MonoScene: Monocular 3D Semantic Scene Completion – Supplementary Material –

Anh-Quan Cao
Inria
anh-quan.cao@inria.fr

Raoul de Charette
Inria
raoul.de-charette@inria.fr

We provide details about the main baselines and MonoScene in Sec. 1, and include additional qualitative and quantitative results in Sec. 2.

Results on image sequences are in the supplementary video: https://youtu.be/qh7La1tRJmE.

1. Architectures details

1.1. Baselines

AICNet [8]. We use the official implementation of AICNet[1]. For the RGB-inferred version, *i.e.* AICNet$^{\text{rgb}}$, we infer depth with the pre-trained AdaBins [2] on NYUv2 [14] and SemanticKITTI [1] from the official repository[2].

3DSketch [4]. We use 3DSketch official code[3]. For 3DSketch$^{\text{rgb}}$, we again use AdaBins (*cf*. above) and convert depth to TSDF with 'tsdf-fusion'[4] from the 3DMatch Toolbox [17].

JS3C-Net [16]. We use the official code of JS3C-Net[5]. For JS3C-Net$^{\text{rgb}}$, we generate the input point cloud by unprojecting the predicted depth (using AdaBins) to 3D using the camera intrinsics. The semantic point clouds, required to train JS3C-Net, are obtained by augmenting the unprojected point clouds with the 2D semantics obtained using the code[6] of [19].

LMSCNet [12]. We use the official implementation of LMSCNet[7]. For LMSCNet$^{\text{rgb}}$, the input occupancy grid is obtained by discretizing the unprojected point cloud.

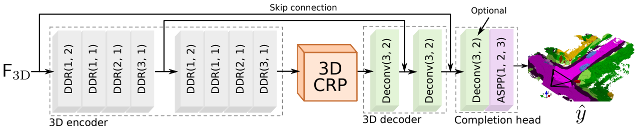

Figure 1. **MonoScene 3D network.** The 3D UNet uses 2 downscale layers with DDR blocks [9] and 2 upscale layers with deconv. The completion head uses ASPP and an *optional* deconv layer. Notations: DDR(dilation, downsample rate), Deconv(kernel size, dilation), ASPP(dilations).

1.2. MonoScene

Fig. 1 details our 3D UNet. Similar to 3DSketch [4], we adopt DDR [9] as the basic building block for large receptive field and low memory cost. The 3D encoder has 2 layers, each downscales by half and has 4 DDR blocks. The 3D decoder has two deconv layers, each doubles the scale. Similar to others [12] the completion head has an ASPP with dilations (1, 2, 3) to gather multi-scale features and an *optional* deconv to reach output size – used in SemanticKITTI only.

For training, MonoScene took 7 hours using 2 V100 32g GPUs (2 items per GPU) on NYUv2 [14] and 28 hours to train using 4 V100 32g GPUs (1 item per GPU) on SemanticKITTI [1].

2. Additional results

2.1. SemanticKITTI

Quantitative performance. We report performance on validation set in Tab. 1. Comparing against the test set performance from the main paper Tab. 1b, we notice MonoScene generalizes better than JS3C-Net$^{\text{rgb}}$ and AICNet$^{\text{rgb}}$ since the validation and test set gap is smaller (-0.42 vs -1.34 and -1.22). We also report the complete SemanticKITTI official benchmark (*i.e.* hidden test set) in Tab. 2 showing that while MonoScene uses only RGB, it still outperforms some of the 3D input SSC baselines.

[1] https://github.com/waterljwant/SSC
[2] https://github.com/shariqfarooq123/AdaBins
[3] https://github.com/charlesCXK/TorchSSC
[4] https://github.com/andyzeng/tsdf-fusion
[5] https://github.com/yanx27/JS3C-Net
[6] https://github.com/YeLyuUT/SSeg
[7] https://github.com/cv-rits/LMSCNet

		SC	SSC																			
Method	SSC Input	IoU	road (15.30%) 🟪	sidewalk (11.13%) 🟪	parking (1.12%) 🟪	other-ground (0.56%) 🟥	building (14.1%) 🟨	car (3.92%) 🟦	truck (0.16%) 🟪	bicycle (0.03%) 🟦	motorcycle (0.03%) 🟦	other-vehicle (0.20%) 🟪	vegetation (39.3%) 🟩	trunk (0.51%) 🟫	terrain (9.17%) 🟩	person (0.07%) 🟥	bicyclist (0.07%) 🟪	motorcyclist (0.05%) 🟪	fence (3.90%) 🟧	pole (0.29%) 🟨	traffic-sign (0.08%) 🟥	mIoU
LMSCNet^{rgb} [12]	$\hat{x}^{occ}_{3D}$	28.61	40.68	18.22	4.38	0.00	10.31	18.33	0.00	0.00	0.00	0.00	13.66	0.02	20.54	0.00	0.00	0.00	1.21	0.00	0.00	6.70
3DSketch^{rgb} [4]	x^{rgb},$\hat{x}^{TSDF}$	33.30	41.32	21.63	0.00	0.00	$\underline{14.81}$	18.59	0.00	0.00	0.00	0.00	**19.09**	0.00	26.40	0.00	0.00	0.00	0.73	0.00	0.00	7.50
AICNet^{rgb} [8]	x^{rgb},$\hat{x}^{depth}$	29.59	43.55	20.55	$\underline{11.97}$	$\underline{0.07}$	12.94	14.71	$\underline{4.53}$	0.00	0.00	0.00	15.37	$\underline{2.90}$	**28.71**	0.00	0.00	0.00	2.52	0.06	0.00	8.31
*JS3C-Net^{rgb} [16]	$\hat{x}^{pts}$	**38.98**	50.49	$\underline{23.74}$	11.94	$\underline{0.07}$	**15.03**	**24.65**	4.41	0.00	0.00	**6.15**	18.11	**4.33**	26.86	$\underline{0.67}$	0.27	0.00	$\underline{3.94}$	$\underline{3.77}$	$\underline{1.45}$	$\underline{10.31}$
MonoScene (ours)	x^{rgb}	$\underline{37.12}$	**57.47**	**27.05**	**15.72**	**0.87**	14.24	$\underline{23.55}$	**7.83**	**0.20**	**0.77**	$\underline{3.59}$	$\underline{18.12}$	2.57	**30.76**	**1.79**	**1.03**	0.00	**6.39**	**4.11**	**2.48**	**11.50**

* Uses pretrained semantic segmentation network.

Table 1. **Performance on SemanticKITTI [1] (validation set).** We report the performance on semantic scene completion (SSC - mIoU) and scene completion (SC - IoU) for RGB-inferred baselines and our method.

Method	Input	IoU	mIoU
3D			
SSCNet [15]	x^{TSDF}	29.8	9.5
TS3D [7]	x^{TSDF}+x^{rgb}	29.8	9.5
TS3D+DNet [1]	x^{TSDF}+x^{rgb}	25.0	10.2
ESSCNet [18]	x^{pts}	41.8	17.5
LMSCNet [12]	x^{occ}	$\underline{56.7}$	17.6
TS3D+DNet+SATNet [1]	x^{occ}	50.6	17.7
Local-DIFs [11]	x^{occ}	**57.7**	22.7
JS3C-Net [16]	x^{pts}	56.6	$\underline{23.8}$
S3CNet [5]	x^{occ}	45.6	**29.5**
2D			
MonoScene	x^{rgb}	34.2	11.1

Table 2. **Complete SemanticKITTI official benchmark (hidden test set)**. Results are taken from [13]. Despite using only single RGB image as input, MonoScene still surpasses some of the SSC baselines with 3D input.

Qualitative performance. In Fig. 2 we also include additional qualitative results. Compared to all baselines, MonoScene captures better landscape and objects (*e.g.* cars, rows 3-9; pedestrian, rows 6, 10; traffic-sign, rows 3, 5). Still, it struggles to predict thin small objects (*e.g.* trunk, row 1; pedestrian, row 3; traffic-sign, row 2, 6), separate far away consecutive cars (*e.g.* row 5, 7, 8), and infer very complex, highly cluttered scenes (*e.g.* rows 9, 10).

Evaluation scope. Tab. 3 reports the performance when considering either only the voxels inside FOV (in-FOV), outside FOV (out-FOV), or all voxels (Whole Scene) as reported in the main paper. Compared to the Whole Scene, the in-FOV performance is higher since it considers visible surfaces, whereas the out-FOV performance is significantly lower since the image does not observe it.

2.2. NYUv2

We show additional qualitative results in Fig. 3. In overall, MonoScene predicts better scene layouts and better objects geometry, evidently in rows 1-4, 6, 9, 10. Still, MonoScene mispredicts complex (*e.g.* bookshelfs, row 1, 4, 6), or rare objects (running machine, row 8). Sometimes, it confuses semantically-similar classes (*e.g.* window/objects, row 6, 8; beds/objects, row 1, 5; furniture/table, row 1, 2) due to the high variance of indoor scene *i.e.* wide range of camera poses, objects have completely different appearances, poses and positions even in the same category *e.g.* beds (rows 1, 5-7, 9); sofa (row 2-4).

2.3. Generalization

Fig. 4 illustrates the predictions of MonoScene, trained on SemanticKITTI, on datasets with different camera setups. We can see the increase in distortion as the camera setups depart from the ones used during training. Furthermore, the domain gap (*i.e.* city, country, etc.) also plays an important role. As MonoScene is trained on the mid-size German city of Karlsruhe, with residential scenes and narrow roads, the gap is smaller with KITTI-360 having similar scenes. The results on nuScenes and Cityscapes suffer both from the camera setup changes and the large metropolitan scenes (*i.e.* Stuttgart - Cityscapes; Singapore, Boston - nuScenes) having wider streets.

	in-FOV		out-FOV		Whole Scene	
	IoU ↑	mIoU ↑	IoU ↑	mIoU ↑	IoU ↑	mIoU ↑
LMSCNet^{rgb} [12]	37.62	8.87	25.36	5.48	34.41	8.17
3DSketch^{rgb} [4]	32.24	7.82	26.50	5.83	33.30	7.50
AICNet^{rgb} [8]	35.69	8.75	25.79	5.61	29.59	8.31
*JS3C-Net^{rgb} [16]	**42.22**	$\underline{11.29}$	$\underline{28.27}$	$\underline{6.31}$	**38.98**	$\underline{10.31}$
MonoScene(ours)	$\underline{39.13}$	**12.78**	**31.60**	**7.45**	$\underline{37.12}$	**11.50**

* Uses pretrained semantic segmentation network.

Table 3. **SemanticKITTI performance (validation set) on in-/out-FOV and the Whole Scene.** We report the performance on the scenery inside (in-FOV), outside (out-FOV) camera FOV, and considering all voxels (Whole Scene). MonoScene is best in most cases, with in-FOV performance logically higher.

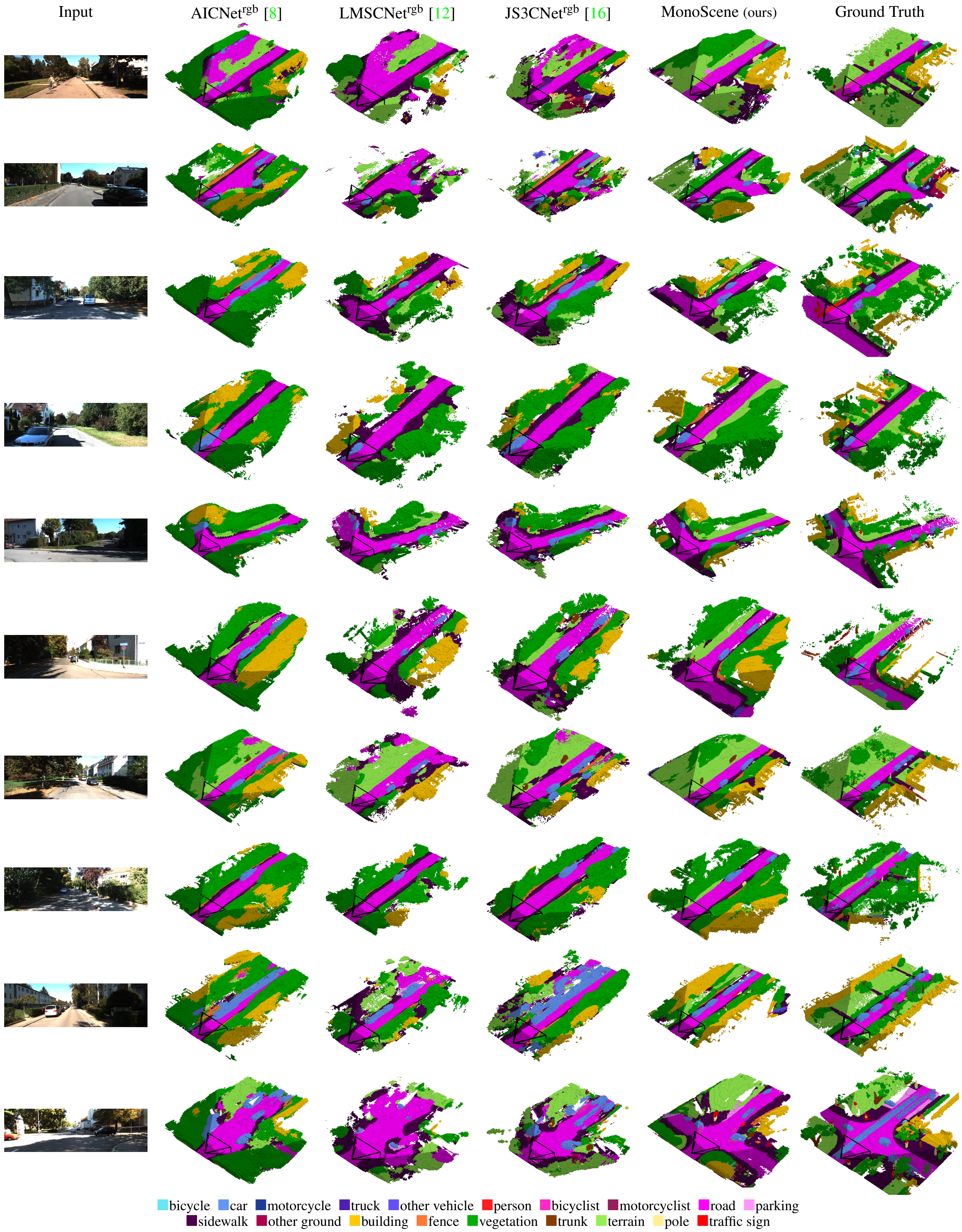
Input
AICNet^{rgb} [8]
LMSCNet^{rgb} [12]
JS3CNet^{rgb} [16]
MonoScene (ours)
Ground Truth
🟦bicycle 🟦car 🟦motorcycle 🟪truck 🟪other vehicle 🟥person 🟪bicyclist 🟪motorcyclist 🟪road 🟪parking
🟪sidewalk 🟥other ground 🟨building 🟧fence 🟩vegetation 🟫trunk 🟩terrain 🟨pole 🟥traffic sign

Figure 2. **Results on SemanticKITTI [1] (validation set).** The input is shown left. Darker voxels represent the scenery outside the viewing frustum (*i.e*. unseen by the image).

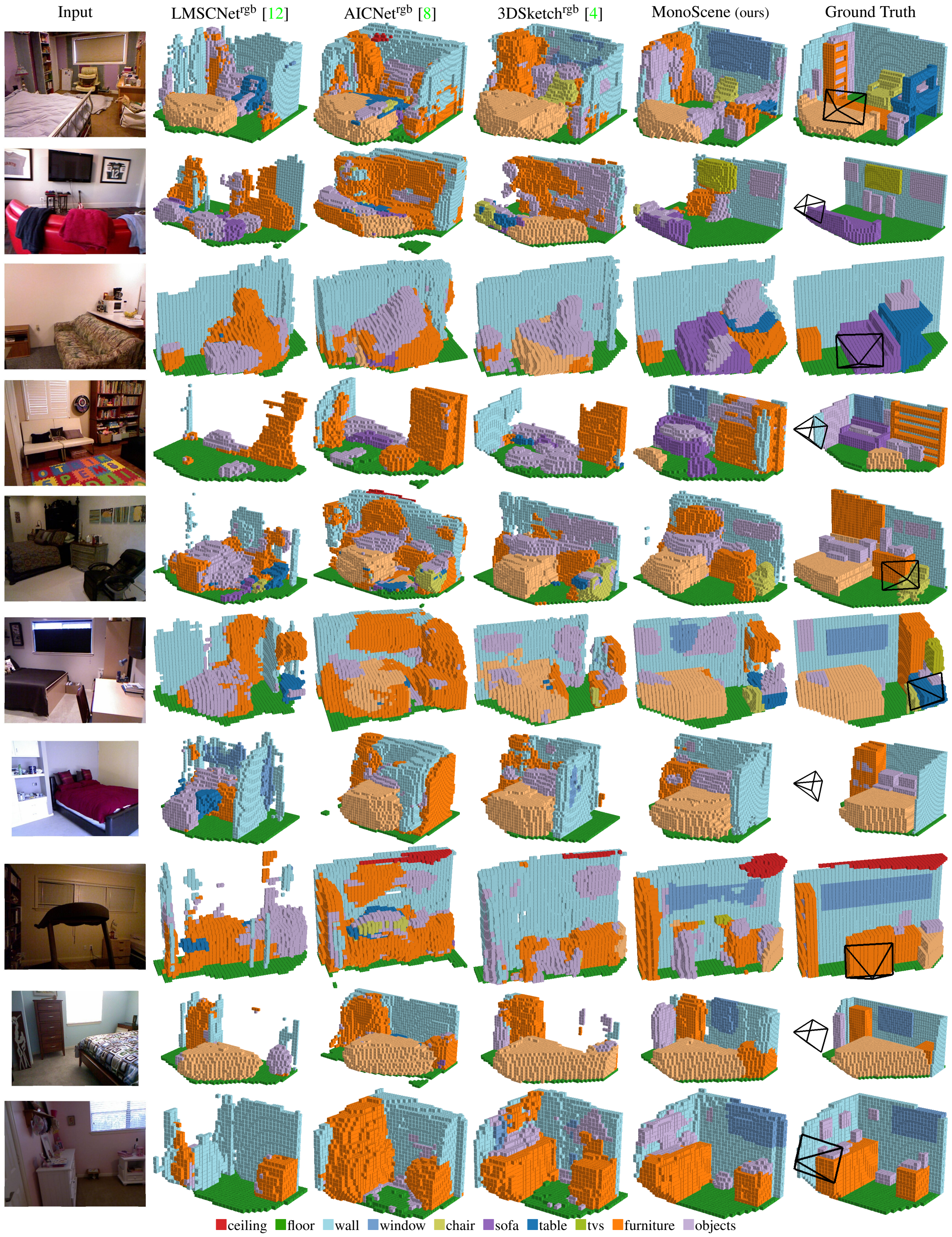

Figure 3. **Results on NYUv2 [14] (test set).** The input is shown leftmost and the camera viewing frustum is shown in the ground truth (rightmost).

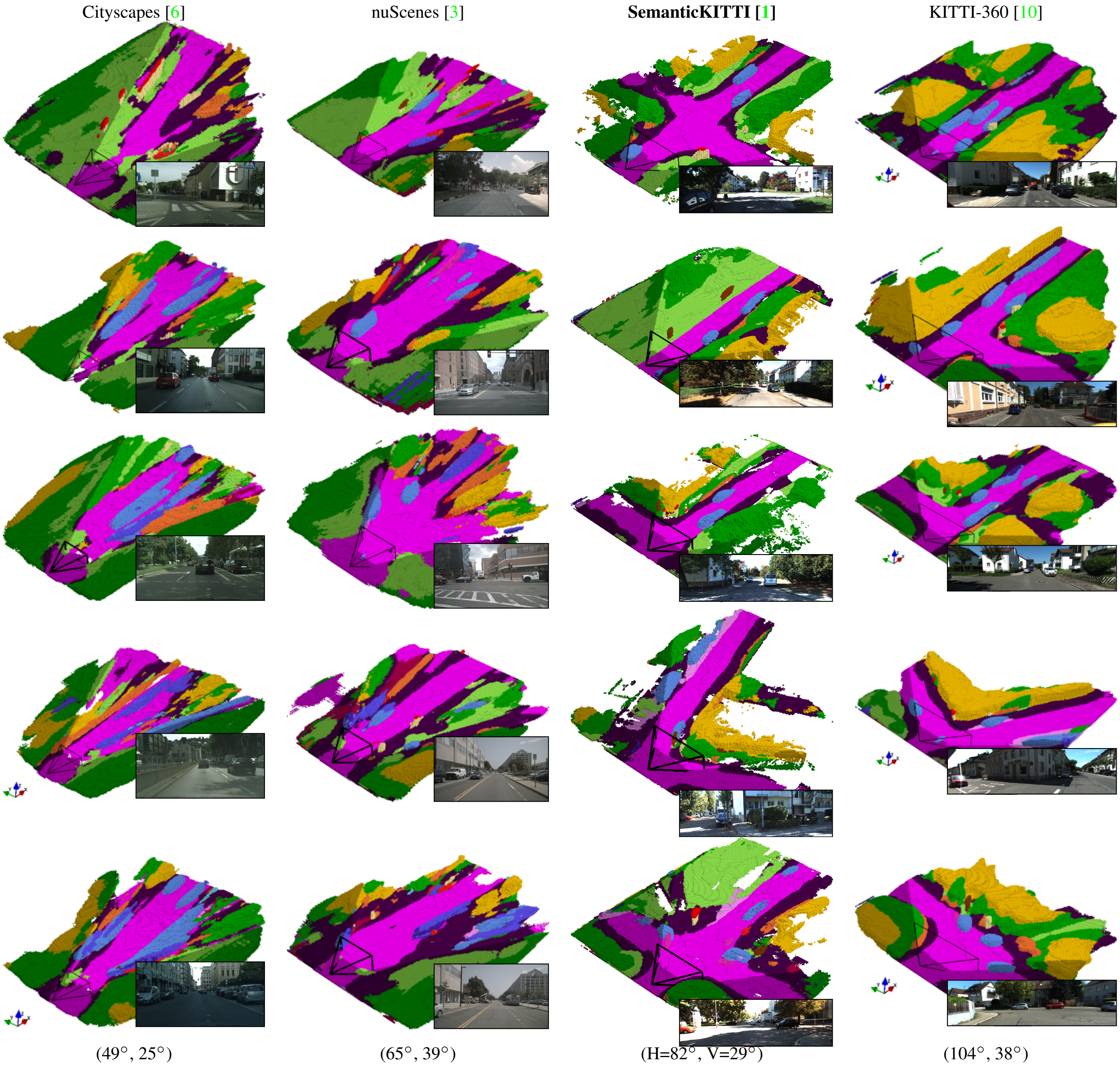

Figure 4. **Domain gap and Camera effects.** Outputs of MonoScene when trained on SemanticKITTI having horizontal FOV of 82°, and tested on datasets with decreasing (left) or increasing (right) FOV. SemanticKITTI and KITTI-360 are recored in mid-size German city of Karlsruhe while nuScenes and Cityscapes are from large metropolitan areas (*e.g*. Stuttgart - Cityscapes; Singapore, Boston - nuScenes) whose streets are much wider, denser and have different landscapes.

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, 2019. 1, 2, 3, 5

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth Estimation using Adaptive Bins. In *CVPR*, 2021. 1

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 5

[4] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *CVPR*, 2020. 1, 2, 4

[5] Ran Cheng, Christopher Agia, Yuan Ren, Xinhai Li, and

Bingbing Liu. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *CoRL*, 2020. 2

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 5

[7] Martin Garbade, Johann Sawatzky, Alexander Richard, and Juergen Gall. Two stream 3d semantic scene completion. In *CVPRW*, 2019. 2

[8] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *CVPR*, 2020. 1, 2, 3, 4

[9] Jie Li, Yu Liu, Dong Gong, Qinfeng Shi, Xia Yuan, Chunxia Zhao, and Ian Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *CVPR*, 2019. 1

[10] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv.org*, 2109.13410, 2021. 5

[11] Christoph Rist, David Emmerichs, Markus Enzweiler, and Dariu Gavrila. Semantic scene completion using local deep implicit functions on lidar data. *T-PAMI*, 2021. 2

[12] Luis Roldão, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *3DV*, 2020. 1, 2, 3, 4

[13] Luis Roldão, Raoul De Charette, and Anne Verroust-Blondet. 3D Semantic Scene Completion: a Survey. *IJCV*, 2021. 2

[14] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV*, 2012. 1, 4

[15] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 2

[16] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*, 2021. 1, 2, 3

[17] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017. 1

[18] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao. Efficient semantic scene completion network with spatial group convolution. In *ECCV*, 2018. 2

[19] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *CVPR*, 2019. 1