TrueCity: Real and Simulated Urban Data for Cross-Domain 3D Scene Understanding

Supplementary Material

Here, we further discuss our experiments. Complementary to this supplementary materials is our submission here: [2], which describes creation of the simulation environment.

7. Laser Scanning Simulation

7.1. Configuration

For the laser scanning simulation, we configure the sensor to match the real-world laser scanning setup.

Parameter	Value
Number of lasers	128
Points generated by all lasers	500,000 points/s
Rotation frequency	20 Hz
Upper FOV	15°
Lower FOV	−25°
Horizontal FOV	360°
Range	100 m

Table 5. Configuration parameters of the LiDAR sensor model.

7.2. Radiometry

As highlighted in the main body of the paper, we do not include any radiometric features in the simulation (e.g., RGB). We opt for this approach due to the limited field-of-view of the acquisition camera, which does not fully cover the field-of-view of the laser scanners. This issue would result in incomplete and non-robust image-to-model projections. A possible solution for model texturing is presented in [47]; however, it is subject to wide-angle facade observation, and the choice of the appropriate image for texturing is based on a heuristic, which introduces further randomness to the scan-to-model domain gap analysis. Nevertheless, we are convinced that future work shall further investigate the impact of radiometry and thus object material on the simulation and resulting domain gap.

7.3. Dynamic Objects

The dynamic objects are not included in the simulation setup, as discussed in the main body of the paper. The large discrepancy between the accurate as-it-happened simulation and actual measurements dictates this choice. Even if the procedural vehicle and pedestrian models can be placed in a scene, the simulation of their realistic trajectories is questionable and would include large heuristics in the simulation, which would make the domain gap analysis hardly tractable. Additionally, due to the speed of the mapping vehicle and the observed vehicles and pedestrians, the dynamic object representation is sparse and mainly noisy. We observe the impact of this phenomenon in Tab. 4, where the segmentation results on real-only data also show a large discrepancy between the baseline models.

8. Experimental Setup

8.1. Baseline Segmentation Models

Point-based models operate directly on unordered points with permutation-invariant set functions and lightweight neighborhood aggregation. In our experiments, we use PointNet [32], PointNet++ [33], and RandLA-Net [20]. PointNet applies per-point MLPs with a symmetric reduction for global context; PointNet++ adds hierarchical sampling and grouping to capture local structure; RandLA-Net improves scalability via random sampling with attentive local aggregation.

Kernel-based models 3D convolutions impose locality and yield predictable receptive-field growth. We use KPConv [48], which places learnable kernel points within each neighborhood and aggregates features with geometry-aligned weights in an encoder-decoder pyramid, preserving small-scale structure while expanding context.

Transformer-based models attention replaces fixed kernels with content-adaptive routing across local neighborhoods and long-range, semantically related regions. We consider region-level attention with Superpoint Transformer [34] and octree-structured attention with OctFormer [51], as well as point-level attention with Point Transformer v1 and v3 [55, 64]. Superpoint Transformer attends over a superpoint graph [23]; OctFormer organizes tokens on an octree for multi-scale attention; Point Transformer uses localized self-attention with relative positional encodings, and v3 tightens geometric invariances and reduces overhead for large-scale outdoor LiDAR.

8.2. Training Setup

Unless otherwise noted, we maintain a unified training setup: all models are trained for 100 epochs with a constant learning rate of 10^{-4}, a mini-batch size of 32, and the AdamW optimizer. Model-specific deviations from this setup are described in Sec. 9. Experiments run on NVIDIA H40, L40, and RTX 6000 Ada Generation GPUs;

to isolate the effect of the real/synthetic composition, we keep the training dynamics (optimizer, schedule, batch size, total epochs, augmentations, and point budget) identical across conditions and architectures. Whenever possible, we rely on the authors' public implementations with minimal changes.

8.3. Evaluation Setup

Evaluation follows the same point budget and normalization as training. We report three standard metrics for semantic segmentation: mean Intersection-over-Union (mIoU), mean Accuracy (mAcc), and Overall Accuracy (OA). For C semantic classes, let n_{ij} denote the number of points of class i predicted as class j, and $n_i = \sum_j n_{ij}$ the total number of points in class i. The metrics are defined as:

$$\mathrm{IoU}_i = \frac{n_{ii}}{n_i + \sum_j n_{ji} - n_{ii}}, \tag{2}$$

$$\mathrm{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \mathrm{IoU}_i, \tag{3}$$

$$\mathrm{Acc}_i = \frac{n_{ii}}{n_i}, \tag{4}$$

$$\mathrm{mAcc} = \frac{1}{C} \sum_{i=1}^{C} \mathrm{Acc}_i, \tag{5}$$

$$\mathrm{OA} = \frac{\sum_{i=1}^{C} n_{ii}}{\sum_{i=1}^{C} n_i}. \tag{6}$$

9. Training Recipe

9.1. PointNet and PointNet++

Each input point cloud is downsampled to 2,048 points via farthest-point sampling to preserve geometric coverage. We apply a fixed augmentation regimen: normalization to the unit sphere, a random rotation about the vertical (z) axis, independent reflections across the x and y axes, isotropic scaling drawn from a uniform distribution, and additive jitter. The networks are trained with cross-entropy under the global settings of Sec. 8.

9.2. RandLA-Net

Training operates on pre-sharded point clouds. For each iteration we form a mini-batch by uniformly sampling a fixed-size subset of 2,048 points per sample; at evaluation time we sweep scenes with sequential, non-overlapping chunks. Augmentations follow the original recipe of [20] (normalization, random z-rotation, axis flips, isotropic scaling, elastic distortion, and jitter). Optimization and schedules are identical to Sec. 8.

9.3. KPConv

We use the reference KPConv segmentation model [48] with the authors' architectural defaults. To ensure parity across methods, each training item is reduced to 2,048 points via farthest-point sampling and processed with the same augmentation regimen as PointNet/PointNet++. The model is optimized with AdamW at a constant learning rate of 10^{-4}; loss weighting follows the effective-number scheme.

9.4. Point Transformer v1 and v3

For Point Transformer v1 [64] and v3 [55], inputs are standardized to 2,048 points per item using farthest-point sampling. We apply the same normalization and geometric augmentations as above to avoid confounding from data processing. Both variants are trained for 100 epochs with AdamW (10^{-4} learning rate) and cross-entropy with class weighting as in Sec. 8. Evaluation uses the identical point budget and normalization.

9.5. OctFormer

We follow the reference implementation of OctFormer [51]. Coordinates are normalized to the unit sphere and encoded into an OCNN octree (depth 8, full_depth 2) with the associated neighborhood structures. The OctFormer backbone feeds an FPN-style head that upsamples multi-scale features and interpolates them to query points for per-point classification. Unless stated otherwise, we sample 2,048 points per item for parity with the other models and train with AdamW at 10^{-4}, cross-entropy (with `ignore_label`, optional class weights and label smoothing), and the global settings of Sec. 8.

9.6. Superpoint Transformer

We adopt the Superpoint Transformer (SPT) [34] in its reference implementation. As per the SPT methodology, each input scene is oversegmented into superpoints, so no further downsampling or chunking is required. Preprocessing and augmentation follow the DALES configuration in the official codebase. To remain faithful to the original setup, training uses stochastic gradient descent (SGD) with a learning rate of 0.01, weight decay of 10^{-4}, and a batch size of 4, which differs slightly from the global setting in Sec. 8.

9.7. Data Augmentation

We acknowledge that the data augmentation is a common technique for improving model performance and increasing its generalization [22]. However, we opted to omit data augmentation in our experimental setup, since it would have introduced an additional variable to the coherent domain gap analysis: The influence of data augmentation would make assessing the impact of real-to-synthetic data hardly tractable.

10. Additional Information about Dataset and Evaluation Result

Class / Metric	PointNet					PointNet++				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
mIoU	6.03	10.74	10.89	13.10	**14.51**	9.72	20.95	23.18	**25.38**	23.39
mAcc	14.58	21.63	21.82	25.35	**25.98**	19.75	29.37	35.22	30.84	**38.03**
OA	30.36	48.10	49.29	47.99	**49.82**	34.39	62.80	**65.36**	63.27	63.15
RoadSurface	7.84	51.92	54.04	49.78	**67.55**	60.90	73.90	**81.20**	80.30	72.50
GroundSurface	16.45	24.11	17.27	21.90	**28.55**	33.20	45.30	**50.40**	49.60	32.40
CityFurniture	0.62	2.68	4.12	**10.80**	8.56	15.20	14.60	21.30	25.50	**46.20**
Vehicle	0.00	0.00	0.00	0.00	0.00	**3.50**	0.00	0.18	0.51	0.01
Pedestrian	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.32**	0.04
WallSurface	39.97	37.01	**45.28**	43.07	30.87	0.20	56.90	63.50	**65.70**	63.20
RoofSurface	0.00	0.06	**0.19**	0.04	0.00	0.00	**0.80**	0.30	0.10	0.30
Door	0.02	0.00	0.00	**0.11**	0.00	0.00	0.00	0.20	0.10	**0.70**
Window	2.57	6.00	1.05	5.38	**6.39**	3.61	7.30	4.50	4.20	**14.60**
BuildingInstallation	0.01	0.26	0.01	**4.53**	0.15	0.00	0.70	3.00	1.70	**4.60**
SolitaryVegetationObject	4.24	1.04	5.78	11.40	**14.79**	0.00	**50.60**	39.30	45.60	14.40
Noise	0.63	5.82	2.93	10.21	**17.26**	0.00	1.30	14.30	30.90	**31.70**

Table 6. Per-class IoU (↑) under different S–R dataset mixtures. Models: **PointNet** [32] and **PointNet++** [33]. Bold marks the best value for each model across mixtures.

Class / Metric	RandLA-Net					KPConv				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
mIoU	8.98	13.25	15.73	16.89	**17.71**	15.84	21.55	28.50	22.33	**29.90**
mAcc	25.62	**31.27**	28.48	29.82	30.97	38.12	33.59	37.79	**42.91**	42.01
OA	35.40	50.32	**59.37**	57.09	54.57	50.07	62.08	61.62	61.92	**62.80**
RoadSurface	3.12	60.53	**70.35**	62.01	60.26	30.64	**69.26**	54.26	56.31	54.59
GroundSurface	23.31	**29.51**	29.41	25.27	26.65	24.67	**36.49**	31.91	33.20	35.96
CityFurniture	2.33	**15.91**	9.99	14.38	12.93	**18.25**	10.09	12.67	1.85	14.57
Vehicle	0.00	0.03	1.03	0.59	**3.27**	0.00	0.00	38.93	**70.33**	63.64
Pedestrian	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WallSurface	45.19	41.29	51.15	**54.18**	53.23	59.97	63.21	68.54	**73.55**	71.03
RoofSurface	**0.32**	0.17	0.29	0.19	0.18	0.82	0.01	0.47	0.00	**1.79**
Door	**18.22**	2.51	1.41	0.16	3.25	0.00	0.00	0.00	0.00	0.00
Window	3.82	2.83	9.51	8.75	**9.72**	1.86	0.43	**18.21**	1.17	2.89
BuildingInstallation	0.94	1.68	2.91	**8.46**	2.90	**7.85**	3.02	0.85	0.85	3.72
SolitaryVegetationObject	10.42	1.91	2.07	10.68	**14.86**	45.74	70.51	**81.45**	0.41	77.13
Noise	0.10	2.64	10.63	17.95	**25.27**	0.31	5.58	**34.73**	30.26	33.43

Table 7. Per-class IoU (↑) under different S–R dataset mixtures. Models: **RandLA-Net** [20] and **KPConv** [48]. Bold marks the best value for each model across mixtures.

Class / Metric	Point Transformer v1					Point Transformer v3				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
mIoU	16.30	19.79	23.43	24.66	**28.89**	14.13	19.29	**25.30**	24.64	25.24
mAcc	26.09	34.39	37.01	35.52	**39.33**	28.07	34.09	43.16	40.33	**44.09**
OA	57.54	60.29	67.54	**68.70**	67.98	53.15	60.22	**65.94**	65.72	60.75
RoadSurface	62.41	61.89	**80.33**	77.93	70.44	64.42	68.85	70.74	**71.17**	42.36
GroundSurface	31.94	37.54	**45.54**	37.09	41.67	32.53	32.66	**33.27**	32.80	27.53
CityFurniture	25.89	19.50	13.51	14.23	**44.83**	5.35	34.86	24.11	17.34	**35.13**
Vehicle	0.00	0.51	4.41	1.51	**6.54**	0.00	0.00	0.00	0.00	0.00
Pedestrian	0.00	0.00	**0.05**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WallSurface	53.15	61.42	64.39	65.32	**67.10**	48.15	54.79	63.42	64.63	**69.69**
RoofSurface	0.10	0.29	**0.92**	0.30	0.08	0.01	0.10	0.10	0.00	**0.36**
Door	0.01	0.13	0.51	**0.66**	0.01	0.00	0.81	**2.77**	0.08	1.62
Window	3.28	5.61	7.61	11.78	**31.29**	3.09	5.17	4.92	4.35	**10.70**
BuildingInstallation	3.08	5.24	6.69	**8.87**	3.41	8.40	2.53	**9.95**	9.30	4.44
SolitaryVegetationObject	15.54	35.35	34.70	43.22	**52.31**	6.72	12.90	60.86	60.80	**70.39**
Noise	0.16	9.98	22.43	**35.04**	28.96	0.84	18.80	33.50	35.19	**40.69**

Table 8. Per-class IoU (↑) under different S–R dataset mixtures. Models: **Point Transformer v1** [64] and **Point Transformer v3** [55]. Bold marks the best value for each model across mixtures.

Class / Metric	OctFormer					Superpoint Transformer				
	100S	75S	50S	25S	0S	100S	75S	50S	25S	0S
mIoU	13.07	14.17	14.22	13.91	**17.65**	14.31	17.01	14.22	**19**.61	15.96
mAcc	22.84	23.99	26.28	26.15	**27.19**	24.28	29.40	26.42	**31.79**	28.29
OA	53.30	55.34	49.71	50.97	**56.28**	54.17	**58.62**	54.63	56.98	54.64
RoadSurface	49.32	51.58	35.34	54.81	**71.98**	**77.74**	73.73	61.19	63.51	53.16
GroundSurface	16.01	16.34	**22.60**	19.60	16.99	18.09	30.35	16.83	**31.35**	25.92
CityFurniture	0.92	2.18	**15.24**	0.01	0.14	1.22	**3.70**	0.85	2.49	0.47
Vehicle	0.00	0.00	0.00	0.07	**20.82**	0.00	0.26	0.55	**4.82**	0.22
Pedestrian	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WallSurface	56.22	**62.14**	60.36	45.79	38.96	38.97	47.32	55.57	56.98	**63.19**
RoofSurface	0.00	0.00	0.00	0.00	0.00	2.11	2.60	2.84	**3.21**	1.25
Door	0.00	0.00	0.00	0.00	0.00	**0.30**	0.09	0.00	0.00	0.00
Window	0.00	0.00	**0.76**	0.00	0.00	**8.94**	3.89	1.43	2.99	0.55
BuildingInstallation	0.00	0.00	**4.02**	0.00	0.00	2.29	0.75	0.01	1.30	**2.63**
SolitaryVegetationObject	34.33	37.85	32.25	26.72	**47.30**	21.54	34.45	16.70	**48.13**	21.12
Noise	0.00	0.00	0.08	**19.93**	15.57	0.46	6.96	14.70	20.50	**23.07**

Table 9. Per-class IoU (↑) under different S–R dataset mixtures. Models: **OctFormer** [51] and **Superpoint Transformer** [34]. Bold marks the best value for each model across mixtures.

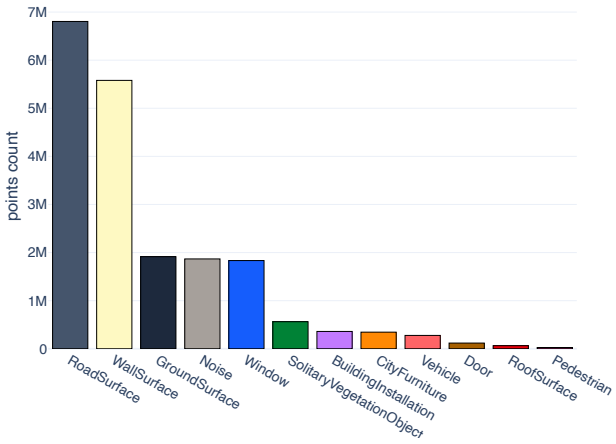

Figure 6. The class distribution of the real-only test set.

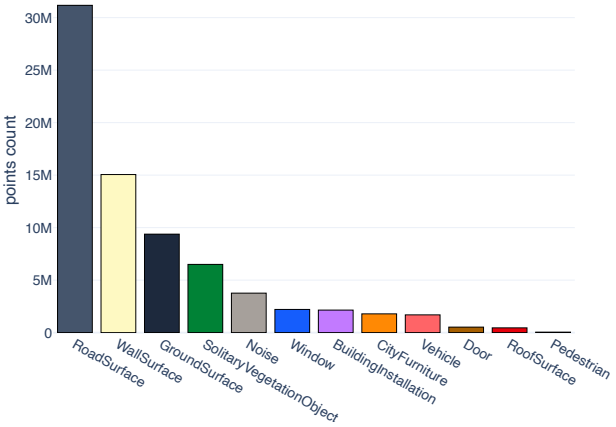

Figure 9. The class distribution of the 0S–100R train set.

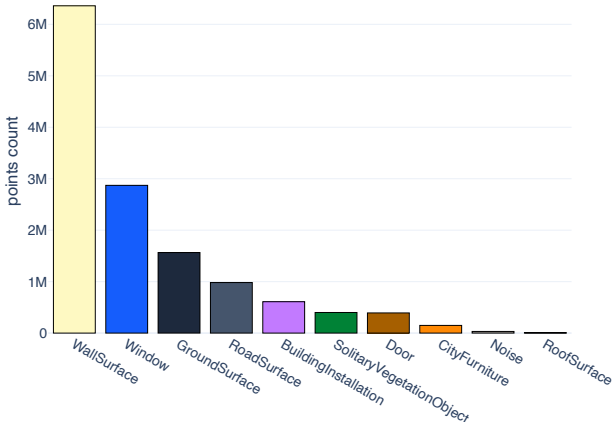

Figure 7. The class distribution of the synthetic-only test set, which is not used in training and evaluation.

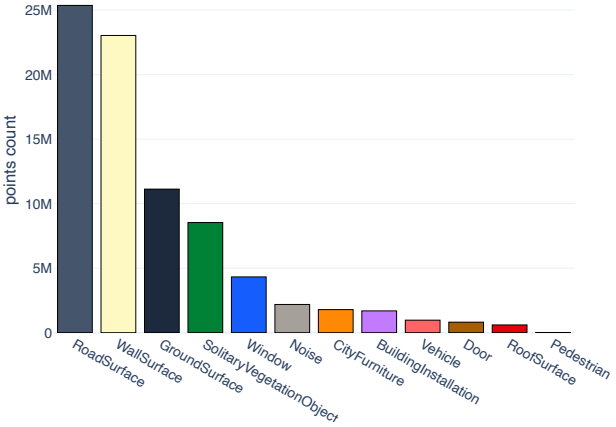

Figure 10. The class distribution of the 25S–75R train set.

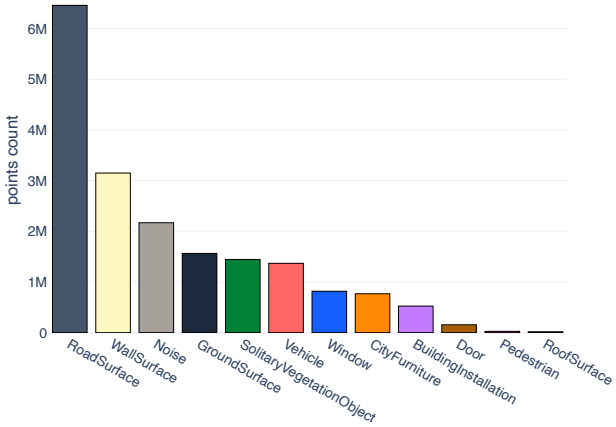

Figure 8. The class distribution of the real-only validation set.

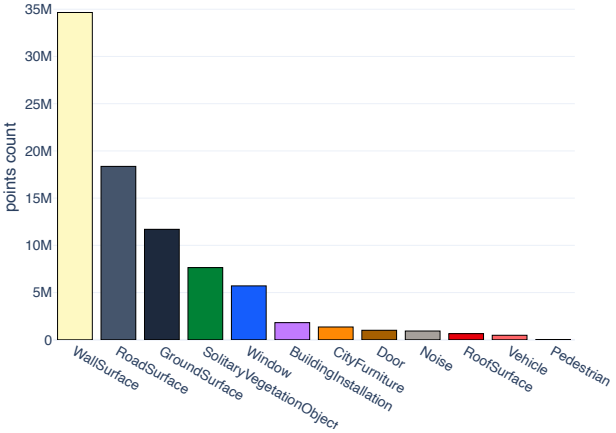

Figure 11. The class distribution of the 50S–50R train set.

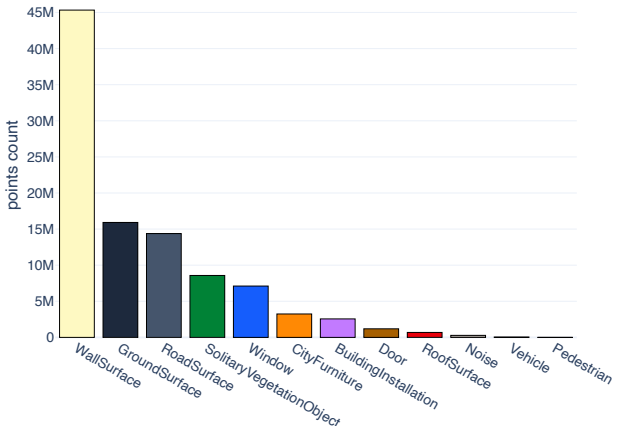

Figure 12. The class distribution of the 75S–25R train set.

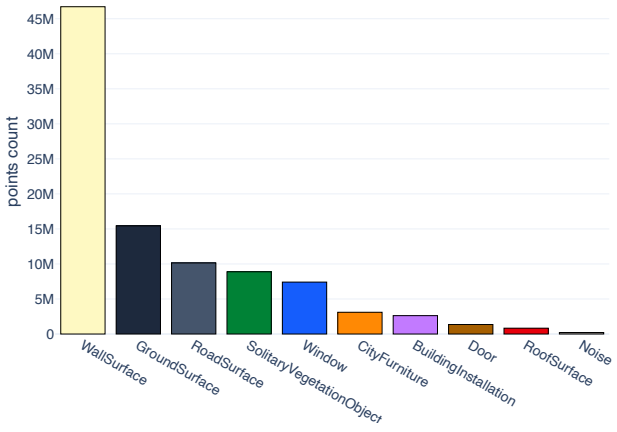

Figure 13. The class distribution of the 100S–0R train set.