R-SCoRe: Revisiting Scene Coordinate Regression for Robust Large-Scale Visual Localization

Supplementary Material

In this supplementary, we first elaborate on the details in the implementation of R-SCoRe. After that, we show additional results and interpret their meaning. Finally, we reflect on the current limitations of R-SCoRe and discuss future work we consider to improve the performance of localization with SCR further and close the gap to feature matching methods completely.

A. Implementation Details

A.1. Local encodings

Pretrained feature extractor. For Dedode [7], we select the top 5,000 keypoints per image using the Dedode-L detector and extract features using the Dedode-B descriptor. For LoFTR [25], we utilize the CNN feature grid after layer 3, which is 8× smaller than the input image. We use the center of each grid cell as the keypoint.

Local encoding PCA. Before training, we run PCA on the local encodings to reduce their dimensionality to 128 entries. As shown in Fig. 1, reducing the feature dimensionality to 128 dimensions preserves over 90% of the variance for different local encoders [2, 7, 25] on various datasets [11, 23, 24]. To enable efficient computation of the PCA on the GPU, we extract approximately 10 million features via sampling from the training images. In order to incorporate all available features, incremental PCA could be used instead. However, we found that sampling achieves similar performance.

Local encoding buffer. We allocate the training buffer with 32 million 128-dimensional features per GPU, across four GPUs, for a total of 128 million features in half-precision floating-point format.

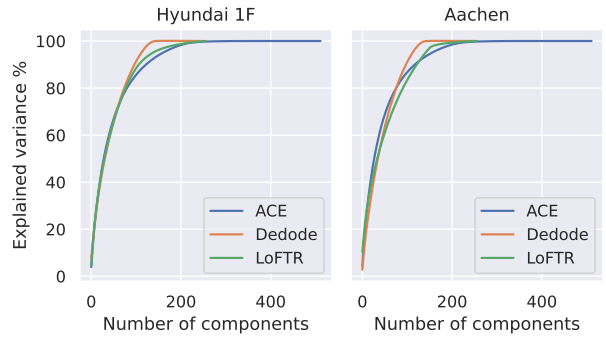

Figure 1. **Local Encoding PCA.** The ratio of variance explained by different numbers of PCA dimensions of local encodings. Reducing the dimensionality to 128 dimensions usually preserves over 90% of the variance.

Image data augmentation. Similar to previous works [2, 27], each image undergoes data augmentation with random resizing, rotation, and color jittering, before we extract local features. Random resizing adjusts the shorter edge, uniformly sampled between 320 and 720 pixels. Rotation is applied uniformly within the range of -15 to 15 degrees, while brightness and contrast are jittered with factors uniformly sampled from [0.9, 1.1].

A.2. Global Encoding Learning with Node2Vec

We use Node2Vec [8] to learn node embeddings for the training images based on the covisibility graph of the scene. Node2Vec performs weighted random walks on the graph and learns embeddings with the Skip-gram [14] objective. The random walk is controlled by two parameters: the return parameter p, and the in-out parameter q. These parameters influence the random walk behavior: the probability of returning to the previous node is proportional to $\frac{1}{p}$, moving farther from the current node is proportional to $\frac{1}{q}$, and staying equidistant to the previous node is proportional to 1.

We use parameters favoring less exploration: $p = 0.25$ and $q = 4$. The embedding dimension is set to 256, aligning with the R^2Former [31] feature dimension used in GLACE [27] to enable a fair comparison in our evaluation.

A.3. Covisibility Graph Construction

We estimate covisibility directly from camera poses using a weighted frustum overlap, following [17, 22]. For each image i, we uniformly sample N_i pixels and unproject each with random depths within $[0, d_{\text{v}}]$, then check visibility $V_k(i \to j)$ from viewing frustum image j. The directed overlap score is computed as:

$$O(i \to j) = \frac{\sum_{k=1}^{N_i} V_k(i \to j)\alpha_k(i,j)}{N_i}, \qquad (1)$$

where $\alpha_k(i,j)$ is the cosine similarity between ray directions. The covisibility graph is constructed by applying a threshold of 0.2 to the harmonic mean of $O(i \to j)$ and $O(j \to i)$. We use maximum viewing frustum depth $d_v = 8$ for indoor scenes and $d_v = 50$ for outdoor scenes.

Recall that Table 6 of the main paper compares covisibility graph construction from frustum overlap to a more

sophisticated version that performs feature matching. For the Aachen Day-Night [23, 24], we observe similar performance and, hence, prefer the simpler algorithm, based on frustum overlap. Here, we shed some light on how covisibility graph construction from feature matching is implemented. First, we perform feature matching between image pairs using SuperPoint [5] and SuperGlue [21], verified against ground truth poses. Second, we consider image pairs covisible that possess 100 or more matched keypoints.

A.4. Network Architecture

We adopt the MLP architecture and position decoder from GLACE [27], enhanced with an additional refinement module. The architecture employs $n = 3$ residual blocks for both the initial output and the refinement module, resulting in a total of six residual blocks. The width of the residual blocks is set to $w = 768$ for the Aachen [23, 24] and Hyundai Department Store [11] 4F datasets, and $w = 1280$ for the Hyundai Department Store [11] B1 and 1F datasets. The hidden width in the residual block is expanded by a factor $m = 2$.

A.5. Training Details

The training is conducted over 100,000 iterations using the AdamW [13] optimizer, with a weight decay set to 0.01. With 4 NVIDIA GeForce RTX 4090, the training takes approximately 4 hours for smaller networks with width $w = 768$ and up to 8 hours for larger networks with width $w = 1280$. For additional acceleration and memory efficiency, our model is trained with mixed precision. Finally, the model weight and bias are saved in a half-precision format to reduce the model size. An exception are the training camera cluster centers, which are saved in single-precision.

B. Additional Results

Encoding	Augmentation	Dept. 1F Val		
R^2Former [31]	Gaussian	42.1	74.5	92.2
R^2Former [31]	Covis	62.0	83.8	94.8
Covis	Covis	72.3	88.7	95.5
Covis	Gaussian	59.1	78.9	90.5

Table 1. **Ablation study of global encodings.** Accuracy at (0.1m, 1°), (0.25m, 2°), and (1m, 5°) thresholds. The isotropic Gaussian data augmentation can also work with our covisibility graph encoding directly, while the best performance is achieved by using our covisibility graph data augmentation.

B.1. Hyundai Department Store Validation Results

The results for the validation set of Hyundai Department Store [11] are shown in Tab. 2. Note that Neumap [26] only provides their result on the validation set. In our main paper we evaluate on the official test set of [11], and, hence, [26] is omitted from the evaluation there. The findings from the validation set are similar to the analysis we conduct in the main paper. While Neumap [26] delivers similar performance to R-SCoRe (using local encodings of Dedode [7]) on 1F and 4F, it significantly trails our method on B1. In addition, R-SCoRe maintains about 6-8$\times$ smaller map sizes and its localization speed appears to be considerably faster than those of Neumap [26].

B.2. Additional Global Encoding Ablation

As shown in Fig. 2, using multiple hypotheses can deliver a significant gain in performance. In general, increasing the number of hypotheses improves the performance, although the gain diminishes when the number of hypotheses becomes larger than 10.

In Tab. 1, we explore whether isotropic Gaussian data augmentation proposed in [27] can also work with our covisibility graph encoding. While we can indeed (*cf*. last row) improve the performance directly, our covisibility graph augmentation delivers better results for either encoding. For the experiment, we use the same standard deviation $\sigma = 0.1$ for the noise as in GLACE [27].

B.3. Network Architecture Ablation

Recall that our model predicts a coarse intermediate and a refined output. Without refinement, our network architecture becomes more similar to the standard SCR pipelines introduced in [2, 27]. To justify our design, we conduct an ablation study using the original network architecture without the refinement module. For a fair comparison, the baseline using the original architecture has the same total depth and width but directly outputs the final coordinate at the end without a coarse to fine refinement. In training, our pipeline with the explicit refinement module achieves a lower median reprojection error and also reduces the training error more rapidly (Fig. 3, left). Similarly, the ratio of inlier training predictions improves more quickly with explicit refinement, but after some time, both pipelines show a similar value (Fig. 3, middle). A closer look at the mean reprojection error (Fig. 3, right) of these inliers shows a significant gap also at the end of training. We conjecture that our pipeline with the explicit refinement module can deliver more accurate predictions. Finally, as shown in Tab. 3, the superior training performance also leads to improved localization accuracy of the pipeline with the explicit refinement module – especially for stricter thresholds. For this evaluation on Aachen Day-Night [23, 24], we employ covisibility graphs computed by frustum overlap.

C. Limitations and Future Work

Throughout our evaluation, we show that R-SCoRe achieves competitive performance on recent large-scale

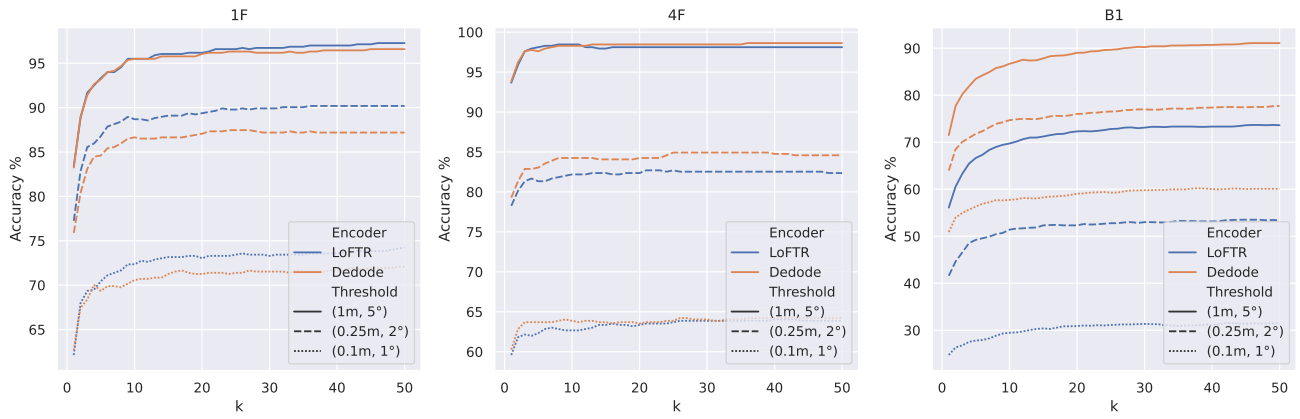

Figure 2. **Comparison of localization accuracy with different number of global hypotheses.** The accuracy at (0.1m, 1°), (0.25m, 2°), and (1m, 5°) thresholds with different numbers of global hypotheses is plotted. Increasing the number of hypotheses improves localization performance, though the performance gain typically plateaus when the number of hypotheses exceeds 10.

	Dept. 1F Validation	Dept. 4F Validation	Dept. B1 Validation
HLoc+D2-Net [6, 20]	(83.2 / 89.2 /94.5) / 398GB	(72.1 / 85.3 / 98.5) / 183GB	(70.2/ 78.0 / 86.1) / 505GB
HLoc+R2D2 [18, 20]	(85.8 / 89.9 / 94.4) / 166GB	(72.6/ 84.6 / 98.3) / 76GB	(71.6/ 78.0 / 86.0) / 210GB
PoseNet [10]	(0.0 / 0.0 / 0.4) / 41MB	(0.0 / 0.0 / 0.2) / 41MB	(0.0 / 0.0 / 0.0) / 41MB
Neumap [26]	(75.5 / 88.2 / 95.8) / 726MB	(70.4 / 85.4 / 99.0) / 431MB	(46.0 /66.5 / 79.8) / 857MB
ESAC (×50) [1]	(49.7 / 71.5 / 84.1) /1.4GB	(45.2 / 69.9 / 85.1) / 1.4GB	(5.4 / 9.1 / 14.2) / 1.4GB
ACE (×50) [2]	(14.2 / 49.9 / 77.8) / 205MB	(29.3 / 80.0 / 96.7) / 205MB	(2.6 / 14.0 / 28.2) / 205MB
GLACE [27]	(4.9 / 24.4 / 53.5) / 42MB	(24.5 / 57.5 / 85.4) / 42MB	(1.0 / 4.5 / 13.8) / 42MB
R-SCoRe (LoFTR* [25])	(72.3 / 88.7 / 95.5) / 127MB	(62.5 / 82.2 / 98.6) / 50MB	(29.4 / 51.3 / 69.6) / 130MB
+ Depth	(74.7 / 89.2 / 95.9) / 127MB	(67.6 / 84.4 / 98.5) / 50MB	(32.4 / 54.4 / 71.0) / 130MB
R-SCoRe (Dedode [7])	(70.6 / 86.6 / 95.5) / 127MB	(63.9 / 84.2 / 98.3) / 50MB	(57.7 / 74.7 / 86.7) / 130MB
+ Depth	(77.1 / 88.6 / 95.6) / 127MB	(68.5 / 84.9 / 98.5) / 50MB	(59.5 / 75.6 / 86.8) / 130MB

Table 2. **Hyundai Department Store Validation Set evaluation.** The percentages of query images within three thresholds: (0.1m, 1°), (0.25m, 2°), and (1m, 5°) and the map size are reported. R-SCoRe achieves competitive accuracy with a small map size. *We use LoFTR [25] outdoor, trained on MegaDepth [12], instead of the indoor model trained on ScanNet [4] for the B1 scene with strong illumination change.

	Aachen Day			Aachen Night		
Original	65.5	82.9	95.3	51.0	78.6	96.9
Refinement	74.8	86.9	96.4	64.3	89.8	96.9

Table 3. **Ablation study of refinement module.** Accuracy at (0.25m, 2°), (0.5m, 5°), and (5m, 10°) thresholds are reported. The explicit refinement module improves the performance, especially for stricter thresholds.

benchmarks, while maintaining very small map sizes. Although we improve on recent SCR methods there still remains a gap – compared to the state-of-the-art feature based methods – in meeting the strictest pose quality thresholds. We conjecture that this limitation may stem from the network's inability to fully generalize and be invariant under extreme input variations, which makes the output coordinate not accurate enough. One potential direction for improvement is integrating our discriminative scene representation with generative models like NeRF [15]. For instance, SCR could provide a robust initialization, which could then be refined by aligning with NeRF-based approaches [3, 28, 29].

Additionally, further reductions in map size could be explored by integrating techniques such as pruning [30], low-rank approximation [19], and quantization [9, 16], which all appear to be applicable to our pipeline in a straightforward manner.

References

[1] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019. 3

[2] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to

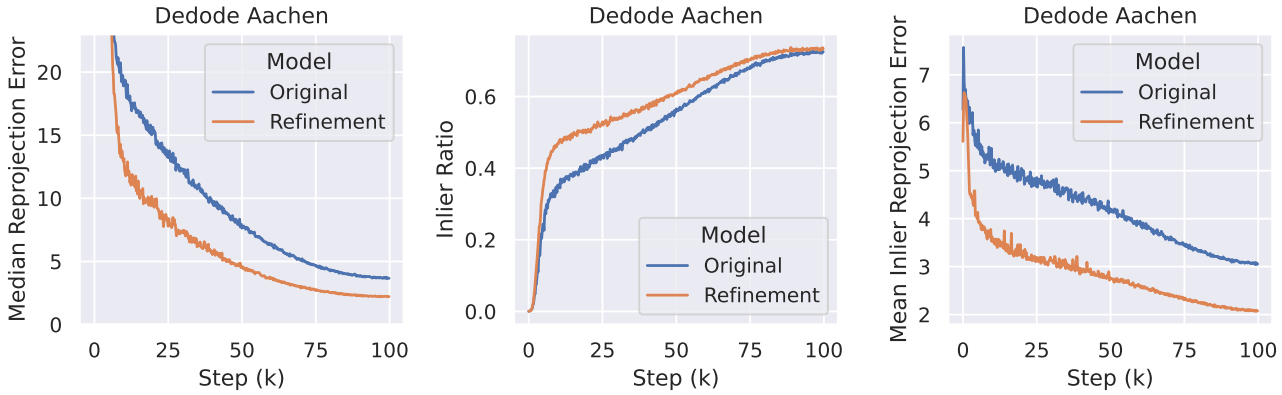

Figure 3. **Ablation study of refinement module.** We present the median reprojection error, the ratio of inlier training predictions with reprojection errors below 10 pixels, and the mean projection error of these inliers.

relocalize in minutes using rgb and poses. In *CVPR*, 2023. 1, 2, 3

[3] Shuai Chen, Yash Bhalgat, Xinghui Li, Jia-Wang Bian, Kejie Li, Zirui Wang, and Victor Adrian Prisacariu. Neural refinement for absolute pose regression with feature synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20987–20996, 2024. 3

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 3

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 2

[6] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *CVPR*, 2019. 3

[7] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. DeDoDe: Detect, Don't Describe — Describe, Don't Detect for Local Feature Matching. In *3DV*. IEEE, 2024. 1, 2, 3

[8] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. 1

[9] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016. 3

[10] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 3

[11] Donghwan Lee, Soohyun Ryu, Suyong Yeon, Yonghan Lee, Deokhwa Kim, Cheolho Han, Yohann Cabon, Philippe Weinzaepfel, Guérin Nicolas, Gabriela Csurka, and Martin Humenberger. Large-scale localization datasets in crowded indoor spaces. In *CVPR*, 2021. 1, 2

[12] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 3

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[14] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 1

[15] 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. In *ECCV*, 2020. 3

[16] Antonio Polino, Razvan Pascanu, and Dan-Adrian Alistarh. Model compression via distillation and quantization. In *6th International Conference on Learning Representations*, 2018. 3

[17] Anita Rau, Guillermo Garcia-Hernando, Danail Stoyanov, Gabriel J Brostow, and Daniyar Turmukhambetov. Predicting visual overlap of images through interpretable non-metric box embeddings. In *ECCV*, 2020. 1

[18] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems*, 2019. 3

[19] Roberto Rigamonti, Amos Sironi, Vincent Lepetit, and Pascal Fua. Learning separable filters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2754–2761, 2013. 3

[20] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 3

[21] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2

[22] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson,

Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In *ECCV*, 2022. 1

[23] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *BMVC*, 2012. 1, 2

[24] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *CVPR*, 2018. 1, 2

[25] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 1, 3

[26] Shitao Tang, Sicong Tang, Andrea Tagliasacchi, Ping Tan, and Yasutaka Furukawa. Neumap: Neural coordinate mapping by auto-transdecoder for camera localization. In *CVPR*, 2023. 2, 3

[27] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Glace: Global local accelerated coordinate encoding. In *CVPR*, 2024. 1, 2, 3

[28] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IROS*, 2021. 3

[29] Qunjie Zhou, Maxim Maximov, Or Litany, and Laura Leal-Taixé. The nerfect match: Exploring nerf features for visual localization. *European Conference on Computer Vision*, 2024. 3

[30] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017. 3

[31] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng Wang. R2former: Unified retrieval and reranking transformer for place recognition. In *CVPR*, 2023. 1, 2