Table 1. Performance comparison in terms of mIoU (%) between DG methods. The best and second best results are highlighted and underlined, respectively. † denotes our re-implementation of the respective method. G, C, B, M and S denote GTA5, Cityscapes, BDDS, Mapillary and SYNTHIA, respectively.

Table 2. Comparison results between ours and Domain Adaptation methods on GTA5→Cityscapes. DA and DG denote Domain Adaptation and Domain Generalization respectively.

Table 3. Comparison results between ours and Domain Adaptation methods on SYNTHIA→Cityscapes.

Appendix A. Evaluation on other DG settings

Existing DG methods [3, 11–13, 20] only focus on three domain generalization settings, i.e., (1) G → C, B, M, & S; (2) S → C, B, M, & G and (3) C → G, S, B, & M while losing the sight of the other two DG settings: (4) B → G, S, C, & M and (5) M → G, S, C, & M. However, recently several studies [2, 6] stress the importance of the last two settings. Therefore, we consider a more comprehensive evaluation which performs generalization from each of them. Results on the last two settings are reported in Tab. 1, suggesting that our model consistently achieves the state-of-the-art results on all settings and backbones.

Appendix B. Comparison with DA methods

Domain Adaptation (DA) methods require access to the target domain to solve domain shift problems. In contrast, our method is designed in Domain Generalization (DG) manner for broad generalization to totally unseen domains without accessing any target domain data. Therefore, the target domain-accessible DA methods have the inherent performance superiority than DG methods which are target domain-agnostic. In order to see whether our approach is up to the performance standard of DA, we compare the results of our method with those reported from several previous state-of-the-art DA methods. From Tab. 2 and Tab. 3, we can see that the generalization performance of our method...
outperforms the adaptation performance of most other techniques. In addition, no target-domain data is needed in our method, resulting in more extensive applicability.

Appendix C. Further Implementation Details

We follow previous work [3, 11] to adopt normalization and whitening at the first two stages of convolution layers, since shallow layers encode more style information [11]. As shown in Fig. 1, for each backbone network, we impose SAN and SAW after stage 1 and stage 2.

As shown in Fig. 1, for each backbone network, we impose SAN and SAW after stage 1 and stage 2.

Appendix D. Computational complexity

As shown in Tab. 4, compared to the baseline, our methods performs domain generalization with negligible addition in both training and inference time. This is because the proposed modules are only implemented in the first two layers of the network, for only four main categories i.e. $C = 4$, see Sec. 5.5 of the main paper. The additional memory overhead from our modules is less than 2G.

Appendix E. More qualitative results

Fig. 2 shows more qualitative results under various unseen domains. We demonstrate the effects of the proposed semantic-aware feature matching by comparing the segmentation results from our proposed approach and the baseline. In the setting of GTA5 → Mapillary, the baseline fails to cope with these weather changes, while ours still shows fair results. Under the illumination changes as shown in GTA5 → BDDS, our method finds the road and sidewalk clearer than the baseline.

References


Figure 1. Detailed Architecture of our approach with the backbone of VGG and ResNet.

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Methods</th>
<th>Memory (G)</th>
<th>Training Time (s)</th>
<th>Inference Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>Baseline</td>
<td>5.28</td>
<td>0.37</td>
<td>48.17</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>7.41</td>
<td>0.39</td>
<td>48.20</td>
</tr>
<tr>
<td>Res-50</td>
<td>Baseline</td>
<td>6.43</td>
<td>0.40</td>
<td>48.84</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>8.04</td>
<td>0.41</td>
<td>48.86</td>
</tr>
<tr>
<td>Res-101</td>
<td>Baseline</td>
<td>8.24</td>
<td>0.43</td>
<td>50.31</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>10.17</td>
<td>0.45</td>
<td>50.37</td>
</tr>
</tbody>
</table>
Figure 2. Qualitative results of our approach generalizing from GTA5 to other four domains.


[12] Xingang Pan, Xiaohang Zhan, Jianping Shi, Xiaoou Tang,


