A Appendix

A.1 Annotation Procedure and Data Division

The seven common land-cover types were developed according to the "Data Regulations and Collection Requirements for the General Survey of Geographical Conditions", i.e., buildings, road, water, forest, agriculture, and background classes. Based on the advanced *ArcGIS* geo-spatial software , all the images were annotated by professional remote sensing annotators. With the division of these images, a comprehensive annotation pipeline was adopted referring to [48]. The annotators labeled all objects belonging to six categories (except background) using polygon features. As for the 18 selected areas, it took approximately 24.6 h to finish the single-area annotations, resulting in a time cost of 442.8 man hours in total. After the first round of labeling, self-examination and cross-examination were conducted, correcting the false labels, missing objects, and inaccurate boundaries. The team supervisors then randomly sampled 600 images for quality inspection. The unqualified annotations were then refined by the annotators. Finally, several statistics (e.g. object numbers per image, object areas, etc.) were computed to double check the outliers. Based on DeepLabV3, preliminary experiments were conducted to ensure the validity of the annotations.

Table 8: The division of the LoveDA dataset

Domain	City	Region	#Images	Train	Val	Test
Urban	Nanjing	Qixia	320	✓		
		Gulou	320	✓		
		Qinhuai	336	✓		
		Yuhuatai	357		✓	
		Jianye	357			✓
	Changzhou	Jintan	320		✓	
		Wujin	320			✓
	Wuhan	Jianghan	180	✓		
		Wuchang	143			✓
Rural	Nanjing	Pukou	320	✓		
		Gaochun	336	✓		
		Lishui	336	✓		
		Liuhe	320		✓	
		Jiangning	336			✓
	Changzhou	Liyang	320			✓
		Xinbei	320			✓
	Wuhan	Jiangxia	374	✓		
		Huangpi	672		✓	
		Total	5987	2522	1669	1796

A.2 Top Performances Compared with Other Datasets

In order to support the "challeangability" of the proposed dataset compared to other land-cover datasets. By investigating the current researches, the top performances on different datasets have been reported in Table 9. The advanced method (HRNet) only achieved the lowest performance on the LoveDA dataset, showing the difficulty of this dataset

Table 9: Top performances compared with other datasets

Dataset	Top mIoU (%)
GID [46]	93.54
DeepGlobe [36]	52.24
ISPRS Potsdam [27]	82.38
ISPRS Vaihingen [27]	79.76
LoveDA	49.79

A.3 Instance Differences Between Urban and Rural Areas

For the LoveDA dataset, the differences between urban and rural areas at the instance level are shown in the Figure 7. Similar with the pixel analysis in §3.3, the instances across domains are imbalanced. Specifically, the urban areas have more buildings and fewer instances of agricultural land. The rural areas have more instances of agricultural land. This also highlights the inconsistent class distribution problem between different domains.

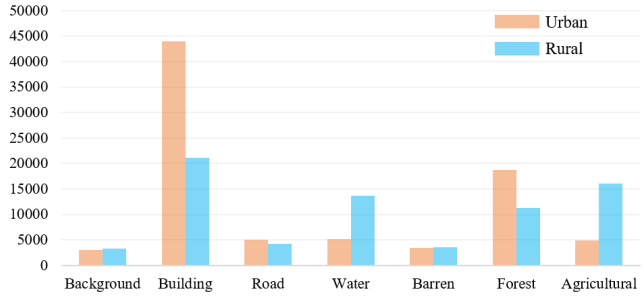


Figure 7: Instance differences between urban and rural areas.

A.4 Implementation Details

All the networks were implemented under the PyTorch framework, using an NVIDIA 24 GB RTX TITAN GPU. The backbones used in all the networks were pre-trained on ImageNet. The number of training iterations was set to $10k$ with a batch size of 16. The eight source images and eight target images were alternately input. The other settings were the same as in the semantic segmentation. As for self-training (ST), the pseudo-generation hyper-parameters remained the same as in the original literature. The classification learning rate was set to 10^{-2}. All the ST-based networks were trained for $10k$ steps including two stages: 1) for the first $4k$ steps, the models were trained only on the source images for initialization; and 2) the pseudo-labels were then updated every $1k$ steps during the remaining training process. Considering the training stability, IAST method was set $8k$ steps for initialization in the **Urban** $\rightarrow$ Rural experiments.

All the networks were then re-implemented following the original literature. The segmentation models followed the default settings in [39], including a modified ResNet50 and atrous spatial pyramid pooling (ASPP)[4]. By using dilated convolutions, the stride of the last two convolution layers was modified from 2 to 1. The final output stride of the feature map was 16.

Following [39], the discriminator was made up of five convolutional layers with a kernel of 4×4 and a stride of 2, where the channel numbers were $\{64, 128, 256, 512, 1\}$, respectively. Each convolution was followed with a Leaky ReLU, and the parameter was set to 0.2. Bilinear interpolation was used for re-scaling the output to the size of the input.

As for the hyperparameter settings, the adversarial scale factor λ was set to 0.001 following [22, 44]. With respect to the two segmentation outputs in [39], λ_1 and λ_2 were set to 0.001 and 0.002, respectively. The weight discrepancy loss was used in CLAN[22], and the default settings were adopted, i.e., $\lambda_w = 0.01$, $\lambda_{local} = 10$, and $\epsilon = 0.4$. FADA [44] adopts the temperature T to encourage a soft probability distribution over the classes, which was set to 1.8 by default. The confidence of pseudo-label θ in PyCDA[18] was set to 0.5 by default. The pseudo-label related hyperparameters for IAST remained the same as in [25]. The target proportion p in CBST was set to 0.1 and 0.5 when transferring to the rural and urban domains, respectively.

A.5 Error Bar Visualization for the UDA Experiments

In order to make the results more convincing and reproducible, we ran all UDA methods five times using a random seed. The error bar visualization for the UDA experiments is shown in Figure 8. The adversarial training methods achieve smaller error fluctuations than the self-training methods. This is because the self-training methods assign and update the pseudo-labels alternately, which brings greater randomness. Hence, for the self-training methods, we suggest that three times more repeats are preferred to provide more convincing results.

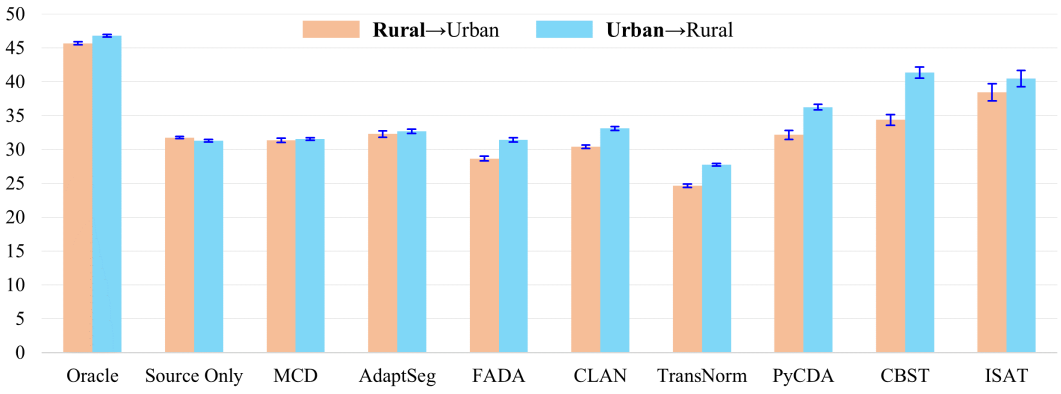
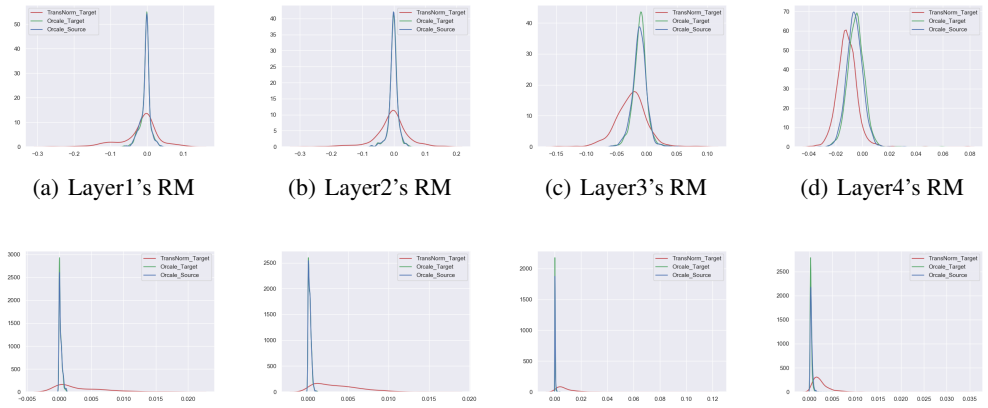


Figure 8: Error bar visualization for the UDA experiments.

A.6 Batch Normalization Statistics in the Different Domains

The batch normalization (BN) statistics are shown in Figure 9. We observe that in the *Oracle* source and target settings, the model has similar BN statistics in both mean and variance. This demonstrates that the gap between the source and target domains does not lie in the BNs, which is different from the conclusion in [47]. Hence, the modification of the BN statistics may have a negative effect, as in TransNorm[47], where the target BN statistics are far different from those of the *Oracle* target model. This observation is consistent with the results listed in Table 6. We speculate that the cause of this failure in the combined simulation dataset UDA experiments[22, 44, 47] is that the source and target domains have large spectral differences, and thus require domain-specific BN statistics. However, the LoveDA dataset is real data obtained from the same sensor at the same time. The spectral difference in the source and target domains is very small (Figure 3(b)), so the BN statistics are very similar (Figure 9).



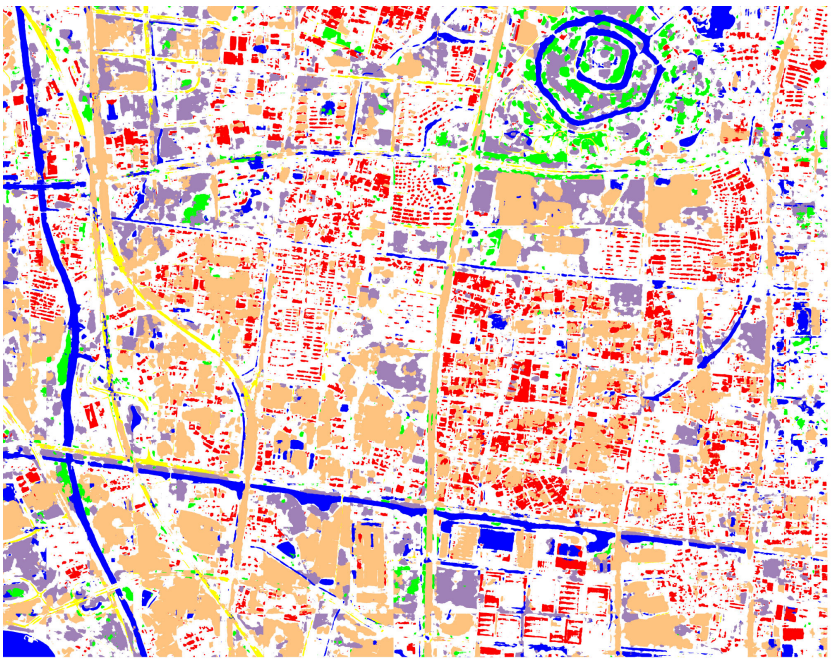
(a) Layer1's RM (b) Layer2's RM (c) Layer3's RM (d) Layer4's RM

(e) Layer1's RV (f) Layer2's RV (g) Layer3's RV (h) Layer4's RV

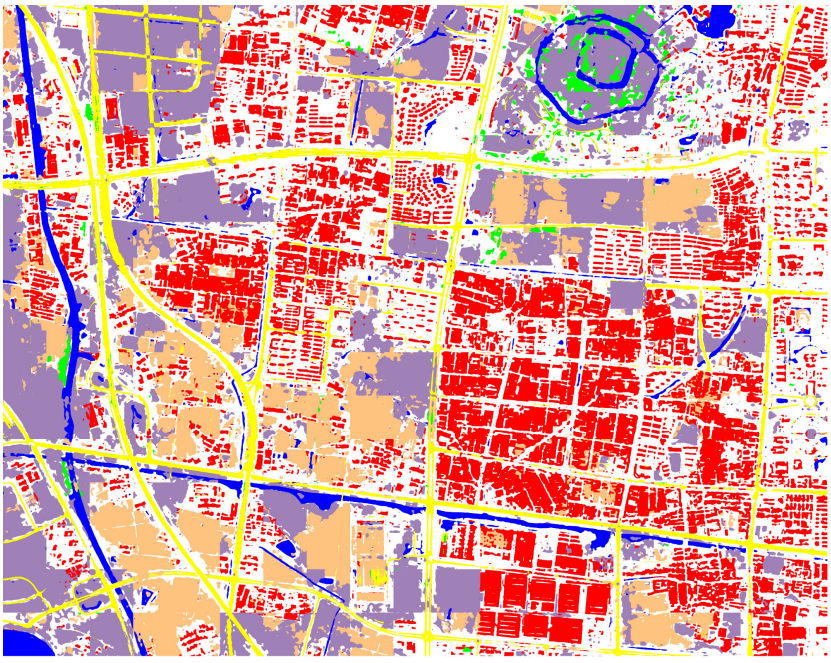
Figure 9: Statistics of the running mean (RM) and running var (RV) of the batch normalization in the different layers of ResNet50. Two *Oracle* models and TransNorm in the **Urban** → Rural experiments are shown.

A.7 Large-scale Visualizations on UDA Test Set

The large-scale visualizations are shown in the Figure 10. Compared with the baseline, CBST can produce better results on large-scale mapping, which highlights the importance of developing UDA methods. However, CBST still has a lot of room for improvement. More tailored UDA algorithms requires to be developed on the LoveDA dataset.



(a) Baseline on Wujin area



(b) CBST on Wujin area

Figure 10: Large-scale visualizations on UDA `Test` set (**Rural** → Urban).