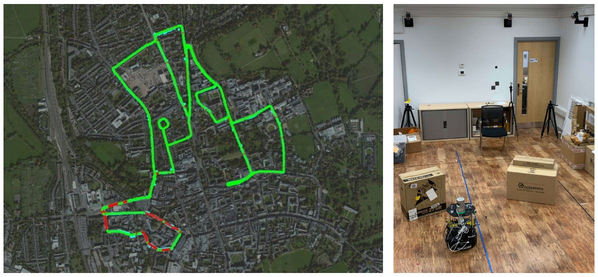
Supplement


Oxford Radar

vReLoc

Figure S1. Overview of the two datasets.

Full-1

Full-2

Full-3

Full-4

Full-6

Full-7

Full-8

Full-9

Figure S2. Visualization of the Oxford Radar dataset. The RGB camera images are for visualization only, and we do not use them in our pipeline.

S1. Supplement Video and Code

We welcome readers to watch our supplemental anonymous video that shows the runtime performance on the Oxford Radar dataset: https://www.youtube.com/watch?v=qplZMOZG-7k

We also welcome readers to run our code at: https://github.com/sijieaaa/HypLiLoc

S2. More About Pose Regression

Visual relocalization refers to the process of determining visual sensor poses from known scene representations such as images, point clouds, key points, features, pose maps, and neural networks.

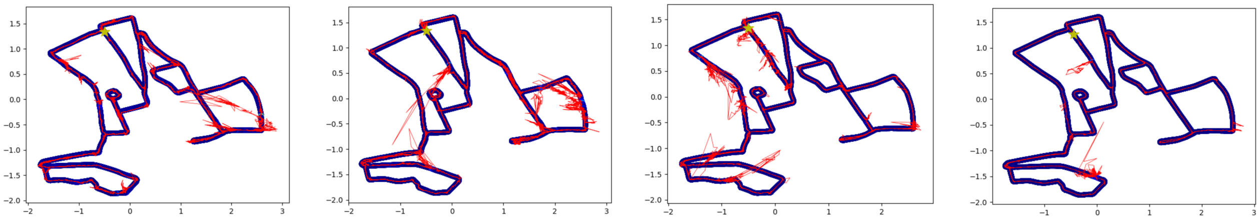
The LiDAR-based pose regression is a type of relocalization pipeline, where the known scene is represented by neural networks in an implicit way, which is different from previous solutions. Given LiDAR point clouds as inputs, the neural network regresses the corresponding poses directly. This formulation is similar to (but operates reversely) the current popular Neural Radiance Fields (NeRFs) [3, 10] that take poses as inputs and outputs the corresponding sensor data. Therefore, the pose regression network can be viewed as another type of neural representation.

We compare typical localization pipelines in Table S1. Structure-based methods achieve the highest global/local pose accuracy, but they suffer from the lowest speed. These types of methods are usually used for offline applications where high-speed inference is not necessary and sufficient computing resources are provided. Visual odometry methods estimate relative poses between frames and serve as a module in the complete SLAM system. The SLAM system provides accurate local pose estimations, but global pose estimations depend on additional information such as loop closure. The retrieval-based methods predict the pose by exhaustively searching the top-matched representations in the database, which is the main cause of high memory consumption and low inference speed. By contrast, pose regression models implicitly represent the scene using neural networks and do not require the database during inference.

S3. Performance After Outlier Filtering

Retrieval-based models [5] require a pre-built database to store candidate scene representations with corresponding poses. Pose regression models do not rely on any database but may suffer from extreme outliers that are far from roads because there is no map or road trajectory information provided. In contrast, the retrieval-based models can only output the locations that are restricted to the trajectory.

This inspires us to explore the possibility of including trajectory information for our regression-based model to further improve its performance. We exclude outlier poses that are far from the database poses over some threshold distances. More specifically, if the regression-based model outputs some pose that is far from all the poses that have been recorded in the trajectory over a threshold distance, we take it as an outlier and discard this output. (Note this



Full-6 Full-7 Full-8 Full-9

Figure S3. More trajectory visualization on the Oxford Radar dataset. The ground truth trajectories are shown in bold blue lines, and the estimated trajectories are shown in thin red lines.

Pipeline	3D Model	Inference Database	Global Acc.	Local Acc.	Speed
Structure-based keypoint matching	need	need	high	high	slow
Visual odometry	-	-	low	high	fast
SLAM	increasing	increasing	low	high	medium
Retrieval	-	need	medium	medium	slow
Pose regression	-	-	medium	medium	fast

Table S1. Comparison of different localization pipelines.

technique is only introduced in the supplement and is not used in the main paper.)

In Table S2, outlier filtering brings 1.25m/0.31° mean error improvements and excludes 27.6% pose estimations. In Table S3, this strategy even supports our network to achieve less than 3m translation error with 16.1% pose estimations dropped out, which is a promising result in the city-wise relocalization task.

In real applications, the exclusion of outliers can be augmented with other techniques like the wheel or LiDAR odometry modules to remedy the dropped poses.

Outlier Thd. (m)	Mean Error (m/°)	Remaining Poses (%)
None	5.82 / 0.97	100.0
25	5.37 / 0.88	99.5
10	5.10 / 0.82	98.6
7	5.03 / 0.79	98.1
5	4.96 / 0.77	97.4
3	4.85 / 0.75	95.2
1	**4.57 / 0.66**	72.4
Difference	(**-1.25 / -0.31**)	(-27.6)
PointNetVLAD [5]	23.59 / 5.87	100.0

Table S2. Performance after filtering pose estimation outliers on Full-8 route of the Oxford Radar dataset.

S4. More Trajectory Visualization

We visualize more of the output trajectories from different routes on the Oxford Radar dataset as shown in Fig. S3.

Outlier Thd. (m)	Mean Error (m/°)	Remaining Poses (%)
None	3.45 / 0.84	100.0
25	3.27 / 0.74	99.5
10	3.18 / 0.71	99.2
7	3.15 / 0.70	99.0
5	3.11 / 0.69	98.8
3	3.05 / 0.68	97.7
1	**2.90 / 0.64**	83.9
Difference	(**-0.55 / -0.20**)	(-16.1)
PointNetVLAD [5]	13.71 / 2.57	100.0

Table S3. Performance after filtering pose estimation outliers on Full-9 route of the Oxford Radar dataset.

S5. Dataset Details

The datasets we used in our experiments include the Oxford Radar dataset and the vReLoc dataset as shown in Fig. S1. For the Oxford Radar dataset, we also visualize the environmental conditions on different routes in Fig. S2. Note that the RGB camera images are for visualization only, and we do not use them in our network. Both of the datasets are available online at:

- https://oxford-robotics-institute.github.io/radar-robotcar-dataset/
- https://github.com/loveoxford/vReLoc

For each dataset, we list the corresponding data split as shown in Table S4 and Table S5.

Scene	Date/Time	Tag	Training	Test
Full-1	2019-01-11-14-02-26	sun	✓	
Full-2	2019-01-14-12-05-52	overcast	✓	
Full-3	2019-01-14-14-48-55	overcast	✓	
Full-4	2019-01-18-15-20-12	overcast	✓	
Full-6	2019-01-10-11-46-21	rain		✓
Full-7	2019-01-15-13-06-37	overcast		✓
Full-8	2019-01-17-14-03-00	sun		✓
Full-9	2019-01-18-14-14-42	overcast		✓

Table S4. Dataset details on the Oxford Radar dataset.

Scene	Tag	Training	Test
Seq-03	static	✓	
Seq-12	walking	✓	
Seq-15	walking	✓	
Seq-16	walking	✓	
Seq-05	static		✓
Seq-06	static		✓
Seq-07	static		✓
Seq-14	walking		✓

Table S5. Dataset details on the vReLoc dataset.

S6. Baseline Models

The baseline models in our comparison include: PointNetVLAD [5], DCP [9], PoseLSTM [6], MapNet [1], AD-MapNet [2], AtLoc+ [7], MS-Transformer [4], PointLoc [8], PosePN [11], PosePN+ [11], PoseSOE [11], and PoseMinkLoc [11].

S7. Codebase

Our codes are developed based on the following repositories:

- https://github.com/ori-mrg/robotcar-dataset-sdk,
- https://github.com/BingCS/AtLoc,
- https://github.com/htdt/hyp_metric.

References

[S-1] Joao F Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8476–8484, 2018. 3

[S-2] Zhaoyang Huang, Yan Xu, Jianping Shi, Xiaowei Zhou, Hujun Bao, and Guofeng Zhang. Prior guided dropout for robust visual localization in dynamic environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2791–2800, 2019. 3

[S-3] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[S-4] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021. 3

[S-5] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4470–4479, 2018. 1, 2, 3

[S-6] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 627–637, 2017. 3

[S-7] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10393–10401, 2020. 3

[S-8] Wei Wang, Bing Wang, Peijun Zhao, Changhao Chen, Ronald Clark, Bo Yang, Andrew Markham, and Niki Trigoni. Pointloc: Deep pose regressor for lidar point cloud localization. *IEEE Sensors Journal*, 22(1):959–968, 2021. 3

[S-9] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019. 3

[S-10] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022. 1

[S-11] Shangshu Yu, Cheng Wang, Chenglu Wen, Ming Cheng, Minghao Liu, Zhihong Zhang, and Xin Li. Lidar-based localization using universal encoding and memory-aware regression. *Pattern Recognition*, 128:108685, 2022. 3