Semantic-guided Camera Ray Regression for Visual Localization

Supplementary Material

Algorithm 2 R-RANSAC Algorithm

Input: $\{\hat{d}_i\}_{i=1}^{N}$, T_R^E (inlier residual threshold), T_R^N (inlier number threshold), N_R (number of samples), It_R^{max} (maximum iteration number);
Output: $\hat{R} \in \mathbb{R}^{3\times3}$;

```
# Initialization
It_R ← 0; # iteration counter
I_best ← ∅; # best inlier set
N_best^in ← 0; # best inlier number
# Main RANSAC Iteration
repeat
    Randomly sample {d̂_i}_i^{N_R} from {d̂_i}_i^N;
    Compute rotation R_temp from {d̂_i}_i^{N_R} as Eq. (3);
    # Collect inliers
    N_temp^in ← 0, # temporary inlier number
    I_temp ← ∅; # temporary inlier set
    for d̂_i ∈ {d̂_i}_i^N do
        Get residual: E_i^{R_temp} = ||d̂_i, R_temp d_i^c||_2^2;
        if E_i^{R_temp} < T_R^E then
            N_temp^in ← N_temp^in + 1;
            d̂_i --into--> I_temp;
        end if
    end for
    if N_temp^in > N_best^in then
        N_best^in ← N_temp^in;
        I_best ← I_temp;
    end if
    if N_best^in > T_R^N · N then
        break
    end if
    It_R ← It_R + 1;
until It_R ≥ It_R^max
# Inlier Refinement
Compute R̂ using all ray directions in I_best as Eq. (3);
```

Algorithm 3 C-RANSAC Algorithm

Input: $\hat{\mathcal{R}}$, T_c^E (inlier residual threshold), T_c^N (inlier number threshold), N_c (number of samples), It_c^{max} (maximum iteration number);
Output: $\hat{c} \in \mathbb{R}^{3\times1}$;

```
# Initialization
It_c ← 0; # iteration counter
I_best ← ∅; # best inlier set
N_best^in ← 0; # best inlier number
# Main RANSAC Iteration
repeat
    Randomly sample {r̂_i}_i^{N_c} from R̂;
    Compute rotation c_temp from {r̂_i}_i^{N_R} as Eq. (4);
    # Collect inliers
    N_temp^in ← 0, # temporary inlier number
    I_temp ← ∅; # temporary inlier set
    for r̂_i = [d̂, m̂] ∈ {r̂_i}_i^N do
        Get residual: E_i^{c_temp} = ||c_temp × d̂ − m̂||_2^2;
        if E_i^{c_temp} < T_c^E then
            N_temp^in ← N_temp^in + 1;
            r̂_i --into--> I_temp;
        end if
    end for
    if N_temp^in > N_best^in then
        N_best^in ← N_temp^in;
        I_best ← I_temp;
    end if
    if N_best^in > T_c^N · N then
        break
    end if
    It_c ← It_c + 1;
until It_c ≥ It_c^max
# Inlier Refinement
Compute ĉ using all rays in I_best as Eq. (4);
```

A. Details of RC-RANSAC

We provide more details about the proposed RC-RANSAC algorithm, including the specific algorithm process and the choice of parameters.

Algorithm flow. The detailed algorithm steps of *R-RANSAC* and *C-RANSAC* are described in Algorithm 2 and Algorithm 3. These two algorithms share many similar steps, and only the residual calculation part is different. The *R-RANSAC* utilizes the L2 distance between direction vectors as the residual. On the other hand, the *C-RANSAC* adopts the L2 distance between the moment vectors as the residual.

Parameter Settings. For *R-RANSAC* in Algorithm 2, we set the inlier residual threshold (°) as $T_R^E = 0.5$, the inlier number threshold as $T_R^N = 0.8$, and the sample number is $N_R = 100$. Finally, the maximum iteration number is $\mathsf{It}_R^{max} = 20$. On the other hand, for *C-RANSAC* in Algorithm 3, we set the inlier residual threshold (cm) as $T_c^E = 5$, the inlier number threshold as $T_c^N = 0.8$, and the sample number is $N_c = 100$. The maximum iteration number is $\mathsf{It}_c^{max} = 20$.

$\mathcal{L}_c$	Med. R Err. (°)	Med. t Err. (cm)	Recall@5°/5cm
✔	0.8	5.6	44.3
✗	0.8	5.7	43.8

Table 7. **Ablation Study about Loss Function.** The experiments are conducted on theMapFree s00006 dataset. We report the median rotation and translation error, along with the recall.

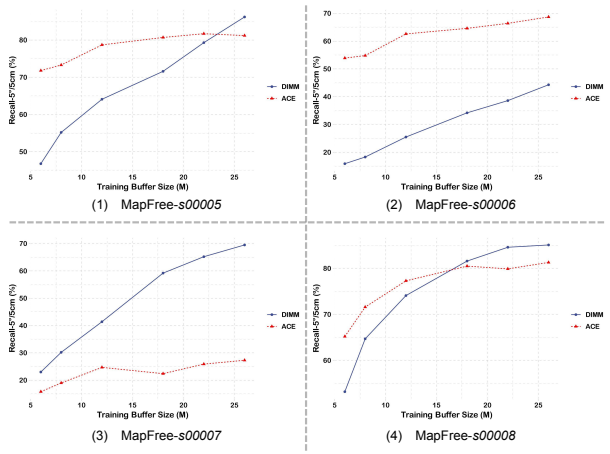

Figure 6. **The observed Training Buffer Scaling Law in CRR.** The experiments are conducted on four scenes of MapFree. It can be seen that large training buffer size leads to greater performance improvement for our method compared to SCR-based ACE.

B. Additional Ablation Study

In this section, we provide two more ablation experiments about our approach, including the training loss and model design of the MLP Mappers.

Training Loss. We validate the effectiveness of center loss in Eq. (16) in the $s00006$ scene of the MapFree dataset. The results are presented in Tab. 7. It is evident that center loss positively impacts network accuracy, particularly enhancing the accuracy of camera translation. This indicates increasing the sensitivity of model to geometric constraints during training can improve overall performance.

The Number and Architecture of MLP Mappers. Additionally, for the number of sub-mappers in our mapper head and the network depth of each sub-mapper, we conducted ablation experiments on scene1 of Indoor-6. With regard to the mapper depth, we modify the MLP depth of the common mapping, but keep the depth of the direction and moment mapping MLP unchanged (both of them are 1-layer MLP). We report the VL Recall@5°/5cm and the corresponding model size. From Tab. 8, it can be observed that a combination of 4 sub-mappers, each composed of MLPs with 4 layers, yields the best accuracy.

Training Buffer Size. During training, we observe a significant impact of the Training Buffer Size (**TBS**) on the model performance. We collect specific data on MapFree $s00005 \sim s00008$ scenes for both our DIMM and SCR method ACE, as illustrated in Fig. 6. It is evident that DIMM accuracy is positively correlated with the TBS. In GDT, the TBS refers to the number of *patch-ray* pairs used for training. The ray parameters corresponding to image patches are almost always distinct with each other in CRR, unlike in SCR where many image patches correspond to the same 3D point. Therefore, a larger TBS implies more training data for CRR-based DIMM, leading to more significant performance improvement compared to SCR-based ACE. We refer to this phenomenon as the CRR-specific Training Buffer Scaling Law. It is important to note that DIMM generally surpasses ACE with the same TBS, showcasing its superiority. We have reported the results of methods with different TBS in previous experiments, while ensuring their consistency with the time and memory costs in Tab. 1. This scaling law also leads to limitations of our method discussed in Suppl. C.

C. Limitation Discussion

In this section, we discuss the limitations of our method. One of the main contributions of this work is the introduction of Camera Ray Regression (CRR) into Visual lLocalization (VL). Thus, we first compare our proposed CRR task with previous SCR tasks. From these task differences, we identify the main limitation of our method: its reliance on a large training data volume due to local ambiguity.

C.1. Comparison between SCR and CRR

Due to the differences in camera pose representations, SCR and CRR task models have to handle ambiguities differently from learning. Specifically, the SCR task requires fitting the mapping from image features to 3D point coordinates. This mapping often includes ambiguity cases where different image features correspond to the same 3D point (due to the view point change or lightning variation) and where similar image features correspond to different 3D points (*e.g.*, repeated textures). The former is more common than the latter. To address this, the model must learn to memory the scene invariance from training data, essentially capturing the scene geometry.

In contrast, in the CRR task, it is rare for different image features to correspond to the same ray, as this would require specific camera movements. However, similar image features corresponding to different ray parameters are common. This includes ambiguities caused by repeated textures (referred to as *global ambiguity*, same as that in SCR) and, more significantly, *local ambiguity* inherent to ray-based representations. In particular, when the camera perspective changes, ray parameters can vary significantly, whereas image features may remain largely unchanged. Due to its commonality, local ambiguity is a key challenge in the CRR task.

Since the ray parameters change with the variation of

	M	1	2	4	6
D_M	2	10.8%/8.8MB	19.1%/11.6MB	32.8%/13.1MB	18.4%/14.6MB
	4	10.1%/9.4MB	19.9%/12.6MB	44.4%/15.2MB	26.9%/17.7MB
	6	9.3%/9.9MB	20.3%/13.6MB	30.9%/17.2MB	31.8%/20.7MB

Table 8. **Ablation Study about the Number and the Depth of MLP Mapper on Indoor-6 Scene1.** The rows are different numbers of mappers (M), while the columns correspond to various mapper depths (D_{Map}). We report the VL Recall@5°/5cm (%) and the head size (MB).

camera perspective, the variation degree of corresponding image patch feature primarily depends on the depth continuity within the image patch, followed by the scene lighting conditions. When depth discontinuity exists within an image patch, camera perspective changes can cause significant variations in image features. In such cases, the model can distinguish different ray parameters by memorizing scene geometry, similar to the SCR task.

However, when depth within an image patch is continuous, the image contents may remain largely unchanged under camera perspective shifts (*e.g.*, flat surfaces such as walls), unless lighting conditions vary significantly. In such scenarios, the model can infer ray parameters by leveraging the geometry constraint and refer to ray estimations from depth-discontinuous patches within the same camera view. This is a strategy we incorporate in our method, referred to the potential geometric constraint learning. Additionally, the model may weakly learn lighting variations in local image patches to help differentiate rays.

From the above analysis, it is evident that the CRR task requires the model not only to memorize scene geometry but also to perceive camera view point changes for local disambiguation. On the other hand, while solving for the camera pose from rays is simpler than using 3D-2D correspondences (linear solution vs. PnP), the computational burden is consequently shifted to the learning process of the model. Fortunately, with powerful feature encoders such as DINO, this learning task becomes feasible. However, compared to the simpler SCR task, CRR also introduces certain limitations due to the above local ambiguity, which we will discuss next.

C.2. Limitation from Local Ambiguity

Since learning to perceive perspective changes is essential, having sufficient training data is particularly important for CRR. In SCR, as different image features can correspond to the same 3D point, a large portion of patch-point pairs in the training data is redundant. Consequently, in GDT, a relatively small training buffer size is sufficient for the model to learn the scene geometric information. An increase in training data size does not significantly improve model performance, *e.g.*, in Fig. 6.

In contrast, for CRR, nearly all patch-ray pairs serve as "new" training data for the model with different ray parameters, providing effective supervision. Meanwhile, the extra requirement for learning perspective variations in CRR necessitates a significantly larger training buffer size than in SCR. This, in turn, increases both training time (*i.e.*, mapping time) and computational costs as shown in Tab. 1.

More importantly, in certain scenarios where the scene scale is large but the available image data is sparse, the CRR task suffers from insufficient training data, such as the Cambridge Landmarks dataset [15]. As a result, our method may fail in these cases. To address this limitation, recent work [18] on using 3DGS for VL task data augmentation offers a potential solution. By generating additional effective patch-ray pairs, these approaches can provide enhanced supervision for CRR model training. We leave this exploration for future work.

D. Additional Qualitative Results

In this section, more visualization results are presented.

D.1. Camera Pose Visualization

Firstly, we provide some camera pose visualization results in Fig. 7. There are 10 random samples from each scene of Indoor-6 in the figure. We show differences between the camera poses from our method and corresponding ground truth. It can be seen that our methods can achieve accurate pose regression. Meanwhile, 3D scene information is hard to obtain from the camera rays regressed by our method, providing effective privacy preserving.

D.2. Semantic Attention Visualization

In addition to visualization results in Fig. 5 and descriptions in Sec. 4.3, we provide more results in Fig. 8 and Fig. 9 to show some interesting cases resulting from the proposed semantic attention module. The color bars on the bottom of the images represent the numerical size of the attention scores from different MLP mappers, as shown in Fig. 5. A comparison of these figures reveals that different image contents correspond to varying attention scores of mappers, which implies a semantic-based spatial segmentation in our model. For example, the fireplace in Fig. 8 elicits a high attention score from the mapper #3 (the blue bar), but low attention from the mapper #2 (the green bar). The sofa in Fig. 9, however, receives high attention from the mapper #2 (the green bar). This suggests that different MLP mappers learn to focus on distinct image regions based on semantic information. The phenomenon naturally emerges during training in our semantic attention module, aligning with our design goal of using multiple MLP mappers to encode scenes based on semantic-based spatial segmentation and adaptively fusing their outputs through the attention mechanism under semantic guidance. On the other hand, the mapper #1 (the red bar) and the mapper #4 (the cyan bar) are consistently activated in both Fig. 8 and Fig. 9. This can be interpreted as these

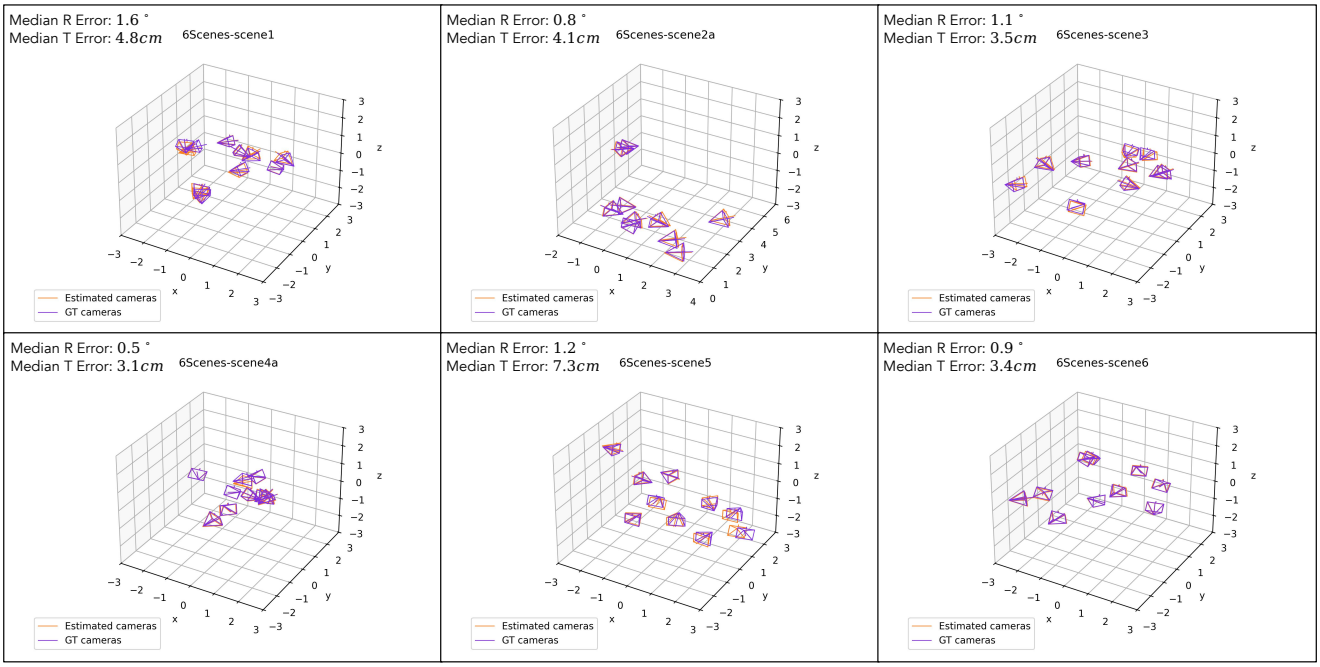

Figure 7. **The qualitative results in Indoor-6 benchmark.** We randomly sample 10 frames from the six scenes to visualize their camera pose comparison between our method and the ground truth. The median rotation and translation errors of these samples are presented as well.

Figure 8. **The qualitative results in Indoor-6 benchmark.** The color bars on the bottom of the images are attention score visualization of MLP mappers, cf. Fig. 5. In these figures, it can be seen that the score of the mapper #2 (the green bar) is minor, compared to the mapper #3 (the blue bar). At the same time, a fireplace in the images seems to be an anchor of high attention scores of mapper #3 (the blue bar).

two mappers may be responsible for global scene memory, implying that a hierarchical scene segmentation also exists in our mapper head. Overall, the proposed semantic attention module facilitates a *soft* multi-mapper ensamble, which implicitly performs hierarchical and semantic-based scene segmentation to organize mapper memory.

D.3. Ray Error Visualization

To further highlight the importance of the global perception in our method, we provide ray error map visualization results in Fig. 10 to compare feature encoders with and without global perception, in addition to Fig. 4. From the figure, it can be seen that the ray errors are concentrated in genuinely challenging areas **after** incorporating global perception, such

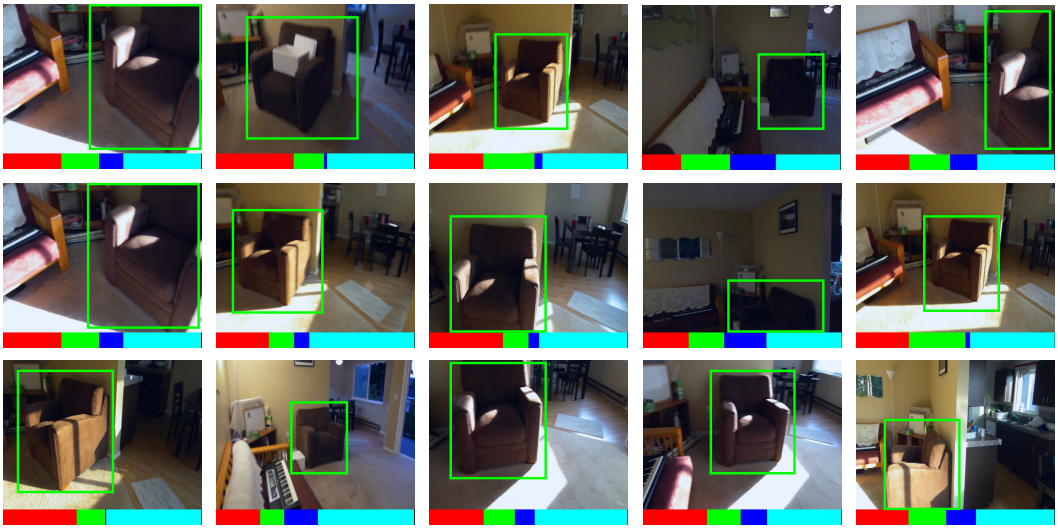

Figure 9. **The qualitative results in Indoor-6 benchmark.** The color bars on the bottom of the images are attention score visualization of MLP mappers, cf. Fig. 5. Comparing to Fig. 8, these figures have quite different contents, and the mapper #2 (the green bar) showcases larger attention scores than the mapper #3 (the blue bar). Similarly, a sofa appears frequently in these images, leads to high attention of the mapper #2 (the green bar).

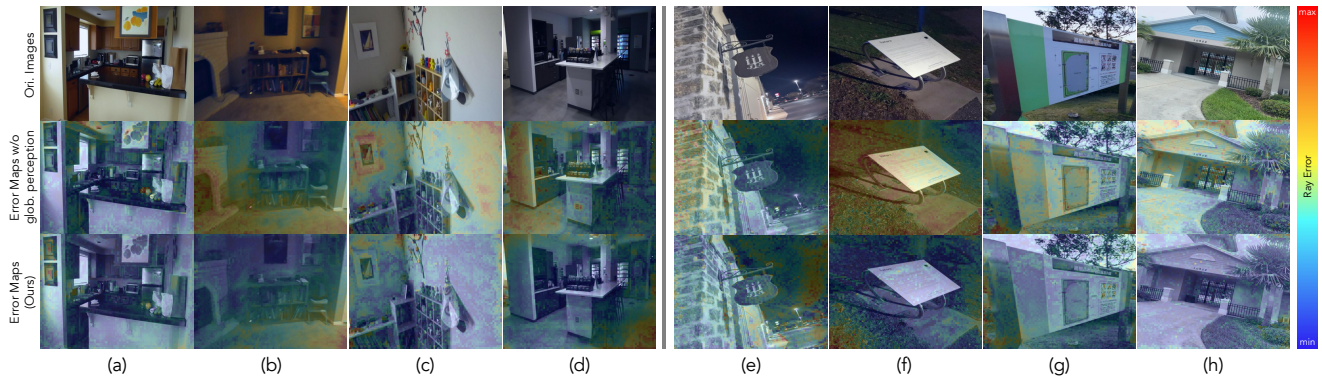

Figure 10. **The ray error distributions with and without global perception in feature encoder.** It is evident that ray errors accumulate in particularly challenging areas after incorporating global perception into the features. The reduction of diffused ray errors can be attributed to the rich semantic information provided by global perception, which also facilitates ray correction through geometric constraints.

as textureless backgrounds with continuous depth (Fig. 10 (b), (c),(d), (e)) and repetitive patterns (Fig. 10 (a), (f), (h)). However, without the global perception, the ray errors spread in much larger regions, *e.g.*, in Fig. 10 (b), (f). This can be interpreted as that the learning of underlying geometric constraints provides effective ray correction (as shown in Fig. 4), which is highly beneficial for the disambiguation in CRR. On the other hand, the remaining ray errors in our method tend to concentrate in regions that exhibit minimal appearance change under varying camera perspectives, which reveals that local ambiguity remains a critical challenge in the CRR task. Therefore, future work could improve the local disambiguation, particularly for textureless and repetitive patterns.