Learning Instance-Specific Adaptation for Cross-Domain Segmentation Supplementary Material

Yuliang Zou[1] Zizhao Zhang[2] Chun-Liang Li[2] Han Zhang[3]
Tomas Pfister[2] Jia-Bin Huang[4]

[1]Virginia Tech [2]Google Cloud AI
[3]Google Brain [4]University of Maryland, College Park

Supplementary Matarial

In this supplementary document, we provide additional experimental results and details to complement the main manuscript. First, we provide the implementation details of our experiments. Second, we describe the three data augmentation strategies we explore. Lastly, we show qualitative results of different cross-domain segmentation settings.

A Implementation details

We conduct all the experiments using the PyTorch framework with one single V100 GPU.

Semantic segmentation. We start with a DeepLabv2 model [1] with a ResNet-101 backbone pre-trained on ImageNet [8]. We follow Chen et al. [2] to pre-train the *source-only* model. In our proposed learning stage, we freeze the model parameters except for the proposed module and train the model with the SGD optimizer with momentum 0.9 and weight decay 5×10^{-4}. We use a learning rate of 2.5×10^{-3} for InstCal-U and a learning rate of 2.5×10^{-2} for InstCal-C. We adopt the polynomial learning rate decay schedule as in Chen et al. [2] and set the total number of training iterations to 80,000. We set the batch size to one.

To compare with recent domain generalizing semantic segmentation methods, we adopt the implementation and off-the-shelf models from RobustNet [4]. We do not conduct pre-training and directly train these off-the-shelf models using the proposed method. For these DeepLabv3+ models, we set the learning rate as 2.5×10^{-3} and reduce the batch size for single GPU training (i.e., 8 for ResNet-50, 4 for ShuffleNetV2 and MobileNetV2). The other training hyper-parameters are kept the same. Please refer to Choi et al. [4] for more details.

Panoptic segmentation. We adopt off-the-shelf models from Panoptic-DeepLab [3], implemented in the PyTorch Detectron2 codebase. We use a learning rate of 2.5×10^{-4}. We set the batch size to 4. The other training hyper-parameters are kept the same. Please refer to the PyTorch implementation for more details.[1]

[1] https://github.com/facebookresearch/detectron2/tree/main/projects/Panoptic-DeepLab

B Strong data augmentation strategies

We provide details for the three strong data augmentation strategies we adopted. **RandAugment [5].** RandAugment samples m operations from a pre-defined list of image augmentation operations and composes them to form the final augmentation for each input data. In this paper, we set $m = 2$, and we only adopt the color-related augmentation operations in the pre-defined list: Identity, AutoContrast, Invert, Equalize, Solarize, Posterize, Color, Brightness, Sharpness. **AugMix [7].** AugMix is proposed to improve model robustness and uncertainty estimation. AugMix constructs three augmentation paths with cascaded one, two, and three augmentation operations for each input data. The three augmented data are then linearly combined, where the combination weights are sampled from a Dirichlet distribution. Finally, the original and augmented images are linearly combined to construct the final augmented image, where the combination weight is sampled from a Beta distribution. We adopt all the augmentation operations, including geometric transforms, and we use the original annotation as the ground truth for the final augmented image. We only use the augmentation but not the Jensen-Shannon divergence consistency loss described in Hendrycks et al. [7]. **DeepAugment [6].** Instead of composing basic image augmentation operators to construct a strong data augmentation, DeepAugment transforms the input image by feeding it into an image-to-image translation network and randomly perturbing the intermediate feature representations. There are three options provided in the official implementation. We adopt the CAE [9] variant, which is initially used for the image compression task.

C Qualitative results

We visualize model prediction results in Figure 1 and Figure 2. For cross-domain semantic segmentation, the proposed method consistently improves over different settings. For cross-domain panoptic segmentation, we can see that the proposed method dramatically improves the background segmentation, such as trees.

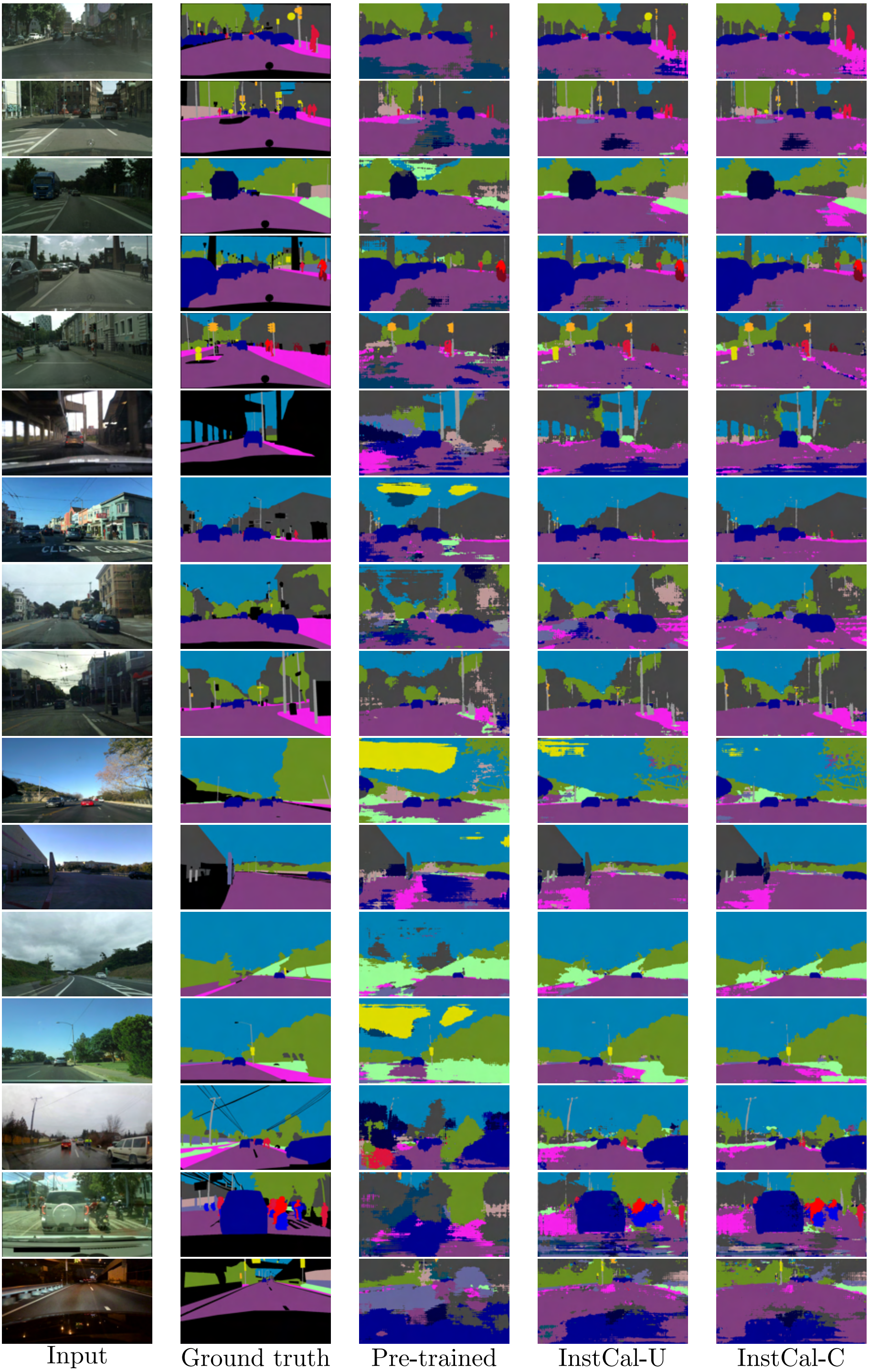

Fig. 1: **Qualitative results of semantic segmentation.** Models are trained on the GTA5 dataset. We treat the Cityscapes, BDD100k, Mapillary, and WildDash2 datasets as the unseen target domains.

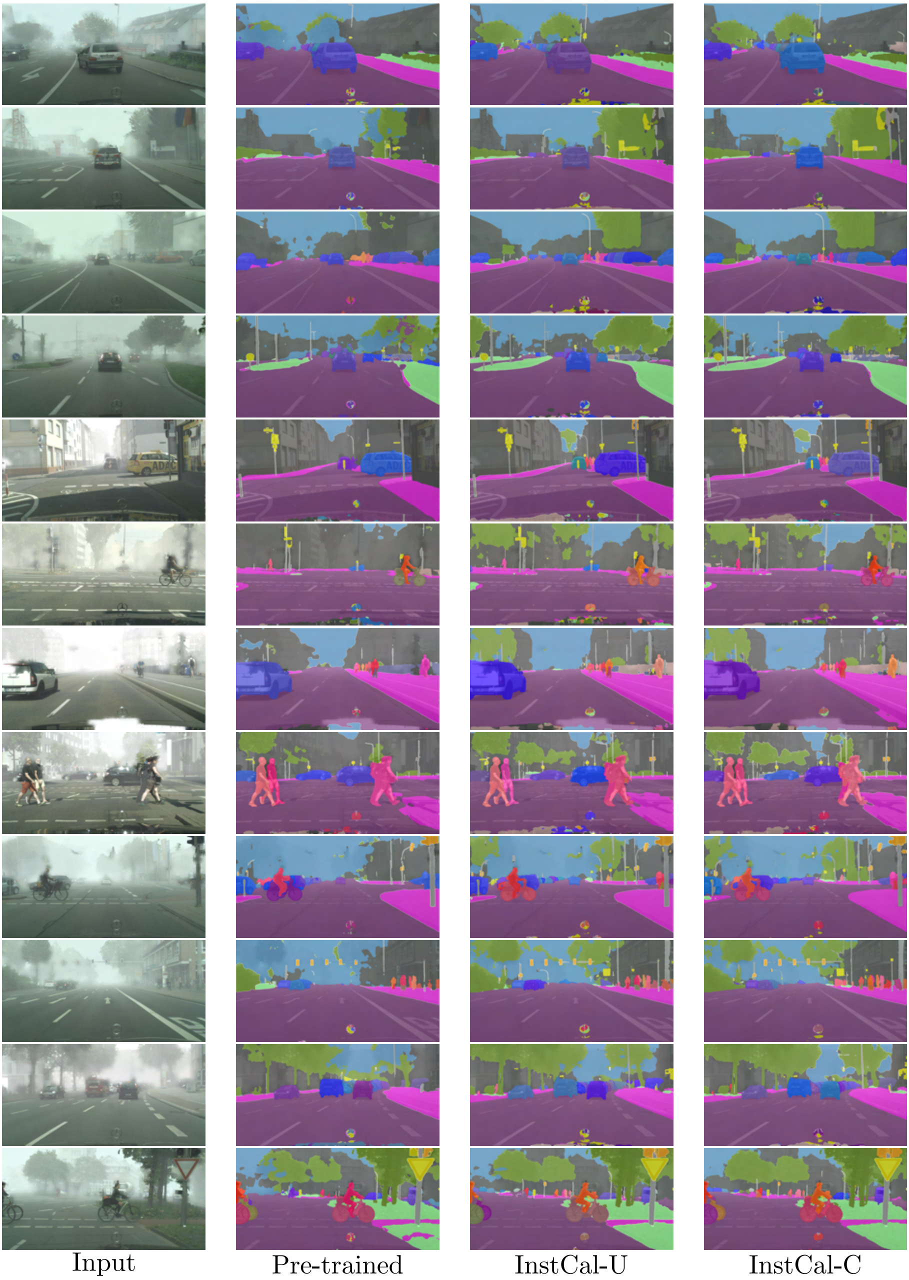

Fig. 2: Qualitative results of panoptic segmentation. Models are trained on the clean Cityscapes dataset and tested on the Foggy Cityscapes dataset.

References

1. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI (2017) 1
2. Chen, M., Xue, H., Cai, D.: Domain adaptation for semantic segmentation with maximum squares loss. In: ICCV (2019) 1
3. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: CVPR (2020) 1
4. Choi, S., Jung, S., Yun, H., Kim, J.T., Kim, S., Choo, J.: Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In: CVPR (2021) 1
5. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: CVPR Workshop (2020) 2
6. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: ICCV (2021) 2
7. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. In: ICLR (2020) 2
8. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015) 1
9. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. In: ICLR (2017) 2