Long-term Visual Map Sparsification with Heterogeneous GNN Supplementary Material

Ming-Fang Chang [1], Yipu Zhao[2], Rajvi Shah[2], Jakob J. Engel[2], Michael Kaess[1], and Simon Lucey[3]

[1]Carnegie Mellon University [2]Meta Reality Labs Research [3]The University of Adelaide

Abstract

In this supplementary material, we provide the details that were not included in the main paper due to space limitation. We will go through the details of the conventional ILP method, the full recall curves of the different configurations compared in the main paper, and more map graph statistics and visualizations.

1. Details of Conventional ILP Method

Here we describe the details of the ILP method we used to generate L_{gt} and as baselines [1–3]. Given N_p 3D points and N_m key frame images in the map, letting $\mathbf{x}$ be the binary point selection vector, we can formulate the following ILP problem:

$$\begin{aligned} minimize\ & \mathbf{q}^T\mathbf{x} + \lambda \mathbf{1}^T\zeta \\ s.t.\ & \mathbf{A}\mathbf{x} + \zeta \geq b\mathbf{1} \\ & \sum_{i=1}^{N_p} \mathbf{x}_i = n_{desired} \\ & \mathbf{x} \in \{0,1\}^{N_p} \\ & \zeta \in \{\{0\} \cup \mathbb{Z}^+\}^{N_m}, \end{aligned} \tag{1}$$

where $\mathbf{q}$ is an assigned weighting vector, $\mathbf{A} \in \mathbb{R}^{N_m \times N_p}$ is the visibility matrix, ζ is the slack variable, b is a tunable variable indicating the desired minimum number of observable 3D points for each map key frame ($b = 30$ was used in the paper as in [1]), and $n_{desired}$ is the desired total 3D point number. The weighting vector $\mathbf{q}$ is computed from the observation count (the number of times a 3D point is matched by a 2D key point in the map building history) of each 3D point. Let c_i denote the observation count of a 3D point with index i, the corresponding q_i for this point is assigned as $max(c_1, c_2, \ldots, c_{N_p}) - c_i$ as in [3]. The idea is to assume the well-observed points in the past to play a more important role in future localization, and decrease the weights of these points in the minimization cost. In a dynamic environment where the physical structures of the world changes, a point that was observed for more times in the past might not exist or as important in the future, even if it has a robust feature descriptor.

2. Full Recall Curves

We present the full recall curves for all the methods compared, for which we only provided linearly interpolated and averaged numbers in Tab. 2 of the main paper due to space limitation. In Fig. 1 we show the full recall curves of g_2 layer using GATConv, GraphConv, and SAGEConv. In Fig. 2 we show the full curves of the proposed combined loss and without either $\mathcal{L}_{KC}$ or $\mathcal{L}_{BCE}$. One small difference between our GATConv and the original version in [4] is the way of merging multi-head attentions. In [4] the authors computed the mean of all the heads and added an nonlinear layer, while we performed simple summation. Empirically we found simple summation outperforms the original version, as shown in Fig. 3.

3. Map Graph Statistics

To better describe the scale of the localization problem we are solving, we listed the number of nodes in each testing map graph and the corresponding processing time (on CPU) for the proposed GNN to process the whole map graph and generate scores. Note that number of images in the map graph is slightly less than Sec. 4 in the main paper because some images were discarded during the mapping process. To perform efficient training on these large map graphs, we iterated through extracted local map graphs during training on GPU, and computed the scores of the whole map graphs at once on CPU with trained weights.

4. More Map Graph Visualization

Here we visualize more samples of local map graphs in Fig. 5. Each of the local map graph was extracted by first sampling a map image, and trace the connected edges to

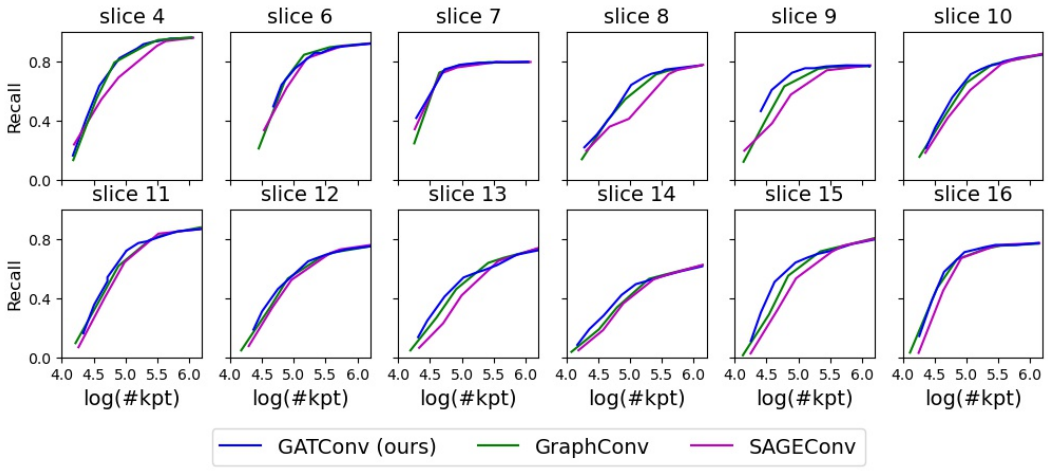

Figure 1. The full recall curves of different g_2 layers.

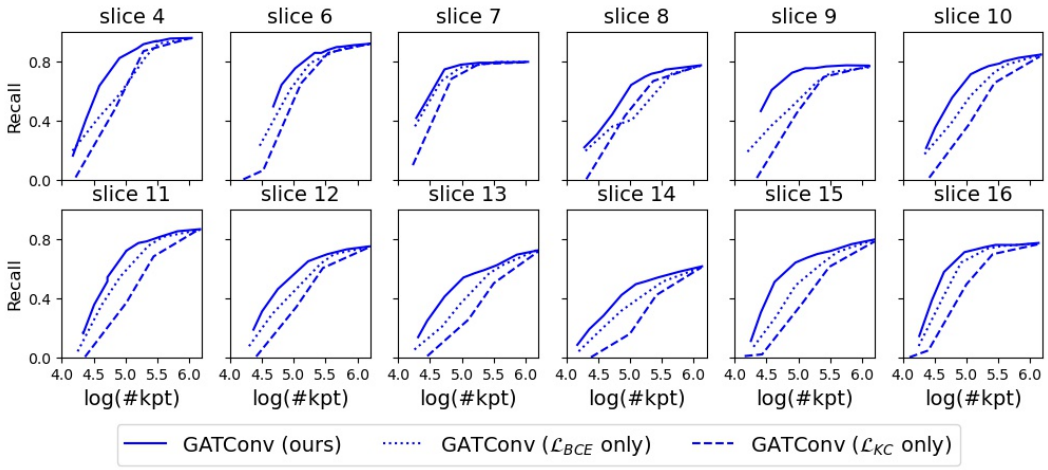

Figure 2. The full recall curves of using different losses.

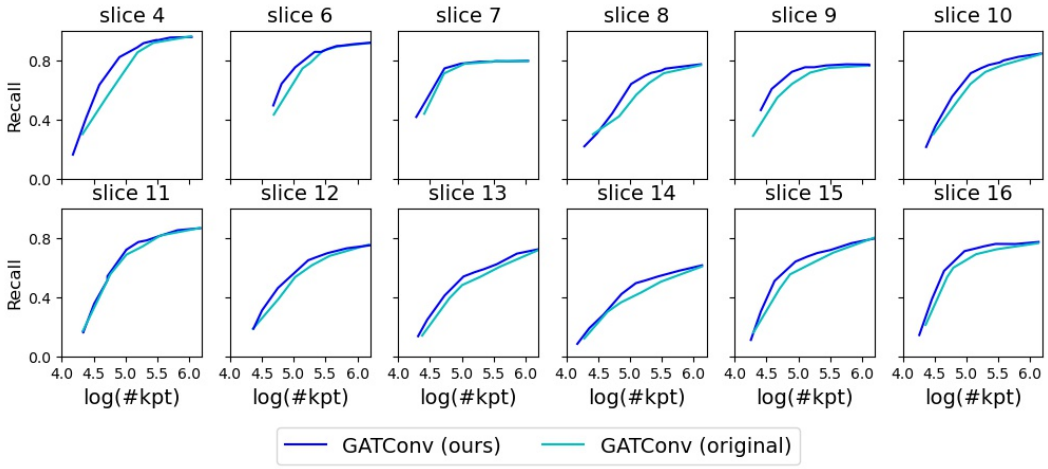

Figure 3. The full recall curves of different GATConv versions.

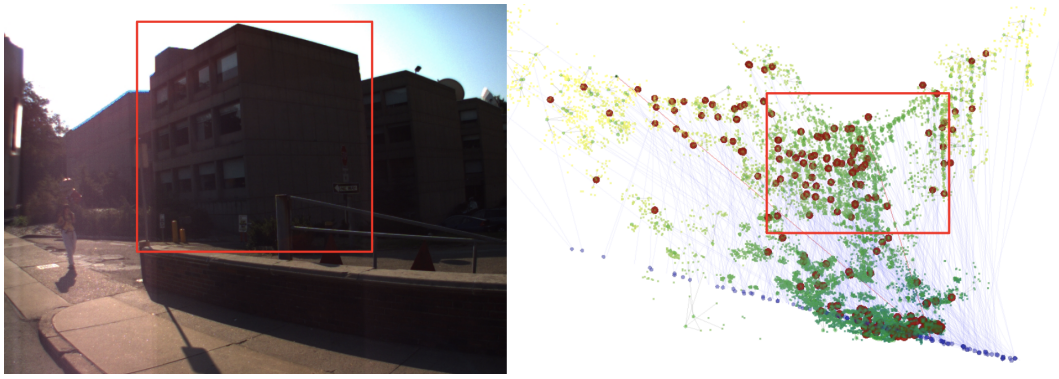

Figure 4. Another visualization of point selection, where the $\mathcal{V}_p$, $\mathcal{V}_m$, $\mathcal{E}_v$, $\mathcal{E}_n$ are shown by green/yellow dots, blue dots, blue lines, and gray lines. The edges were downsampled for better visualization clarity. Assigning $n_{desired} = 200$, our method selects the red dots from the green dots (the $\mathcal{V}_p$ connected to the key points in the sampled map image). On the left, we show the corresponding image to the local map graph. The corresponding parts are shown by red and orange boxes.

Table 1. Test map graph sizes stats and the processing time. The graph size is represented by the number of nodes and edges. Each $\mathcal{V}_k$ carries an R2D2 descriptor with dimension 128.

slice	time (ms)	# $\mathcal{V}_p$	# $\mathcal{V}_m$	# $\mathcal{V}_k$	# $\mathcal{E}_c$	# $\mathcal{E}_k$	# $\mathcal{E}_n$
4	3,188.20	326,797	1,260	2,072,834	2,072,834	2,072,834	3,267,970
6	5,261.06	508,844	1,642	3,632,027	3,632,027	3,632,027	5,088,440
7	2,982.98	320,258	1,196	2,034,125	2,034,125	2,034,125	3,202,580
8	3,440.97	377,380	1,423	2,465,051	2,465,051	2,465,051	3,773,800
9	3,563.63	387,988	1,154	2,531,802	2,531,802	2,531,802	3,879,880
10	4,023.90	433,398	1,356	2,830,219	2,830,219	2,830,219	4,333,980
11	3,930.64	421,625	1,394	2,859,143	2,859,143	2,859,143	4,216,250
12	4,143.14	439,425	1,454	2,960,501	2,960,501	2,960,501	4,394,250
13	4,395.27	456,772	1,409	3,275,294	3,275,294	3,275,294	4,567,720
14	3,806.34	427,022	1,392	2,629,698	2,629,698	2,629,698	4,270,220
15	4,474.99	469,165	1,418	3,156,230	3,156,230	3,156,230	4,691,650
16	3,527.82	370,006	1,319	2,617,829	2,617,829	2,617,829	3,700,060

acquire all the information needed by our GNN as described in the main paper. Besides the map graphs, we can also observe the variety of objects in this dataset from Fig. 5.

References

[1] Marcin Dymczyk, Simon Lynen, Michael Bosse, and Roland Siegwart. Keep It Brief: Scalable Creation of Compressed Localization Maps. In *IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2015. 1

[2] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual-inertial localization revisited. *Int. J. of Robotics Res.*, 2020. 1

[3] Hyun Soo Park, Yu Wang, Eriko Nurvitadhi, James C. Hoe, Yaser Sheikh, and Mei Chen. 3D Point Cloud Reduction Using Mixed-Integer Quadratic Programming. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2013. 1

[4] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *Int. Conf. Learn. Represent.*, 2018. 1

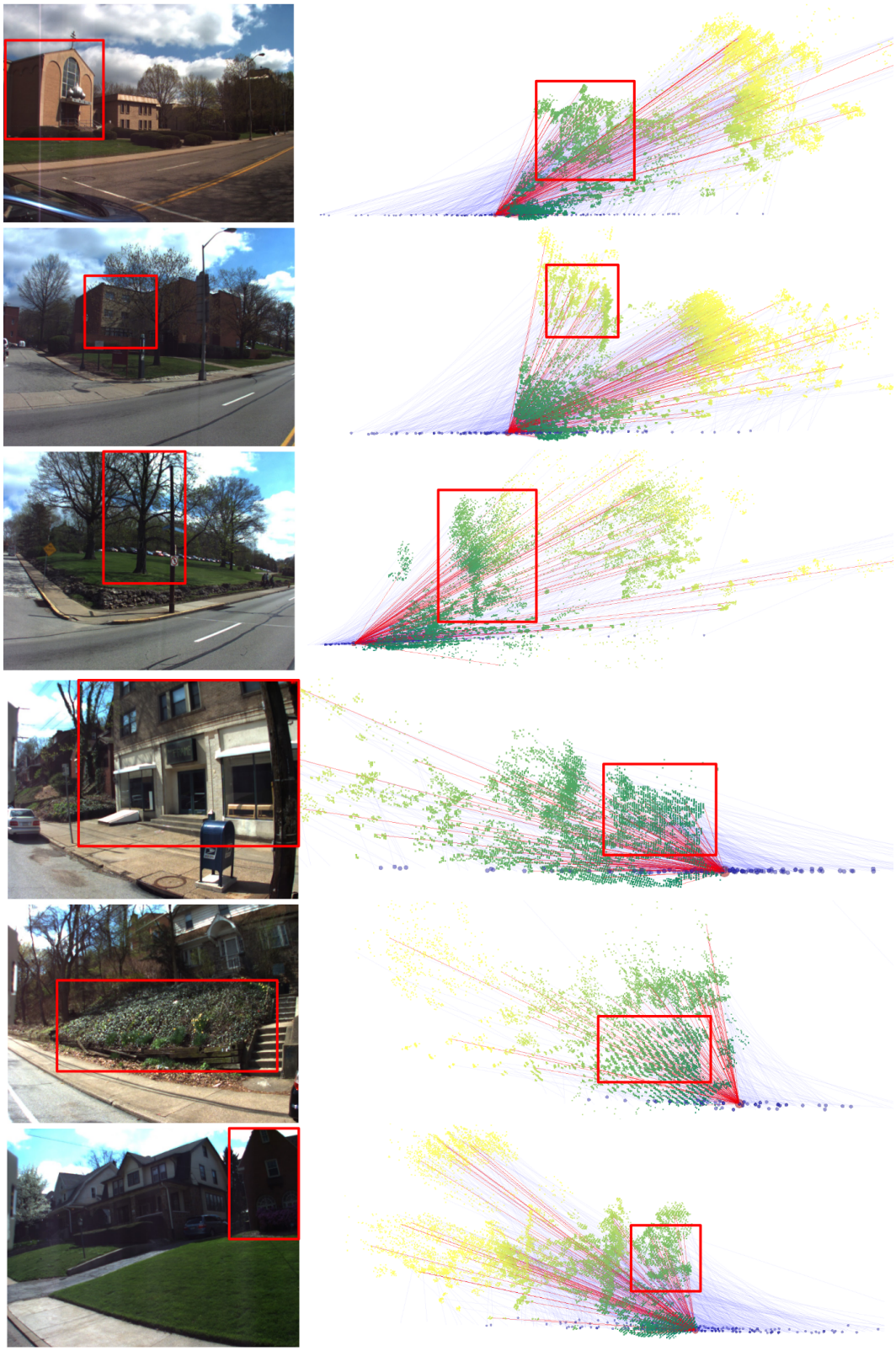

Figure 5. More local graph visualizations. The upper three graphs were extracted from camera 0, and the lower three graphs were extracted from camera 1. The corresponding parts of the images and the graphs are shown by red boxes. The $\mathcal{V}_p$, $\mathcal{V}_m$, $\mathcal{E}_v$ from the map are shown by green/yellow dots, blue dots, and blue lines. The $\mathcal{E}_v$ connected to the sampled $\mathcal{V}_m$ are shown by red lines. These map graphs provide all the information needed to predict scores for the $\mathcal{V}_p$ connected by the red $\mathcal{E}_v$ set.