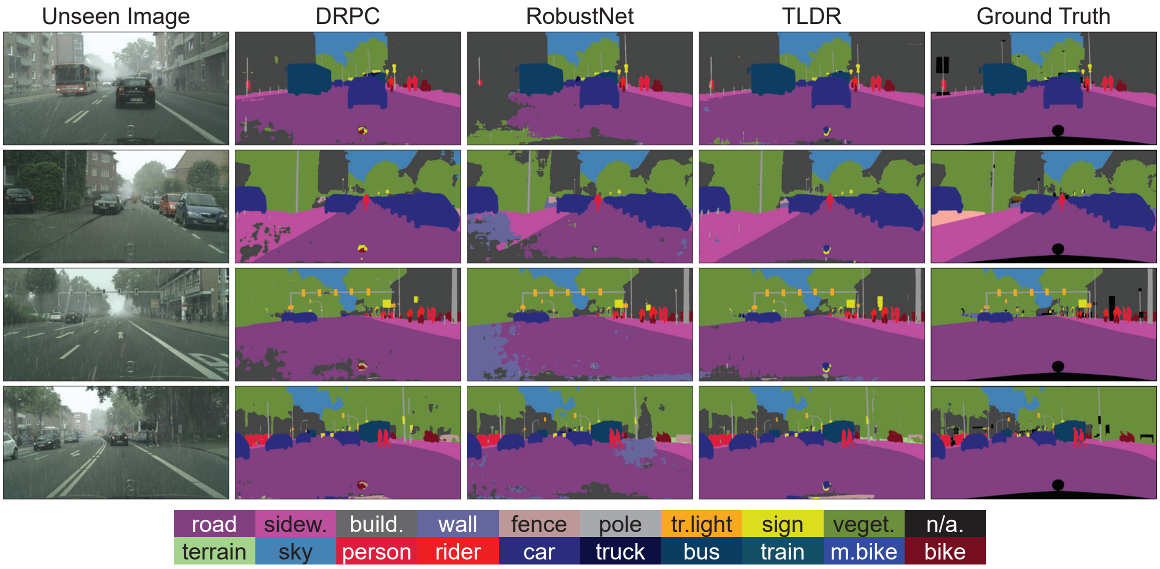


Figure S3. Qualitative results of DGSS methods [2, 19] and our TLDR on Cityscapes→RainCityscapes.

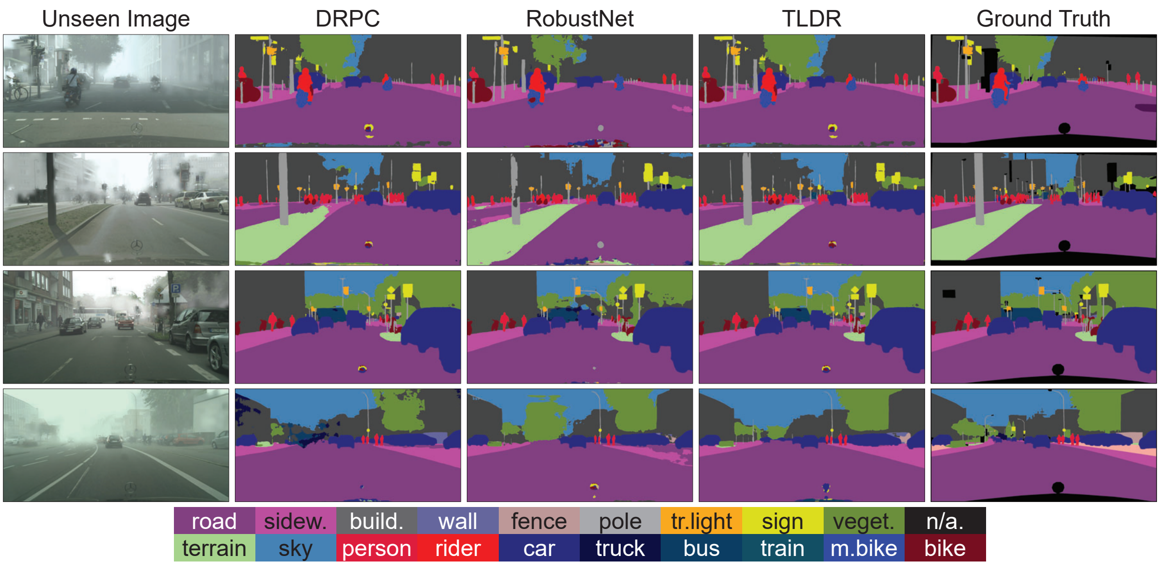


Figure S4. Qualitative results of DGSS methods [2, 19] and our TLDR on Cityscapes→Foggy Cityscapes.

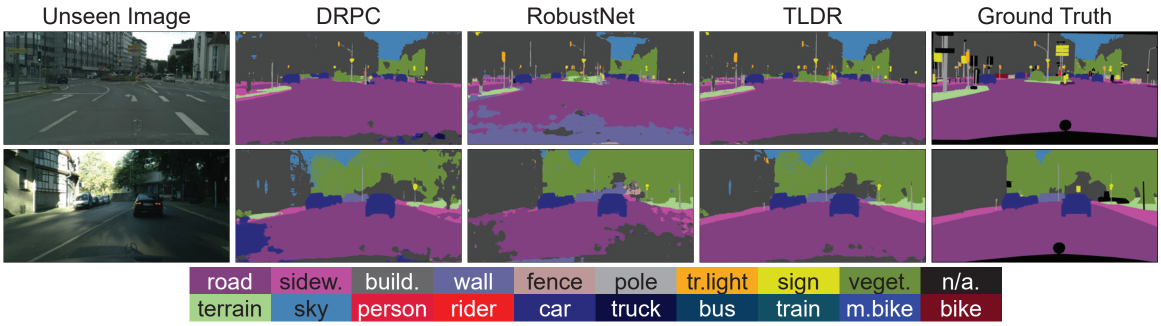


Figure S5. Qualitative results of DGSS methods [2, 19] and our TLDR on GTA→Cityscapes.

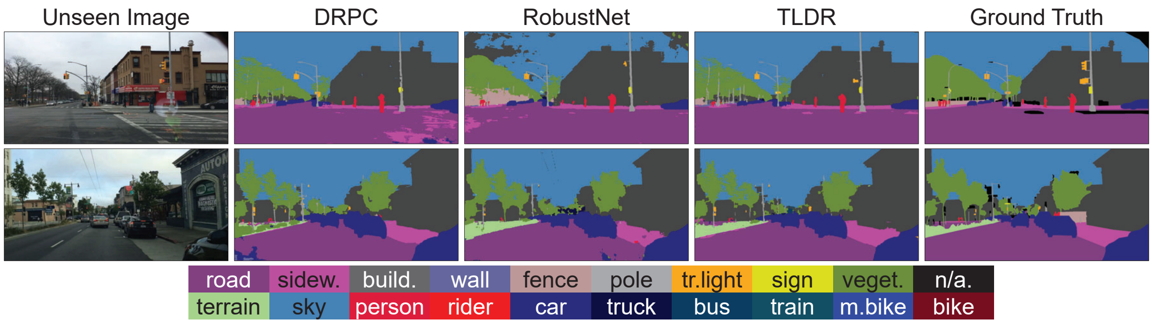


Figure S6. Qualitative results of DGSS methods [2, 19] and our TLDR on GTA→BDD.

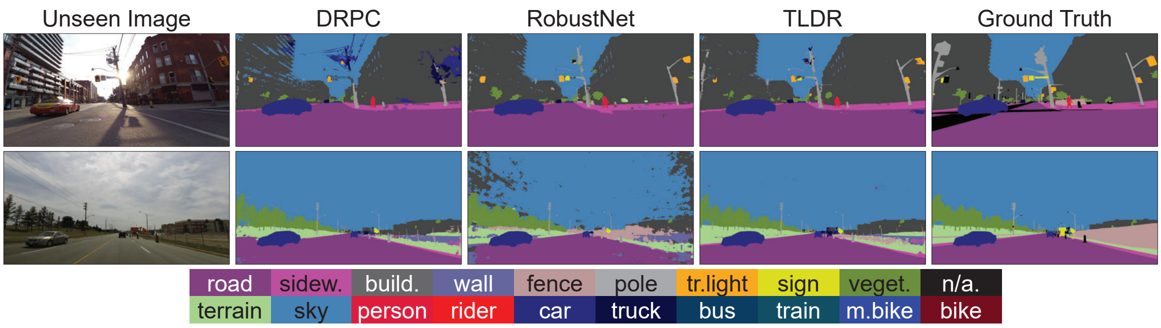


Figure S7. Qualitative results of DGSS methods [2, 19] and our TLDR on GTA→Mapillary.

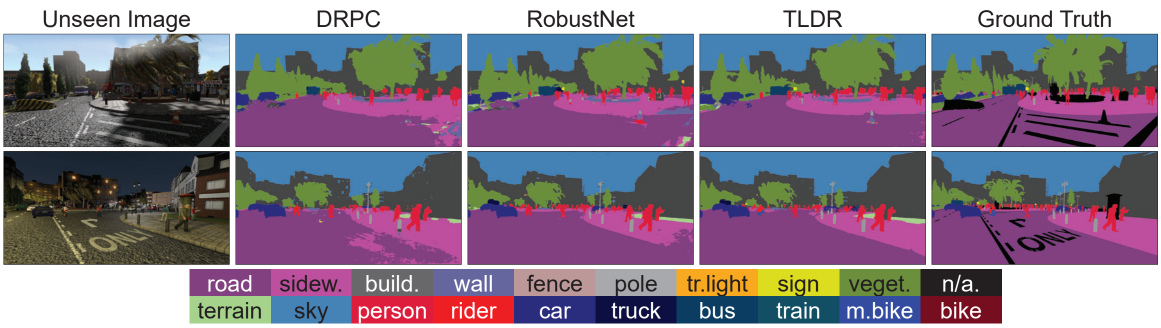


Figure S8. Qualitative results of DGSS methods [2, 19] and our TLDR on GTA→SYNTHIA.

References

[1] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, (6):679–698, 1986.

[2] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11580–11590, 2021.

[3] 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.

[4] Patrick Esser, Robin Rombach, and Bjorn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9223–9232, 2020.

[5] David V Foster and Peter Grassberger. Lower bounds on mutual information. *Physical Review E*, 83(1):010101, 2011.

[6] 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.

[7] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8022–8031, 2019.

[8] Md Amirul Islam, Matthew Kowal, Patrick Esser, Sen Jia, Bjorn Ommer, Konstantinos G Derpanis, and Neil Bruce. Shape or texture: Understanding discriminative features in cnns. *arXiv preprint arXiv:2101.11604*, 2021.

[9] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. WildNet: Learning Domain Generalized Semantic Segmentation from the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9936–9946, 2022.

[10] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017.

[11] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[12] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4990–4999, 2017.

[13] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of the European conference on computer vision (ECCV)*, pages 102–118, 2016.

[14] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[15] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision (IJCV)*, 126:973–992, 2018.

[16] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(11), 2008.

[17] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:12077–12090, 2021.

[18] 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 2636–2645, 2020.

[19] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2100–2110, 2019.

[20] Yuyang Zhao, Zhun Zhong, Na Zhao, Nicu Sebe, and Gim Hee Lee. Style-hallucinated dual consistency learning for domain generalized semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 535–552, 2022.