# ADeLA: Automatic Dense Labeling with Attention for Viewpoint Shift in Semantic Segmentation (Supplementary Material)

## 1. Summary

This supplemental material provides more details on experiments and datasets described in the main paper. We also include additional experiments to analyze further and justify our method.

In Sec. 2, we provide more sample data of our collected datasets. In Sec. 3, we add more details of our training process. In Sec. 4, we show more experimental results, including standard deviation of our benchmarking results in Tab. 5 of the main paper, and qualitative results to demonstrate the ability of our method to generalize to real-world images.

## 2. Additional samples from our dataset

We show more samples from our dataset in Fig. 5.

## 3. Additional training details

Here we present more details about data augmentations used in our training for the view transformation network $\psi$, namely hue perturbation and color permutation.

### 3.1. Hue perturbation

The hue jittering factors are uniformly sampled from the interval $[-0.3, 0.3]$ in all experiments. Please refer to the first four columns in Fig. 2 for an illustration.

### 3.2. Color permutation

To apply color permutation, we first split the range of the 8-bit color values, i.e., $[0, 255]$ into $B$ intervals of equal length, and map each color value to the number of the interval that this value falls in, i.e., quantization (Fig. 1). We perform this quantization for each channel of the color images. To permute, we simply generate a random permutation of the set $\{1, 2, ..., B\}$, which represents a one-one mapping between the intervals. We then convert the images into a permuted one using the colors indexed by this random permutation. Examples of the color permutation are shown in Fig. 2 (last two columns).

In our experiment, we use $B = 8$ and apply color permutation only to $x_V$ and $\bar{x}_Q$. Ideally, we can set $B = 256$, but a smaller $B$ can help improve the computational efficiency. Hue perturbation can be considered as a global shift in the H component in the HSV space. It may still be possible for the network to learn the shift by observing $x_K$ and $x_V$. Compared to hue perturbation, color permutation further increases the randomness of the color change and helps the network predict appearance information solely from $x_V$.


Figure 1. An example of color quantization.

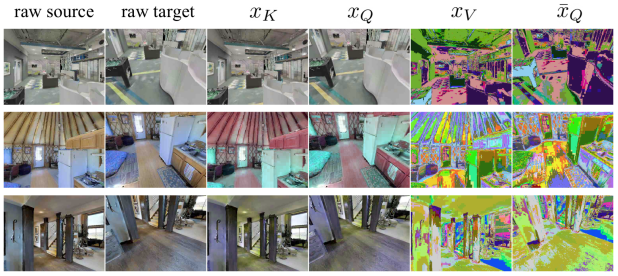


Figure 2. Data augmentation during training of the view transformation network. Left to right: original source and target color images (1st, 2nd columns); the hue perturbed key and query images ($x_K, x_Q$); the hue perturbed and color permuted value and ground-truth output images ($x_V, \bar{x}_Q$).

## 4. Additional experiments

We demonstrate more visual comparisons of our method with other top-performing baselines in Fig. 3.

### 4.1. Randomness in the training process

We report the randomness measured by the standard deviation of the mIoUs of the baselines in Tab. 1.

| Type | Method | Target Domains | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 10° | 20° | 30° | 40° | 50° | 60° | 70° | 80° | 90° |
| Baseline | Target Only | 0.0497 | 0.3552 | 0.0355 | 0.1502 | 0.2832 | 0.1090 | 0.1262 | 0.2495 | 0.1168 |
| | Source Only | 0.2828 | 0.3185 | 0.2712 | 0.0441 | 0.0490 | 0.0795 | 0.0717 | 0.0502 | 0.0386 |
| | UNet [6] | 0.2203 | 0.1504 | 0.1801 | 0.2470 | 0.6315 | 0.3635 | 0.3180 | 0.3233 | 0.1801 |
| Dense Corresp. Est. | RAFT [7] | 0.4650 | 0.2685 | 0.2757 | 0.9286 | 0.1513 | 0.1301 | 0.0755 | 0.0681 | 0.1358 |
| | MFNet [14] | 0.4257 | 0.2540 | 0.0917 | 0.2274 | 0.0458 | 0.1353 | 0.0458 | 0.0351 | 0.0839 |
| | DICL [9] | 0.1888 | 0.6264 | 0.9430 | 0.3081 | 0.0874 | 0.0709 | 0.0577 | 0.0252 | 0.0153 |
| UDA (unpaired) | ProDA [12] | 0.3995 | 0.2301 | 0.6920 | 0.4940 | 0.1732 | 0.1823 | 0.1418 | 0.0781 | 0.0874 |
| | CLAN [4] | 0.2108 | 0.2501 | 0.3502 | 0.2030 | 0.2307 | 0.1229 | 0.0436 | 0.0163 | 0.0100 |
| | CAG [13] | 0.6429 | 0.3958 | 0.4453 | 0.3989 | 0.4809 | 0.0635 | 0.0306 | 0.0265 | 0.0321 |
| | FDA [11] | 0.6191 | 0.4400 | 0.0404 | 0.1803 | 0.1007 | 0.0513 | 0.0721 | 0.1234 | 0.0950 |
| | PLCA [1] | 0.2641 | 0.3963 | 0.5151 | 0.2474 | 0.2934 | 0.2943 | 0.2318 | 0.2046 | 0.1062 |
| | LTIR [2] | 0.1930 | 0.3951 | 0.0351 | 0.0709 | 0.3863 | 0.0814 | 0.0872 | 0.0361 | 0.1015 |
| | CCM [3] | 0.2532 | 0.0257 | 0.2353 | 0.1670 | 0.1979 | 0.0152 | 0.0293 | 0.0534 | 0.0383 |
| | Advent [8] | 0.2306 | 0.2273 | 0.1141 | 0.0597 | 0.1452 | 0.0332 | 0.0300 | 0.0462 | 0.0348 |
| | Intrada [5] | 0.0304 | 0.1850 | 0.1008 | 0.0590 | 0.1267 | 0.0343 | 0.0332 | 0.0087 | 0.0375 |
| UDA (paired) | ProDA [12] | 3.8214 | 0.2651 | 0.2904 | 0.2078 | 0.2002 | 0.0716 | 0.0308 | 0.0515 | 0.3029 |
| | CLAN [4] | 1.6087 | 0.3407 | 0.2871 | 0.1721 | 0.1387 | 0.1637 | 0.1400 | 0.2108 | 0.2316 |
| | CAG [13] | 3.5663 | 2.1186 | 5.4703 | 1.5237 | 0.6600 | 0.1649 | 0.2204 | 0.1837 | 0.2730 |
| | FDA [11] | 1.0055 | 0.5012 | 1.1540 | 1.3336 | 0.6200 | 0.3637 | 0.3404 | 0.4518 | 0.4293 |
| | PLCA [1] | 0.5098 | 0.4753 | 0.3769 | 0.6816 | 0.3012 | 0.1489 | 0.1734 | 0.0917 | 0.0724 |
| Novel View Syn. | Appflow [17] | 0.3223 | 0.4770 | 0.7736 | 0.3592 | 0.1790 | 0.1375 | 0.1401 | 0.0611 | 0.1882 |
| | Synsin [10] | 0.3535 | 0.3703 | 0.1750 | 0.2450 | 0.2203 | 0.2515 | 0.1604 | 0.1323 | 0.0833 |
| Info. Trans. | ADeLA | 0.3857 | 0.2057 | 0.3065 | 0.3507 | 0.1385 | 0.0949 | 0.3299 | 0.0498 | 0.1888 |

Table 1. Standard deviations of different methods.

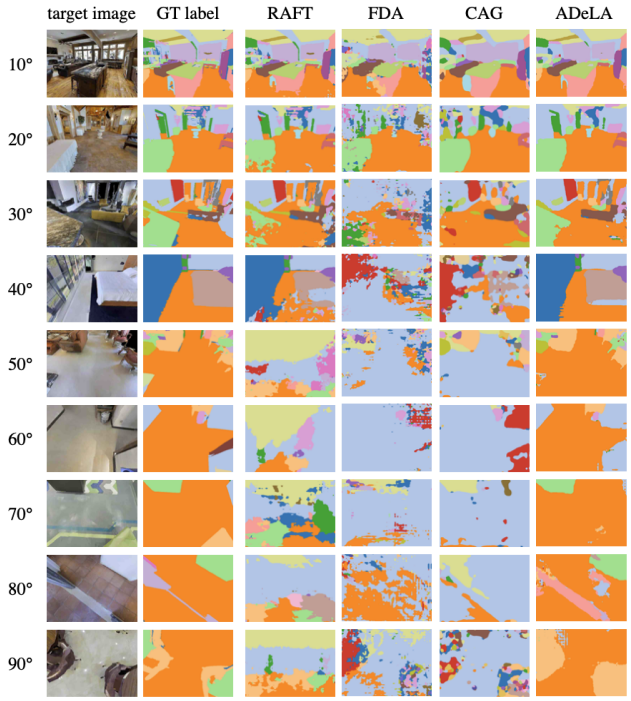


Figure 3. More qualitative results of our method compared with others on all target domains.

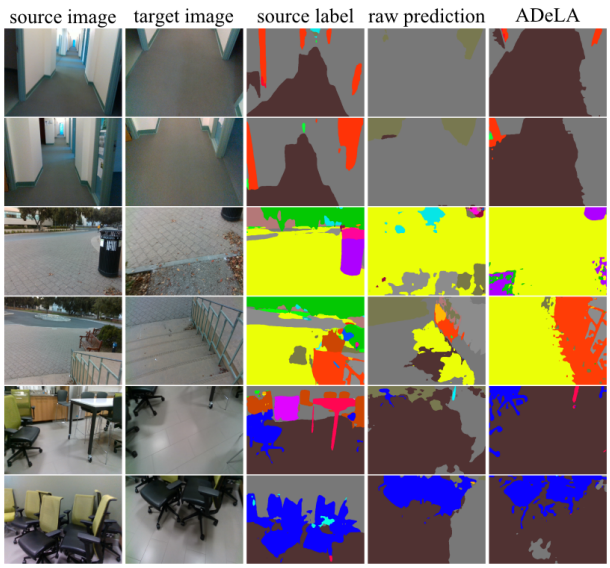


Figure 4. Qualitative results of our model (5th) on real-world images. The source domain pseudo labels (3rd) are acquired from an off-the-shelf semantic segmentation model, raw predictions (4th) are obtained by directly applying the off-the-shelf model to target domain images.

### 4.2. Effectiveness of constraining early predictions in training $\psi$

We train network $\psi$ with loss function shown in Eq. 7. The loss forces early outputs $x_Q^l$ to be similar to the final perturbed target view image $\bar{x}_Q$. We study the role of early

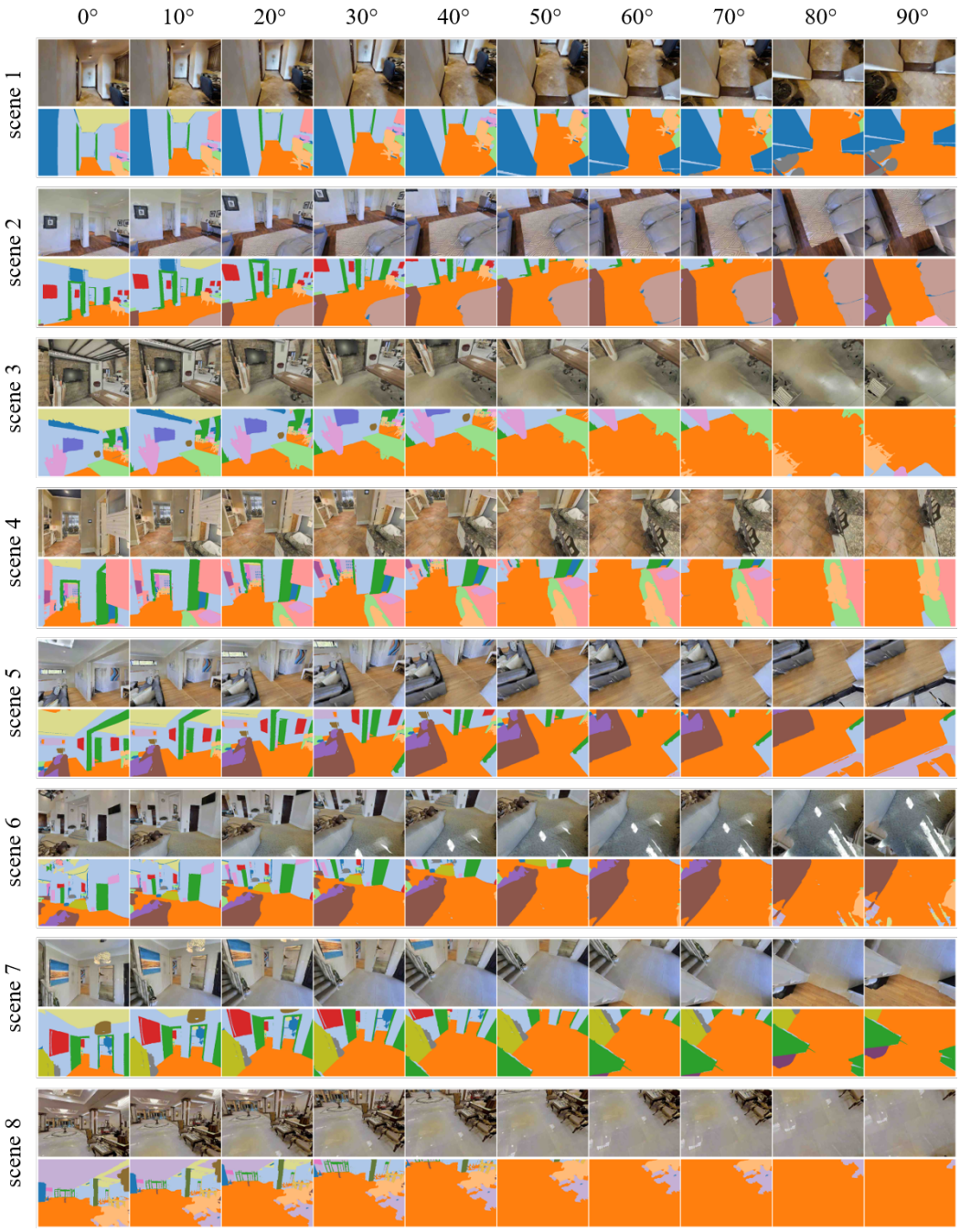


Figure 5. More samples from our dataset. For each scene, the 1st row shows color images, and the 2nd row shows the corresponding semantic segmentation.

supervision with 0° as the source domain and 30° as the target domain. As seen in Tab. 2, the constraint on the early outputs significantly improves the convergence rate of the training process and the accuracy of the trained model.

| | w/o early supervision | w/ early supervision |
|---|---|---|
| mIoU | 32.5 | 42.7 |
| converence rate | 35 epochs | 20 epochs |

Table 2. Effectiveness of early supervision with source domain 0° and target domain 30°.

### 4.3. Generalization on real-world data

To test how the trained network generalizes to real-world data, we collect some sample videos using a custom-made gantry shown in Fig 6.

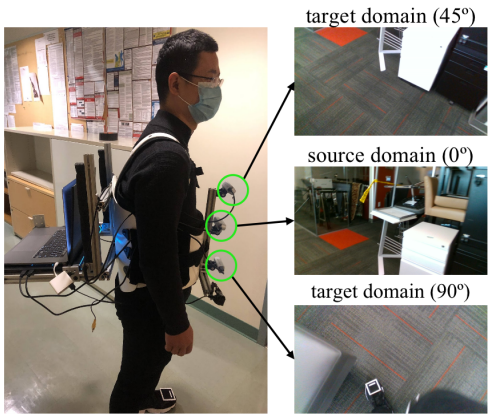


Figure 6. Data collection gantry. The platform has three different cameras pitching $0^\circ, 45^\circ, 90^\circ$ respectively.

Due to the lack of semantic annotations in the source domain (forward view) for our collected real data, we use an off-the-shelf semantic segmentation network (trained on the ADE-20k [15, 16][1] dataset) to provide the source domain labels. We then apply the view transformation network on these source domain labels to get the hallucinated labels on the target view.

The qualitative results are presented in Fig. 4. As observed in the figure, our model can correctly transfer labels of the floor and wall (1st and 2nd row), trashcan and handrail (3rd and 4th row), and chairs and tables (5th and 6th row) to their respective target views, whereas directly applying the pretrained model on the target domain images generates much noisy and even incorrect predictions. This demonstrates the ability of our model to generalize to the real world.

## References


[1] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander G. Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In *Advances in neural information processing systems*, 2020. 2

[2] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12975–12984, 2020. 2

[3] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *European Conference on Computer Vision*, pages 440–456. Springer, 2020. 2

[4] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019. 2

[5] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020. 2

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[7] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020. 2

[8] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 2

[9] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Kaihao Zhang, Pan Ji, and Hongdong Li. Displacement-invariant matching cost learning for accurate optical flow estimation. *arXiv preprint arXiv:2010.14851*, 2020. 2

[10] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 2

[11] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. 2

[12] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. *arXiv preprint arXiv:2101.10979*, 2, 2021. 2

[13] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *arXiv preprint arXiv:1910.13049*, 2019. 2

[14] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I Chang, Yan Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6278–6287, 2020. 2

[15] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4

[16] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic under-


[1] github.com/CSAILVision/semantic-segmentation-pytorch

standing of scenes through the ade20k dataset. *International Journal on Computer Vision*, 2018. 4

[17] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016. 2