OFVL-MS: Once for Visual Localization across Multiple Indoor Scenes —Supplementary Material—

In this supplementary material, we provide additional information to further understand our proposed OFVL-MS. In Sec. 1, we present more details on network architecture. In Sec. 2, we further detail the experimental setups for all datasets. In Sec. 3, we provide a detailed analysis of the incremental experiments. In Sec. 4, we introduce the collection and fabrication process of the proposed LIVL dataset. In Sec. 5, we provide various qualitative results on benchmarks.

1. Network Architecture

OFVL-MS rescales the image to 640×480 as input and predicts the scene coordinates and uncertainty feature maps with the resolution of 60×80.

Backbone. As mentioned in the main paper, we utilize ResNet [3] with slight modifications as the backbone to extract features from the input images. Specifically, we remove the max pooling layer, average pooling, and fully connected layer, and change the stride of four residual blocks as (1,1,2,2) to ensure that the resolution of intermediate features F_n is $H/8 \times W/8$, where H and W mean the height and width of the original image. Besides, the output channels of pre-layer and four residual blocks of all OFVL-MS families are set to $(16, 64, 128, 256, 512)$, as shown in Fig. 1.

Regression Layer. Our regression layer is designed as a fully connected layer, as shown in Fig. 2. The intermediate features F_n are fed into a set of convolutional layers to predict scene coordinates $\hat{D}_n$ and uncertainty $\hat{U}_n$.

2. Implementation Details

2.1. Gradient Estimation of Scores $s_{n,i}$

In the backward pass of the network, the gradient of the scores $s_{n,i}$ is formulated as:

$$\nabla_{s_{n,i}} L_{n,i} = \frac{\partial L_{n,i}}{\partial \Theta(s_{n,i})} \frac{\partial \Theta(s_{n,i})}{\partial s_{n,i}}. \tag{1}$$

Since the gradient of the indicator function $\Theta(\cdot)$ is zero at almost all points, the scores $s_{n,i}$ cannot be directly optimized using gradient descent. To avoid this dilemma, we utilize a straight gradient estimator [1] and modify the gradient of $s_{n,i}$ as:

$$\nabla_{s_{n,i}} L_{n,i} = \frac{\partial L_{n,i}}{\partial \Theta(s_{n,i})}. \tag{2}$$

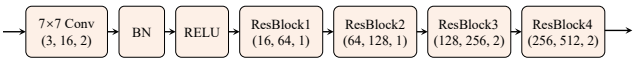


Figure 1: The modified architecture of ResNet that OFVL-MS families use.

2.2. Training Details.

We view the visual localization of each scene as an individual task. Since each task has its own dataset domain, we use multiple GPUs to optimize these tasks, where the task-shared parameters are optimized in the global group while the task-specific parameters are optimized in the task-specific group, which is implemented by defining multiple communication groups in DistributedDataParallel of PyTorch. For each scene, we utilize 2 GPUs for training.

OFVL-MS employs Adamw solver for optimization with an initial learning rate of 1.4×10^{-3} and weight decay of 0.05 for 200000 iterations with a batch size of 4 when training 7-Scenes dataset [10]. Besides, we apply the cosine annealing policy to adjust learning rate with warmup of 250 iterations. For data augmentation, we follow HSCNet [6] and apply affine transformations to each training image. Technically, we translate, rotate, and scale the image by values uniformly sampled from $[-20\%, 20\%], [-30°, 30°], [0.7, 1.5]$ respectively. We also augment the images with additive brightness changes uniformly sampled from $[-20, 20]$.

For 12-Scenes dataset [11], OFVL-MS utilizes Adamw with an initial learning rate of 2.4×10^{-3} for 200000 iterations with a batch size of 4. Considering the training trajectories almost coincide with test trajectories in 12-Scenes

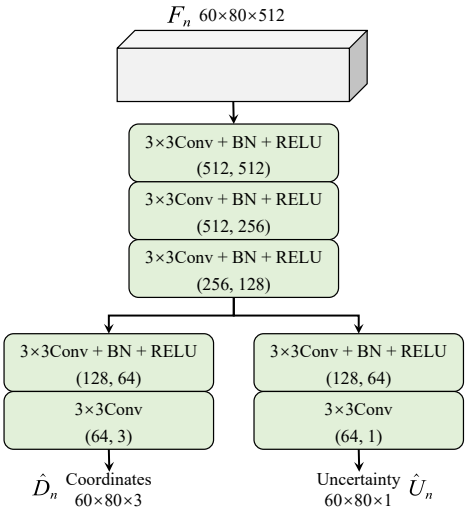


Figure 2: **Regression Layer Architecture.**

	Runtime (ms)	Med. Err.	Acc.
DSAC++ [2]	68.2	0.036,1.10	74.4
HSCNet [6]	81.7	0.03,0.9	84.8
OFVL-MS18	**58.4**	0.025,0.80	85.20
OFVL-MS34	71.6	0.023,0.74	87.37
OFVL-MS50	86.9	**0.021,0.69**	**88.72**

Table 1: **The runtime** that each method occupies on 7-Scenes dataset.

dataset, we do not employ data augmentation and set weight decay to 0.

We follow [6] and define the evaluation metrics as: (i) the median positional errors. (ii) the median rotational errors. (iii) the percentage of images whose positional and rotational errors less than $5cm$ and 5°. The positional errors Δt and rotational errors ΔR are formatted as:

$$\begin{aligned} \Delta t &= ||\hat{t} - t||_2, \\ \Delta R &= \frac{||Rod(\hat{R}^T R)||_2}{\pi} \times 180, \end{aligned} \tag{3}$$

where $\hat{t}$ means predict translation vector, t means ground truth translation vector, $\hat{R}$ means predict rotation matrix, R means ground truth rotation matrix, $Rod(\cdot)$ means the Rodriguez formula used to transform the rotation matrix into a rotation vector.

2.3. Pose Estimation

OFVL-MS utilizes the same parameters setting as in [2]. The threshold of reprojection errors is set to 10 pixels to reject outliers. Moreover, OFVL-MS selects 256 groups 2D pixel coordinates-3D scene coordinates corresponding to refine until convergence for a maximum of 100 iterations.

2.4. Run Time

We utilize 2 NVIDIA Tesla V100 for training each scene and leverage distributed training to realize visual localization across scenes, which takes about 17/31/32 hours when training OFVL-MS18/34/50 on both 7-Scenes and 12-Scenes datasets.

At test time, it takes about $60ms, 70ms$, and $85ms$ for OFVL-MS18, OFVL-MS34, and OFVL-MS50 to localize an image. Scenes coordinates prediction takes about $25 - 45ms$ depending on the network size and pose optimization takes about $30 - 60ms$.

As shown in Tab. 1, OFVL-MS families achieve superior localization performance with fast inference speed.

3. Generalize to New Scenes

We conduct two experiments to verify the capability of OFVL-MS to generalize to New Scenes. EXP1: we utilize the model trained on 12-Scenes and conduct the generalization experiments on 7-Scenes. EXP2: we utilize the model trained on 7-Scenes and conduct the generalization experiments on 12-Scenes. We will not compare the parameters since it has been illustrated in the original paper.

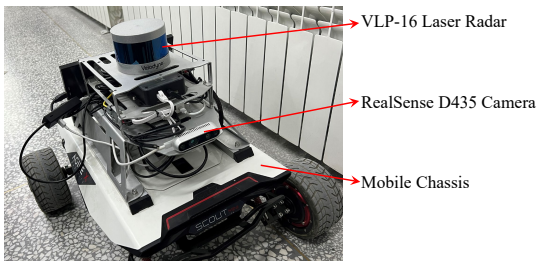


Figure 3: The dataset collection equipment.

For EXP1, we freeze the task-shared parameters trained on 12-Scenes, and add an additional regression layer for each scene of 7-Scenes to predict the scene coordinates. As shown in Tab. 2, despite generalizing to a new scene, OFVL-MS34/50† still outperforms HSCNet and FDANet by $0.82\%/1.93\%$ and $1.95\%/3.06\%$ in terms of 5cm-5° accuracy. Compared with state-of-the-art method VS-Net, OFVL-MS34/50† achieve lower positional and rotational errors. Besides, it is astonishing to find the localization accuracy of OFVL-MS34/50† in certain scenes (e.g. pumpkin, redkitchen, and stairs) is higher than that of OFVL-MS34/50, indicating that low-level geometric information learned from 12-Scenes is beneficial for better scene parsing of 7-Scenes.

For EXP2, we freeze the task-shared parameters trained on 7-Scenes, and add an additional regression layer for each scene of 12-Scenes to predict the scene coordinates. As shown in Tab. 3, it is interesting to find that OFVL-MS18/34/50† achieve better localization performance compared with OFVL-MS18/34/50 in many scenes, further indicating that the related tasks benefit from the shared informative features. Furthermore, OFVL-MS families yield poor localization accuracy when generalizing to 5b scene, resulting in inferior performance overall.

4. LIVL Dataset

As shown in Fig. 3, the dataset collection equipment contains a mobile chassis, a RealSense D435 camera, and a VLP-16 laser radar. LIVL dataset records RGB-D images and corresponding camera poses of four different indoor environments, as shown in Fig. 4.

Specifically, we utilize the ROS system to record RGB images and aligned depth images with corresponding timestamp T_1 through subscribing $/camera/color/image\_raw$ and $/camera/aligned\_depth\_to\_color/image\_raw$ topics provided by RealSense D435 camera. Furthermore, we obtain point clouds with timestamp T_2 through subscribing

		Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs	Average
SCoordNet [14]	**Err.**	0.019,0.63	0.023,0.91	0.018,1.26	0.026, 0.73	0.039,1.09	0.039,1.18	0.037,1.06	0.029,0.98
	Acc.	—	—	—	—	—	—	—	—
HSCNet [6]	**Err.**	0.02,0.7	0.02,0.9	0.01,0.9	0.03,0.8	0.04,1.0	0.04,1.2	0.03,0.8	0.03,0.9
	Acc.	**97.5**	96.7	**100.0**	86.5	59.9	65.5	**87.5**	84.8
FDANet [12]	**Err.**	0.018,0.64	0.018,0.73	0.013,1.07	0.026,0.75	0.036,0.91	0.034,1.03	0.041,1.14	0.026,0.89
	Acc.	95.70	96.10	99.20	88.08	65.65	78.32	62.80	83.69
VS-Net [4]	**Err.**	**0.015,0.5**	0.019,0.8	0.012,0.7	**0.021,0.6**	0.037,1.0	0.036,1.1	0.028,0.8	0.024,0.8
	Acc.	—	—	—	—	—	—	—	—
OFVL-MS18	**Err.**	0.021,0.67	0.018,0.67	0.010,0.56	0.030,0.83	0.033,0.96	0.035,1.02	0.031,0.89	0.025,0.80
	Acc.	96.20	97.55	98.90	81.725	67.15	75.06	79.80	85.20
OFVL-MS34	**Err.**	0.019,0.63	0.017,0.65	**0.008,0.53**	0.027,0.74	0.031,0.93	0.032,1.01	0.027,0.69	0.023,0.74
	Acc.	97.40	96.60	**100.0**	85.58	67.50	77.14	87.40	87.37
OFVL-MS50	**Err.**	**0.015,0.50**	**0.015,0.59**	**0.008**,0.56	0.023,**0.63**	0.030,0.86	0.031,0.99	0.026,0.76	**0.021,0.69**
	Acc.	97.10	**99.40**	**100.0**	**89.53**	68.80	**81.48**	84.70	**88.72**
OFVL-MS18†	**Err.**	0.019,0.64	0.021,0.85	0.011,0.70	0.036,0.96	0.033,0.96	0.037,1.13	0.046,1.30	0.029,0.93
	Acc.	95.90	89.35	93.60	77.075	63.90	69.68	53.60	77.59
OFVL-MS34†	**Err.**	0.018,0.67	0.019,0.67	0.009,0.67	0.026,0.69	0.033,0.92	**0.030,0.97**	0.028,0.67	0.023,0.75
	Acc.	97.95	95.95	98.90	85.42	68.15	77.92	75.80	85.62
OFVL-MS50†	**Err.**	0.017,0.60	0.018,0.70	0.010,0.67	0.024,0.64	**0.028,0.78**	0.031,0.99	**0.022,0.59**	**0.021**,0.71
	Acc.	96.65	97.00	97.10	86.53	**70.90**	79.10	80.00	86.75

Table 2: **Incermental experiments**. † indicates using the task-shared parameters trained on 12-Scenes to conduct generalization experiments on 7-Scenes.

		Kitchen-1	Living-1	Bed	Kitchen-2	Living-2	Luke	Gates 362	Gates 381	Lounge	Manolis	Floor5a	Floor5b	Average
DSAC++ [2]	**Err.**	—	—	—	—	—	—	—	—	—	—	—	—	—
	Acc.	**100**	**100**	99.5	99.5	**100**	99.5	**100**	96.8	95.1	96.4	83.7	95.0	96.8
HSCNet [6]	**Err.**	0.008,0.4	0.011,0.4	**0.009**,0.4	0.007,0.3	0.010,0.4	0.012,0.5	0.010,0.4	0.012,0.6	0.014,0.5	0.011,0.5	0.012,0.5	0.015,0.5	0.011,0.5
	Acc.	**100**	**100**	**100**	**100**	**100**	96.3	**100**	99.1	**100**	**100**	98.8	97.3	99.3
FDANet [12]	**Err.**	0.009,0.30	0.011,0.26	0.013,0.46	0.007,0.27	0.014,0.26	0.019,0.61	0.011,0.38	0.011,0.43	0.015,0.37	0.015,0.35	0.020,0.34	0.026,0.41	0.014,0.37
	Acc.	**100**	**100**	**100**	**100**	**100**	99.2	**100**	**100**	**100**	**100**	**100**	95.7	99.6
OFVL-MS18	**Err.**	0.012,0.39	0.012,0.32	0.012,0.62	0.010,0.39	0.012,0.45	0.015,0.62	0.012,0.54	0.016,0.67	0.012,0.37	0.012,0.53	0.014,0.43	0.016,0.44	0.013,0.48
	Acc.	**100**	**100**	**100**	**100**	**100**	94.7	**100**	97.2	99.7	99.7	99.8	93.1	98.7
OFVL-MS34	**Err.**	**0.003**,0.17	0.007,0.21	0.014,0.40	0.004,0.17	**0.005,0.17**	**0.009,0.32**	**0.007,0.28**	**0.009,0.35**	0.007,**0.16**	0.007,0.28	**0.007,0.25**	**0.009,0.22**	**0.007,0.25**
	Acc.	**100**	**100**	**100**	**100**	**100**	99.20	**100**	**100**	**100**	**100**	**100**	**100**	**99.9**
OFVL-MS50	**Err.**	0.007,0.27	0.006,0.13	0.013,0.43	0.005,0.22	0.005,0.22	0.013,0.48	0.008,0.31	0.011,0.45	0.008,0.24	0.008,0.33	0.010,0.33	0.012,0.29	0.008,0.30
	Acc.	**100**	**100**	**100**	**100**	**100**	96.0	**100**	99.6	**100**	**100**	**100**	97.8	99.5
OFVL-MS18†	**Err.**	0.005,0.29	0.004,0.16	0.018,0.87	**0.003**,0.16	0.005,0.20	0.016,0.55	0.009,0.37	0.012,0.48	**0.006**,0.22	0.006,0.31	0.013,0.51	0.021,0.48	0.009,0.38
	Acc.	**100**	**100**	84.8	**100**	**100**	98.4	**100**	99.9	**100**	**100**	98.5	79.5	96.7
OFVL-MS34†	**Err.**	0.005,**0.16**	**0.003,0.11**	0.012,0.54	**0.003,0.14**	**0.005**,0.18	0.011,0.43	0.014,0.37	0.015,0.52	**0.006**,0.18	**0.005,0.22**	0.013,0.52	0.019,0.43	0.009,0.31
	Acc.	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.7	**100**	**100**	98.6	68.9	97.3
OFVL-MS50†	**Err.**	0.005,0.18	0.005,0.19	**0.009,0.35**	**0.003**,0.15	0.005,0.18	0.013,0.43	0.009,0.36	0.013,0.36	0.009,0.30	0.006,0.24	0.010,0.29	0.017,0.54	0.008,0.29
	Acc.	**100**	96.8	**100**	**100**	**100**	99.8	**100**	**100**	96.9	**100**	**100**	85.7	98.3

Table 3: **Incermental experiments**. † indicates using the task-shared parameters trained on 7-Scenes to conduct generalization experiments on 12-Scenes.

$/velodyne\_points$ topic provided by VLP-16 laser radar. Then, we generate final RGB-D images and corresponding point clouds through aligning T_1 and T_2. Ultimately, We utilize the LiDAR-based SLAM system A-LOAM [13] to compute the ground truth pose.

For each scene, four sequences are recorded, in which three sequences are used for training and one sequence for testing. K544: a room spanning about $12 \times 9m^2$ with 3109 images for training and 1112 images for testing. Floor5: a hall spanning about $12 \times 5m^2$ with 2694 images for training and 869 images for testing. Parking lot1: a parking lot spanning about $8 \times 6m^2$ with 2294 images for training and 661 images for testing. Parking lot2: a parking lot spanning about $8 \times 8m^2$ with 2415 images for training and 875 images for testing. This dataset is challenging for visual localization since it contains substantial lighting, motion blur,

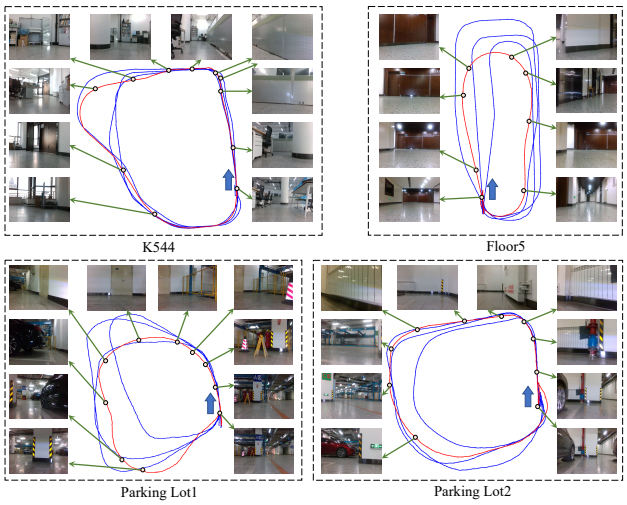


Figure 4: **Our new public dataset LIVL for visual localization.** The dataset comprises four scenes, including K544, Floor5, Parking lot1, and Parking lot2. The blue lines denote training trajectories and the red lines denote test trajectories.

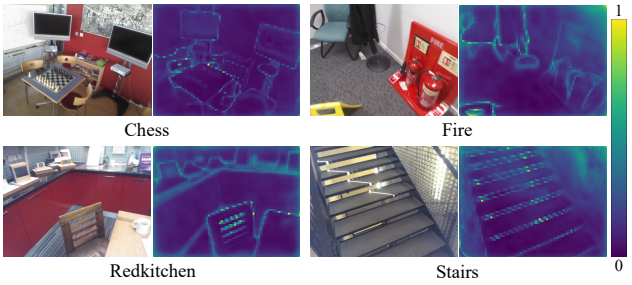


Figure 5: **The visualization of the predicted uncertainty.** OFVL-MS predicts large uncertainty at the object boundaries where the deep discontinuity occurs.

sparse texture, and glass structures.

We utilize 2 GPUs when training OFVL-MS families on each scene of LIVL dataset. We employ the Adamw solver for optimization without weight decay. The initial learning rate is set to 8×10^{-4} with cosine annealing. Furthermore, we do not employ data augmentation and set total iterations as $200k$ with a batch size of 4.

5. Additional Qualitative Results

5.1. Uncertainty Visualization

The uncertainty modeling quantifies the noise coming from data and model [5]. As shown in Fig. 5, we visualize the predicted uncertainty maps, in which the large uncertainty occurs at the object boundaries due to the depth discontinuity, proving that uncertainty modeling is crucial for OFVL-MS to realize precise scene parsing.

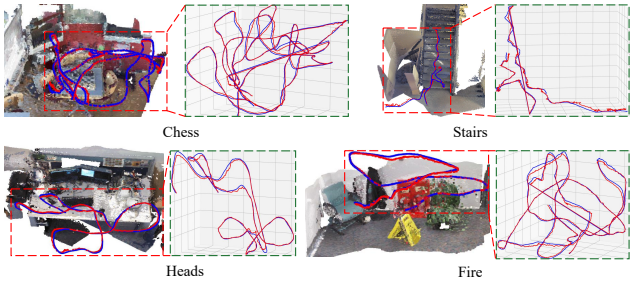


Figure 6: **The visualization of camera trajectories.** The blue lines denote the ground truth trajectories, and the red lines denote the predicted trajectories.

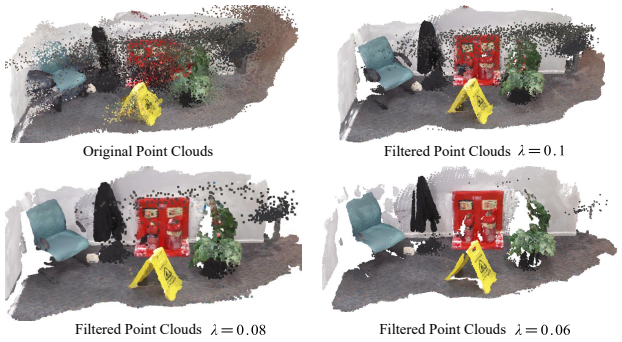


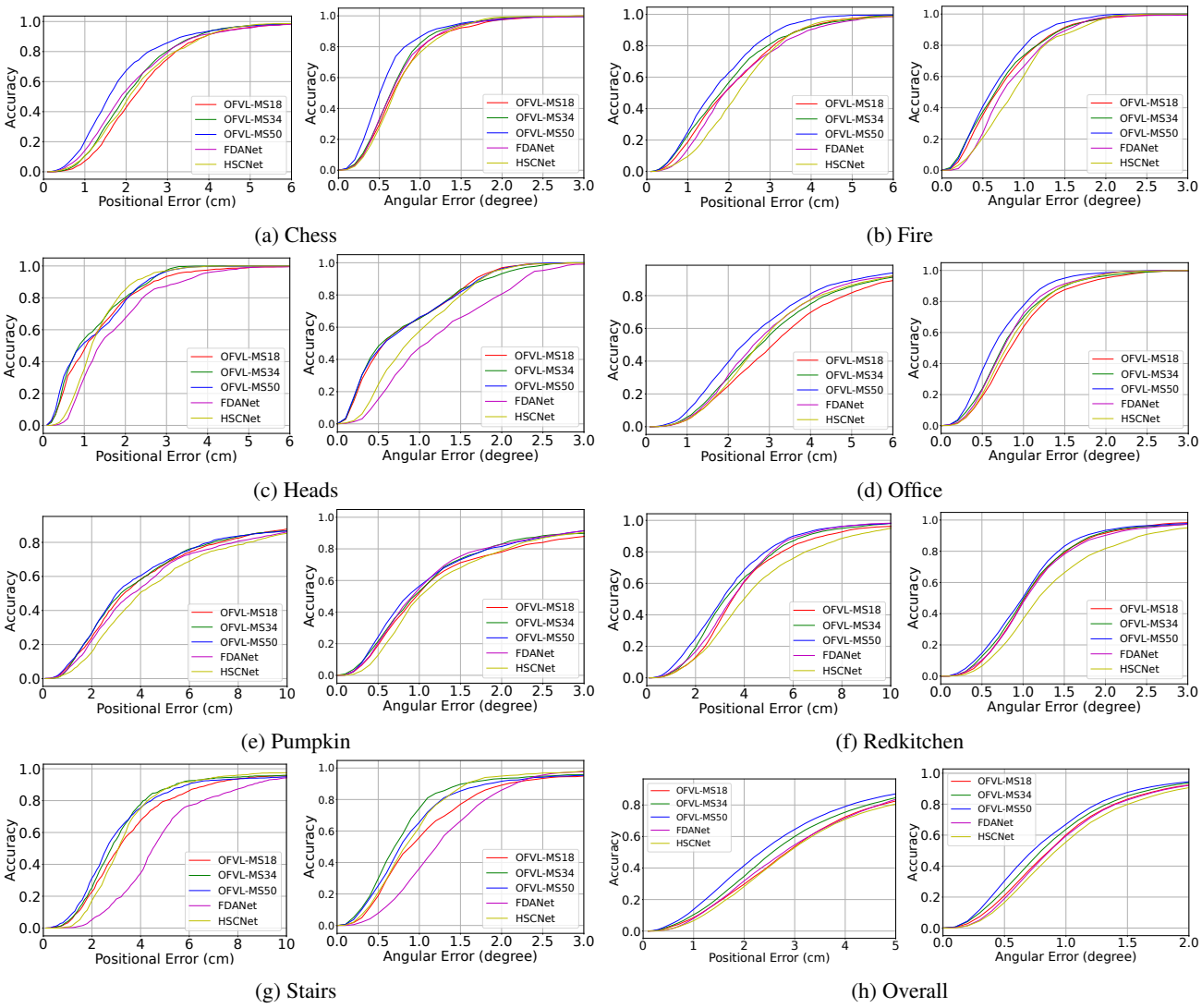
Figure 7: **The visualization of point clouds.** We obtain neater point clouds after filtering out the points with uncertainties larger than λ.

5.2. Trajectories Visualization

As shown in Fig. 6, we visualize the predicted camera trajectories and ground truth to conduct qualitative analysis. It can be observed that the predicted trajectories are very close to the ground truth trajectories, which demonstrates the strong ability of OFVL-MS to realize accurate visual localization.

5.3. Scene Point Clouds Visualization

Following KFNet, OFVL-MS families learn the uncertainties of the predicted scene coordinates to quantify the errors coming from measurement noise and process noise. To validate the effectiveness of learning uncertainties, we filter the point clouds whose uncertainties are larger than λ and visualize the filtered point clouds, as shown in Fig. 7. We can observe that OFVL-MS suppresses the noise and generates neater point clouds with the uncertainties increasing.



(a) Chess (b) Fire

(c) Heads (d) Office

(e) Pumpkin (f) Redkitchen

(g) Stairs (h) Overall

Figure 8: **The cumulative pose errors distribution** of all scenes in 7-Scenes dataset.

5.4. Cumulative Pose Errors Distribution

As shown in Fig. 8, we illustrate the cumulative pose error distribution of all scenes in 7-Scenes dataset. We can observe that OFVL-MS families achieve the best localization accuracy in almost all scenes. Besides, FDANet [12] and OFVL-MS18 realize inferior performance because of the weak capability to differentiate similar image patches caused by limited receptive fields since they both utilize ResNet18 as backbone.

6. Limitations

In this work, we confine the scenes to indoor environments as joint training for many scenes necessitates particularly exact monitoring signals of ground truth scene coordinates; otherwise, shared parameters would not be throughly optimized. The obtained ground truth scene coordinates in outdoor scenes contain a significant number of outliers, making training challenging, especially for joint training. As shown in Tab. 4, OFVL-MS34 achieves certain inferior performance compared with typical structure-based methods and SCoRe based methods. We argue that this phenomenon is induced by inadequate optimization for the task-shared parameters. In our future work, we hope to include outdoor scenes in our work.

References

[1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 1

[2] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proceed-*

Cambridge	Great Court	K.College	Old Hospital	Shop Facade	St M.Church	Average
DSAC++ [2]	0.40, 0.20	0.18, 0.30	0.20, 0.30	0.06, 0.30	0.13, 0.40	0.19, 0.30
Active Search[9]	-	0.42, 0.55	0.44, 1.01	0.12, 0.40	0.19, 0.54	0.29, 0.63
SCoordNet [14]	0.43, 0.20	0.16, 1.29	0.18, 0.29	0.05, 0.34	0.12. 0.36	0.13, 0.32
VS-Net [4]	0.22, 0.10	0.16, 0.20	0.16, 0.30	0.06, 0.30	0.08, 0.30	0.14, 0.24
HSC-Net [6]	0.28, 0.20	0.18, 0.30	0.19, 0.30	0.06, 0.30	0.09, 0.30	0.16, 0.28
PixLoc [8]	0.30, 0.14	0.14, 0.24	0.16, 0.32	0.05, 0.23	0.10, 0.34	0.15, 0.25
HLoc[7]	0.76, 0.30	0.34, 0.40	0.43, 0.60	0.09, 0.40	0.16, 0.50	0.36, 0.31
HLoc+SuperGlue [7]	0.10, 0.07	0.07, 0.11	0.13, 0.24	0.03, 0.14	0.04, 0.12	0.07, 0.14
OFVL-MS34	0.46, 0.31	0.28, 0.53	0.25, 0.49	0.16, 0.56	0.24, 0.61	0.28, 0.50

Table 4: **The median translation and rotation errors of different localization methods on Cambridge dataset**.

ings of the IEEE conference on computer vision and pattern recognition, pages 4654–4662, 2018. 2, 3, 6

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[4] Zhaoyang Huang, Han Zhou, Yijin Li, Bangbang Yang, Yan Xu, Xiaowei Zhou, Hujun Bao, Guofeng Zhang, and Hongsheng Li. Vs-net: Voting with segmentation for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6101–6111, 2021. 3, 6

[5] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016. 4

[6] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11983–11992, 2020. 1, 2, 3, 6

[7] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 6

[8] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3247–3257, 2021. 6

[9] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016. 6

[10] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 1

[11] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 323–332. IEEE, 2016. 1

[12] Tao Xie, Kun Dai, Ke Wang, Ruifeng Li, Jiahe Wang, Xinyue Tang, and Lijun Zhao. A deep feature aggregation network for accurate indoor camera localization. *IEEE Robotics and Automation Letters*, 7(2):3687–3694, 2022. 3, 5

[13] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015. 3

[14] Lei Zhou, Zixin Luo, Tianwei Shen, Jiahui Zhang, Mingmin Zhen, Yao Yao, Tian Fang, and Long Quan. Kfnet: Learning temporal camera relocalization using kalman filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4919–4928, 2020. 3, 6