Supplement for Aerial Lifting

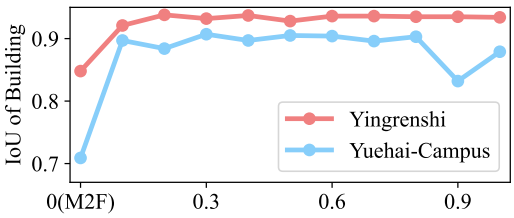


Figure I. Effect of altitude offset in semantic fusion strategy.

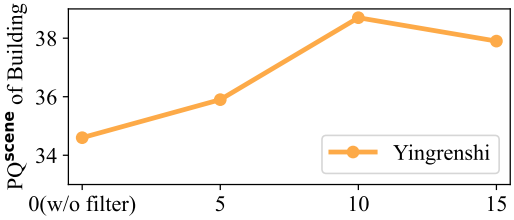


Figure II. Effect of geometry-guided instance filtering threshold.

A. More Details for the Method

Implementation Details We implemented our method with PyTorch [62] and used the Adam optimizer [38] with a learning rate of 0.001 for the hash-grid and 0.01 for the semantic and instance MLPs. We combine the tri-plane features [99] and a cuboid hash-grid proposed by StreetSurf [28], as a backbone for geometry reconstruction. The hash-grid was trained with a hash level of $L = 16$, the highest resolution of $R = 8192$, and a hash table size of $T = 2^{22}$. The architectures of the semantic and instance networks are identical, each consists of a 5-layer MLP with 128 channels. Moreover, for a scene represented by a volume of [0, 1], we raise the altitude of all cameras by displacing each camera in the opposite direction of the camera's focal point with an offset of 0.3, during the scale-adaptive semantic label fusion. The geometry-guided instance filtering threshold is empirically set to 10 meters (in physical space) for all testing scenes.

A.1. Cross-view Instance Label Grouping

Figure III illustrates cases of Cross-view Instance Label Grouping on the *Longhua-1* and *Yingrenshi* datasets. For example, in the SAM instance label, mask blocks A, B, and C belong to the same building but are segmented as three distinct instances, introducing ambiguity in the supervision label during training. This can result in two pixels from the same building being incorrectly labeled as different instances. In contrast, with our cross-view instance label grouping, separated blocks of the same instance are merged into the same group (e.g., group1: {A, B, C}; group2: {D, E}), guiding the training of the instance field more effectively. Specifically, during training, we randomly select a single instance from each group (e.g., blocks A and D in the SAM label) to reduce the occurrence of conflicts. The pseudo-code for the proposed cross-view instance label grouping is shown in Algorithm 1.

A.2. Mask2Former Semantic Label Mapping

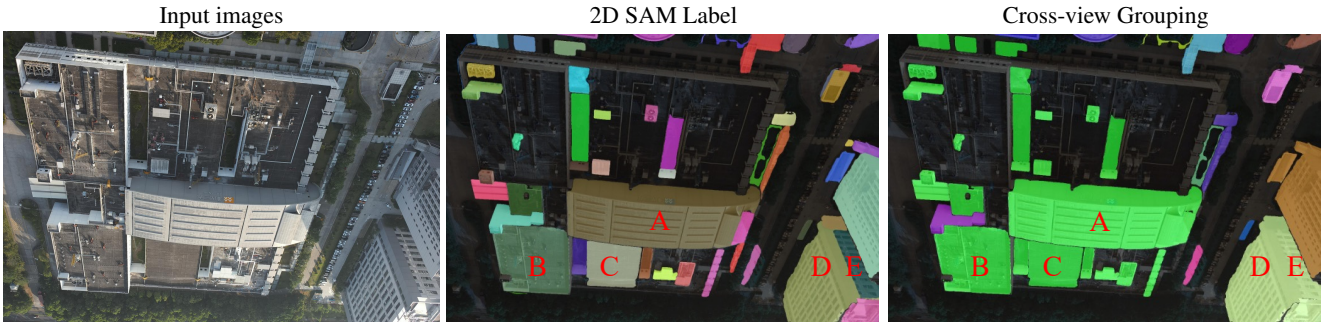
Following Panoptic-Lift [70], we employ the universal 2D segmentation method, Mask2Former [13], to obtain semantic labels and utilize the implementation[1] with test-time augmentation. The original Mask2former provides pretrained models on various datasets, including COCO [48], Cityscapes [17], ADE20K [102], *et al*. We observed that the model trained on the ADE20K dataset (swin_large_IN21k model) demonstrates robust performance for semantic segmentation of aerial images. For training, we map the ADE20K classes (150 classes in total) into four categories: Building, road, car, and tree. Additionally, we marked the category from the 150 classes that may not appear in aerial images as *Cluster* (*e.g.*, indoor objects), mitigating interference from inaccurate segmentation results of Mask2former.

Moreover, during the processing of the scale-adaptive semantic label fusion, the images with the original size are cropped into four parts to obtain segmentation results, as feeding the entire image may lead to out-of-memory errors. Then, the car and tree segmentation results from the downscaled images of Mask2Former are substituted.

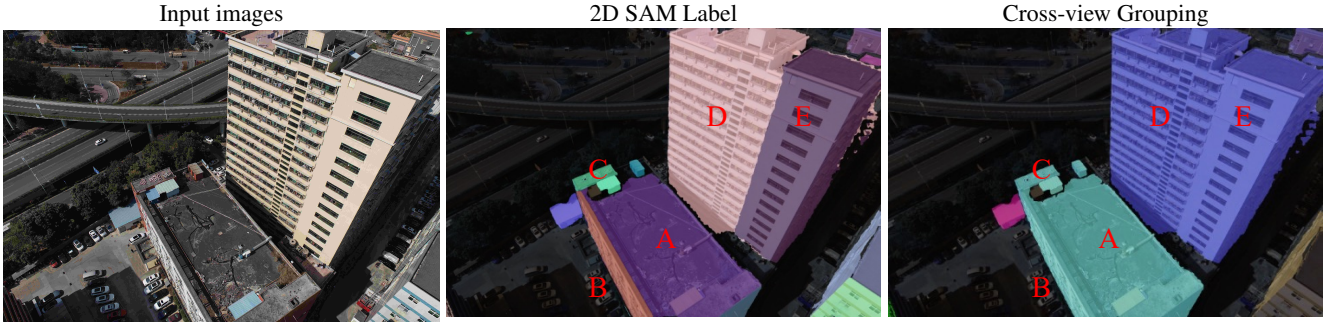
A.3. Effect of Hyper-parameter

Setting of far view in semantic fusion. For a scene represented by a volume of [0, 1], we raise the altitude of all

[1] An implementation of Mask2former with test-time augmentation: https://github.com/nihalsid/Mask2Former.



(a) *Longhua-1*



(b) *Yingrenshi*

Figure III. Illustration of Cross-view Instance Label Grouping. Different colors represent different instances. SAM [39] produces over-segmented masks and an instance might be segmented into different blocks (*e.g.*, A, B, and C belong to the same building but are incorrectly divided into different instances). Our cross-view instance label grouping alleviates this problem, reducing the conflict of 2D instance supervision during training.

Algorithm 1 Pseudo code for the cross-view instance label grouping strategy.

Require: N images with SAM masks $\mathbf{H}_i$ for each i-th view
Ensure: Cross-view guidance map U_i for each view

for $i = 1$ to N **do**
  Project SAM masks from all other views onto the i-th view: $\{\mathbf{H}_{j\to i} | j = 1, \ldots, N, j \neq i\}$
  for each instance mask $\mathbf{H}_i^k$ in the i-th view **do**
    for each instance mask $\mathbf{H}_{j\to i}^l$ in projected mask $\mathbf{H}_{j\to i}$ **do**
      if $\frac{|\mathbf{H}_i^k \cap \mathbf{H}_{j\to i}^l|}{\min(|\mathbf{H}_i^k|, |\mathbf{H}_{j\to i}^l|)} > \tau$ **then**
        expanded mask: $\mathbf{H}_{i\cup j}^k$.append($\mathbf{H}_{j\to i}^l$)
      end if
    end for
    Combine all $\mathbf{H}_{i\cup j}^k$ to form cross-view mask U_i^k: $U_i^k = \bigcup_{j\neq i} \mathbf{H}_{i\cup j}^k$
  end for
  Organize cross-view masks in ascending order based on area
  Sequentially layer cross-view masks onto map $H \times W$ to form U_i
end for

cameras by displacing each camera in the opposite direction of the camera's focal point with an offset of 0.3. Figure I shows that the scale-adaptive semantic fusion consistently enhances the results of Mask2Former and remains effective across various offset values. This strategy is inspired by the observation that large object recognition benefits from a larger receptive field, leading to more reliable segmentation.

Geometry-guided instance filtering. The geometry-guided instance filtering threshold is empirically set to 10 meters (in physical space) for all testing scenes. Figure II shows that applying the filtering can improve results against w/o filter, and the filtering works effectively in the range of

[5, 15] meters.

B. Details for the Instance Field Optimization

Our building instance segmentation method is built upon the 2D image segmentation method. In selecting the base model for 2D image segmentation, we considered SAM [39] and Detectron [87]. During our testing, we found that Detectron did not perform well on aerial images for building segmentation. Consequently, we experimented with Detectron2-SpaceNet [43], which is fine-tuned on the SpaceNet dataset [81] and based on the Mask-RCNN model from Detectron [87]. While its segmentation performance showed improvement, it did not generalize well to diverse urban scenes (refer to Figure 3 of the main paper). Therefore, we decided to build our model upon the SAM model.

As stated in the paper, we utilize two methods to achieve the 3D building instance segmentation: *linear assignment* from Panoptic-Lift [70] and *contrastive learning* from Contrastive-Lift [4].

Linear Assignment[2] 2D machine-generated instance labels are noisy and view-inconsistent, for which Panoptic-Lift [70] proposes to map them into 3D surrogate identifiers, and finds out the most compatible injective mapping by solving a linear assignment problem. Let $U(\boldsymbol{r})$ denotes the instance segment label of the pixel casting ray $\boldsymbol{r}$, $\boldsymbol{R}_k$ the subset of rays in $\boldsymbol{R}$ that belong to 2D instance $k \in K_{\mathbf{I}}$, and $K_R \subseteq K_{\mathbf{I}}$ the subset of 2D instances that are represented in the batch of rays $\boldsymbol{R}$, the optimal injective mapping is then given by:

$$\Pi_{\boldsymbol{R}}^{\star} = \underset{\Pi_{\boldsymbol{R}}}{\operatorname{argmax}} \sum_{k \in K_R} \sum_{\boldsymbol{r} \in \boldsymbol{R}_k} \frac{\tilde{S}(\boldsymbol{r})_{(\Pi_{\mathbf{I}}(U(\boldsymbol{r})))}}{|\boldsymbol{R}_k|} \tag{7}$$

where $\tilde{S}(\boldsymbol{r})_{(\Pi_{\mathbf{I}}(U(\boldsymbol{r})))}$ denotes the $\Pi_{\mathbf{I}}(U(\boldsymbol{r}))$-th element of the instance label vector $\tilde{S}(\boldsymbol{r})$.

Thus the instance loss can be formulated as follows:

$$\mathcal{L}_{\text{instance}} = -\frac{1}{|\boldsymbol{R}|} \sum_{\boldsymbol{r} \in \boldsymbol{R}} w_r \log \tilde{S}(\boldsymbol{r})_{(\Pi_{\boldsymbol{R}}^{\star}(U(\boldsymbol{r})))} \tag{8}$$

where w_r is the prediction confidence.

Contrastive Learning[3]

Instead of aligning labels extracted from multiple views, Contrastive-Lift [4] directly learns embeddings from the noisy 2D machine-generated labels via optimizing a contrastive loss and acquires the instance segments by simply clustering the embeddings. The instance loss can be formulated as follows:

$$\mathcal{L}_{\text{instance}} = \mathcal{L}_{\text{sf}} + \mathcal{L}_{\text{conc}} \tag{9}$$

where the first item $\mathcal{L}_{\text{sf}}$ is the contrastive loss using a slow-fast learning scheme, and the second item $\mathcal{L}_{\text{conc}}$ is the concentration loss used to encourage the embeddings to form concentrated clusters for each object.

Specifically, given the two non-overlapping subsets $\boldsymbol{R}_1$and $\boldsymbol{R}_2$ partitioned from rays in $\boldsymbol{R}$, the contrastive loss function is:

$$\mathcal{L}_{\text{sf}} = -\frac{1}{|\boldsymbol{R}_1|} \cdot \sum_{\boldsymbol{r} \in \boldsymbol{R}_1} \log \frac{\sum_{\boldsymbol{r}' \in \boldsymbol{R}_2} \mathbf{1}_{U(\boldsymbol{r})=U(\boldsymbol{r}')} \exp\left(\operatorname{sim}\left(\tilde{S}(\boldsymbol{r}), S(\boldsymbol{r}'); \gamma\right)\right)}{\sum_{\boldsymbol{r}' \in \boldsymbol{R}_2} \exp\left(\operatorname{sim}\left(\tilde{S}(\boldsymbol{r}), S(\boldsymbol{r}'); \gamma\right)\right)} \tag{10}$$

where $\mathbf{1}$ is the indication function, $\operatorname{sim}(x, x'; \gamma) = \exp\left(-\gamma \left\|x - x'\right\|^2\right)$is used to compute the similarity between embeddings in Euclidean space, and $S(\boldsymbol{r}')$ is the instance label inferred by the slowly-updated embedding field [4].

And the concentration loss function is:

$$\mathcal{L}_{\text{conc}} = \frac{1}{|\boldsymbol{R}_1|} \sum_{\boldsymbol{r} \in \boldsymbol{R}_1} \left\| \tilde{S}(\boldsymbol{r}) - \frac{\sum_{\boldsymbol{r}' \in \boldsymbol{R}_2} \mathbf{1}_{U(\boldsymbol{r})=U(\boldsymbol{r}')} S(\boldsymbol{r}')}{\sum_{\boldsymbol{r}' \in \boldsymbol{R}_2} \mathbf{1}_{U(\boldsymbol{r})=U(\boldsymbol{r}')}} \right\|^2 \tag{11}$$

For clustering, we use HDBSCAN [55] clustering in experiments following Contrastive-Lift [4]. We sample all rays in the testing images and then randomly select 200000 pixels of building for clustering, with the minimum cluster size set to 1000.

C. More Details for the Dataset

C.1. Dataset Selection

We evaluate our method on UrbanBIS dataset [92], which provides 3D building-level instance annotations and 3D semantic segmentation annotations of six categories, including buildings, roads, cars, and trees. We select four regions with a high density of building instances and various architecture styles, namely *Yingrenshi, Yuehai-Campus, Longhua-1, and Longhua-2*. Figure IV shows the bird's eye view of the four mentioned areas, which are covered by a diverse range of architectural instances.

C.2. Ground-truth Label for Evaluation

2D ground-truth label for each view is acquired by projecting the 3D point cloud annotations. Specifically, UrbanBIS dataset [92] provides 3D point cloud annotations for semantic and instance segmentation, along with 2D aerial images. However, individual 2D annotations for semantic and instance segmentation are not provided for each image, and camera poses for projections onto 2D images are also not

[2] We utilize the official implementation from Panoptic-Lift: https://github.com/nihalsid/panoptic-lifting.

[3] We utilize the official implementation from Contrastive-Lift: https://github.com/yashbhalgat/Contrastive-Lift.

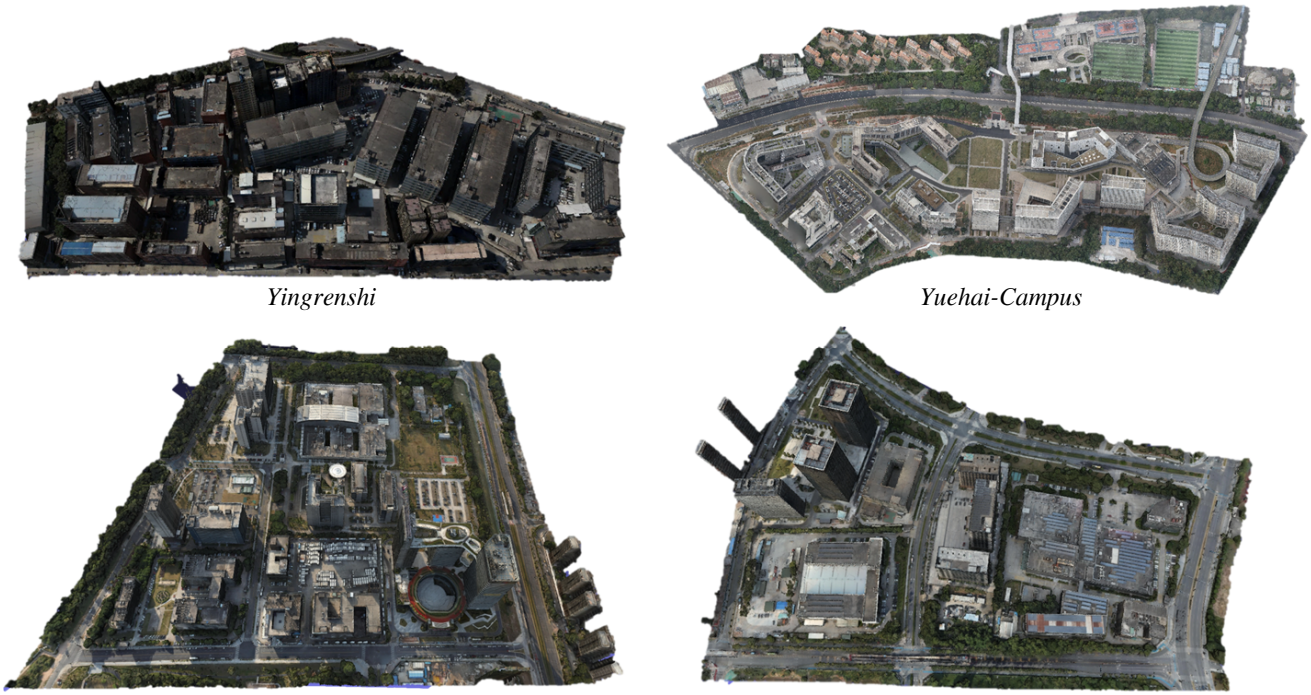


Figure IV. Bird's eye view of the *Yingrenshi*, *Yuehai-Campus*, *Longhua-1* and *Longhua-2* areas in UrbanBIS dataset [47].

available. Moreover, the point cloud given by UrbanBIS is sparse, resulting in unsatisfactory projections on 2D images.

To address these limitations, we reconstruct a dense point cloud and corresponding camera poses from the 2D aerial images. Subsequently, we register the reconstructed point cloud with the annotated UrbanBIS point cloud using CloudCompare and annotate our reconstructed points by employing KD-tree [3] algorithm to find out the labels of the nearest annotated points in the UrbanBIS point cloud from ours. This process allows us to obtain annotations for the dense point cloud with known camera poses, which are then projected onto 2D images, yielding 2D image annotations.

It is important to note that we made modifications to the original annotations for two reasons. Firstly, the ground-truth annotations are not sufficiently accurate, mainly regarding missing annotations of cars. Secondly, for the Yuehai-Campus area, UrbanBIS has not provided corresponding 2D aerial images so far. We then utilized images from the UrbanScene dataset [47], which covers the Yuehai-Campus region but with a significant time gap compared to the UrbanBIS dataset [92]. Consequently, there are substantial discrepancies in the distribution of cars and trees between the UrbanBIS point cloud and our point cloud reconstructed from UrbanScene dataset [47]. As modifying annotations on the point cloud would be time-consuming, we use the labeling tool ISAT [95] to fix the 2D testing image annotations.

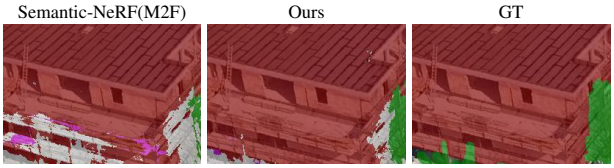


Figure V. Visual results on UAVid dataset.

Table I. Comparison on UAVid dataset.

Method	Sequence #14		Sequence #31	
	mIoU	Building	mIoU	Building
Mask2former	64.9	74.9	57.8	73.3
Semantic-NeRF (M2F)	69.7	91.5	58.0	87.5
Ours	**74.1**	**92.8**	**61.9**	**88.8**

D. More Results

D.1. Semantic Segmentation on UAVid Dataset

To further evaluate the effectiveness of our method, we conduct experiments on the UAVid dataset, which contains 2D semantic labels for sparse frames. We conducted additional evaluations of semantic segmentation by choosing two video sequences for which the camera trajectory has a wide coverage area and can be reconstructed using COLMAP. The results presented in Table I and Figure V

Method	Yingrenshi	Yuehai-Campus	Longhua-1	Longhua-2
RandLA-Net [34]	42.7	39.4	50.2	54.7
Ours	**62.9**	**55.7**	**66.5**	**66.0**

Table II. Comparison with the point-based method. The reported values are 3D mIoU.

highlight that our method outperforms the baseline methods, demonstrating the effectiveness of our semantic fusion strategy.

D.2. Comparison with Point-based Method

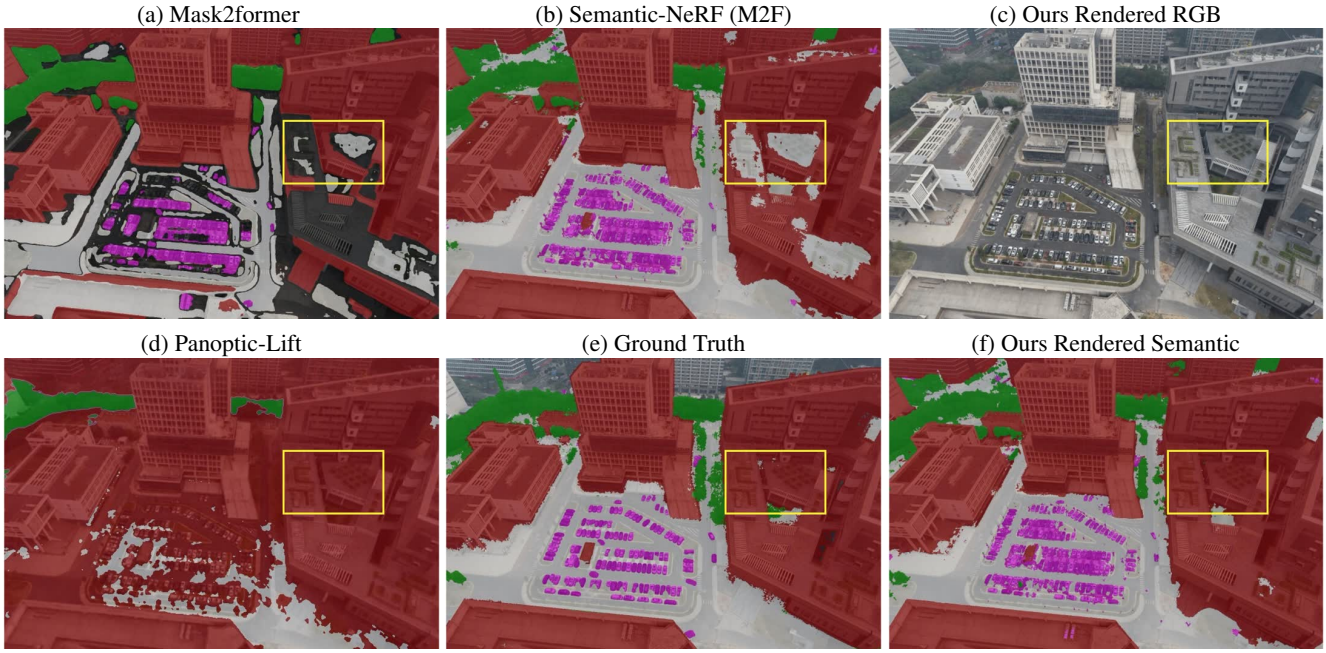
As there are no existing NeRF methods for aerial understanding, we adapt Semantic-NeRF to have the same backbone as ours to have a fair comparison. To further validate the effectiveness of our method, we compare it with the SOTA point-based method [31] on point cloud segmentation (during inference, the input is the GT point cloud). The model is trained on the SensatUrban dataset [33]. For our method, we query the 3D point coordinates in the semantic field to obtain its predicted category. Table II shows that our method achieves more accurate results.

D.3. More Visualization Results

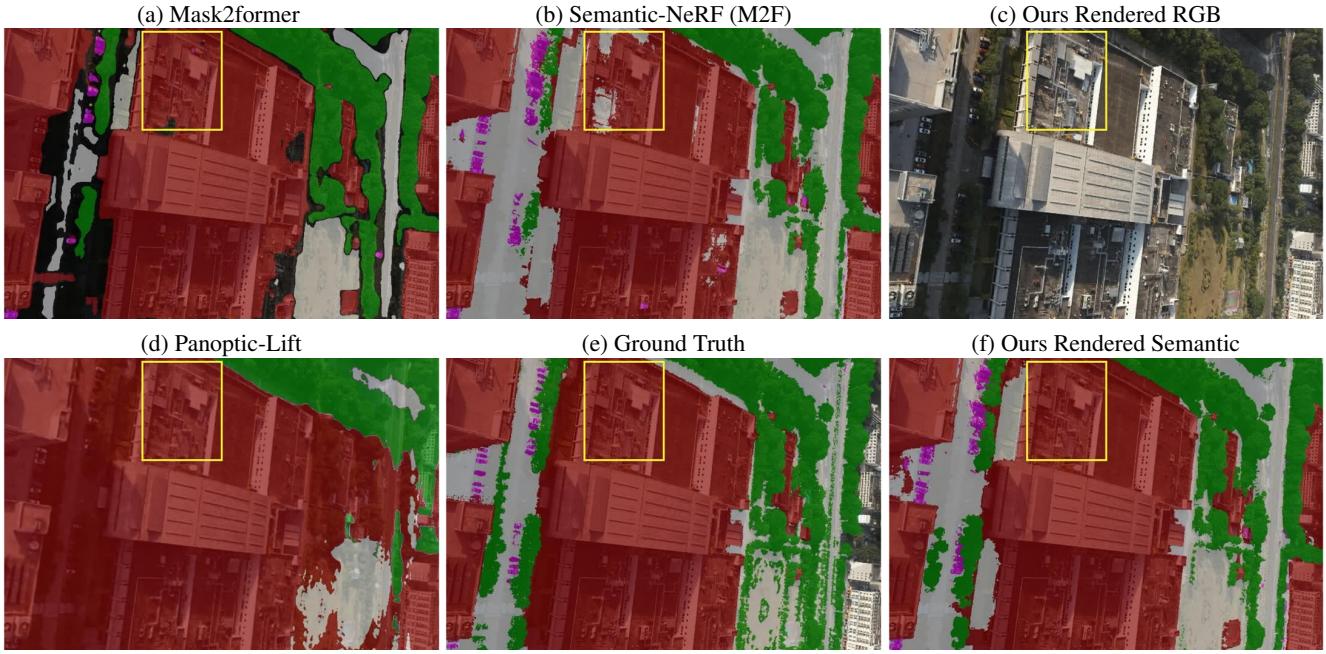
Figure VI shows more qualitative semantic segmentation results on the UrbanBIS dataset.

E. Limitation

Our method relies on a pre-trained 2D segmentation model and the SAM model to generate 2D labels. The failure of 2D methods will affect the final results. Moreover, our method needs a per-scene optimization for scene parsing.



(1) Comparison on *Yuehai-Campus*



(2) Comparison on *Longhua-1*

Figure VI. Qualitative comparison of semantic segmentation on *Yuehai-Campus* and *Longhua-1* from UrbanBIS dataset (*Building*: Red, *Road*: White, *Car*: Violet, *Tree*: Green, unrecognized areas of Mask2former: Black). The areas without masks have no GT annotation in (e). Moreover, we present the novel-view synthesis results of our method.