A Appendix

A.1 Additional Guides

1. Submission introducing new datasets must include the following in the supplementary materials:
   (a) **Dataset documentation and intended uses. Recommended documentation frameworks include datasheets for datasets, dataset nutrition labels, data statements for NLP, and accountability frameworks.** The datasheet for LoveDA dataset is provided in the supplementary material.
   (b) **URL to website/platform where the dataset/benchmark can be viewed and downloaded by the reviewers.** The code and dataset were shared at: Google Drive
   (c) **Author statement that they bear all responsibility in case of violation of rights, etc., and confirmation of the data license.** The authors state that they bear all responsibility in case of violation of rights, and confirmation of the data license.
   (d) **Hosting, licensing, and maintenance plan. The choice of hosting platform is yours, as long as you ensure access to the data (possibly through a curated interface) and will provide the necessary maintenance.** The hosting plan follows the provided datasheet in the supplemental material. We will publish the LoveDA dataset on Codalab.
2. To ensure accessibility, the supplementary materials for datasets must include the following:
   (a) **Links to access the dataset and its metadata. This can be hidden upon submission if the dataset is not yet publicly available but must be added in the camera-ready version. In select cases, e.g when the data can only be released at a later date, this can be added afterward. Simulation environments should link to (open source) code repositories.** The code and dataset were shared at: Google Drive
   (b) **The dataset itself should ideally use an open and widely used data format. Provide a detailed explanation on how the dataset can be read. For simulation environments, use existing frameworks or explain how they can be used.** Each instance in the dataset contains an image and corresponding semantic mask that are 1024 by 1024 pixels in PNG format.
   (c) **Long-term preservation: It must be clear that the dataset will be available for a long time, either by uploading to a data repository or by explaining how the authors themselves will ensure this.** We will publish the LoveDA dataset on Codalab. All questions and comments can be sent to Junjue Wang: kingdrone@whu.edu.cn. All changes to the dataset will be announced through the LoveDA mailing list.
   (d) **Explicit license: Authors must choose a license, ideally a CC license for datasets, or an open source license for code (e.g. RL environments).** The LoveDA dataset will be released under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license (CC BY-NC-SA 4.0).
   (e) **Add structured metadata to a dataset's meta-data page using Web standards (like schema.org and DCAT): This allows it to be discovered and organized by anyone. If you use an existing data repository, this is often done automatically.** The dataset is provided with the guideline of data division.
   (f) **Highly recommended: a persistent dereferenceable identifier (e.g. a DOI minted by a data repository or a prefix on identifiers.org) for datasets, or a code repository (e.g. GitHub, GitLab,...) for code. If this is not possible or useful, please explain why.** The persistent dereferenceable identifier and code repository will be added after the dataset is open source. The dataset will be submitted at IEEE DataPort and the code will be released at GitHub.
3. **For benchmarks, the supplementary materials must ensure that all results are easily reproducible. Where possible, use a reproducibility framework such as the ML reproducibility checklist, or otherwise guarantee that all results can be easily reproduced, i.e. all necessary datasets, code, and evaluation procedures must be accessible and documented.** The code, dataset, pre-trained model parameters, and executable scripts have been provided to ensure reproducibility.

4. **For papers introducing best practices in creating or curating datasets and benchmarks, the above supplementary materials are not required.**

A.2 Dataset Annotation Procedure

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 [42]. The annotators labeled all objects belonging to six categories (except background) using polygon features. As for the 10 selected areas, it took approximately 24.6 h to finish the single-area annotations, resulting in a time cost of 246 man hours in total. After the first round of labeling, self-examination and cross-examination was 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.

A.3 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 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.

All the networks were then re-implemented following the original literature. The segmentation models followed the default settings in [35], 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 [35], 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 [21, 38]. With respect to the two segmentation outputs in [35], λ_1 and λ_2 were set to 0.001 and 0.002, respectively. The weight discrepancy loss was used in CLAN[21], and the default settings were adopted, i.e., $\lambda_w = 0.01$, $\lambda_{local} = 10$, and $\epsilon = 0.4$. FADA [38] 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[16] was set to 0.5 by default and the parameters in IAST remained the same as in [23]. The target proportion p in CBST was set to 0.3 and 0.5 when transferring to the rural and urban domains, respectively.

A.4 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 7. 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.

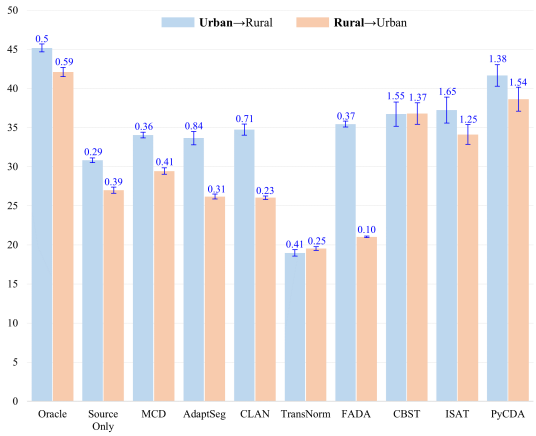
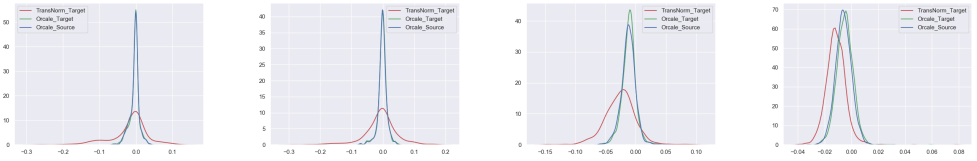


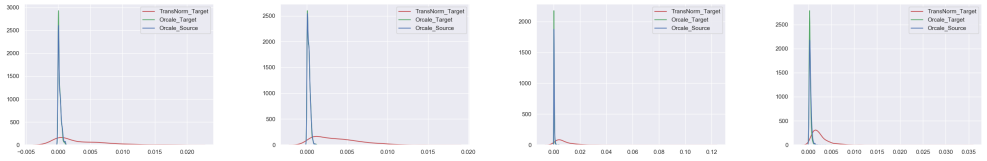
Figure 7: Error bar visualization for the UDA experiments.

A.5 Batch Normalization Statistics in the Different Domains

The batch normalization (BN) statistics are shown in Figure 8. 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 [41]. Hence, the modification of the BN statistics may have a negative effect, as in TransNorm[41], 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 4.2. We speculate that the cause of this failure in the combined simulation dataset UDA experiments[21, 38, 41] 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 8).



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



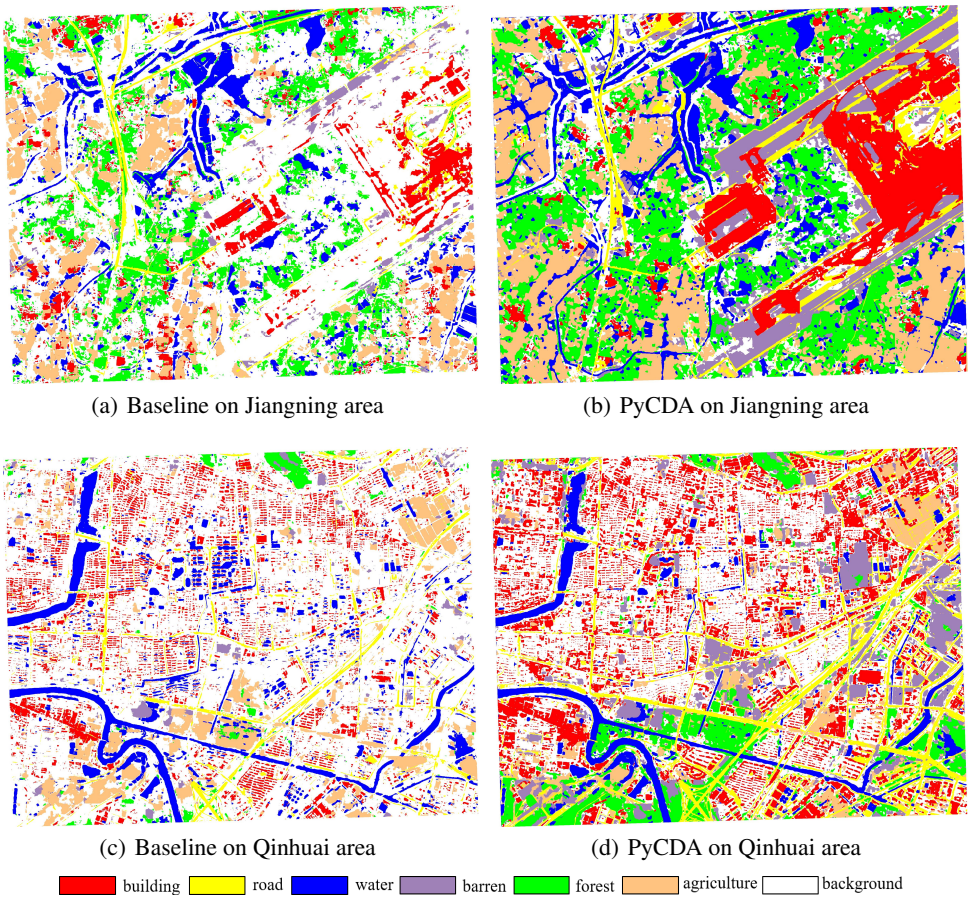
(e) Layer1's running var (f) Layer2's running var (g) Layer3's running var (h) Layer4's running var

Figure 8: Statistics of the running mean and running var of the batch normalization in the different layers of ResNet50. Two *Oracle* models and TransNorm in the **Urban** → Rural experiments are shown.

A.6 Large-scale Visualizations on UDA Test Set

The large-scale visualizations are shown in the Figure 9. Compared with the baseline, PyCDA can produce better results on large-scale mapping, which highlights the importance of developing UDA

methods. However, PyCDA still has a lot of room for improvement. More tailored UDA algorithms requires to be developed on the LoveDA dataset.



(a) Baseline on Jiangning area (b) PyCDA on Jiangning area

(c) Baseline on Qinhuai area (d) PyCDA on Qinhuai area

Figure 9: Large-scale Visualizations on UDA Test Set.

A.7 Broader Impact

This work offers a free and open dataset with the purpose of advancing land-cover semantic segmentation in the area of remote sensing. We also provide two benchmarked tasks with three considerable challenges. This will allow other researchers to easily build off of this work and create new and enhanced capabilities. The authors do not foresee any negative societal impacts of this work. A potential positive societal impact may arise from the development of generalizable models that can produce large-scale high-spatial-resolution land-cover mapping accurately. This could help to reduce the manpower and material resource consumption of surveying and mapping.