Text2Loc: 3D Point Cloud Localization from Natural Language

Supplementary Material

A. Overview

In this supplementary material, we provide more experiments on the KITTI360Pose dataset [12] to demonstrate the effectiveness of our Text2Loc and show more insights we gathered during the development. We first present thorough ablation experiments to study the impact of the proposed CCAT on the fine localization performance in Sec. B. In Sec. C, we provide qualitative results of top-3 candidate submaps retrieved and localization performance when changing one sequence in the query textural descriptions. Next, we describe implementation details about our network architecture in Sec. D and analysis of the proposed PMC module in Sec. E. Finally, Sec. F shows more visualizations of point cloud localization from text descriptions.

B. More analysis of Cascaded Cross-Attention Transformers

In this section, we first explore the performance of different numbers of Cascaded Cross-Attention Transformers (CCAT) in our fine localization network. We further provide a comparison to study the difference between our CCAT and Hierarchical Cross-Attention Transformer (HCAT) in [37].

Number of CCAT. We insert CCAT one by one before the MLP layer in Text2Loc. '0' means using a single Cross Attention Transformer (CAT) to fuse text and 3D point cloud features. Table 7 shows the localization performance of our Tex2Loc with different numbers of CCAT units. As seen from the table, Text2Loc achieves the best performance with 2 CCAT units. When the number expands to 3, the performance degrades. This implies that the text-submap feature fusion is sufficient with fewer CCAT units. On the other hand, when the number is set to 1, the performance decreases. Therefore, we set the fixed number of CCAT as 2 in our network.

Difference with HCAT. Recent work CASSPR [37] has explored the integration of 3D point-wise features with voxelized representations through a designed Hierarchical Cross-Attention Transformer (HCAT). In HCAT, two parallel Cross Attention Transformers (CAT1 and CAT2) process inputs from different branches (point and voxel), each serving as query and key respectively. In contrast, our Cascaded Cross-Attention Transformer (CCAT) employs a sequential, cascaded structure to merge text and point cloud cross-modal information. Notably, in our CCAT, the second CAT utilizes the output of the first CAT as its key and value, distinguishing it from the parallel architecture of HCAT. Table. 8 presents a performance comparison of different modules within our Text2Loc architecture. Utilizing the proposed CCAT, we observed an approximate 4% increase in retrieval accuracy at top 10 on the test set. This table demonstrates a consistently superior performance of our CCAT compared to the HCAT used in [37].

Number of CCAT	Localization Recall ($\epsilon < 5m$) ↑					
	Validation Set			Test Set		
	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$
0	0.28	0.57	0.66	0.26	0.51	0.60
1	0.36	0.67	0.77	0.32	0.59	0.69
2	**0.37**	**0.68**	**0.77**	**0.33**	**0.61**	**0.71**
3	0.35	0.67	0.77	0.32	0.59	0.69

Table 7. Localization performance for Text2Loc with different numbers of CCAT on the KITTI360Pose benchmark. '0' means using a single Cross Attention Transformer (CAT) to fuse text and 3D point cloud features.

Methods	Localization Recall ($\epsilon < 5m$) ↑					
	Validation Set			Test Set		
	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$
HCAT [37]	0.35	0.66	0.75	0.32	0.59	0.68
CCAT (Ours)	**0.37**	**0.68**	**0.77**	**0.33**	**0.61**	**0.71**

Table 8. Performance comparison of different modules within our Text2Loc architecture on the KITTI360Pose benchmark.

Motivation of CCAT. The motivation for the CCAT module in fine localization arose from the challenge of target position regression based on the text descriptions. Encoding accurate textual features is crucial for regression since the model directly predicts target positions, without any text-instance matcher. We thus design a cascade structure to enhance text features with the information from retrieved point clouds. The HCAT [37] module, in contrast, aims to compensate for the quantization losses for the LiDAR-based place recognition task. HCAT should ensure that each branch is useful in isolation, thus preventing one branch from dominating over the other.

C. Visualization of robustness analysis

Fig. 7 visualizes some qualitative results for Sec. 6.4. For each instance, we display the original query text descriptions along with the top 3 retrieved submaps and their final predicted locations at the top, followed by modified queries (highlighted in red) and their results at the bottom. In the first example, we cannot find the positive submaps in the top-3 matches, leading to a complete localization failure. In the second example, even though we identify the positive

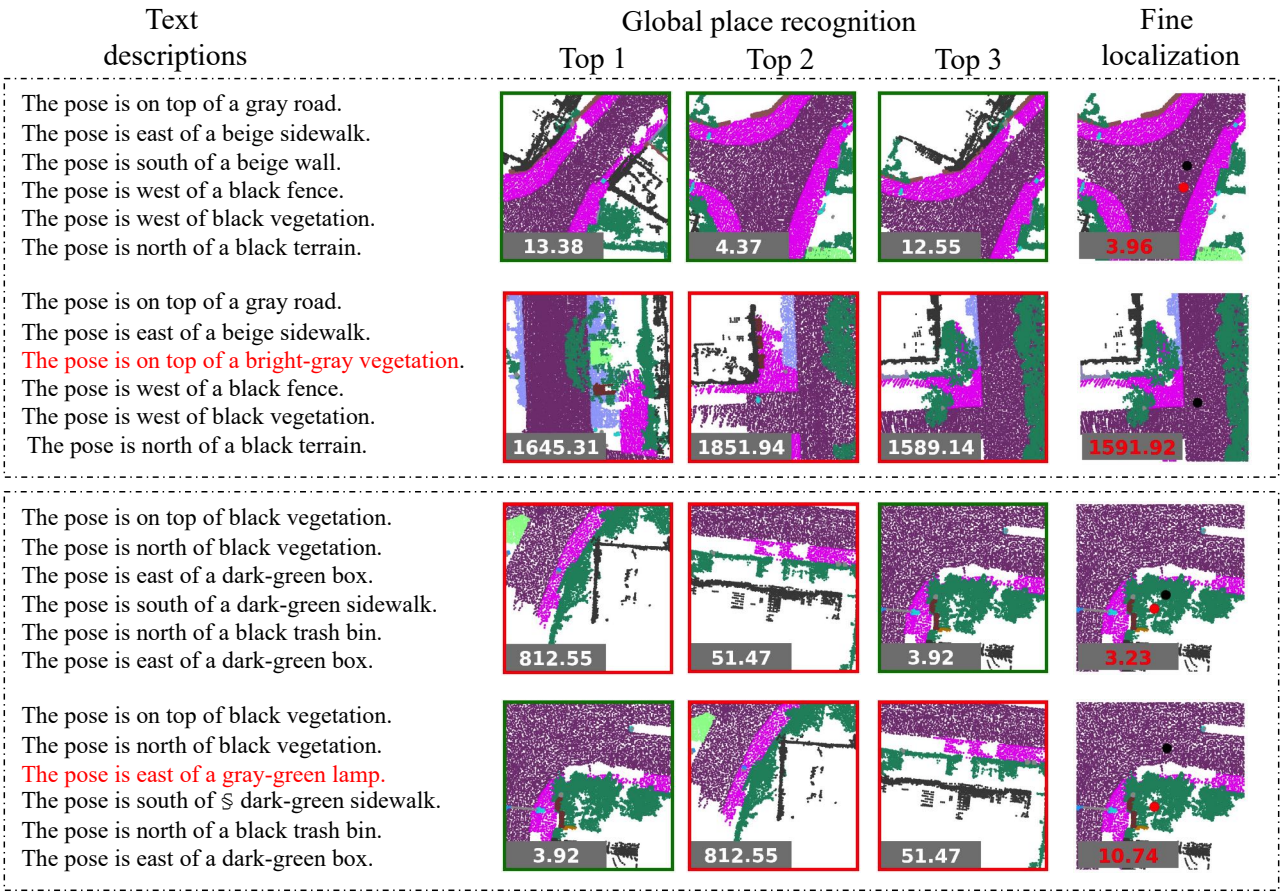

Figure 7. Robust analysis of our Text2Loc on the KITTI360Pose Benchmark. We present the top-3 retrieved submaps in global place recognition and the final predicted location for both the original query text descriptions and the modified queries (in red).

submaps in the global place recognition, the exact localization is still off. The results are consistent with our expectation that accurate text embedding is essential for predicting the target location in fine localization.

D. Implementation Details

We train the model with Adam optimizer for the text-submap global place recognition with a learning rate (LR) of 5e-4. The model is trained for a total 20 epochs with batch size 64, and we follow a multi-step training schedule wherein we decay LR by a factor of 0.4 at each 7 epoches. The temperature coefficient τ is set to 0.1. We consider each submap to contain a constant 28 object instances. The intra- and inter-text encoder in the text branch has 1 encoder layer respectively. We utilize PointNet++ [20] from [12] to encode every individual instance within the submap. In all quantitative results relating to global place recognition, we adopt the definition of the ground truth (GT) submap as [12], where it refers to the submap in the database that contains textual descriptions of targets, with its center point closest to the target. For the fine localization network, we train the model with an LR of 3e-4 for 35 epochs with batch size 32. To make a fair comparison, we set the embedding dimension for both text and submap branch as 256 in global place recognition and 128 in fine localization. The code is available for reproducibility.

Transformer in global place recognition. Formally, each transformer with max-pooling in the proposed intra- and inter-text encoder can be formulated as follows:

$$\begin{aligned} \mathbf{F}_T &= \text{Max-pooling} \circ \text{Transformer}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \\ &= \text{Max-pooling} \circ \left[\widetilde{\mathbf{F}}_T + \text{FFN}\left(\widetilde{\mathbf{F}}_T\right)\right], \\ \widetilde{\mathbf{F}}_T &= \mathbf{Q} + \text{MHSA}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right), \end{aligned} \tag{6}$$

where $\mathbf{Q} = \mathbf{K} = \mathbf{V} = F_t \in \mathbb{R}^{N_t \times d}$ represent the query, key, and value matrices.

Within the MHSA layer, self-attention is conducted by projecting $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ using h heads, with our choice being $h = 4$. More precisely, we initially calculate the weight

matrix using scaled dot-product attention [32], as in Eq. 7:

$$\text{Attention}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\text{Softmax}\left(\frac{\mathbf{Q}\cdot\mathbf{K}^{T}}{\sqrt{d_k}}\right)\mathbf{V}, \quad (7)$$

Subsequently, we compute the values for the h heads and concatenate them together as follows:

$$\text{Multi-Head}(\mathbf{Q},\mathbf{K},\mathbf{V})=[\,\text{head1},\ldots,\text{head}h]\,\mathbf{W}^{O}, \quad (8)$$

$$\text{head}i=\text{Attention}\left(\mathbf{Q}\mathbf{W}i^{Q},\mathbf{K}\mathbf{W}i^{K},\mathbf{V}\mathbf{W}i^{V}\right), \quad (9)$$

where $\mathbf{W}_{i}^{Q,K,V,O}$ denote the learnable parameters.

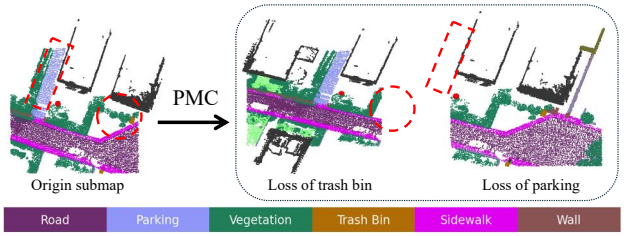

Figure 8. Visualization of lost instances due to our PMC.

E. Analysis of PMC module

PMC can be seen as a data augmentation. However, this augmentation is not suitable for the previous text-instance matcher in Text2Pos [12] and RET [33] since PMC can lead to the loss of object instances in certain submaps (see Fig. 8 above); thereby, solely integrating the PMC into Text2Pos results in performance degradation. Conversely, adding more training submaps by PMC benefits our Text2Loc since we adopt a matching-free strategy without any text-instance matches.

F. More visualization results

In this section, we visualize more examples of correct point cloud localization from text descriptions and failure cases in Fig. 9. For (a) and (b), Text2Loc successfully retrieves all positive submaps within the top-3 results during global place recognition. We observe that these top-3 retrieved submaps display a high degree of semantic similarity to both the ground truth and each other. In cases of (c) - (e), despite some of the top-3 submaps being negatives retrieved by our text-submap place recognition, Text2Loc effectively localizes the text queries within a $5\,\text{m}$ range after applying the fine localization network. It demonstrates our fine localization network can improve the localization recall, which turns such wrong cases in place recognition into a successful localization.

We also present some failure cases where all retrieved submaps are negative. For example, in case (g), the query text description contains an excessive number of objects of the same category 'Pole'. This description ambiguity poses a significant challenge to our place recognition network, leading to the retrieval of incorrect submaps. In the future, We hope to investigate more precise and accurate text descriptions, like integrating specific landmark information, including street names, zip codes, and named buildings, into text-based localization networks.

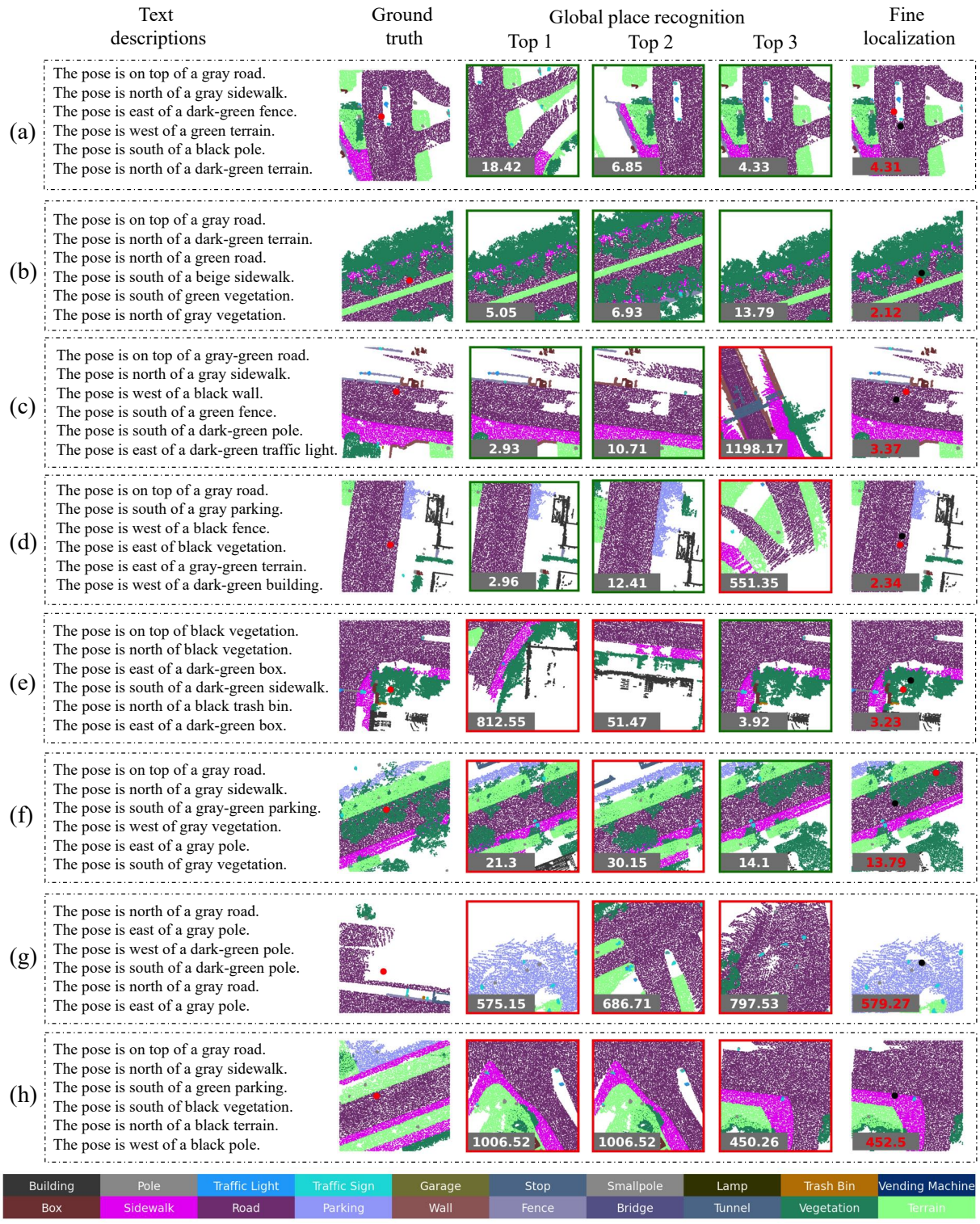

Figure 9. Qualitative localization results on the KITTI360Pose dataset: In global place recognition, the numbers in top3 retrieval submaps represent center distances between retrieved submaps and the ground truth. Green boxes indicate positive submaps containing the target location, while red boxes signify negative submaps. For fine localization, red and black dots represent the ground truth and predicted target locations, with the red number indicating the distance between them.